Discrete-Event Simulation and Integer Linear Programming for Constraint-Aware Resource Scheduling

Seung Yeob Shin, Hari Balasubramanian, Yuriy Brun,Philip L. Henneman, Leon J. Osterweil

Abstract—This paper presents a method for scheduling resources in complex systems that integrate humans with diverse hardware and software components, and to study the impact of resource schedules on system characteristics. The method uses discrete-event simulation and integer linear programming, and relies on detailed models of the system's processes, specifications of the capabilities of the system's resources, and constraints on the operations of the system and its resources. As a case study, we examined processes involved in the operation of a hospital emergency department, studying the impact staffing policies have on such key quality measures as patient length of stay, numbers of handoffs, staff utilization levels, and cost. Our results suggest that physician and nurse utilization levels for clinical tasks of 70% result in a good balance between length of stay and cost. Allowing shift lengths to vary and shifts to overlap increases scheduling flexibility. Clinical experts provided face validation of our results. Our approach improves on the state of the art by enabling using detailed resource and constraint specifications effectively to support analysis and decision-making about complex processes in domains that currently rely largely on trial and error and other ad hoc methods.

Index Terms—discrete-event simulation, linear programming, resource planning

1 Introduction

Our society has become increasingly dependent on complex human-intensive systems that integrate human resources with diverse hardware and software components. As a result, correctness of system performance, safety, and efficiency have become correspondingly important. For example, such systems are responsible for keeping airplanes safely separated from each other, oversee the delivery of healthcare to patients in clinical settings, and support electric power grids. The incorrect or unsatisfactory performance of these systems can lead to waste, damage to critical infrastructure, and even loss of life. Providing desired assurances about the speed, correctness, reliability, and efficiency of these systems has become a critical societal need. But the size and complexity of these systems greatly complicates our ability to provide these kinds of assurances.

The behavior of these systems is complicated considerably by reliance on many different kinds of human and other resources, diverse goals and optimization objectives, and a combinatorial explosion of contingencies and exceptional conditions that may arise during execution. Because these systems integrate the contributions of humans, they are sensitive to differences in the skill levels and various idiosyncrasies of those humans, including the possibility that different humans may perform differently under identical conditions. The performance characteristics of these systems are likely to vary considerably depending on the conditions under which they operate. Thus, a system that is heavily loaded may behave quite differently from the same system operating under a light load. Further, all of these dimensions of complexity are often interdependent, increasing the challenge of understanding and optimizing such systems. For example, some systems' performance may degrade when under heavy load, but that degradation may be mitigated by substituting highly skilled humans into key roles, or by eliminating certain exceptional conditions.

This complexity poses significant challenges to efforts to reason definitively about key system characteristics, such as safety, correctness, speed, and efficiency. For example, the throughput of a system cannot be determined without taking into account the characteristics of human performers, the scheduling of the human and other resources' availabilities, and the possible system execution sequences, including those taken in response to exceptions and contingencies. The safety of such systems cannot be assured without being able to reason about all execution possibilities, but also all possible actions that might be taken by human participants, under all possible load conditions. Because of the interrelatedness of all of these factors, straightforward analytic approaches must all too often make simplifying assumptions that limit the value of their conclusions.

In this paper, we propose using a combination of discrete-event simulation, and integer linear programming to develop resource schedules and to accurately estimate key system properties, while optimizing desired constraints. Our approach allows enforcing complex constraints on the resources, including variations in human skills and interactions between humans.

Discrete-event simulation research has been one of the most promising approaches to studying the behavior of such complex systems. But here too oversimplification due to overly conservative assumptions can raise serious questions about the validity of

- S. Y. Shin, Y. Brun, and L. J. Osterweil are with the School of Computer Science, University of Massachusetts, Amherst, MA, 01003, USA.
E-mail: {shin, brun, ljo}@cs.umass.edu
- H. Balasubramanian is with Mechanical and Industrial Engineering, University of Massachusetts, Amherst, MA, 01003, USA.
E-mail: hbalasubraman@ecs.umass.edu
- P. L. Henneman is with the Department of Emergency Medicine, Tufts-Baystate Medical Center, Springfield, MA, 01199, USA.
E-mail: philip.henneman@baystatehealth.org

simulation results that have been obtained, thereby limiting their value. Prior work on resource scheduling has not accounted for the detailed models of resources, and constraints on those resources necessary to model complex, human-intensive systems. Queuing models, for example, have only partially addressed variations in resource load [13], [16], [19], [20], [21], and cannot apply to the flow of complex systems [6], [9], [14]. In view of these concerns, our work has focused on the development and application of discrete-event simulations that account for (1) detailed process models that include specifications of how to deal with such complications as concurrency and responding to exceptional conditions, (2) detailed specifications of the characteristics of resources (including human resources) and the ways in which their efforts are applied to process performance, and (3) specifications of how system load can be expected to vary over time.

Specifically, this paper presents a novel approach for creating precise and detailed discrete-event simulations of complex, human-intensive systems and their performance in resource-constrained environments. Our approach consists of three steps. First, the approach uses discrete-event simulation to compute resource requirements, such as how many of each resource are necessary to be present at each time in the discrete-event simulation to meet user-specified resource utilization requirements. Second, our approach uses deterministic integer linear programming (ILP) to produce a resource schedule that satisfies those resource requirements and user-specified constraints on resource utilization. Third, our approach again uses the discrete-event simulation to compute how the resource schedule affects statistical estimates of the system's runtime properties. Our approach combines the rigor of mathematical programming with the complex detail and realism of a discrete-event simulation. While previous projects have developed and applied discrete-event simulation based on detailed models of processes and resources, our work is unique in the breadth and depth of detail in the models, and the incorporation of resource constraints. As a result, our approach more accurately models human-intensive systems, particularly under resource-constrained conditions, and allows for precise measurements of system properties and developing resource requirements and schedules that optimize a wide variety of objectives.

The main contributions of our work are a novel approach for using discrete-event simulation and integer programming to analyze complex, human-intensive systems and develop resource schedules in resource-constrained environments, and an evaluation of this approach in the healthcare domain. Our approach is mathematically rigorous and precise. Unlike prior work, our approach:

- incorporates constraints on the resources, and their capabilities, use, and allocation policies;
- handles the complex realism of real-world, resource-constrained systems, including multiplicity of resources, time-varying events that trigger resource requirements, detailed processes of resource use, and empirically derived, stochastic time distributions of durations of process steps;
- accounts for a desired, target resource utilization ranges and constraints on resources scheduling, such as "resources of a given type may only be utilized between 60% and 75% of the time, for no more than 8 hours per day;"
- accurately models resource interactions, and the resource scheduling interplay in simultaneously scheduling all resources, accounting for all relevant constraints; and
- allows flexibility in which system properties are optimized, including often conflicting measures.

While our technique is general and applies broadly to resource-dependent systems, for exposition and evaluation, we focus in this paper on healthcare systems, and, in particular, on hospital emergency departments (EDs). EDs, like many resource-dependent systems have significant constraints on their resource use, and variability in system requirements. For example, EDs commonly have five-fold variation in patient arrival rates throughout a 24-hour period, and staffing and resource scheduling decisions need to be responsive to such variation while simultaneously considering the impact on conflicting objectives, such as patient waiting time, utilization of the doctors and nurses, delays in care, and staffing costs.

We evaluate our scheduling approach by applying it to detailed models of EDs. The approach identifies several interesting findings. First, staffing composed of variable-length shifts that are allowed to overlap requires less staff salaries, and enables greater staff utilization. Second, staffing composed of long, single-length shifts that do not overlap (i.e., start at the same time), results in fewer patient handoffs (the transitioning of a patient's care to a new nurse or doctor) but incurs significant deviations from desired utilization ranges. Allowing flexible shift start times significantly reduces this problem. Further, while the increase in handoffs incurred by overlapping shifts increases the patient length of stay, that increase is only marginal.

Our approach enables not only computing resource schedules, but also exploring how constraints, requirements on the resources, and allocation policies impact critical system properties. In the ED domain, the approach enables exploring, in a principled manner, the effects of doctor and nurse assignment policies, patient admittance policies, shift scheduling policies, requirements on a single doctor and nurse handling all of a patient's procedures, on the length of stay of the patient, patient handoffs, hospital financial efficiency, etc.

The rest of the paper is organized as follows. Section 2 provides a detailed review of the relevant research. Section 3 presents the methodological details of our simulation optimization approach, first overviewing the discrete-event simulation engine, then describing the integer linear programming formulation and the scheduling algorithm, and finally, using the simulation optimization framework. Section 4 evaluates our approach by applying it to the ED scenario and (1) computing the hourly staff utilization, and (2) the interplay between utilization, staffing costs, length of stay, and handoffs. Finally, Section 5 summarizes our contributions and future research directions.

2 Related Work

The problem of scheduling resources in complex environments has been extensively studied, with a considerable focus on hospital emergency department operations. This section describes this prior work, and its limitations. Overall, despite the large number of these prior studies, in our view, none of them has attempted to model all of the relevant aspects of the scheduling problem simultaneously, and in sufficient detail.

Bergh et al. [10] provide a comprehensive literature survey of personnel scheduling problems. They examined 291 relevant articles published from 2004 to 2012, and found that 93 of these papers are related to healthcare. Their work indicates that the staffing problem in healthcare has been widely studied, but that many issues remain to be addressed adequately. These issues include: (1) Lack of integration of the dynamic complexity of the staffing, such as demand forecasting, hiring and firing, machine scheduling, and multiple hospital locations. (2) Lack of integration of the details of the staff, such as differences in skills, flexibility in contracts, and breaks. (3) Lack of consideration for the staffing and equipment constraints. (4) Not handling uncertainty sources, such as nondeterminism in decision making, in scheduling, and in the demand. (5) Insufficient testing of the robustness of solutions to noise, uncertainty, schedule flexibility, etc. (6) Consistent lack of implementation and empirical study of effects of the proposed algorithms. (7) Lack of scientific comparison between the approaches.

Analytical approaches are widely used in studying ED staffing, since closed form expressions are relatively easy to calculate and require less data than simulation approaches. Green et al. [14] show how queuing models work effectively to create staffing for various patient arrival rates in an ED. Their work focuses on the time lag between a patient's arrival and their actual treatment. While a useful rough-cut approach, this work simplifies the ED care process considerably, and only calculates the physician schedules. By contrast, our approach accounts for the fact that patients with different acuities have different care needs, that the patient care process consist of multiple steps, and that nurse and other resources' schedules also affect lag times.

Cochran et al. [6] introduce a multi-class queuing network analysis for the capacity planning of both beds and staff. Their approach incorporates five types of patients characterized by resource utilization and priority, non-exponential service time distributions, and nine patient care areas. Based on their queuing network, they can determine staff level for each area to satisfy a quality measurement, such as the number of patient walkouts. This work does not take into account the need to schedule physicians, nurses, and equipment simultaneously, and instead uses statistical service time distributions, which we believe is a less accurate scheduling method than our approach.

Li et al. [18] propose an analytical framework that models the complexity of an ED by incorporating a variety of flow controls, such as split, re-entrant, closed, and parallel queueing. The paper presents a method for redistributing limited resources and mitigating bottlenecks. However, the approach proposed by Li et al. does not go as far as our approach in modeling the complexities of the ED. For example, their approach does not model constraints that patients need to be cared for by the same physician and nurse — at least until the end of that physician's and nurse's shifts.

Many researchers have noted the need to model multiple, complex constraints to schedule ED resources accurately. Some have even stressed the need to model variations in numbers of working hours per week, days-off regulations, and staff salaries. Carter et al. [5], for example, address the problem of scheduling ED physicians in the presence of real-world constraints from six hospitals in the greater Montreal, Canada area. They attempt to create schedules that improve the quality of patient care, while satisfying physicians' vacation schedule requirements and assuring that there is a minimum of 16 hours between any pair of a physician's shifts. However, this work neither considers the simultaneous scheduling of other resources, such as nurses and clerical assistants, nor more complicated sets of constraints, and it is unclear if it can be extended to handle those two common features of EDs.

