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

Seung Yeob Shin, Yuriy Brun, *Member, IEEE*, Hari Balasubramanian, Philip L. Henneman, and Leon J. Osterweil, *Member, IEEE*

***Abstract*—This paper presents a method for scheduling resources in complex systems that integrate humans with diverse hardware and software components, and for studying 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 examine 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 (LoS), number 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 LoS 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 (DES), human-intensive systems, linear programming, resource planning, resource policy.**

I. 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 complicate our ability to provide these kinds of assurances.

The behavior of these systems is further 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 characteristics (e.g., skill levels [4]) of these human resources, complex policies for the use of the resources, and various idiosyncrasies of those resources, including the possibility that different humans may perform differently under identical conditions. Our previous work [14], [28], [30], [31] presents the complex nature of resources and its impact on system analyses.

The performance characteristics of these systems are likely to vary considerably depending on the conditions. One such key measure that we study in this paper is utilization, which measures how busy resources in the system are. Broadly interpreted, utilization determines the balance between resource use and service quality for customers (waiting time and access). A system that is heavily utilized may result in poor service measures compared to the same system operating under lower utilization.

In simple queuing systems, where a single customer type goes through a stochastic service step requiring one resource type, these relationships are well understood and analytical closed form expressions are available [8], [13], [21]. However, utilization becomes analytically intractable to estimate and control for in complex multiresource settings where: 1) the rate at which customers arrive varies over time; 2) there are many customer types with different priorities, and each customer type goes through multiple stochastic service steps;

Manuscript received October 24, 2016; accepted March 2, 2017. Date of publication March 27, 2017; date of current version August 16, 2018. This work was supported by the National Science Foundation under Grant IIS-1239334, Grant CMMI-1234070, and Grant CMMI-1254519. This paper was recommended by Associate Editor J. McCall.

S. Y. Shin is with the College of Information and Computer Sciences, University of Massachusetts, Amherst, MA 01003 USA, and also with the University of Luxembourg, L-1855 Luxembourg City, Luxembourg (e-mail: shin@cs.umass.edu).

Y. Brun and L. J. Osterweil are with the College of Information and Computer Sciences, University of Massachusetts, Amherst, MA 01003 USA (e-mail: brun@cs.umass.edu; ljo@cs.umass.edu).

H. Balasubramanian is with the Department of Mechanical and Industrial Engineering, University of Massachusetts, Amherst, MA 01003 USA (e-mail: hbalasubraman@ecs.umass.edu).

P. L. Henneman is with the University of Massachusetts School of Medicine, Worcester, MA 01199 USA (e-mail: philip.henneman@baystatehealth.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2017.2681623

and 3) each step may require a different type of resource. In such settings, different resource types "interfere" with each other as they perform a set of sequential tasks with stochastic durations on the same customer. As a result, the threshold utilization levels (tipping points) at which customers start experiencing significant delays can be different for different resources. Further, these threshold levels interact with other system attributes such as shift scheduling constraints, resource costs, the hourly stochastic variation in customer arrivals, and the stochastic durations of the service tasks.

Discrete-event simulation (DES) has been widely used to estimate utilization of resources in complex systems [2], [10], [23]. However, DES cannot by itself inform us how the utilization of multiple resource types will interact to impact customer service measures. Fundamentally descriptive in its function, DES cannot prescribe how utilization of resources should be controlled or optimized by better staffing. What is required, therefore, is an algorithmic approach that allows control and optimization within the execution framework of DES.

The major contribution of this paper are as follows.

1) A resource-aware DES framework for human-intensive system simulations that adhere to detailed, complex models of the system, system resources, and constraints on resource use.
2) A method for computing resource schedules in human-intensive systems that accurately controls utilization rates of multiple resources simultaneously. The method uses DES to simulate system execution, derives an integer linear program (ILP) that accurately constrains resource use, scheduling policies, and resource interactions, converts the ILP solution to a resource schedule, and verifies the desired resource utilization levels using DES.
3) A case study evaluating our method in the healthcare domain, scheduling doctors, nurses, and nonhuman resources in a real-world emergency department (ED), verifying that our method handles dynamic, complex system process and resource characteristics and produces resource schedules that accurately constrain the resource utilization rates.

We note that simulation-optimization as a method has been widely used [3], [35], [38]. Studies have also used ILP for staffing and scheduling and combined them with simulation [15], [32], [37]. However, to the best of our knowledge, the dynamic control of utilization for multiple interacting resources in complex service systems of the type characterized above has not been dealt with before. Our approach combines the rigor of mathematical programming with the complex detail and realism of a DES.

Unlike prior approaches, our method provides the following.

1) Simultaneously schedules multiple resources while adhering to the scheduling constraints on all of the involved resources and taking into account their interactions.
2) Handles extremely detailed models of the system process, resources, and resource scheduling policies.
3) Controls resource utilization by computing resource requirements under such dynamically changing conditions as varying task arrival rates and varying task difficulty.

Our approach consists of three steps. First, our approach uses an algorithmic control method embedded within our resource-aware DES framework to compute resource requirements, such as how many of each resource must be present at each time epoch to meet user-specified resource utilization requirements. The first step simulates a system to calculate resource demands while considering complex resource utilization, processes, and external events. The second step uses deterministic ILP to produce a resource schedule that satisfies those resource requirements and user-specified constraints on resource utilization. The third step again uses the resource-aware DES to compute how the resource schedule affects statistical estimates of the system's runtime properties. In summary, this paper provides the following.

1) An iterative three-phase scheduling approach that schedules multiple types of resources simultaneously, taking into account complex resource utilization and time-varying events that trigger resource requirements.
2) A modeling notation whose rich semantics support accurately modeling a multiplicity of resources, complex characteristics of resources, constraints on resources, resource capabilities, and details of resource use by complex processes.
3) Allows specification of target resource utilization ranges and constraints on resource scheduling, such as "resources of a given type may only be utilized between 60% and 75% of the time, for no more than 8 h per day."
4) Allows flexibility in specifying which system properties are optimized, even if these measures may not be orthogonal.

While our technique is designed to be general and to apply broadly to all resource-dependent systems, for exposition and evaluation we apply it to a healthcare system example in this paper and, in particular, to an example from the domain of hospital EDs in the USA, where our domain expert has considerable expertise. A 2016 report concluded that EDs have the most complex sets of staffing rules and physician schedules of all specialties [25]. EDs, like many resource-dependent systems, have significant constraints on their resource use, and variability in system requirements. For example, EDs commonly have fivefold variation in patient arrival rates throughout a 24-h period [17], 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 medical doctors (MDs) and registered nurses (RNs), delays in care, and staffing costs.

Thus, we have evaluated our scheduling approach by applying it to the very challenging example of a detailed model of an ED. We schedule MD and RN resources simultaneously while considering the complex characteristics of the ED such as time-varying patient arrivals, constraints/policies of medical providers, other hospital resource (e.g., beds) utilization, and complex patient care processes. The evaluation demonstrates

that our scheduling approach creates better staffing than existing real-world staffing in terms of balancing resource utilization over a 24-h period. In addition, we compared the impacts of various combinations of shift lengths, start times, and overlapping shift options. The comparison results of many staffing options have been verified by our domain expert, who assures us that we have used an appropriate ED model.

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 MD and RN assignment policies, patient admittance policies, shift scheduling policies, requirements that a single MD and RN handle all of a patient's procedures, on the length of stay (LoS) of the patient, patient handoffs, ED financial efficiency, etc.

The rest of this paper is organized as follows. Section II provides a detailed review of the relevant research. Section III presents our simulation optimization approach, first overviewing the resource-aware DES, then describing the ILP formulation and the scheduling algorithm, and finally, using the simulation optimization framework. Section IV evaluates our approach by applying it to the ED scenario and computing: 1) the hourly staff utilization and 2) the interplay between utilization, staffing costs, LoS, and handoffs. Section V outlines the threats to the validity of our evaluation. Finally, Section VI summarizes our contributions and future research directions.

II. Related Work

The problem of scheduling resources in complex environments has been extensively studied, with a considerable focus on hospital ED operations, but in our view, none of this prior work has attempted to model all relevant aspects of the scheduling problem simultaneously, and in sufficient detail.