As with our approach, Brunner et al. [3], [4] study the possibility of scheduling flexible shifts for physicians. Their work allows full flexibility of shift starting times and shifts lengths, and includes the assignment of breaks and the use of planned overtime, while conforming to constraints defined as part of general labor agreements. Ferrand et al. [12] provide a method to build cyclic physician schedules that can be repeated throughout the year. Their schedules can incorporate holidays, work assignments, and vacation requests. Stolletz et al. [27] introduce an optimization technique to schedule physicians with flexible shifts. Their approach supports balancing the work times and the number of on-call service assignments over all physicians. Kazemian et al. [17] introduce a deterministic integer-programming-based healthcare provider shift design to minimize patient handoffs. While considering complex shift constraints, unlike our work, all of these approaches rely on relatively coarse approximations of ED processes, and ignore the scheduling of nurses and clerical workers. These factors can greatly affect key ED measures, such as patient waiting time and staff utilization, which puts into question the applicability of these approaches in the real world.

Understandably, it is difficult to incorporate the full range of the complex issues arising in ED resource scheduling into analytical models. Hence, discrete-event simulation has been widely used in studying ED resource allocation [1], [7], [11], [22], [23], [25], [26]. Wang et al. [29] use a simulation model to identify potential changes in operational policies to reduce patients' length of stay. They suggest reassignment of nurse jobs, combining registration with triage, adding float nurses, mandatory requirement of physician's visit within 30 min, and the simultaneous improvement of durations of the most sensitive procedures to decrease the length of stay. Zeng et al. [31] use a simulation model to study the ED of a community hospital. Based on their sensitivity analysis, they suggest that adding nurses and CT scanners can reduce patient waiting times and length of stay. They also suggest that a team nursing policy (creating pooled capacities) can significantly improve ED efficiency. Brenner et al. [2] use simulation to identify bottlenecks and investigate the optimal numbers of human and equipment resources.

In contrast to our work, these discrete-event simulations do not model time-varying arrival rates, which are particularly important to the accuracy of staffing models. We use a combination of integer linear programming and discrete-event simulation to support resource scheduling in the presence of time-varying arrival rates. Sinreich et al. [8] also propose the combination of discrete-event simulation and integer programming to study staff scheduling, using simulation to first identify the bottleneck resource and the required number of units of this resource, and then reschedule the start times of the bottleneck resource's shifts to better fit the resource requirements. Iterating the steps of identifying a bottleneck resource and optimizing the resource's shift start times can approximate the optimal staffing of the resources. This paper also presents an algorithm for transferring shifts between similar resources, such as fast-track physicians and surgical physicians, to improve staffing of the bottleneck resource. However, in scheduling one resource at a time, this approach does not, for example, take into consideration complex interactions between resources. By contrast, our work schedules multiple resources simultaneously, accounting for interactions between resources, and procedures that require multiple, constrained resources. Additionally, unlike Sinreich et al.'s approach, our work allows more flexibility in shifts than fixed, 8-hour shifts.

Izady et al. [15] propose a heuristic iterative approach to determining the minimal hour-by-hour staff levels needed in an ED. Their approach combines a queueing model of non-stationary infinite server networks, the square root staffing law, and simulation to achieve the UK government target that 98% of patients should be discharged, transferred, or admitted within 4 hours of their arrivals. Their technique first calculates required staffing levels using an offered load analysis [21] and the square root staffing law [16] with non-stationary infinite server networks. Then, the technique uses simulation to test if the derived staffing satisfies the government target. If it does not, the technique reruns the algorithm by adjusting the target delay probability of a resource. While this approach incorporates multiple types of resources, interactions between resources, and different patient routing based on the patient types, it takes into consideration only a limited number of constraints that typically characterize ED operations. For example, their model does not allow consideration of ED operations for which a patient must be cared for by the same doctor and nurse that were assigned when the patient was first placed in a bed. In addition, this technique does not support analysis of staff utilization under time varying arrival rates. By contrast, our approach handles both detailed constraints and varying arrival rates.

Zeltyn et al. [30] use a simulation model of an ED to propose staffing levels over several different time horizons ranging from several hours to several months. This work presents the modified offered-load approximation staffing algorithm. The approach first simulates ED models hypothesizing the availability of infinite resources to estimate the workloads of busy resources, and then uses these estimates to calculate staffing demands through offered load analysis, finally evaluating the through simulation. The simulation-based offered load analyses show significant improvements over a commonly used, rough cut capacity-planning technique [28], as measured by waiting time to be first seen by a physician. However, by contrast to our approach, this work only on finding staffing demand levels, and has not used to study the influence of variability in shift starting times and shift lengths. In addition, this work does not provide insight into resource utilization levels.

In summary, previous work has made various simplifying and restrictive assumptions in studying how resource utilization approaches affect such key quality measures as patient waiting time and resource utilization levels. Some of these techniques have not considered variable arrival rates. Some support minimal (or no) resource utilization constraints. Some have failed to consider the effects of one resource type on another. Some rely on coarse process models that poorly describe the patient care process, or simplistic resource models that poorly describe the complexity of the involved resources. Our work builds on these earlier efforts by addressing these shortcomings. In our work, we combine unusually detailed models of processes and resources, a highly flexible approach to specifying constraints, variable arrival rates, and flexibility in human resource shift policies.

3 Approach

As indicated above, our work centers on using a powerful capability for simulating the operations of a hospital ED. We begin this section with a detailed explanation of this simulation capability, focusing first on our approaches to specifying the ED process and the ED resources, and then focusing on incorporating into our simulations various constraints and context conditions.

3.1 ED Process Modeling

Our approach falls broadly under the heading of model-based simulation, centering on the use of a detailed model of the process by which patients are treated in an example ED. We used the Little-JIL language to create this model. Little-JIL process definitions are based upon the notion of functional decomposition of a high level process into a hierarchy of steps. Little-JIL has well-defined semantics based upon finite state machine definitions, and is supported by a tool suite that includes a graphical editor that renders process definitions as visualizations such as is shown in Figure 1

The central semantic element of a Little-JIL definition is the step. Steps are connected by edges to parents (above) and children (below), with edges also specifying the flow of arguments between parents and children. Parent steps both define scopes, and also specify the flow of control between children. The legend in Figure 1 indicates three different control flow possibilities: sequential (children performed in left-to-right order), parallel (children performed in any order, possibly concurrently), and choice (only one of the children selected for performance). Each step also incorporates a specification of needed resources (e.g. doctor, nurse, x-ray machine) to be allocated at run time (see Figure 3). It is useful to note that these specifications can set up contentions that can further constrain execution order, for example by making concurrent performance either possible or impossible.

Our ED process model was developed based on the advice of a domain expert with extensive experience as an emergency physician and ED manager at the Baystate Medical Center, in Springfield, MA, USA. Figure 1 illustrates one small part of this very detailed process definition, namely the patient testing process for an acuity-level-four patient. Thus, Figure 1 specifies that `AL4Test` is a parallel step, which means that a lab test process, `AL4LabProc`, can be performed in parallel with the other tests, although contention for needed resources (in this case the patient) may make concurrency impossible. As noted in the legend of Figure 1, steps may have prerequisites that may be used by our simulations to specify the relative frequency with

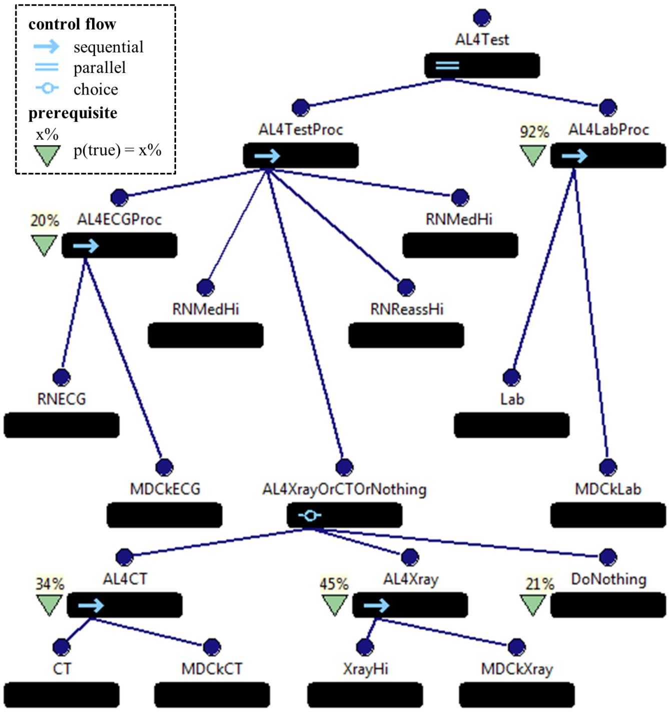


Fig. 1. The Little-JIL definition of the patient testing process, which is part of the care an acuity-level-four patient undergoes in an ED

which exceptions should be thrown, or which of the alternatives specified as the children of a choice steps is the one that should be selected. Thus, the pre-requisite on `AL4LabProc` means that 92% of acuity-level-four patients require the lab test. For the other tests, a nurse checks a patient's `ECG` first, `RNECG`, and then a doctor checks the ECG result, `MDCkECG`, because `AL4ECGProc` controls its child steps sequentially. After the ECG test, a nurse gives a medication to the patient, `RNMedHi`, and then the patient will be transferred to either the CT or the x-ray room. This behavior is represented by the `AL4XrayOrCTOrNothing` choice step which means only one of its child steps will be executed, with the choice being made by the agent who performs the parent step.

The use of Little-JIL to define the Baystate ED process has made it quite easy to define and represent visually some challenging, yet key, features of the process. Thus, for example:

Allowing For Process Variation: The Little-JIL choice step makes it easy to show that patients can arrive in either of two ways: (1) critical patients always arrive by ambulance and are immediately placed in a bed, while (2) other patients arrive using their own transportation and are assigned a bed-placement acuity level. After patients are placed in beds, they are classified into one of six treatment acuity levels. The process each patient goes through varies based on this treatment acuity level. This is easily represented, again using the Little-JIL choice step, where the acuity level artifact is input to the choice step, and is used to select which of its six substeps (each representing the treatment of a patient having the corresponding treatment acuity level) is to be performed.

Supporting Human Decision Making: The choice step kind also makes it easy to represent places where agents (especially human agents) are able to use their judgment to make choices, and what these choices are. As just noted, Figure 3 shows the representation of the doctor‘s ability to choose either a CT or an X-Ray for the patient. In addition the choice step was useful in concisely and precisely defining the way in which patients are triaged by a triage nurse (TrRN). In this case, the TrRN assigns each patient a bed-placement triage level (an integer between 1 and 5), which is an output from the step performed by the TrRN. This assigned triage level is then used as an input to a bed allocation step, which uses the TrRN‘s judgment to determine whether a bed is to be allocated now, or whether the allocation is to be deferred.

Concurrency: Some steps in some of the treatment processes can be performed in parallel, and, indeed, further concurrency arises because the entire treatment process is performed once for each patient in the ED. This creates the potential for contention for resources such as MDs, RNs, and X-Ray machines, and makes the clear and precise specification of the exact nature of the concurrency particularly important. The Little-JIL parallel step makes it quite easy to specify this concurrency, and supports a very clear visual depiction of the concurrency that seems readily accessible to our ED domain experts.

Exception Management: Emergency Department operations rarely follow a straight path through predefined sequences of steps. Instead exceptional and non-nominal situations arise continually. Thus, for example, the lack of needed resources (e.g. beds, nurses) may necessitate changes in treatment sequences or substitution of resources, and treatment procedures that prove ineffective may necessitate new diagnostic procedures and diagnoses. The Little-JIL exception management facilities, featuring scoped handling of typed exceptions has proven to be particularly effective in defining clearly and precisely even difficult exception management scenarios.

Our full detailed process definition contains 164 steps (of which Figure 3 depicts only a small part). While this seems to be a large number of steps, our view is that it is a relatively modest number in view of the size and complexity of the ED process that is defined.

3.2 ED Resource Modeling

While the clear and precise representation of ED process steps and artifact flows is clearly a key challenge in this work, our view is that the clear and precise representation of resources is at least equally important and equally challenging. In Little-JIL resource specification is orthogonal to process activity specification, meaning that the resource specification is separate from the activity specification. On the other hand, the two are precisely linked to each other, with each Little-JIL step incorporating a precise specification of the resources and agent needed to perform the step, and each resource specification incorporating an enumeration of the steps whose execution it is able to support. In order to simulate the performance of a Little-JIL step, a specification of the resources and agent needed is passed to a resource manager, which then assigns appropriate resources to the step, if such resources are currently available. The allocation decision is guided strongly by inspection of the allocation status of each resource, as well as by specified allocation policies and constraints.

3.2.1 Resource Specification in Little-JIL

A Little-JIL resource is modeled as the composition of a set of *attributes* and a set of *capabilities*. A resource‘s attributes describe its inherent nature, and its capability set is the set of the steps for which the resource can participate in performing. Some examples of attributes include the resource‘s age, experience, job title, and place in the ED staff reporting structure. As will be seen these attributes may be used in deciding which resource is most likely to be most suitable for assignment to a given step at a given time.

One particularly important item of information to be used in deciding which resources to allocate is the resource‘s work shift. In our work this is specified by an attribute pair, namely `shiftStart` and `shiftEnd`. A resource will not be allocated to a task unless the allocation is taking place at a time that is between the values of these two attributes.

Two other attributes that are particularly important are the resource‘s `reservation_capacity` and `assignment_capacity` attributes, whose values are used to determine the resource‘s ability to take on new tasks. The `assignment_capacity` attribute is fixed during each simulation, quantifying the maximum amount of effort that a resource can provide at any given time. This is used to ensure that the resource manager does not overload any resource by assigning it to more steps than the resource can handle. Overloading is prevented by maintaining an accumulated effort attribute as a running total of the amount of effort that the resource is expending in performing all of the tasks to which it is currently assigned. In considering assigning an additional task to a resource, the resource manager determines when the effort required by the new task will exceed the resource‘s `assignment_capacity`. Note, however, that our model follows the common practice of allowing some resources (e.g. MDs and RNs) to take responsibility for more patients than they can be actively engaged in treating at the same time. Thus, for example, an MD might have a `assignment_capacity` of 1 (meaning that the MD can be doing only one task at a time), but may still be allowed to be handling more than one patient. On the other hand in most clinical settings there is a limit to the number of patients that the MD can be allowed to handle. This limit is quantified as the value of the `reservationt_capacity` attribute . This attribute is used to limit the amount of effort that the resource can take responsibility for. In most cases, reservation capacity is greater than assignment capacity because a resource should be able to handle a larger number of activities over a period of time than it can handle at any one time.

A resource specification also incorporates a specification of capabilities, namely the steps that the resource is able to participate in performing, and the circumstances under which this is possible. Thus, for example, an MD‘s capability list would presumably include capabilities for prescribing medications and ordering tests, while a Triage Nurse‘s capabilities list would include steps for assigning bed-placement priorities. On the other hand, recognizing that exceptional situations may necessitate exceptional behaviors, each capability also includes the specification of a `guard`, a Boolean expression defined over the values of the dynamically-changing values generated by the simulation, that is used to specify the circumstances under which the resource can be assigned to participate in the performance of the step for which resources are being requested. For example, a guard may specify (although not shown in this example) that a doctor may give injections, but only if no nurse is available, or that an RN may prescribe medications, but only if no doctor is available and only if the patient‘s condition is particularly serious.

Resource allocations must also take into account various kinds of specified constraints. Our simulations allow for the specification of constraints that govern the resolution of resource contention (e.g. when the same resource instance can satisfy multiple resource requests) and activity contention (e.g. when there are multiple resource instances that are capable of providing the capability requested by a given step). First-come first-serve (`FIFO`) is an example of a built-in policy that can be specified for resolving resource contention. A custom-built policy that specifies the use of the least utilized resource first (`LeastUtilizedFirst`) (see Figure 2) is an example of an activity contention policy that is used in our simulation. Our resource manager incorporates built-in policies that specify assignment based on the priority of a request (`Priority`), which resource was least-recently-used (`LRU`), and which resource was most-recently-used (`MRU`). In addition to these, we created custom policies that are based upon various functions over the dynamic variables of the process. Thus, Figure 2 specifies two custom allocation policies, `SickestFirst` and `LeastUtilizedFirst`.

Figure 2 shows an example of specification of how an MD resource is specified. MD resource attributes include `shiftStart` and `shiftEnd` as integer variables. Those attributes are used in `guard` (`time >= shiftStart && time < shiftEnd`) of reservation and assignment to specify when the listed capabilities (`MDCkECG, MDCkCT, MDCkXray, MDCkLab`) are available for reservation and assignment requests. By allowing reservation guard and assignment guard to be specified differently, it enables to speicify the `New patient constraint`, which specify that MDs stop accepting new patients 1 hour before their shifts end. Forcing this constraint only requires a modest change to the reservation guard (`time >= shiftStart && time < shiftEnd-3600`) for the MD resource.

Two capacity attributes, `reservation_capacity` and `assignment_capacity`, in Figure 2 are 1 as their default values. In this example, `effort_needed` of the reservation is 0, so MDs can see multiple patients; however, MDs are allowed to perform only one step of patient care since `effort_needed` of the assignment is 1. If an ED constraints that MDs see at most 4 patients, it requires to change of `reservation_capacity` to 4 and `effort_needed` of the reservation to 1.

3.2.2 Resource Request Specification in Little-JIL

The final piece of the resource model is the way in which resources are requested. Consistent with our previously explained need for the separation of resource allocation into reservation and assignment, Similarly, we separate resource requests from process activities into two types, a reservation request and an assignment request. If an activity requests a resource, and that resource’s allocation constraints (guard and capacity) cannot be

Attribute Declaration

```
<declare-attribute
  name="shiftStart" type="integer" />
<declare-attribute
  name="shiftEnd" type="integer" />
```

Resource Model

```
<resource type="MD">
<attribute name="shiftStart" value="" />
<attribute name="shiftEnd" value="" />
<capacity
  reservation_capacity="1"
  assignment_capacity="1"/>
<capability name="TreatByMD, MDCkECG, MDCkCT,
    MDCkXray, MDCkLab">
<reservation
  guard="time >= shiftStart && time < shiftEnd"
  contention_policy=
    "SickestFirst : ProblemSpecific"
  selection_policy=
    "LeastUtilizedFirst : ProblemSpecific"
  effort_needed="0" />
<assignment
  guard="time >= shiftStart && time < shiftEnd"
  contention_policy=
    "SickestFirst : ProblemSpecific"
  selection_policy=
    "LeastUtilizedFirst : ProblemSpecific"
  effort_needed="1" />
</capability>
</resource>
```

Fig. 2. The `MD` resource model, specifying attributes, capabilities, and allocation policies.

satisfied, the activity will not be able to proceed until a suitable resource becomes available. The activity generates a resource request for each resource it needs. The resource requests are specified as follows (Figure 3 shows several examples of resource requests):

Reservation Request:
reserved-resource: capability, count, [`replaceable`,] `blocking` | `nonblocking`

Assignment Request:
resource: capability, `blocking` | `nonblocking` [, reserved-resource]

Both reservation and assignment requests for resources ask for an available resource that performs a particular capability. Which resource is returned depends on the dynamic state of the process. For example, a doctor may be assigned to drawing a patient's blood, but only when all nurses are fully allocated, and only when the blood draw task is considered to require a small amount of effort and a low skill level. In cases with high skill and effort level requirements, it would be unwise to allocate a doctor, and our request model supports the use of blocking the request (see the `blocking` and `nonblocking` keywords in the request definitions) to ensure that only fully qualified resources are allocated to the activity.

Step	Resource request specification
Treat	reserved_nurse: TreatByRN, 1, `replaceable`, `blocking`
Treat	reserved_doctor: TreatByMD, 1, `replaceable`, `blocking`
RNECG	nurse: RNECG, `blocking`, reserved_nurse
RNMedHi	nurse: RNMedHi, `blocking`, reserved_nurse
RNReassHi	nurse: RNReassHi, `blocking`, reserved_nurse
MDCkECG	doctor: MDCkECG, `blocking`, reserved_doctor
MDCkCT	doctor: MDCkCT, `blocking`, reserved_doctor
MDCkXray	doctor: MDCkXray, `blocking`, reserved_doctor
MDCkLab	doctor: MDCkLab, `blocking`, reserved_doctor
CT	ct_room: CT, `blocking`
XrayHi	x-ray_room: XrayHi, `blocking`

Fig. 3. Resource request specifications. Each step in Figure 1 has a resource request specification associated with it. `Treat` step is not appeared in Figure 1. This scoping step reserves a doctor and nurse resources when a patient arrives in an ED.

Finally, the `replaceable` keyword in the reservation request means that a resource may be replaced by another, under certain situations. (See `reserved_nurse` and `reserved_doctor` in Figure 3) For example, a doctor may need to be replaced when leaving for dinner, while other resource reservation requests may accept no substitutions. This feature of the request enables this language to be used to model very complicated resource management policies and constraints, such as quarantining an entire emergency department, and issuing handoffs among doctors.

Our simulations also incorporate more complicated constraints that are necessary to support the accurate simulation of the actual workings of a real hospital ED. Examples are the `Same MD-RN constraints`, which specify that a patient assigned to a specific bed is to be cared for by the same MD and RN throughout the patient's stay, unless shift changes necessitate the assignment of a different MD or RN. This constraint is specified by restricting assignment of a resource from reserved resources. (e.g., `nurse` resource requests are restricted by `reserved_nurse` in Figure 3.)

3.3 Specification of a Simulation

As the goal of our research is to evaluate different approaches to staffing and resource allocation in a hospital ED, it was necessary to set up and perform ED simulations that assumed different distributions of resources, having different characteristics, and constrained by different policies and constraints. Thus, a simulation run consisted of a specifications of the process activity and artifact flow (specified as described in 3.1, specifications of the resources, requests and resource allocation policies) specified as described in 3.2), and specifications of actual patient care scenarios. The key components of a patient care scenario are: (1) a specification of the flow of patients into the ED, including the rate of arrival of patients of different acuity levels over a 24-hour time period, (2) a specification of the resources that are available for assignment over the 24-hour time period, and (3) a specification of the characteristics of the performance (e.g. the amount of time taken, and the probability of the throwing of an exception) of each resource instance in carrying out each of the

```
<poisson-messages type="ed.PatientArrivalMessage"
     start="1" end="3600" mean="529"/>
<poisson-messages type="ed.PatientArrivalMessage"
     start="3601" end="7200" mean="657"/>
<poisson-messages type="ed.PatientArrivalMessage"
     start="7201" end="10800" mean="772"/>
```

Fig. 4. Specification of the arrival distributions of the patients from 0 to 3 hours (simulation time unit is second). Patient arrival rates over the 24-hour period are specified by the Poisson distribution, based on actual arrival rates at the Baystate Medical Center, in Springfield, MA, USA. The ED handles, on average, 270 patients per day.

steps in the process. We use Little-JIL/JSim discrete-event simulator [24] with extending the resource specification described in 3.2.