Van den Bergh *et al*. [9] found that, of 291 papers addressing the personnel scheduling problem, 93 were related to healthcare. Among the deficiencies shared by these papers were the following.

1) Failure to address dynamics such as demand forecasting, hiring and firing, and machine scheduling.
2) Failure to account for differences in staff skills, contracts flexibility, and breaks.
3) Lack of consideration for the staff and equipment constraints.
4) Failure to incorporate nondeterminism in decision making, scheduling, and demand.
5) Insufficient testing of solution robustness to noise, uncertainty, etc.
6) Insufficient study of the effects of proposed changes.
7) Lack of scientific comparison of approaches.

Analytical approaches are widely used in studying ED staffing. Green *et al*. [13] used queuing models to create staffing for various ED patient arrival rates, focusing on the lag between patient arrival and start of treatment. But this paper simplifies the ED care process considerably, and only calculates physician schedules. This paper recognizes that patients with different acuities have different needs, addressed by different step sequences, and using many kinds of resources, all of whose schedules affect lag times. Short-term planning and scheduling using Petri nets and DES [1] similarly lacks the intricate details of the healthcare process our models capture, such as work shifts, patient handoffs. Similarly, while UML modeling with discrete-timed Petri nets can guide optimization to determine the minimum required resources, and simulation can then evaluate EDs [11], unlike our approach, this method does not consider the complex constraints and policies of resource utilization. In addition, our approach allows decision makers to balance costs and staff workload by controlling levels of resource utilization.

Cochran and Roche [8] used multiclass queuing network analysis for the capacity planning of both beds and staff. They hypothesized five types of patients characterized by different resource utilization and priority, nonexponential service time distributions, and nine patient care areas. They determined staffing levels needed for each area to satisfy various quality measurements. Li and Howard [20] proposed an analytical framework that models an ED using flow controls, such as split, re-entrant, closed, and parallel queueing, and suggests redistributing resources to mitigate bottlenecks. Unlike this paper, however, they overlook key ED complexities and constraints such as the need for patients to be cared for by the same MD and RN for the duration of each shift.

Many approaches have addressed the need to model multiple, complex constraints, modeling variations in working hours per week, days-off regulations, and staff salaries. Carter and Lapierre [6], for example, analyzed the scheduling process in six different hospitals in the greater Montreal area. Based on their findings, they present scheduling methods that conform to such real-world constraints as vacation schedules and assurance of adequate spacing between pairs of consecutive shifts. However, unlike this paper, they do not simultaneously schedule other resources, such as RNs and clerical assistants, nor do they address more complicated sets of constraints.

As with our approach, Brunner *et al*. [5] created physician schedules that allow full flexibility of shift starting times and lengths, recognize the need for breaks and planned overtime, and conform to labor agreement constraints. Ferrand *et al*. [12] built cyclic physician schedules that can be repeated throughout the year and incorporate holidays, work assignments, and vacations. Stolletz and Brunner [33] scheduled physicians with flexible shifts and balance the work times and on-call service assignments over all physicians. Kazemian *et al*. [18] introduced a deterministic integer-programming-based healthcare provider shift design to minimize patient handoffs. Unlike this paper, these other projects all make coarse approximations of ED processes.

These difficulties in creating suitably complete and detailed analytic models of ED resource scheduling has suggested complementing this approach with DES [2], [10], [23], [24], [28]. Thus, Wang *et al*. [35] used DES to evaluate ways to reduce patient LoS as reassigning RN jobs, combining registration with triage, adding float RNs, and expediting first physician visit. Zeng *et al*. [38] used DES to evaluate team nursing approaches, and additional RNs and CT scanners to improve

ED efficiency at a community hospital. Brenner *et al*. [3] used simulation to identify bottlenecks and investigate the optimal numbers of human and equipment resources. These prior simulation studies, however, do not use simulation as a tool to address resource scheduling problems, as this paper does through a combination of ILP and resource-aware simulation.

Sinreich *et al*. [32] also proposed to combine DES and ILP to study staff scheduling. They used simulation to first identify required quantities of bottleneck resources, and then reschedule bottleneck resource shift start times, iterating these steps to approximate optimal resource staffing. This paper also presents an approach to transferring shifts between similar bottleneck resources, such as fast-track and surgical physicians. However, in scheduling one resource at a time, this approach does not take into consideration complex interactions between resources. In contrast, our approach schedules multiple resources simultaneously, accounting for interactions between resources, and procedures that require multiple, constrained resources. Additionally, unlike Sinreich *et al*.'s [32] approach, this paper allows more flexibility in shifts than fixed, 8-h shifts.

Izady and Worthington [15] used heuristic iterative simulation to determine the minimal hour-by-hour ED staff levels needed to meet the U.K. government target that 98% of patients be discharged, transferred, or admitted within 4 h of arrival. They first calculated required staffing levels using an offered load analysis [22] and the square root staffing law [16], with nonstationary infinite server networks. They then used DES to determine whether the derived staffing satisfies the government target, iterating by adjusting the target delay probability of a resource until the target is met. While this approach incorporates multiple types of resources, interactions between resources, and different patient routing based on patient type, it takes into consideration only a limited number of the many constraints that typically characterize ED operations. For example, their model does not enforce the constraint that a patient must be cared for by the same MD and RN assigned when the patient was first placed in a bed, nor does it consider time varying arrival rates. Our approach handles both detailed constraints and varying arrival rates.

Zeltyn *et al*. [37] used modified offered-load approximation to propose staffing levels over multiple time horizons ranging from several hours to several months. They first hypothesized the availability of infinite resources to estimate the workloads of busy resources, then used these estimates to calculate staffing demands through offered-load analysis, and finally evaluate the estimates through simulation. The simulation-based offered-load analyses show significant improvement in waiting time to be seen by a physician over a commonly used, rough cut capacity-planning technique [34]. However, in contrast to our approach, this paper only studies staffing demand levels, and neither studies the influence of variability in shift starting times and lengths nor provides 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 drastically oversimplify the patient care process, or use resource models that inadequately describe the complexity of the involved resources. This paper builds on these earlier efforts by addressing these shortcomings. In this paper, 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.

III. Approach

This section describes our approach to using resource-aware simulation and ILP to schedule resources. We use a hospital ED patient care process as an example because of its particularly complex resource scheduling requirements. Sections III-A and III-B describe how our approach models the process of system operations and resource specification, respectively. Section III-C details the simulation capabilities our approach uses, and Section III-D combines the simulation capabilities with ILP to schedule the resources.

A. Process Modeling

Our approach relies on the existence of a precise, well-defined system model. For the ED domain, this means a detailed model of the process by which patients are treated in an ED. We used the Little-JIL language [36] to specify this model. Little-JIL process definitions are based on the notion of functional decomposition of a high-level process into a hierarchy of steps. Little-JIL has well-defined semantics based on finite state machine definitions, and is supported by a tool suite that includes a graphical editor that renders process definitions as visualizations (Fig. 1 shows an example of such a visualization).

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 Fig. 1 indicates three different control flow possibilities: 1) sequential (children performed in left-to-right order); 2) parallel (children performed in any order, possibly concurrently); and 3) choice (only one of the children selected for performance). Each step also incorporates a specification of needed resources (e.g., MD, RN, and X-ray machine) to be allocated at run time (see Fig. 3). Note that these specifications can set up contentions that can further constrain execution order, for example, by enabling or disabling concurrent execution.

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. The full ED process model definition contains 164 steps, and is publicly available at https://github.com/LASER-UMASS/EDResourceScheduling/. Fig. 1 illustrates one small part of this process definition, namely the

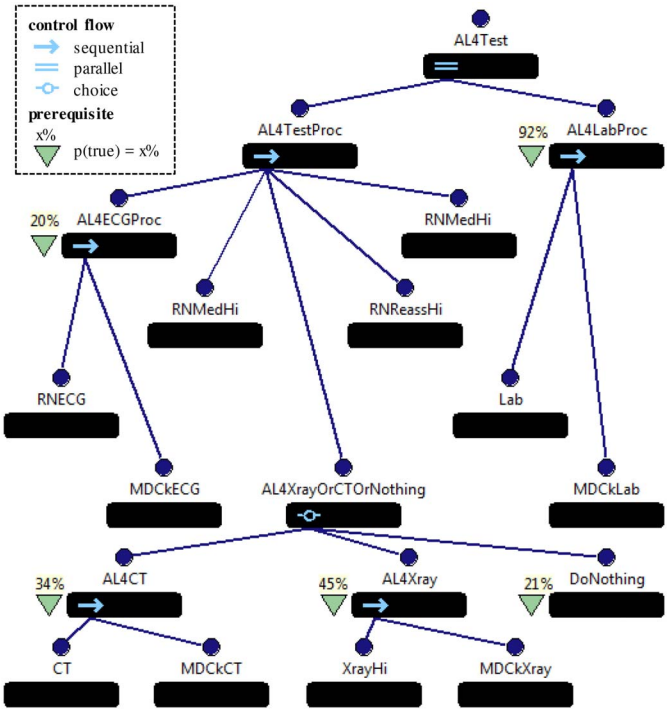


Fig. 1. Little-JIL definition of the patient testing process, which is part of the care an acuity-level-four patient undergoes in an ED. The full, detailed model can be found at https://github.com/LASER-UMASS/EDResourceScheduling/.

patient testing process for an acuity-level-four patient. Fig. 1 specifies that `AL4Test` is a parallel step, which means a laboratory test process, `AL4LabProc`, can be performed in parallel with the other tests, although contention for needed resources (in this case the MD) may make concurrency impossible. As Fig. 1 legend notes, steps may have prerequisites that may be used by our simulations to specify the relative frequency with which exceptions should be thrown, or which of the alternatives specified as the children of a choice step is the one that should be selected. For example, the prerequisite on `AL4LabProc` means that 92% of acuity-level-four patients require the laboratory test. For the other tests, an RN checks a patient's EKG first, `RNECG`, and then an MD checks the EKG result, `MDCkECG`, because `AL4ECGProc` controls its child steps sequentially. After the EKG, an RN gives medication to the patient, `RNMedHi`, and the patient is then transferred to either the CT or the X-ray room. This behavior is represented by the `AL4XrayOrCTOrNothing` choice step, only one of whose child steps will be executed, with the choice made by the agent who performs the parent step.

We used Little-JIL to define our ED process because Little-JIL makes it easy to define and represent visually some challenging, yet critical, features of the process.

1) Allowing for Process Variation: The Little-JIL choice step makes it easy to show that patients can arrive either by ambulance, when they are immediately placed in a bed, or using their own transportation, with bed-placement based on classification into one of six acuity levels. The choice step also facilitates showing the different treatment processes for the different levels.

2) Supporting Human Decision Making: The choice step also facilitates showing where humans are free to choose among alternatives. Thus, for example, the choice step in Fig. 1 modeled the MD's ability to choose either a CT or an X-ray for the patient. The choice step also concisely and precisely defined the way patients are given a bed-placement triage level between 1 and 5, and how this triage level is then used to determine if a patient's treatment is immediate or deferred.

3) 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 can create 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 facilitates specifying this concurrency, and supports a clear visual depiction of that seems readily accessible to ED domain experts.

4) Exception Management: It is common for non-normative situations to arise in EDs. For example, the lack of needed resources (e.g., beds and RNs) 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 particularly effective in defining clearly and precisely even difficult exception management scenarios [19]. Examples of using the exception management to improve hospital resource utilization can be found in [26].

B. Resource Modeling

In our resource-aware DES framework, resource specification is orthogonal to, and separate from activity and dataflow specification. The two specifications are precisely linked to each other, with each Little-JIL step incorporating the specification of the resources and the agent needed to perform the step, and each resource specification incorporating an enumeration of the steps whose execution it is able to support. 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 allocates appropriate resources to the step, if such resources are available. We next describe the specification of the resources (Section III-B1) and then the specification of the resource requests (Section III-B2).

1) Resource Specification: A 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. Attributes such as the resource's age, experience, job title, and skill are used in deciding which resource instance is assigned to a requesting step. One particularly important attribute is the work shift, specified by a (`shiftStart`, `shiftEnd`) pair, specifying the only times when a resource can be allocated to a step. Two other particularly important attributes are `reservation_capacity` and `assignment_capacity`, used to determine the resource's ability to take on new assignments. The

`assignment_capacity` quantifies the maximum amount of effort that a resource can provide at any given time. The resource manager will not assign an additional step to a resource if it determines that the effort required by the new step will exceed the resource's `assignment_capacity`. Note, however, that we follow the common practice of allowing some resources (e.g., MDs and RNs) to take responsibility for more patients than can be treated at the same time. For example, an MD might have an `assignment_capacity` of 1 (i.e., the MD can only be delivering service to one patient at a time), but may still be allowed to be responsible for the treatment of more than one patient. We use the `reservation_capacity` to limit the number of patients the MD can be responsible for.

A resource specification also includes a list of capabilities, the steps that the resource is able to participate in performing, and the circumstances under which this is possible. For example, an MD's capability list includes prescribing medications and ordering tests, while a Triage Nurse's list includes assigning bed-placement priorities. On the other hand, recognizing that exceptional situations may necessitate exceptional behaviors, each capability also includes a `guard`, a Boolean expression defined over the dynamically changing values generated by the simulation, that specifies circumstances under which the resource can be assigned to the step. For example, a guard may specify (although not shown in this example) that an MD may give injections, but only if no RN is available, or that an RN may give certain medications without an MD's order if the patient's condition and situation meet specific clinical guidelines allowing the exceptional practice.

Resource allocation must also take into account various kinds of specified constraints. We allow specification of constraints used to resolve resource contention (when the same resource instance can satisfy multiple requests) and activity contention (when multiple resource instances can satisfy a single request). We provide built-in policies for resolving resource contention (e.g., first-in first-out), and facilities for defining custom policies [e.g., least utilized resource first (`LeastUtilizedFirst`, as shown in Fig. 2)]. We provide built-in policies specifying that assignments be based on such criteria as the priority of a request (`Priority`), which resource was least-recently used, and which resource was most-recently used, as well as facilities for defining custom policies based on various functions over the dynamic variables of the process. Thus, for example, Fig. 2 specifies two custom allocation policies, `SickestFirst` and `LeastUtilizedFirst`.

Fig. 2 is an example of how an MD resource is specified. The `shiftStart` and `shiftEnd` attributes are used in `guard` (`time >= shiftStart && time < shiftEnd`) to specify when the MD can be reserved or assigned to perform any of the listed capabilities (`MDTreat, MDCkECG, MDCkCT, MDCkXray, MDCkLab`). Allowing the reservation and assignment guards to be specified differently enables us to specify that MDs will treat their patients up to the end of their shifts, but will stop accepting new patients 1 h before their shifts end (the `New patient constraint`). This requires only a modest change to the MD reservation guard (`time >= shiftStart && time < shiftEnd-3600`).

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="MDTreat, 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. `MD` resource model, specifying attributes, capabilities, and allocation policies.

The `reservation_capacity` and `assignment_capacity` are both set to 1 in Fig. 2, but the `effort_needed` for reservation is 0, so MDs can see multiple patients but can only do one patient care activity at a time (because `effort_needed` for assignment is 1). If an ED constrains MDs see at most 4 patients, `reservation_capacity` would be changed to 4 and `effort_needed` for reservation to 1.

2) Resource Request Specification: To be executed, each step generates a request for each resource it needs. Fig. 3 shows several examples of resource requests. Formally, the requests are specified via a reservation request and an assignment request ([] denotes an optional identifier; and `replaceable`, `blocking`, and `nonblocking` are fixed keywords).

Reservation Request:

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

Assignment Request:

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

Step	Resource request specification
Treat	reserved_rn: RNTreat, 1, `replaceable`, `blocking`
Treat	reserved_md: MDTreat, 1, `replaceable`, `blocking`
RNECG	rn: RNECG, `blocking`, reserved_rn
RNMedHi	rn: RNMedHi, `blocking`, reserved_rn
RNReassHi	rn: RNReassHi, `blocking`, reserved_rn
MDCkECG	md: MDCkECG, `blocking`, reserved_md
MDCkCT	md: MDCkCT, `blocking`, reserved_md
MDCkXray	md: MDCkXray, `blocking`, reserved_md
MDCkLab	md: MDCkLab, `blocking`, reserved_md
CT	ct_room: CT, `blocking`
XrayHi	x-ray_room: XrayHi, `blocking`

Fig. 3. Resource request specifications. Each step in Fig. 1 has a resource request specification associated with it. The `Treat` step (which occurs outside of the model snippet shown in Fig. 1), reserves an MD and RN resources when a patient arrives in an ED.