Figure 4 shows a specification of patient arrival rates at the ED. Patient arrival rates over the 24-hour are specified by the Poisson distribution. It specifies `mean` time between patient arrivals from `start` to `end` times. Simulation time unit is interpreted as a second in our simulation. Patient arrival is represented as a discrete-event named `ed.PatientArrivalMessage`, so the event will be generated based on the specified Poisson distributions. Note that arrival rates vary over the 24-hour period. The specifications used in our simulations were based upon observations taken at the Baystate Medical Center, in Springfield, MA, USA.

Among the arrived patients (`ed.PatientArrivalMessage` events), critical patients are the sickest (acuity level six), while the others are categorized into the remaining five acuity levels. Figure 5 specifies how to decide critical or non-critical patients among the arrived patients. Because `Treat` step is the first step to handle an arrived patient in our ED process model, it has the linear probability distribution to specify 2% of the arrived patients are critical patients. The specification means when `Treat` step is `posted` to be started by a simulation agent, the agent starts the step immediately since it requires no time consumption to start by `fixed value="0"`. Then, the agent sets an artifact value of `isCriticalPatient` as `true` with 2% of linear-probability.

Figure 2 shows an example of the specification of an `MD` type of resource. Given the type specification, Figure 6 specifies three examples of `MD` instances. The numbers of available MD and RN resources vary over 24 hours, and these numbers are achieved by having the MD and RN resources work in shifts. Typically, an MD or an RN will work one of three different 8-hour shifts each starting at a fixed time, although our simulations have suggested that greater flexibility in the start times and durations of shifts could lead to improved staff utilization. As noted above, the attribute specification feature of Little-JIL resource specification makes this straightforward. The full model (omitted for exposition), also defines the `RN`, `TrRN`, `clerk`, `bed`, `x-ray room`, and `CT room` resources.

Estimates of the amounts of effort required to perform each step for each acuity level are specified by triangular distributions. Figure 7 shows an example specifying the time distributions of three leaf steps, `RNECG`, `MDCkECG` and `RNMedHi`, of the process shown in Figure 1. For instance, when `RNECG` step is `started`, it takes simulation time (second) based on the triangular distribution of (min=233, mode=313, max=472) to `complete` its execution. The time distributions of all steps in our ED process model are determined based on data of Baystate Medical Center, in Springfield, MA, USA.

```
<step name="Treat"> <posted> <group>
<start> <fixed value="0" /> </start>
<linear-probability>
  <case chance="2">
    <set-field parameter="patientInfo">
    <field name="isCriticalPatient">
    <boolean value="true" /></field></set-field>
  </case>
  <case chance="98">
    <set-field parameter="patientInfo">
    <field name="isCriticalPatient">
    <boolean value="false" /></field></set-field>
  </case>
</linear-probability>
</group> </posted> </step>
```

Fig. 5. Specification of critical and non-critical patient distributions. Among the arrived patients, 2% of them are considered as critical patients.

```
<instantiate type="MD" number="3" />

<!-- Shift: 0AM-8AM -->
<instance type="MD" id="1"
  set_attribute="shiftStart" value="0" />
<instance type="MD" id="1"
  set_attribute="shiftEnd" value="28800" />
<!-- Shift: 8AM-4PM -->
<instance type="MD" id="2"
  set_attribute="shiftStart" value="28800" />
<instance type="MD" id="2"
  set_attribute="shiftEnd" value="57600" />
<!-- Shift: 4PM-0AM -->
<instance type="MD" id="3"
  set_attribute="shiftStart" value="57600" />
<instance type="MD" id="3"
  set_attribute="shiftEnd" value="0" />
```

Fig. 6. Specification to instantiate three MD resources. Three 8-hour shifts are specified for the MD resource instances. Time unit of the specification is second.

3.4 Simulation-based Staffing Optimization

The core of our model-based approach to ED simulation was to use the JSim [25] system, which automatically creates simulations from process specifications such as those described in the previous subsection. These various specifications provide data this is necessary to support our simulations, but is not sufficient.

```
<step name="RNECG"> <started>
<complete><triangular-range min="233" mode="313"
    max="472" /></complete>
</started> </step>

<step name="MDCkECG"> <started>
<complete><triangular-range min="11" mode="36"
    max="88" /></complete>
</started> </step>

<step name="RNMedHi"> <started>
<complete><triangular-range min="181" mode="448"
    max="856" /></complete>
</started> </step>
```

Fig. 7. These time distributions of steps are modeled from data of Baystate Medical Center, in Springfield, MA, USA. Second time unit is used in triangular distribution.

Because the goal of our work was to determine how various characteristics of ED operation varied depending upon different levels of staff utilization, it was also necessary to determine what staffing levels could be expected to result in specified staff utilization levels. This was somewhat complicated by the variation in demand for the services of an ED over a 24-hour period of operation. Accordingly we devised a three-stage approach to creating and running our simulations.

In the first stage, we used JSim to generate an ED simulation, but we assumed that there was an infinite supply of all necessary resources. As this ED simulation executed, we added and removed resources as required in order to sustain our utilization targets are for each hour in the simulated 24-hour day. We call this the Staffing Demands Algorithm. These computations were made during execution of the simulation, while considering all other specifications such as time varying patient arrivals, ED processes for each acuity, resource interactions and other patient flow constraints. At the end of this simulation, we had obtained the number of resources d_b^k required in time interval b to maintain the utilization target. For our case-study b is the index for hour of the day, and k refers type of staff – in our case, either MDs or RNs. The exact details of how d_b^k is computed is provided in Section 3.4.1.

In the second stage, we use d_b^k, $b = (0,1),(1,2),...,(23,0)$, as input to a deterministic integer linear program (ILP) whose purpose was to obtain the minimum cost staffing schedule. The decision variables of the ILP determine The number of type k resources to be scheduled in hour b of a 24-hour day. The number scheduled must be greater than or equal to d_b^k. Additionally, hospitals may have their own restrictions such shift lengths and starting times. For example, it is possible that in some EDs doctors can work only 8 hour shifts that start at 8 am, 4 pm and midnight (i.e. no overlapping shifts); in other cases, overlaps in shifts may be allowed, and shift lengths might also be 6, 8 or 12 hours in length. The output of this second stage was x_b^k, the number of resources of type k that need to be scheduled in hour b to minimize the total cost of salaries while meeting the required constraints. The details of the integer linear program are provided in Section 3.4.2.

In the third stage, the staff schedule computed from the second stage ILP was used to specify the exact numbers of MD and RN resources that are to be available for each hour in the 24-hour day simulation. We then ran simulations using these staffing levels, and the other modeling information as described in the previous subsection in order to determine operational information such as patients‘ length of stay, waiting times, contribution margin, and the actual utilization levels of the resources (which, depending on the staffing constraints, may be different from original targets), the number of patient handoffs etc.

For our studies we ran batteries of simulations that were based upon many different hypotheses about staff utilization, shift length variation, shift overlapping, etc. Each of these different hypotheses required carrying out all three of the stages just described. Each produced operational information that could then be used as the basis for comparing and evaluating the effects of these different choices of staff utilization levels and scheduling approaches. The results are presented in Section 4.

We now provide details of how we carried out each of these three stages in our simulation approach.

3.4.1 *Staffing Demands Algorithm*

As the number of patient arrivals varies by the hour, the number of resources required will also need to be varied so that the utilization of the resource remains within the pre-specified utilization limits. The goal of the staffing demands algorithm is to dynamically compute this distribution of required numbers of resources. Before we describe the algorithm, we distinguish between resource sets as follows. We assume that there are some resources that are available for use during the entire duration of the bth hour. In addition, there may also be other resources that may be used for some portion of the hour, because they continue to be used in order to finish a task that was begun during the previous hour. For instance, if a doctor‘s shift ends before completing an x-ray check for a patient, the doctor will complete the x-ray check task thereby providing some amount of MD resource during an hour that is beyond the MD‘s original shift. Thus these additional incremental amounts of resource availability must be added to the resource levels provided by scheduled resources in order to accurately determine the levels of resources required for each hour in the 24-hour day. The notations in the algorithm for staffing demands are:

i time interval between resource adjustment

l lower utilization limit to trigger incrementation of the number of resources required for this time interval

u upper utilization limit to trigger decrementation of the number of resources required for this time interval

b time block tuple (f,t) from time f to t where $|t-f| = i$

k resource type

r_b sum of busy periods for resource r during time block b

d_b^k staffing demand, the number of required staff, for resource r of type k during time block b

R^k the set of resources of type k for which we want to determine required staffing levels

R_b^k the set of resources of type k that are used during time block b, $R_b^k \subseteq R^k$

A_b^k the set of resources of type k that are available to be assigned during time block b, $A_b^k \subseteq R_b^k$, $A_{(0,i)}^k = R^k$ where $(0,i)$ is the first time block

Figure 8 describes the staffing demands algorithm. R_b^k is the set of used resources of type k during time block b. A_b^k is the set of available resources of type k during time block b which means that only resource $r \in A_b^k$ is available to be assigned during time block b. Further, $A_b^k \subseteq R_b^k$.

For example, Figure 9 shows the execution of the algorithm for a single instance. As per lines 1-12 of the algorithm, the resource utilization during time block $b = (t-i,t)$ is calculated. Here, $R_b^k = \{r_1, r_2, r_3\}$ since resources r_1, r_2 and r_3 are used from time $t-i$ to t. However, $A_b^k = r_2, r_3$ because resource r_1 left at time c (some amount of resource utilization in this time period was attributable to the overflow into this time period of some work begun during the previous period). Therefore, $r_{1\ b} = c - t - i$. After the algorithm calculates $util$ at line 12, it determines how many resources are required to satisfy the desired range of utilization levels l and u (lines 14-20). If the calculated $util$ is between l and u, it means that resources in A_b^k are utilized as expected. However, if $util$ is outside the limits l and u, the algorithm calculates staffing demands d_b^k based on the mid-point of the utilization range $mid = (l+u)/2$ and actual assigned sub-periods num during time $t-i$ to t.

In addition to calculating staffing demands d_b^k, the algorithm adjusts the numbers of resources A_{nb}^k for next time block, i.e. from t to $t+i$. The adjustment assumes that there are no dramatic changes in patient arrivals in this next period. Therefore, we use d_b^k to decide the size of A_{nb}^k.

3.4.2 *Staffing via Integer Linear Programming (ILP)*

In this section, we present our ILP-based staffing approach, which minimizes total staff salaries, while meeting (a) hourly constraints on staff numbers calculated by the previously-described staffing demands algorithm, and (b) constraints on allowed shift lengths and shift start times. The ILP-based staffing approach divides a day into several discrete time blocks (each an hour in length in our case-study). The ILP parameters are listed below.

B a set of time blocks in a day

L^k a set of shift lengths for resource type k

$S_{b,l}^k$ a set of time blocks in a shift for resource type k where shift starting time block $b \in B$, shift length $l \in L^k$

d_b^k staffing demands, the number of required staff, for resource type k during each time block $b \in B$

$p_{b,l}^k$ a staffing pattern for resource type k of a hospital
1 if a shift begins a time block $b \in B$ and its shift length is $l \in L$;
0 otherwise

c^k staffing cost per hour for resource type k

For instance, Figure 10 shows the ILP parameter values needed to determine MD staffing levels. Parameter B divides a day into twenty four time blocks. Parameter $p_{b,l}^{MD}$ establishes three eight-hour, non-overlapping shifts a day for MDs. Doctor staffing demand d_b^{MD} is assumed to have been derived using our previously-described simulation-based algorithm.