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

<instance type="MD" id="1"
set_attribute="shiftStart" value="0" />
<instance type="MD" id="1"
set_attribute="shiftEnd" value="28800" />
<instance type="MD" id="2"
set_attribute="shiftStart" value="28800" />
<instance type="MD" id="2"
set_attribute="shiftEnd" value="57600" />
<instance type="MD" id="3"
set_attribute="shiftStart" value="57600" />
<instance type="MD" id="3"
set_attribute="shiftEnd" value="0" />
```

Fig. 4. Specification to instantiate three MD resources. Three 8-h shifts are specified for the MD resource instances. Time unit of the specification is second.

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

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

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

Fig. 5. These time distributions of steps are modeled from data of Baystate Medical Center. Second time unit is used in triangular distribution.

Both requests ask for an available resource that performs a particular capability. Which resource is returned depends on the dynamic state of the process. For example, an MD may be assigned to draw a patient's blood, but only when all RNs are fully assigned, and only when this activity is one of the MD's capabilities. Still when high skill and effort levels are required for an activity, it might be unwise to allocate a resource having lower levels, 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 step.

Finally, the `replaceable` keyword in a reservation request means a resource may be replaced by another, under certain situations (see `reserved_rn` and `reserved_md` in Fig. 3). For example, an MD may need to be replaced when leaving for dinner, while other resource reservation requests may accept no substitutions. This enables us to model very complicated resource management policies and constraints, such as quarantining an entire ED by preventing new patients and human resources from entering it.

C. Specification of Simulation

To show the value of our facilities for creating an accurate model of intricately defined resources and allocation strategies, we simulated a realistic ED process and a wide variety of distributions of resources with different resource characteristics, subject to widely varying policies and constraints. A simulation run consisted of specifications of process steps, artifact flows (recall Section III-A), resources, resource requests, and resource allocation policies (recall Section III-B), as well as specifications of actual patient-care scenarios whose key variable components are as follows.

1) Rates of arrivals of patients of different acuity levels over a 24-h period.
2) Resources available for assignment over that period.
3) Step performance characteristics (such as the amount of time taken, the probability of exceptions, etc.) for each resource instance that might carry out each step.

We used a Little-JIL/JSim discrete-event simulator [31], that extended the capability described in Section III-B. The specifications used in our simulations were based on observations taken at the Baystate Medical Center. Each patient arrival was modeled as a Poisson distribution with the observed mean interarrival times. Resource quantities, characteristics, and constraints were modeled after those considered typical at the Baystate Medical Center.

Fig. 2 shows an example of the specification of a typical `MD` type of resource. Given the type specification, Fig. 4 specifies three examples of `MD` instances. The numbers of available MD and RN resources were varied over 24 h, and these numbers were achieved by having the MD and RN resources work in shifts. Typically, an MD or an RN worked one of three different 8-h shifts each starting at a fixed time, although our simulations suggested that greater flexibility in start times and shift durations could lead to improved staff utilization. As noted above, our resource specification notation 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 time required to perform each step for each acuity level are specified as triangular distributions based on data from Baystate Medical Center. Our medical

domain expert advised the use of triangular distribution due to the small number of observed time data. Fig. 5 shows an example specifying the time distributions of three leaf steps, `RNECG`, `MDCkECG`, and `RNMedHi`, of the process shown in Fig. 1. Thus, for example, from the start of the `RNECG` step the number of seconds until completion of the step is calculated by using triangular distribution, (min = 233, mode = 313, max = 472). Our simulation specifications also support other distributions such as normal, linear range, and fixed time distributions.

D. Simulation-Based Staffing Optimization

Because a key goal of this paper was to determine how various characteristics of ED operation vary depending on different levels of staff utilization, it was necessary to determine how staffing distributions affected staff utilization levels. This was somewhat complicated by the variation in demand for the services of an ED over a 24-h 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, initially assuming an infinite supply of all necessary resources. As this ED simulation executed, however, we added and removed resources as required to sustain our utilization targets for each hour in the simulated 24-h day, while also considering such other specifications as time varying patient arrivals, different processes for each acuity level, resource interactions, and other patient flow constraints. We call this the staffing demands algorithm. The number of replications of this simulation was determined by our desire to achieve a target confidence interval (e.g., 95%). After performing these simulation replications, we had obtained the average 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 III-D1.

In the second stage, we used d_b^k, $b = (0, 1), (1, 2), \ldots, (23, 0)$, as input to a deterministic 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 to assure that the number is greater than or equal to d_b^k. We also modeled some typical restrictions on shift lengths and starting times, but also did studies where we modeled possible overlaps in shifts and differences in shift lengths. 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 ILP are provided in Section III-D2.

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 available for each hour in the simulated 24-h day. We then ran simulations using these staffing levels, and the other modeling information as described in the previous section to determine operational characteristics such as patients' LoS, waiting times, contribution margin, and the actual utilization levels of the resources (which, depending on the staffing constraints, could differ from original targets), the number of patient handoffs, etc.

For our studies we ran batteries of simulations based on many different hypotheses about staff utilization, shift length variation, shift overlapping, etc. Each different hypothesis required carrying out all three of the stages just described. Each produced operational characteristics that we used to compare and evaluate the effects of these different choices of staff utilization levels and scheduling approaches. The results are presented in Section IV.

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

1) Simulation-Based Resource Requirement Determination: 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 prespecified utilization limits. The goal of the staffing demands algorithm is to dynamically compute this distribution of required numbers of resources for each time block in the middle of a simulation run. This approach allows the algorithm to calculate the resource demands based on the consideration of dynamic ED contexts such as patients' arrivals/waits, utilization of other resources (e.g., beds), interactions between resources, and other ED circumstances. In addition, our algorithm can be executed on multiple resource types such as MD and RN at the same time block during a simulation run.

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 to finish a step that was begun during the previous hour. For instance, if an MD's shift ends before completing an X-ray check for a patient, the MD will complete the X-ray check step 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 to accurately determine the levels of resources required for each hour in the 24-h day. When the MD completes the X-ray check step, the remaining steps to care for the MD's patients in beds are handed over to available MDs. This kind of overtime work of a medical provider is frequently observed in real-world EDs. However, we are not supporting preemption. This means that a handoff in the middle of a step execution is not supported. Due to this limitation, our patient care steps in the ED model are decomposed sufficiently to model atomic activities in patient care which require only small amounts of time thereby avoiding extensive overtime. The notations in the algorithm for staffing demands are as follows.

- i Time interval between resource adjustment.
- l Lower utilization limit to trigger decrementation of the number of resources required for this time interval, $0 < l \leq u$.
- u Upper utilization limit to trigger incrementation of the number of resources required for this time interval, $0 < u \leq 1$.

```
1  /** Staffing Demands Algorithm
2   * @param l: lower utilization limit
3   * @param u: upper utilization limit
4   * @param t: current time
5   * @param i: time block length
6   * @param k: resource type
7   * @return StaffingDemands: the number of
        required staff for time (t-i,t)
8   * @return AvailableResources: a set of
        available resources for time (t,t+i)
9   */
10 StaffingDemands,AvailableResources
11 calculateSaffingDemands (
12   double l, double u,
13   Time t, Time i,
14   ResourceType k) {
15
16   // calculate resource utilization
17   TimeBlock b = (t-i,t);
18   double denominator = 0;
19   double numerator = 0;
20   for (∀ Resource r : r ∈ R_b^k) {
21     denominator += (r ∈ A_b^k) ? i : r_b;
22     numerator += r_b;
23   }
24   double utilization=numerator/denominator;
25
26   // calculate staffing demands
27   StaffingDemands d_b^k = 0;
28   if (utilization ≥ l &&
29       utilization ≤ u) {
30     d_b^k = count(A_b^k);
31   } else {
32     double middle = (l+u)/2;
33     d_b^k = round(numerator/(middle*i));
34   }
35
36   // select available resources
37   TimeBlock nb = (t,t+i);
38   AvailableResources A_nb^k = {};
39   while(count(A_nb^k) != d_b^k) {
40     Resource r : r ∈ R^k;
41     A_nb^k = A_nb^k ∪ {r};
42   }
43
44   return d_b^k,A_nb^k
45 }
```

Fig. 6. 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 $nb = (t, t+i)$.

- 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.

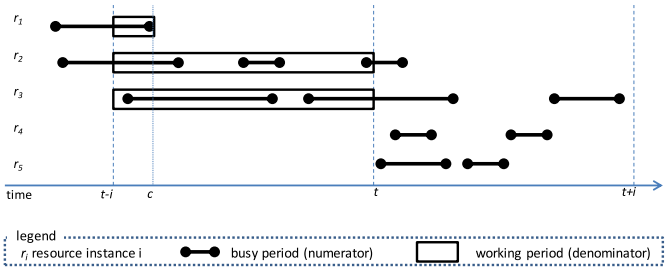


Fig. 7. Instance of the algorithm execution for staffing demands: $R_b^k = \{r_1, r_2, r_3\}$, $A_b^k = \{r_2, r_3\}$, $d_b^k = 3$, $R_{nb}^k = \{r_2, r_3, r_4, r_5\}$, and $A_{nb}^k = \{r_3, r_4, r_5\}$, where $b = (t-i, t)$, $nb = (t, t+i)$

- R^k A set of resources of type k for which we want to determine required staffing levels.
- R_b^k A set of resources of type k that are used during time block b, $R_b^k \subseteq R^k$.
- A_b^k A 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.

Fig. 6 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, Fig. 7 shows the execution of the algorithm for a single instance. As per lines 16–24 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 *utilization* at line 24, it determines how many resources are required to satisfy the desired range of utilization levels l and u (lines 26–34). If the calculated *utilization* is between l and u, it means that resources in A_b^k are utilized as expected. However, if *utilization* is outside the limits l and u, the algorithm calculates staffing demands d_b^k based on the mid-point of the utilization range $middle = (l+u)/2$ and actual assigned subperiods *numerator* 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 $t+i$ (lines 36–42). 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.

2) Staffing via Integer Linear Programming: In this section, we present our ILP-based staffing approach, which minimizes total staff salaries, while meeting: 1) hourly constraints on staff numbers calculated by the previously described staffing demands algorithm and 2) 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.

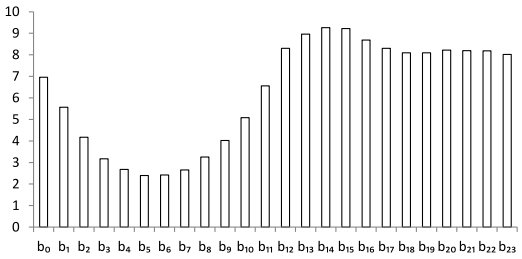
- B A set of time blocks in a day.
- L^k A set of shift lengths for resource type k.

B: $\{b_0, b_1, b_2, b_3, b_4, b_5, \ldots, b_{19}, b_{20}, b_{21}, b_{22}, b_{23}\}$, $b_i = (i, i+1)$

L^{MD}: $\{8\}$

$S^{MD}_{b,l}$: $S^{MD}_{b_0,8} = \{b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7\}, \ldots,$
$S^{MD}_{b_{23},8} = \{b_{23}, b_0, b_1, b_2, b_3, b_4, b_5, b_6\}$

d^{MD}_b:



$p^{MD}_{b,l}$: $p^{MD}_{b_7,8} = p^{MD}_{b_{15},8} = p^{MD}_{b_{23},8} = 1, \forall b \in B \setminus \{b_7, b_{15}, b_{23}\}, p^{MD}_{b,8} = 0$

c^{MD}: USD188/hour

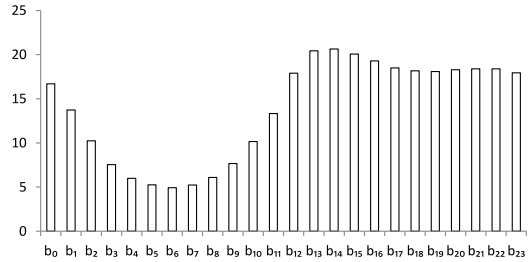
Fig. 8. Parameter values for MD staffing. B: 24 time blocks. L^{MD}: 8-h shift length. $S^{\mathrm{MD}}_{b,l}$: time blocks in l length shift b. d^{MD}_b: staffing demands per each time blocks driven by 70%–80% utilization target. $p^{\mathrm{MD}}_{b,l}$: nonoverlapped three, 8-h shifts. c^{MD}: salary per hour.

B: $\{b_0, b_1, b_2, b_3, b_4, b_5, \ldots, b_{19}, b_{20}, b_{21}, b_{22}, b_{23}\}$, $b_i = (i, i+1)$

L^{RN}: $\{6, 8, 12\}$

$S^{RN}_{b,l}$: $S^{RN}_{b_0,6} = \{b_0, b_1, b_2, b_3, b_4, b_5\}, \ldots,$
$S^{RN}_{b_{23},6} = \{b_{23}, b_0, b_1, b_2, b_3, b_4\},$
$S^{RN}_{b_0,8} = \{b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7\}, \ldots,$
$S^{RN}_{b_{23},8} = \{b_{23}, b_0, b_1, b_2, b_3, b_4, b_5, b_6\},$
$S^{RN}_{b_0,12} = \{b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}\}, \ldots,$
$S^{RN}_{b_{23},12} = \{b_{23}, b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}\}$

d^{RN}_b:



$p^{RN}_{b,l}$: $\forall b \in B, \forall l \in L^{RN}, p^{RN}_{b,l} = 1$

c^{MD}: USD55/hour

Fig. 9. Parameter values for RN staffing. B: 24 time blocks. L^{RN}: 6-, 8-, 12-h shift lengths. $S^{\mathrm{RN}}_{b,l}$: time blocks in l length shift b. d^{RN}_b: staffing demands per each time blocks driven by 60%–70% utilization target. $p^{\mathrm{RN}}_{b,l}$: three different shift lengths, and the staffing pattern allows a shift to start at any time. c^{RN}: salary per hour.

- $S^k_{b,l}$ 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^k_b Staffing demands, the number of required resource, for resource type k during each time block $b \in B$.
- $p^k_{b,l}$ 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, Fig. 8 shows the ILP parameter values needed to determine MD staffing levels. Parameter B divides a day into 24 time blocks. Parameter $p^{\mathrm{MD}}_{b,l}$ establishes three 8-h, nonoverlapping shifts a day for MDs. The staffing pattern parameter $p^{\mathrm{MD}}_{b,l}$ can encode any staffing pattern of various shift lengths and start times; however, this section demonstrates only these three shifts of MDs for exposition. MD staffing demand d^{MD}_b 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 MDs and RNs to work a 6-, 8-, or 12-h shift; further, they may also allow shifts to start at any time. To accommodate this additional flexibility, the ILP parameters for RNs can be set as in Fig. 9.