Alternatively, if an ED administration desires more flexibility to meet hourly variation in demands, they may allow nurses to

```
/** Staffing Demands Algorithm
 * @param upper: upper utilization limit
 * @param lower: lower utilization limit
 * @param t: current time
 * @param i: time block length
 * @param k: resource type
 * @return StaffingDemands: the number of
     required staff for time block (t-i,t)
 * @return AvailableResources: a set of available
      resources for time block (t,t+i)
 */
StaffingDemands,AvailableResources
calculateSaffingDemands (
  double upper, double lower,
  Time t, Time i,
  ResourceType k) {

  // calculate resource utilization
  TimeBlock b = (t-i,t);
  double denominator = 0;
  double numerator = 0;
  for (∀ Resource r : r ∈ $R_b^k$) {
    denominator += (r ∈ $A_b^k$) ? i : $r_b$;
    numerator += $r_b$;
  }
  double utilization = numerator / denominator;

  // calculate staffing demands
  StaffingDemands $d_b^k$ = 0;
  if (utilization ≥ lower &&
      utilization ≤ upper) {
    $d_b^k$ = count($A_b^k$);
  } else {
    double middle = (upper+lower)/2;
    $d_b^k$ = round(numerator/(middle*i));
  }

  // select available resources
  TimeBlock nb = (t,t+i);
  AvailableResources $A_{nb}^k$ = {};
  while(count($A_{nb}^k$) != $d_b^k$) {
    Resource r : r ∈ $R^k$;
    $A_{nb}^k$ = $A_{nb}^k$ ∪ {r};
  }

  return $d_b^k$,$A_{nb}^k$
}
```

Fig. 8. Given the upper and lower utilization limits, the algorithm calculates how many resources of type k are required during time block $b = (t-i,t)$ and adjusts the number of available resources of type k to be assigned for next time block $nb = (t, t+i)$.

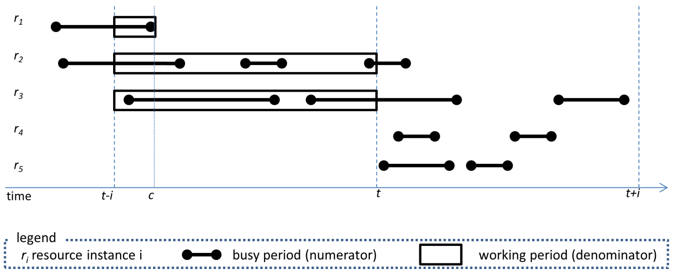
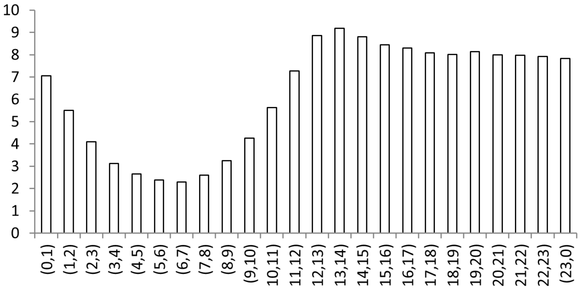


Fig. 9. An instance of the algorithm execution for staffing demands: $R_b^k = \{r_1, r_2, r_3\}$, $A_b^k = \{r_2, r_3\}$, $d_b^k = 3$, $R_{nb}^k = \{r_2, r_3, r_4, r_5\}$, $A_{nb}^k = \{r_3, r_4, r_5\}$ where $b = (t-i, t), nb = (t, t+i)$

B: $\{(0,1), (1,2), (2,3), \ldots, (21,22), (22,23), (23,0)\}$
L^{MD}: $\{8\}$
$S_{b,l}^{MD}$: $S_{(0,1),8}^{MD} = \{(0,1), (1,2), \ldots, (6,7), (7,8)\}, \ldots, S_{(23,0),8}^{MD} = \{(23,0), (0,1), \ldots, (5,6), (6,7)\}$
d_b^{MD}:



$p_{b,l}^{MD}$: $p_{(7,8),8}^{MD} = p_{(15,16),8}^{MD} = p_{(23,0),8}^{MD} = 1, \forall b \in B, b \notin \{(7,8), (15,16), (23,0)\}, p_{b,8}^{MD} = 0$
c^{MD}: USD188/hour

Fig. 10. Parameter values for doctor staffing. B: twenty four time blocks. L^{MD}: 8-hour shift length. $S_{b,l}^{MD}$: time blocks in l length shift b. d_b^{MD}: staffing demands per each time blocks driven by 70%–80% utilization target. $p_{b,l}^{MD}$: non-overlapped three, 8-hour shifts. c^{MD}: salary per hour.

work a six, eight or twelve hour shift; further, they may also allow shifts to start at any time. To accommodate this additional flexibility, the ILP parameters can be set as in Figure 11.

The decision variable in the ILP is $x_{b,l,i}^k$, which determines the number of a particular staff type (MD or RN) needed in each hour of a shift.

$x_{b,l,i}^k$ the number of staff k in time block i in a shift
the shift starts at a time block b and its length is $l \in L^k$

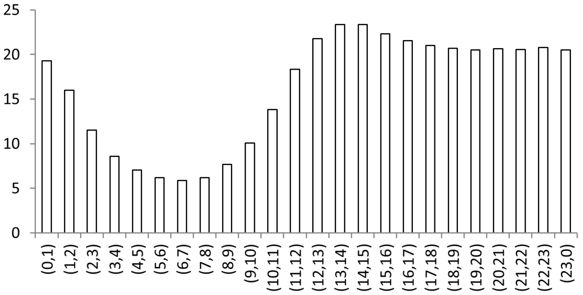
The ILP-based staffing fulfills staffing demands d_b^k, the minimum number of required staff during each time block b. Therefore, the constraint equations (1), (2), (3) always hold true for a staffing solution.

$$\sum_{b \in B} \sum_{l \in L^k} p_{b,l}^k \cdot x_{b,l,i}^k \geq d_i^k, \forall i \in B \tag{1}$$

$$x_{b,l,i}^k = x_{b,l,j}^k, \forall i \in B, \forall j \in B \tag{2}$$

$$x_{b,l,i}^k = 0, \forall i \notin S_{b,l}^k \tag{3}$$

B: $\{(0,1), (1,2), (2,3), (3,4), (4,5), \ldots, (18,19), (19,20), (20,21), (21,22), (22,23), (23,0)\}$
L^{RN}: $\{6, 8, 12\}$
$S_{b,l}^{RN}$: $S_{(0,1),6}^{RN} = \{(0,1), (1,2), \ldots, (4,5), (5,6)\}, \ldots, S_{(23,0),6}^{RN} = \{(23,0), (0,1), \ldots, (3,4), (4,5)\}$,
$S_{(0,1),8}^{RN} = \{(0,1), (1,2), \ldots, (6,7), (7,8)\}, \ldots, S_{(23,0),8}^{RN} = \{(23,0), (0,1), \ldots, (5,6), (6,7)\}$,
$S_{(0,1),12}^{RN} = \{(0,1), (1,2), \ldots, (10,11), (11,12)\}, \ldots, S_{(23,0),12}^{RN} = \{(23,0), (0,1), \ldots, (9,10), (10,11)\}$
d_b^{RN}:



$p_{b,l}^{RN}$: $\forall b \in B, \forall l \in L, p_{b,l}^{RN} = 1$
c^{RN}: USD55/hour

Fig. 11. Parameter values for nurse staffing. B: twenty four time blocks. L^{RN}: 6,8,12-hour shift lengths. $S_{b,l}^{RN}$: time blocks in l length shift b. d_b^{RN}: staffing demands per each time blocks driven by 60%–70% utilization target. $p_{b,l}^{RN}$: three different shift lengths, and the staffing pattern allows a shift to start at any time. c^{RN}: salary per hour.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
$x_{(7,8),8,i}^{MD}$								9	9	9	9	9	9	9	9									
$x_{(15,16),8,i}^{MD}$																8	8	8	8	8	8	8	8	
$x_{(23,0),8,i}^{MD}$	8	8	8	8	8	8	8																	8

Fig. 12. Doctor staffing solution of Figure 10. Three separate shifts, 7–14, 15–22 and 23–6. Each shift has 9, 8 or 8 doctors.

Equation (4) is the objective function of the ILP-based staffing problem. The ILP objective function aims to minimize total staffing costs per day.

$$\min \sum_{b \in B} \sum_{l \in L^k} \sum_{i \in B} c^k \cdot x_{b,l,i}^k \tag{4}$$

To illustrate how the ILP works, a doctor staffing solution for Figure 10 is calculated in Figure 12. There are three separate shifts, 7–14, 15–22 and 23–6. Each shift has 9, 8, or 8 doctors, respectively.

However, the nurse staffing solution in Figure 13 looks very different from the doctor staffing in Figure 12. This is because the ILP parameters for nurse staffing in Figure 11 allow three different shift lengths, and a shift can start at any time (i.e. overlap in shifts are allowed). Therefore, nurse staffing in Figure 13 very closely approximates actual nurse staffing demands d_b^{RN} in Figure 11. We return to this key point while discussing the results of actual simulations carried out as the third stage of our simulation study.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
$x^{RN}_{(0,1),8,i}$	2	2	2	2	2	2	2	2																
$x^{RN}_{(1,2),8,i}$		1	1	1	1	1	1	1	1															
$x^{RN}_{(7,8),6,i}$								3	3	3	3	3	3											
$x^{RN}_{(8,9),6,i}$									4	4	4	4	4	4										
$x^{RN}_{(9,10),6,i}$										3	3	3	3	3	3									
$x^{RN}_{(10,11),12,i}$											4	4	4	4	4	4	4	4	4	4	4	4		
$x^{RN}_{(11,12),6,i}$												1	1	1	1	1	1							
$x^{RN}_{(11,12),8,i}$												3	3	3	3	3	3	3	3					
$x^{RN}_{(12,13),12,i}$													4	4	4	4	4	4	4	4	4	4	4	4
$x^{RN}_{(13,14),12,i}$	4													4	4	4	4	4	4	4	4	4	4	4
$x^{RN}_{(14,15),6,i}$															4	4	4	4	4	4				
$x^{RN}_{(15,16),12,i}$	2	2	2													2	2	2	2	2	2	2	2	2
$x^{RN}_{(19,20),12,i}$	3	3	3	3	3	3	3													3	3	3	3	3
$x^{RN}_{(20,21),6,i}$	4	4																			4	4	4	4
$x^{RN}_{(21,22),8,i}$	1	1	1	1	1																	1	1	1
$x^{RN}_{(22,23),6,i}$	3	3	3	3																			3	3

Fig. 13. Nurse staffing solution of Figure 11. Sixteen overlapped shifts. Shifts have different lengths and start times. Total number of nurses 46 and nurses working hours 392.

4 Experiments

This section describes how we used our three-stage simulation approach to study the effects of basing the staffing levels used to operate an example Emergency Department upon different choices of staff utilization levels. For these simulation studies we used the process model presented in Section 3.1. We also used the resource type specifications presented in Section 3.2, instantiating from these types 2 triage nurses, 5 clerks, 48 beds, 2 x-ray rooms and 4 CT rooms. This ED resource distributions was based on data from Baystate Medical Center, in Springfield, MA, USA. The number of simulation replications that we executed for these studies was determined so as to obtain 95% confidence intervals and a half-width that is within 2% of the mean of staff utilizations. For each replication, we simulated 72 hours of operations of the ED, using only the output of the middle 24-hours in our analysis to ensure that each replication had adequate amounts of warm-up and wind-down times, but that these times did not influence our mean estimates.

In addition to measuring the actual utilization levels for the MDs and RNs in our simulations, we also measured the average Length of Stay (LOS) for patients and the contribution margin (the total revenue derived from treating all patients in the simulated 24-hour period minus staffing and fixed costs). The minimum possible LOS is measured as 116 minutes when there are no resource contentions to perform steps in our ED processes. We also quantified the impact of staff shift scheduling on an important secondary measure, the number of patient handoffs.

We begin by presenting results of the staffing demands algorithm. Recall that the staffing demands algorithm was designed to yield the number of MDs and RNs to be staffed in each hour of the 24-hour day, while ensuring that utilization falls in a prespecified range. We tested the staffing demands algorithm for various combinations of MD and RN utilization limits such as 50%-60%, 60%-70%, and 70%-80% (lower utilization limit - upper utilization limit). We selected these utilization ranges based on our domain expert's advice, expecting that lower ranges would waste personnel time and reduce contribution margin, while higher ranges would increase LOS to an unacceptable level.

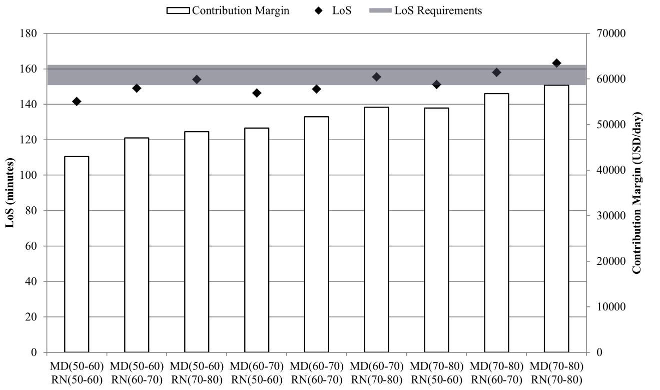


Fig. 15. Simulation results of patient's LoS, contribution margin according to each utilization boundary. `LoS Requirements` is the LoS objective, so that four staffing, MD(50-60) RN(70-80), MD(60-70) RN(70-80), MD(70-80) RN(50-60) and MD(70-80) RN(60-70) satisfy it.

The results of the combinations tested are shown in Figure 14. Note that we have assumed here that the ED is able to change the MD and RN staffing levels every hour as needed. This is equivalent to allowing shifts to be as short as only one hour. Therefore the number of MDs and RNs available for each hour is equal exactly the output of the staffing demands algorithm.

Further, Figure 15 shows the average LOS and contribution margin for each of these combinations. Figure 15 compares LOS for a number of different combinations of staffing utilization levels. The figure uses a gray band to indicate average LOS that lies between 130%-140% of the minimum possible value (116 minutes). This Figure shows that our simulation results indicate that four staffing solutions, MD(50%–60%) RN(70%–80%), MD(60%–70%) RN(70%–80%), MD(70%–80%) RN(50%–60%), and MD(70%–80%) RN(60%–70%), satisfy that LOS objective. Among them, MD(70%–80%) RN(60%–70%) staffing maximizes the contribution margin at 55,113 USD/day. This suggests how our approach could be used by an administrator to evaluate the impact of staffing by adjusting input parameters such as utilization limits and shift lengths.

4.1 Impact of Shift Length and Overlap

To consider the impact of shift length and overlap in shift schedules, we ran simulations in which the allowed shift lengths for RNs could be 6, 8 or 12 hours long in a 24-hour period. We also compared the case where shifts are not allowed to overlap (i.e. the case where shifts are not required to all start and stop at the same time, but are allowed to start an stop at any time) versus the case where they are allowed to overlap. Figure 16 shows RN staffing solutions – output of the ILP-based staffing algorithm in Section 3.4.2. For comparison we also present curves generated by the staffing demands algorithm. Notice that the staffing

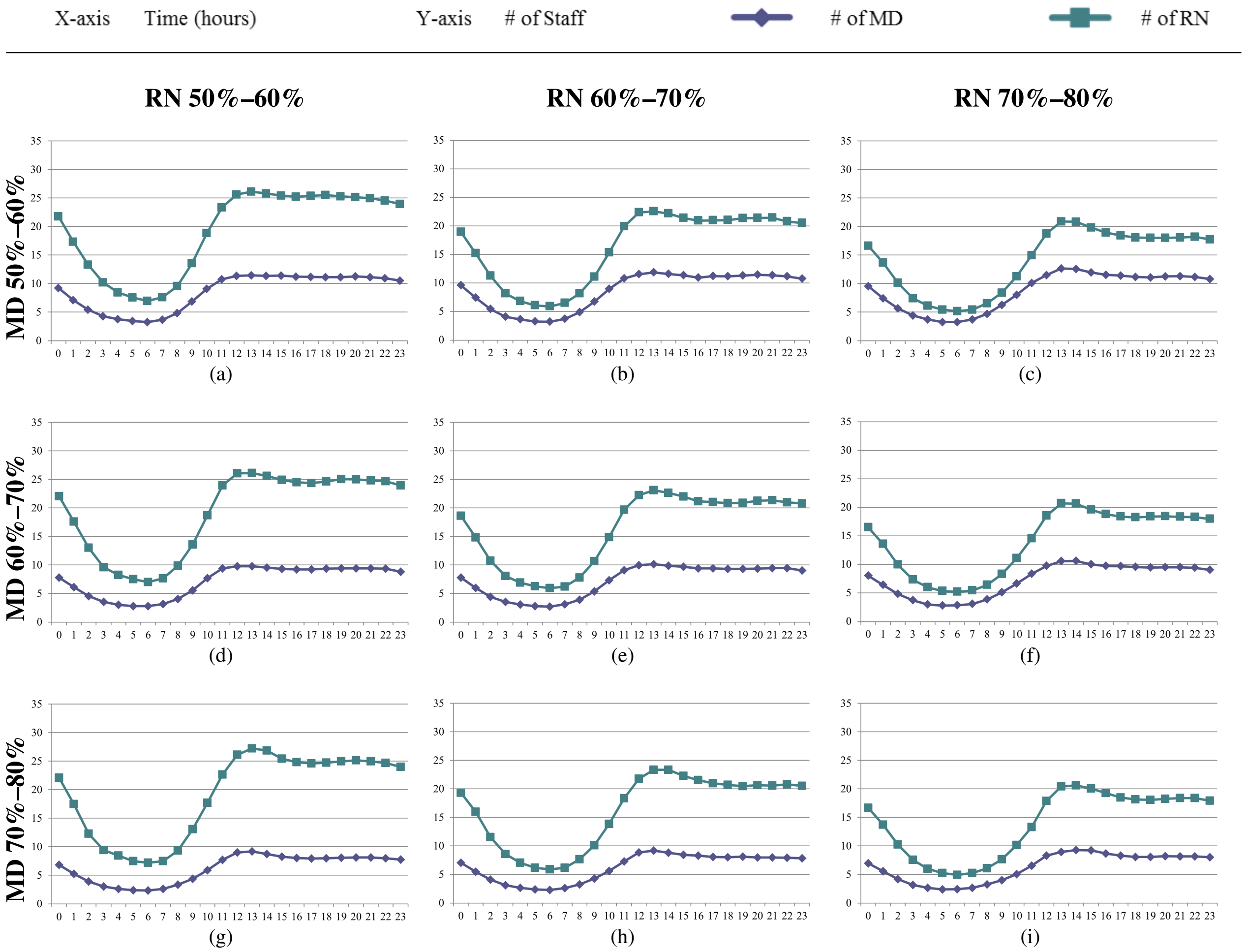


Fig. 14. MD and RN staffing demands curves using the staffing demands algorithm in Figure 8. For example, (a) is MD and RN staffing demands curves when MD's and RN's lower and upper utilization limits are set as 50% and 60%, respectively.

result produced by the ILP for a given shift length and overlap constraint is always equal to or higher than the staffing demands curve.

Figure 16(g) and Figure 16(c) show that a 12-hour shift length cannot cover the staffing demands as closely as 6-hour or 8-hour shift lengths. When we compare the staffing curves of overlapped shifts, Figure 16(a), 16(b) and 16(c) with non-overlapped shifts, Figure 16(e), 16(f), and 16(g), the staffing with overlapped shifts covers the staffing demands curves more closely than the staffing with non-overlapped shifts. Figure 16(d) shows the staffing that allows overlapped shifts of 6, 8 or 12-hour lengths. We call this staffing `flexible` staffing since it allows an ED administrator the greatest flexibility. Notice that such staffing is almost indistinguishable from the staffing demands curve.

In Figure 17, we see the average LOS of the different shift length and overlap combinations. Notice that the LOS differs somewhat across the different staffing options but not significantly. Figure 17 shows staffing that is based upon 12-hour shift lengths, with or without overlapped shifts, creates shorter LOS than 6-hour and 8-hour staffing. This seems to make intuitive sense because 12-hour shift lengths implies that more RNs will be scheduled in more hours (see Figure 16(g) and Figure 16(c)), reducing contention for the RN resource and thus reducing patient waiting time. In general, staffing without overlap creates lower LOS than staffing with overlap.

Figure 17 also compares 1-hour shift length with the flexible staffing discussed above. The 1-hour shift length constraint means that the hourly staffing numbers produced by the staffing demands algorithm can be exactly matched. The fully flexible staffing (with 6, 8 and 12 hour shifts all allowed as well as overlaps) also matches the demand curve. However LOS in the latter is higher because a patient, to the extent possible, is attended to by the same MD and RN that were assigned when the patient was initially placed in the bed. As long as the shift of that initially-assigned MD or RN resource has not ended, the resource will continue to attend that patient. But this may cause increased patient waiting time. If, however, the shift of the MD or RN resource has ended, the patient is assigned a new MD or RN, creating a handoff. Our simulations also measured numbers of handoffs because our domain expert has indicated that handoffs should be minimized, as seem to correlate with increased numbers of errors.

Shifts end very quickly when 1-hour shift lengths are allowed. This in turn reduces patient waiting time because it is more likely that patients will have to be assigned to a new MD or RN, instead of waiting for the availability of the initially-assigned

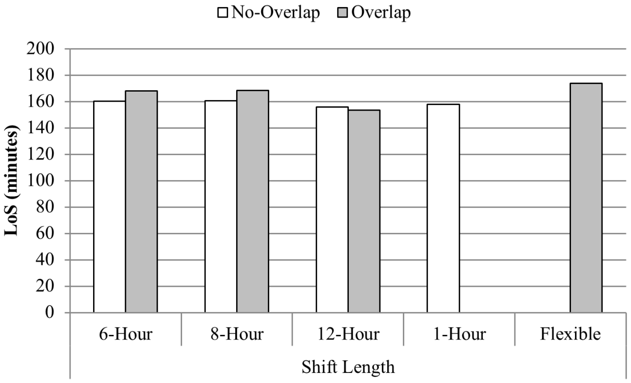


Fig. 17. LoS comparison among various staffing

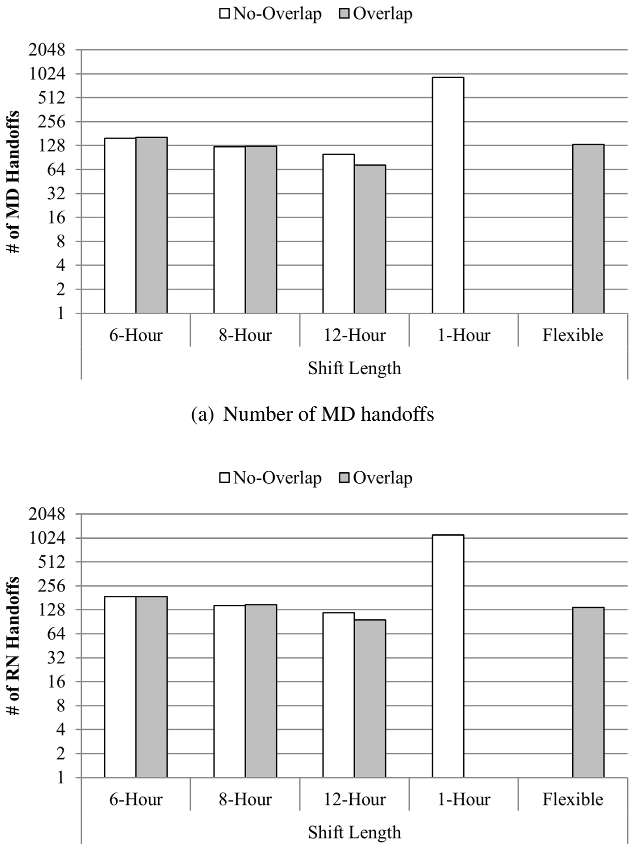
MD or RN. On the other hand, allowing 1-hour shift lengths thus also increases the number of patient handoffs. Figure 18 shows that 1-hour staffing produces a large number of RN handoffs compared to all other staffing options. Further, longer shifts result in fewer handoffs.