The decision variable in the ILP is $x^k_{b,l,i}$, which determines the number of a particular staff type (e.g., MD or RN) needed in each hour of a shift. The number of $x^k_{b,l,i}$ is equal to $|B| \times |L^k| \times |B|$ for all b, l, i combinations. We use the simplex method to solve the ILP, NP-hard problem.

- $x^k_{b,l,i}$ 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^k_b, the minimum number of required staff during each time block b. Equation (1) is the objective function of the ILP-based staffing problem. The ILP objective function aims to minimize total staffing costs per day

$$\min \sum_{b \in B} \sum_{l \in L^k} \sum_{i \in B} c^k \cdot x^k_{b,l,i}. \tag{1}$$

The objective equation (1) is subject to the constraint equations (2)–(4). First, constraint equation (2) enforces that the number of scheduled staff k is always greater than or equal to the number of required staff k for all time blocks while satisfying a given staffing pattern. Second, constraint equation (3) states that the same number of staff k members are working in a shift. Last, constraint equation (4) assures that a staff k member is working on only the staff's shift

$$\sum_{b \in B} \sum_{l \in L^k} p^k_{b,l} \cdot x^k_{b,l,i} \geq d^k_i, \quad \forall i \in B \tag{2}$$

$$x^k_{b,l,i} = x^k_{b,l,j}, \quad \forall i \in S^k_{b,l}, \forall j \in S^k_{b,l} \tag{3}$$

$$x^k_{b,l,i} = 0, \quad \forall i \in B \setminus S^k_{b,l}. \tag{4}$$

i	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}	b_{17}	b_{18}	b_{19}	b_{20}	b_{21}	b_{22}	b_{23}
$x^{MD}_{b_7,8,i}$								9	9	9	9	9	9	9	9									
$x^{MD}_{b_{15},8,i}$																8	8	8	8	8	8	8	8	
$x^{MD}_{b_{23},8,i}$	8	8	8	8	8	8	8																	8

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

To illustrate the ILP, Fig. 10 shows an MD staffing solution for Fig. 8. There are three separate shifts: 7–14, 15–22, and 23–6. Each shift has 9, 8, or 8 MDs, respectively.

However, the RN staffing solution in Fig. 11 looks very different from the MD staffing in Fig. 10. This is because the ILP parameters for RN staffing in Fig. 9 allow three different shift

i	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}	b_{17}	b_{18}	b_{19}	b_{20}	b_{21}	b_{22}	b_{23}
$x^{RN}_{b_0,8,i}$	2	2	2	2	2	2	2	2																
$x^{RN}_{b_1,8,i}$		1	1	1	1	1	1	1	1															
$x^{RN}_{b_7,6,i}$								3	3	3	3	3	3											
$x^{RN}_{b_8,6,i}$									4	4	4	4	4	4										
$x^{RN}_{b_9,6,i}$										3	3	3	3	3	3									
$x^{RN}_{b_{10},12,i}$											4	4	4	4	4	4	4	4	4	4	4			
$x^{RN}_{b_{11},6,i}$												1	1	1	1	1	1							
$x^{RN}_{b_{11},8,i}$												3	3	3	3	3	3	3						
$x^{RN}_{b_{12},12,i}$													4	4	4	4	4	4	4	4	4	4	4	4
$x^{RN}_{b_{13},12,i}$	4													4	4	4	4	4	4	4	4	4	4	4
$x^{RN}_{b_{14},6,i}$															4	4	4	4	4	4				
$x^{RN}_{b_{15},12,i}$	2	2	2													2	2	2	2	2	2	2	2	2
$x^{RN}_{b_{19},12,i}$	3	3	3	3	3	3	3													3	3	3	3	3
$x^{RN}_{b_{20},6,i}$	4	4																			4	4	4	4
$x^{RN}_{b_{21},8,i}$	1	1	1	1	1																	1	1	1
$x^{RN}_{b_{22},6,i}$	3	3	3	3																			3	3

Fig. 11. RN staffing solution of Fig. 9. Sixteen overlapped shifts. Shifts have different lengths and start times. Total number of RNs 46 and RNs working hours 392.

lengths, and a shift can start at any time (i.e., overlap in shifts are allowed). Therefore, RN staffing in Fig. 11 very closely approximates actual RN staffing demands d_b^{RN} in Fig. 9. We return to this key point while discussing the results of actual simulations carried out as the third stage of our simulation study.

IV. Evaluation

This section describes how we used our scheduling approach to study the effects of different staffing levels and policies on the operational characteristics of our example ED. Other experiments that describe the application of our modeling capabilities can be found in [14], [28], [30], and [31]. These papers demonstrate how to use our approach to model various hospital resource utilization flexibly. For this paper, we used the process model presented in Section III-A, and the resource type specifications presented in Section III-B, instantiating from these types 2 triage nurses, 5 clerks, 48 beds, 2 X-ray rooms, and 4 CT rooms. This ED resource distribution was based on Baystate Medical Center data. We executed enough simulation replications to obtain 95% confidence intervals and a half-width within 2% of the mean of staff utilizations. For each replication, we simulated 72 h of ED operations, using only the output of the middle 24 h 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. We used Amazon EC2 to create a virtual server (c4.8xlarge type) to run our simulations. The server instance took about 41 min for 100 simulation replications.

In these simulations we measured actual utilization levels for MDs and RNs, the average patient LoS, the ED contribution margin, and the impact of staff shift scheduling on the number of patient handoffs. We defined LoS as the time from a patient's arrival to departure, and ED contribution margin as the total revenue (using Medicare reimbursement levels)

Fig. 12. MD and RN staffing demands curves using the staffing demands algorithm in Fig. 6. (a) MD and RN staffing demands curves when MD's and RN's lower and upper utilization limits are set as 50% and 60%, respectively. (b) MD 70%–80% and RN 70%–80%. We omitted other combinations of resource utilization ranges to enhance the clarity of the figure.

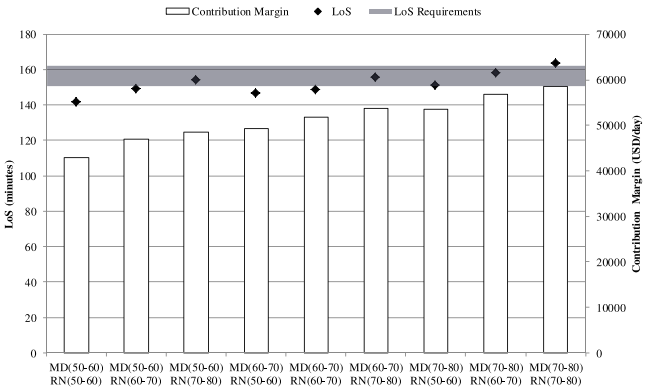


Fig. 13. Simulation results of patient's LoS, contribution margin according to each utilization boundary. `LoS Requirements` is the LoS objective, so that four staffing, MD(50–60) RN(70–80), MD(60–70) RN(70–80), MD(70–80) RN(50–60), and MD(70–80) RN(60–70) satisfy it.

derived from treating all patients in the simulated 24-h period minus costs for staffing, supplies, and medications.

We begin by presenting results of the staffing demands algorithm, which yields the number of MDs and RNs needed in each hour of the day to ensure that utilization falls in a prespecified range. We tested this algorithm for various combinations of MD and RN utilization levels suggested by our domain expert. Fig. 12 shows some example results where we have assumed that staffing levels can change every hour as needed, allowing shifts to be as short as 1 h, but assuring that the number of MDs and RNs is equal exactly to the results produced by the staffing demands algorithm.

Fig. 13 compares LoS for a number of different combinations of staff utilization levels. The figure uses a gray band to indicate average LoS that lies between 130% and 140% of the minimum possible value (116 min). The average LoS range was provided by our domain expert and is used as an objective for staff scheduling. We define the minimum average LoS by assuming an infinite supply of all resources, causing all wait times to be zero. The calculation of the minimum average LoS is then straightforward, using only the static ED process structure to identify the steps whose execution times are to be summed to obtain the minimum. The figure shows that four staffing solutions, MD(50%–60%) RN(70%–80%), MD(60%–70%) RN(70%–80%), MD(70%–80%) RN(50%–60%), and

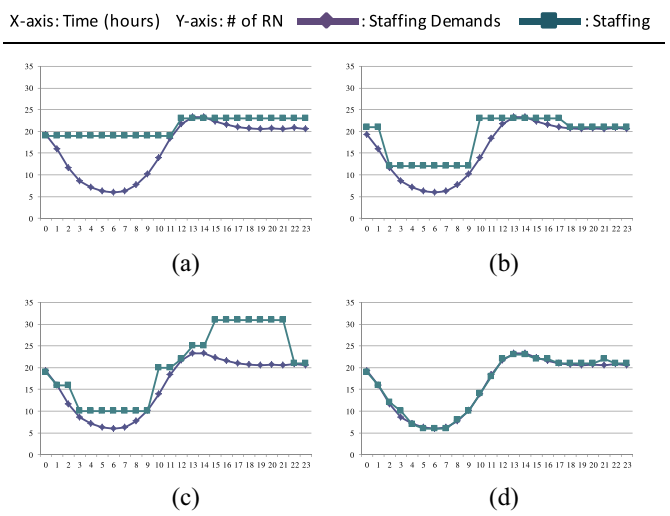


Fig. 14. ILP solutions of various RN staffing patterns. (a) and (c) 12-h shifts disallowing and allowing shift overlap, respectively. (b) 8-h shift disallowing shift overlap. (d) Allowing for combinations of overlapping 6-, 8-, and 12-h shifts. Staff salaries (USD/day): (a) 27 720, (b) 24 640, (c) 27 060, and (d) 21 560. We omitted other combinations such as 6-h shifts and allowing and disallowing shift overlap to improve the clarity of the figure.

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.

A. Impact of Shift Length and Overlap

To consider the impact of shift length and overlap, we ran simulations that allowed RN shift lengths to be 6, 8, or 12 h, and compared the effect of prohibiting shifts to overlap (i.e., start and stop at the same time) to the case where overlap is allowed. Fig. 14 shows RN staffing solutions output from the ILP-based staffing algorithm in Section III-D2, compared to results generated by the staffing demands algorithm. Note that the results produced by the ILP for a given shift length and overlap constraint are always equal to or higher than the staffing demands curve.

Fig. 14(a) and (c) shows that 12-h shift lengths cannot cover staffing demands as closely as 8-h shift lengths, and comparing Fig. 14(c) to Fig. 14(a), shows that overlapping shifts cover the staffing demands curves more closely than nonoverlapping shifts. Fig. 14(d) shows that staffing with overlapped shifts of 6-, 8-, or 12-h lengths (flexible staffing) is almost indistinguishable from the staffing demands curve itself.

Fig. 15 shows that average LoS for the different shift length and overlap combinations does not differ significantly, although Fig. 15 shows staffing based on 12-h shift lengths, with or without overlapped shifts, creates shorter LoS than 6- and 8-h staffing. This makes intuitive sense because 12-h shifts allow more RNs to be scheduled in more hours [see Fig. 14(a) and (c)], reducing RN contention and thus reducing patient waiting time. In general, staffing without overlap creates lower LoS than staffing with overlap. Fig. 15 also shows that flexible staffing results in longer LoS than 1-h staffing. This is because contention created by the same MD and RN constraint causes the patient to have to wait more frequently.

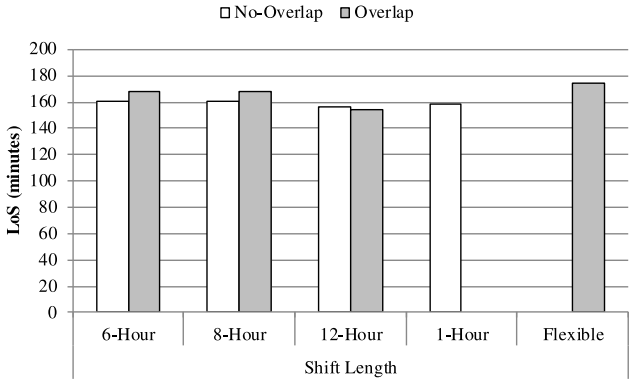


Fig. 15. LoS comparison among various staffing.

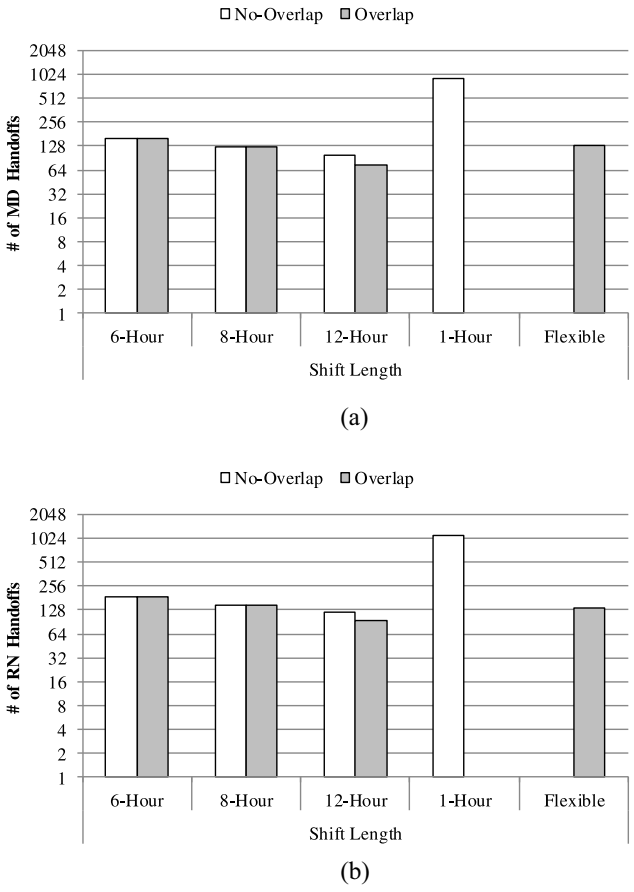


Fig. 16. Number of handoffs comparison among various staffing. Number of (a) MD handoffs and (b) RN handoffs.

Fig. 16 compares the number of handoffs resulting from various staffing options. Shorter shifts (especially 1-h shifts) necessitate more handoffs, which should be minimized, as our domain expert believes they lead to increased errors. Indeed, Fig. 16 shows that longer shifts result in fewer handoffs and that 1-h staffing produces a larger number of RN handoffs than all other staffing options.

Finally, Fig. 17 shows mean utilization levels for all the staffing options. It shows that, because overlapped staffing is closer to the staffing demands curve than staffing without

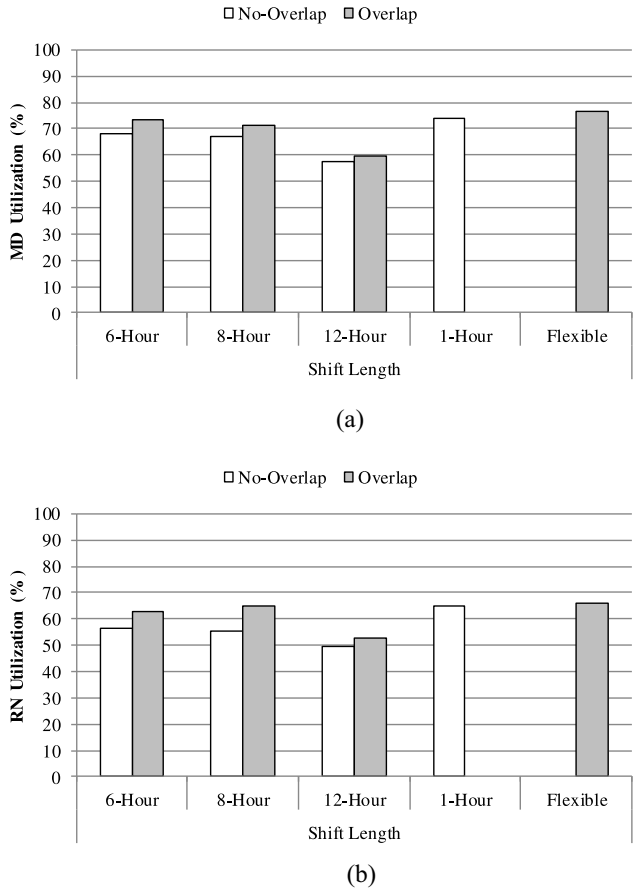


Fig. 17. Utilization comparison among various staffing. (a) MD utilization. (b) RN utilization.

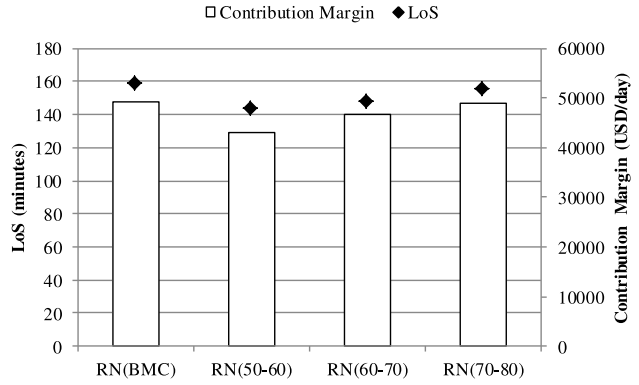


Fig. 18. Patient's LoS and contribution margin comparison: RN(BMC) RN staffing of Baystate Medical Center, RN(50–60) RN staffing derived by utilization limits 50%–60%, RN(60–70) RN staffing derived by utilization limits 60%–70%, and RN(70–80) RN staffing derived by utilization limits 70%–80%.

overlap, overlapping leads to higher utilization. In addition, 6- and 8-h staffing show subtle differences in utilization, but 12-h staffing shows lower utilization both for overlapping and nonoverlapped staffing.

B. Comparison With Baystate Staffing Schedule

Fig. 18 depicts RN staffing data for Baystate Medical Center [RN(BMC)], while RN(50-60), RN(60-70), and

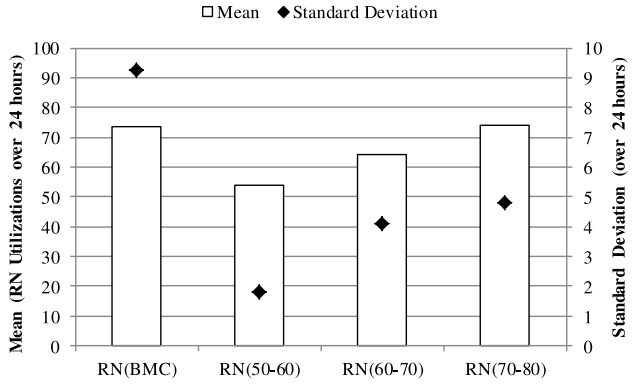


Fig. 19. Utilization comparison: RN(BMC) RN staffing of Baystate Medical Center, RN(50–60) RN staffing derived by utilization limits 50%–60%, RN(60–70) RN staffing derived by utilization limits 60%–70%, and RN(70–80) RN staffing derived by utilization limits 70%–80%.

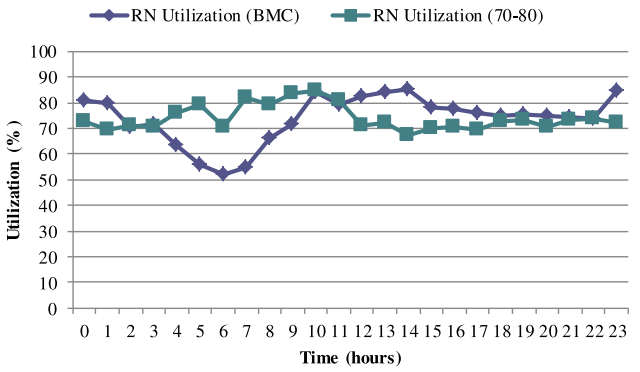


Fig. 20. Utilization comparison over 24 h a day: RN utilization (BMC) RN utilization of Baystate Medical Center, RN utilization (70–80) RN utilization derived by utilization limits 70%–80%.

RN(70-80) represent the RN staffing results obtained from our scheduling studies. Fig. 18 compares LoS and contribution margin. As can be seen, when we set the utilization range to 70%–80% RN(70-80), the simulation results obtained are similar to RN(BMC) staffing. Note also that the simulations designed to assure lower staff utilization levels provide interesting contrasts. For example, LoS is 20 min lower in RN(50-60) but so is the contribution margin.

Fig. 19, comparing average RN utilizations for a 24-h day, and their standard deviations, shows that the variation in RN utilization levels is much lower in staffing results generated by our approach, and also for the RN(70-80) case, which has approximately the same average as RN(BMC).