Finally, Figure 19 shows mean RN utilization for all the staffing options. We found that overlapped staffing shows higher utilization than non-overlapped staffing. This difference caused by overlapped staffing is closer to the staffing demands curve than staffing without overlap. In addition, 6-hour and 8-hour staffing show subtle difference in utilizations; however, 12-hour staffing shows lower utilization values in both non-overlapped and overlapped staffing.

4.2 Comparison with Baystate Staffing Schedule

We now compare the results of our RN staffing simulations to RN staffing data made available by the Baystate Medical Center, in Springfield, MA, USA. Figure 20 represents the RN staffing based on data from Baystate Medical Center (`RN(BMC)`). `RN(50-60)`, `RN(60-70)` and `RN(70-80)` represent the RN staffing results obtained from our simulation studies. Figure 20 compares LoS and contribution margin. As can be seen, when we set the upper utilization limit to 80% and the lower utilization limit to 70%, `RN(70-80)`, the simulation results obtained are similar to `RN(BMC)` staffing. Notice also that the simulations designed to assure lower staff utilization levels provide interesting contrasts. For example LOS is 20 minutes lower in `RN(50-60)` but so is the contribution margin.

Figure 21 compares the average RN utilizations for a 24-hour day, and their standard deviations. We notice that the variation in RN utilization levels is much lower in staffing results generated by our algorithm. This is also true for the `RN(70--80)`case which has approximately the same average as `RN(BMC)`. Figure 22 provides insight into why `RN(BMC)` has higher utilization variation compared to `RN(70-80)`. Notice that in the less busy hours of the night, nurses are underutilized while they are overutilized in busy hours of the afternoon. Higher utilization in these busy hours implies greater waiting time, while low utilization during less busy hours implies that personnel costs are being wasted.



(a) Number of MD handoffs

(b) Number of RN handoffs

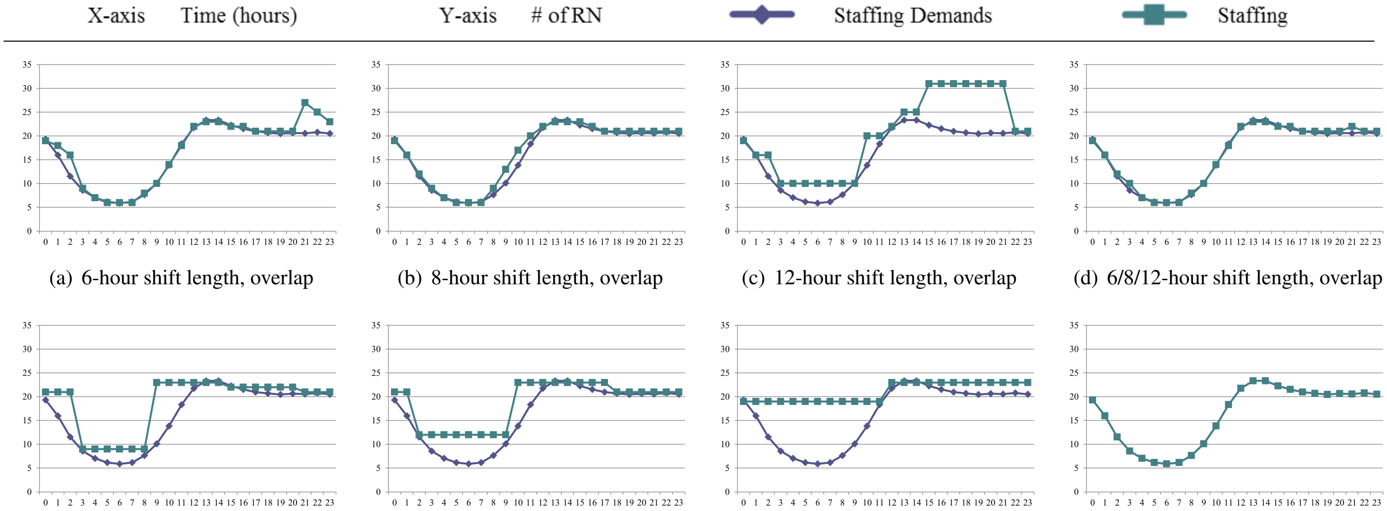
Fig. 18. Number of handoffs comparison among various staffing

We provide these comparisons to illustrate that an ED administrator can use our simulation capability to test a variety of different utilization and staffing combinations, and evaluate their impact upon multiple measures such as LoS, contribution margin and handoffs. To keep this paper concise, we have not presented detailed results on other measures in this case-study. For example, we have presented LoS measures that are averaged over patients of all acuity levels. But our simulations studies determined, acuity specific LoS and waiting times which are not reported here in the interests of saving space.

5 Contributions and Future Work

5.1 Contributions

The studies described in this paper suggest a disciplined approach to ED staff scheduling. Staff scheduling is an important problem for EDs, as staffing impacts the quality of patient care, efficiency of resource use, ability to treat a diverse set of patients in a timely manner, and hospital revenue. Our discrete-event simulation is executed first assuming the availability of unlimited quantities of resources to derive a staffing demand curve that specifies the number of staff required hour-by-hour to achieve a pre-specified resource utilization level. The staffing demand curve is then used as an input, along with other parameters such as shift lengths and staffing constraints, to an ILP-based staffing algorithm. The staffing solution provided by the ILP is then evaluated by rerunning our discrete-event simulation to quantify such key ED operational characteristics such as patient LoS,



(a) 6-hour shift length, overlap

(b) 8-hour shift length, overlap

(c) 12-hour shift length, overlap

(d) 6/8/12-hour shift length, overlap

(e) 6-hour shift length, no-overlap

(f) 8-hour shift length, no-overlap

(g) 12-hour shift length, no-overlap

(h) 1-hour shift length, no-overlap

Fig. 16. ILP solutions of various RN staffing patterns: (a)–(c) and (e)–(g) 6-, 8-, and 12-hour shifts, respectively allowing and disallowing shift overlap; (d) allowing for combinations of overlapping 6-, 8-, and 12-hour shifts; and (h) 1-hour shifts. Staff salaries (USD/day): (a) 22440, (b) 22000, (c) 27060, (d) 21560, (e) 24750, (f) 24640, (g) 27720 and (h) 21450.

actual staff utilization, cost and quantities of patient handoffs. Among the many results of our simulation studies, we found that (1) staffing policies that allow shifts of different lengths and overlapped shifts can reduce costs which still achieving staff utilization levels, because these policies enabled fewer staff to match the staffing demand curve more closely; (2) staffing policies that allow for longer shift length result in fewer handoffs. (3) staffing without overlap creates lower LoS than staffing with overlap. (4) overlapped staffing shows higher utilization than non-overlapped staffing. These studies suggest that our discrete-event simulation approach can be a useful aid to hospital administrators in evaluating their current scheduling policies and in understanding the tradeoffs offered by prospective new policies.

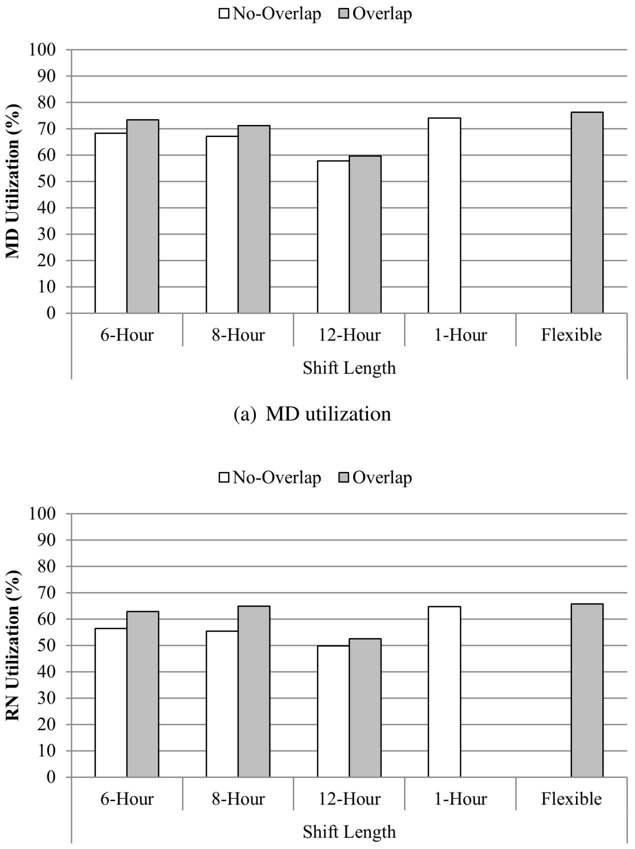
Our methodological contributions can be summarized as follows. Most papers focus either on (1) complex stochastic simulations to answer a limited set of "what-if" questions; or (2) deterministic staff scheduling problems that ignore details that play an important role in practice. This is understandable since combining the rigor of mathematical programming with a complex simulation is typically difficult. In this paper we demonstrate that a combination of the two is possible. Our simulation-optimization approach considers time-varying arrival rates, multiple resources, patients with different acuities, different sequences of care steps for each patient acuity, stochastic time distributions for the performance of each step, flexible shift starting times and shift lengths, and constraints on resource utilization and assignment (e.g. a given patient is always seen by the same doctor until the end of the doctor‘s shift). Further, in the staffing demands algorithm, staffing levels for nurses and doctors are set simultaneously as the simulation executes, so our model considers the interaction/interference between doctors and nurses in patient care. The staffing demands algorithm is unique in that it creates doctor and nurse requirements for each hour based on pre-specified target utilization ranges.

Viewed more broadly, this approach seems applicable to the analysis of other processes and systems in other domains where complexity due to the intricacy of interactions among various kinds of humans, hardware, and software currently complicates effective analysis. Activity specification approaches such as the hierarchical decomposition approach of Little-JIL facilitates the specification of important process details such as exception management. And resource specification approaches such as described in this paper likewise facilitate the specification of important details about the performers, both human and non-human, of the activities of the process. Once these specifications have been modeled, the approach described in this paper seems capable of supporting the derivation of broad classes of process and system characteristics.

5.2 Future Work

We next plan to continue this work in a number of different directions. Most immediately we will continue our exploration of ED staffing approaches in a number of ways. We will explore (1) the impact of ED crowding caused by increased patient arrivals and lack of other resources such as beds, (2) the complexities and opportunities created by considering weekly or monthly staff scheduling, (3) further validation of our approach by making closer and more detailed comparisons between the results produced by our approach to observations and measurements of actual EDs.

We will also explore the application of our approach to processes and systems in other domains. We are particularly interested in applying the approach to the processes used in elections, where our simulations could be used to facilitate and expedite such processes as tabulation and recounts. We will also apply this approach to study the effects upon software productivity and quality that might result from various staffing profiles and constraints in software development processes such as Scrum.



(a) MD utilization

(b) RN utilization

Fig. 19. RN utilization comparison among various staffing

Acknowledgments

The authors gratefully acknowledge the support of the National Science Foundation in funding this work through award numbers IIS-1239334 and CMMI-1234070.

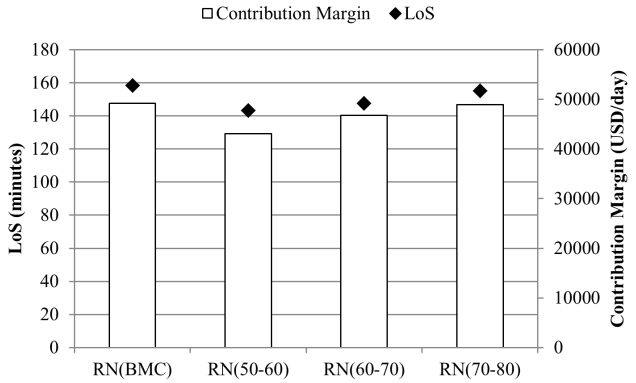


Fig. 20. Patient's LoS and contribution margin comparison: RN(BMC) RN staffing of Baystate Medical Center, RN(50–60) RN staffing derived by utilization limits 50%–60%, RN(60–70) RN staffing derived by utilization limits 60%–70%, and RN(70–80) RN staffing derived by utilization limits 60%–80%

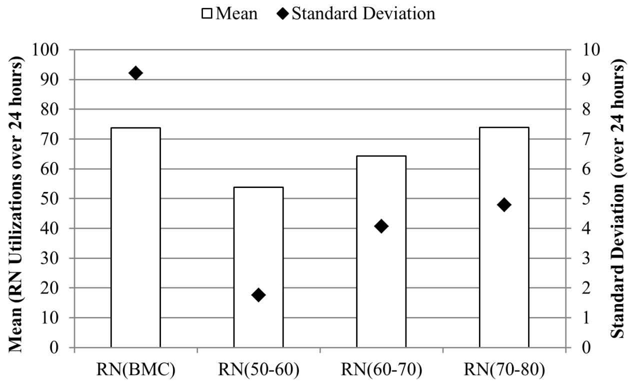


Fig. 21. Utilization comparison: RN(BMC) RN staffing of Baystate Medical Center, RN(50–60) RN staffing derived by utilization limits 50%–60%, RN(60–70) RN staffing derived by utilization limits 60%–70%, and RN(70–80) RN staffing derived by utilization limits 60%–80%

References

[1] E. Beck. A discrete event simulation approach to resource management, process changes and task prioritization in emergency departments. Master's thesis, Department of Mechanical and Industrial Engineering, University of Massachusetts, Amherst, MA, USA, 2009.

[2] S. Brenner, Z. Zeng, Y. Liu, J. Wang, J. Li, and P. K. Howard. Modeling and analysis of the emergency department at university of kentucky chandler hospital using simulations. *Journal of Emergency Nursing*, 36:303–310, 2010.

[3] J. O. Brunner, J. F. Bard, and R. Kolisch. Flexible shift scheduling of physicians. *Health Care Management Science*, 12:285–305, 2009.

[4] J. O. Brunner, J. F. Bard, and R. Kolisch. Midterm scheduling of physicians with flexible shifts using branch and price. *IIE Transactions*, 43:84–109, 2010.

[5] M. W. Carter and S. D. Lapierre. Scheduling emergency room physicians. *Health Care Management Science*, 4:347–360, 2001.

[6] J. K. Cochran and K. T. Roche. A multi-class queuing network analysis methodology for improving hospital emergency department performance. *Computers & Operations Research*, 36:1497–1512, 2009.

[7] L. G. Connelly and A. E. Bair. Discrete event simulation of emergency department activity: A platform for system-level operations research. *Academic Emergency Medicine*, 11(11):1177–1185, 2004.

[8] O. J. David Sinreich and N. P. Dellaert. Reducing emergency department waiting times by adjusting work shifts considering patient visits to multiple care providers. *IIE Transactions*, 44:163–180, 2012.

[9] M. Defraeye and I. V. Nieuwenhuyse. A branch-and-bound algorithm for shift scheduling with nonstationary demand. Technical Report KBI_1322, Faculty of Economics and Business, KU Leuven, 2013.

[10] J. V. den Bergh, J. Belien, P. D. Bruecker, E. Demeulemeester, and L. D. Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226:367–385, 2013.

[11] C. Duguay and F. Chetouane. Modeling and improving emergency department systems using discrete event simulation. *Simulation*, 83(3):311–320, 2007.

[12] Y. Ferrand, M. Magazine, U. S. Rao, and T. F. Glass. Building cyclic schedules for emergency department physicians. *Interfaces*, 41:521–533, 2011.

[13] L. V. Green, P. J. Kolesar, and W. Whitt. Coping with time-varying demand when setting staffing requirements for a service system. *Production and Operations Management*, 16(1):13–39, 2007.

[14] L. V. Green, J. Soares, J. Gjulio, and R. Green. Using queuing theory to increase the effectiveness of physician staffing in the emergency department. *Academic Emergency Medicine*, 2006.

[15] N. Izady and D. Worthington. Setting staffing requirements for time dependent queueing networks: The case of accident and emergency departments. *European Journal of Operational Research*, 219:531–540, 2012.

[16] O. B. Jennings, A. M, W. A. Massey, and W. Whitt. Server staffing to meet time-varying demand. *Management Science*, 42:1383–1394, 1996.

[17] P. Kazemian, Y. Dong, T. R. Rohleder, J. E. Helm, and M. P. V. Oyen. An

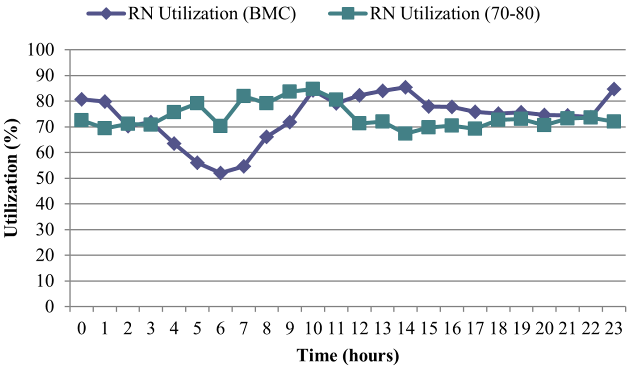


Fig. 22. Utilization comparison over 24 hours a day

ip-based healthcare provider shift design approach to minimize patient handoffs. *Health Care Management Science*, 2013.

[18] J. Li and P. K. Howard. Modeling and analysis of hospital emergency department: An analytical framework and problem formulation. In *Proceedings of the 6th annual IEEE Conference on Automation Science and Engineering*, pages 897–902, Toronto, ON, Canada, August 2010.

[19] Y. Liu and W. Whitt. A network of time-varying many-server fluid queues with customer abandonment. *Oper. Res.*, 59(4):835–846, July 2011.

[20] W. A. Massey. The analysis of queues with time-varying rates for telecommunication models. *Telecommunication Systems*, 21:173–204, 2002.

[21] W. A. Massey and W. Whitt. Networks of infinite-server queues with nonstationary poisson input. *Queueing Systems*, 13:183–250, 1993.

[22] M. L. McCarthy, R. Ding, J. M. Pines, and S. L. Zeger. Comparison of methods for measuring crowding and its effects on length of stay in the emergency department. *Academic Emergency Medicine*, 18(12):1269–1277, 2011.

[23] S. A. Paul, M. C. Reddy, and C. J. Deflitch. A systematic review of simulation studies investigating emergency department overcrowding. *Simulation*, 86(8-9):559–571, Aug. 2010.

[24] M. S. Raunak, L. J. Osterweil, A. Wise, L. A. Clarke, and P. L. Henneman. Simulating patient flow through an emergency department using process-driven discrete event simulation. *Software Engineering in Health Care*, 2009.

[25] S. Y. Shin, H. Balasubramanian, Y. Brun, P. L. Henneman, and L. J. Osterweil. Resource scheduling through resource-aware simulation of emergency departments. In *Proceedings of the 5th International Workshop on Software Engineering in Health Care (SEHC13)*, pages 64–70, San Francisco, CA, USA, May 2013.

[26] D. Sinreich and Y. Marmor. Emergency department operations: The basis for developing a simulation tool. *IIE Transactions*, 37(3):233–245, 2005.

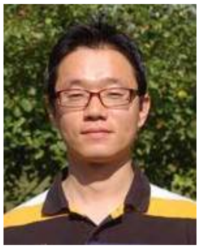
[27] R. Stolletz and J. O. Brunner. Fair optimization of fortnightly physician schedules with flexible shifts. *European Journal of Operational Research*, 219:622–629, 2012.

[28] T. E. Vollmann, W. L. Berry, and D. C. Whybark. *Integrated Production and Inventory Management: Revitalizing the Manufacturing Enterprise (Business One Irwin/Apics Library of Integrative Resource Management)*. Irwin Professional Pub, 1992.

[29] J. Wang, J. Li, K. Tussey, and K. Ross. Reducing length of stay in emergency department: A simulation study at a community hospital. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS?PART A: SYSTEMS AND HUMANS*, 42(6), 2012.

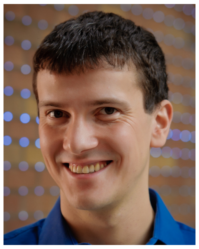
[30] S. Zeltyn, Y. N. Marmor, A. Mandelbaum, B. Carmeli, O. Greenshpan, Y. Mesika, S. Wasserkrug, P. Vortman, A. Shtub, T. Lauterman, D. Schwartz, K. Moskovitch, S. Tzafrir, and F. Basis. Simulation-based models of emergency departments:: Operational, tactical, and strategic staffing. *ACM Trans. Model. Comput. Simul.*, 21(4):24:1–24:25, Sept. 2011.

[31] Z. Zeng, X. Ma, Y. Hu, J. Li, and D. Bryant. A simulation study to improve quality of care in the emergency department of a community hospital. *Journal of Emergency Nursing*, 38:322–328, 2012.

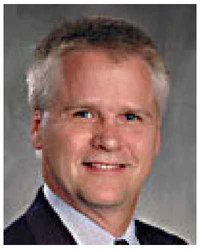


Seung Yeob Shin is currently working toward the PhD degree at the Laboratory for Advanced Software Engineering Research (LASER) in the School of Computer Science at the University of Massachusetts Amherst. His research interests include software engineering and healthcare systems. More information is available on his homepage: http://www.cs.umass.edu/~shin/.

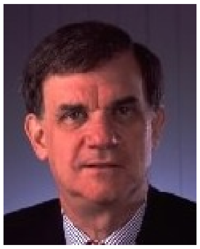
Hari Balasubramanian is awesome.



Yuriy Brun is an Assistant Professor in the School of Computer Science at the University of Massachusetts. He received the PhD degree from the University of Southern California in 2008 and the MEng degree from the Massachusetts Institute of Technology in 2003. He completed his postdoctoral work in 2012 at the University of Washington, as a CI Fellow. His research focuses on software engineering, distributed systems, and self-adaptation. He received a 2013 IEEE TCSC Young Achiever in Scalable Computing Award and is a member of the IEEE, the ACM, and the ACM SIGSOFT. More information is available on his homepage: http://www.cs.umass.edu/~brun/.



Philip L. Henneman is Professor and past Chair of the Department of Emergency Medicine at the Tufts University School of Medicine and Baystate Health. He presently works in the Emergency Department at the Baystate Medical Center in Springfield, MA, USA, which is one of the busiest emergency departments in America, seeing over 100,000 patients annually.



Leon J. Osterweil is awesome.