Fig. 20 provides insight into why RN(BMC) has higher utilization variation compared to RN(70-80), showing that RNs are underutilized in the less busy hours of the night, but are overutilized in the busy hours of the afternoon. Higher utilization in these busy hours implies increased LoS, while low utilization during less busy hours implies that personnel costs are being wasted. However, as can be seen in Fig. 20, our scheduling approach suggests RN staffing that better balances the level of resource utilization over a 24-h period, an important goal in hospital resource management.

To keep this paper concise, we have not presented many other details of our results. For example, we have presented

LoS measures that are averaged over patients of all acuity levels, but our simulations have determined acuity-level-specific LoS and waiting times. These are not reported here to save space. Further results showing the implications of a variety of other ED operations policies can be found in [29].

V. Threats to Validity

While our approach is designed to apply broadly to resource-dependent systems, our findings and suggestions in Section IV may have been affected by the following threats to the validity.

A. Construct Validity Threats

Our staff scheduling approach uses levels of resource utilization as an input to an algorithm for calculating a staffing demand curve. As can be seen in Fig. 6, the utilization level is calculated for each time block by counting the busy periods for each resource. For this project we used as input resource utilization level based on our domain expert's observations and analysis of real-world resource utilization. However, the degree to which this input utilization level matches real staff utilization is unclear because measuring staff utilization in the real world is expensive, time consuming, and requires observers familiar with the exact nature of staff activities.

In addition, staff activities vary considerably, ranging from actual patient care, to hand-washing, meal breaks, walking, and answering telephone calls. Additional variability derives from long-term management policies such as vacation time allowances, and short-term issues such as sickness. The accuracy with which each of these different kinds of activities and policies of resource utilization must be modeled will have to vary depending on the accuracy required of the model and the simulation results it is expected to produce.

B. Internal Validity Threats

Our activity coordination model includes only clinical patient care activities such as patient assessment and treatment, drawing blood and so on. We chose this level of abstraction because it made it easier for our domain expert to specify process details, sequencing, timing, and so forth. Adding nonclinical activities such as meal breaks seemed to significantly increase the domain expert's difficulty in specifying these details. However, these activities definitely influence the performance of patient care in an ED. The extent to which these nonclinical tasks must be specified in an ED process model requires further investigation.

Patient care in an ED incorporates a variety of constraints on resource management (e.g., a patient should be always seen by the same MD until the end of the MD's shift). We performed extensive testing to verify that our ED models correctly incorporate and adhere to those constraints, but testing is inherently incomplete and our implementation may contain bugs that may affect the results. Our ongoing work addresses this potential limitation by studying the application of static analysis techniques to verify the correct application of constraints during simulation.

C. External Validity Threats

Due to lack of data on how patient arrivals vary over the days of the week and the seasons of a year, this paper focuses on staffing for a hypothetical day whose characteristics are derived from mean behaviors measured over the days of a year. Reflecting in our simulations the variation that occurs over the different days of a year requires consideration of such additional constraints as days-off, holidays, labor regulations and so on. To incorporate these additional constraints, the constraints in the ILP in Section III-D2 would have to be carefully modified. Our simulation capabilities, however, should be able to model this variation if given specifications of how patient arrivals vary over any given specified time interval.

We have designed our approach to apply broadly to systems with complex resources, but we have thus far applied it only to the ED healthcare domain. By studying other human-intensive systems, (e.g., blood transfusion [7], elections [27], and software development [39]), we found that characteristics of patient care in an ED are representative of those in other domains, which suggests that our approach may also be applicable to such other domains. However, to strengthen the generality of the approach, additional application domains should be studied.

VI. Contributions and Future Work

We have combined DES and ILP in a three-stage approach to studying resource scheduling for complex systems, using ED staff scheduling as a case study. In the first stage DES is used, 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 prespecified resource utilization level. In the second stage the staffing demand curve is used, along with other parameters such as shift lengths and staffing constraints, as input to an ILP-based staffing algorithm. In the third stage, the staffing solution provided by the ILP is evaluated by rerunning our DES to quantify key ED operational characteristics such as patient LoS, actual staff utilization, cost and quantities of patient handoffs. Among the many results of our simulation studies, we found the following.

1) Staffing policies that allow shifts of different lengths and overlapped shifts can reduce costs while 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 lengths result in fewer handoffs.
3) Staffing without overlap creates lower LoS than staffing with overlap.
4) Overlapped staffing shows higher utilization than nonoverlapped staffing.

These studies suggest that our DES approach can be useful to hospital administrators in evaluating scheduling policies and in understanding the tradeoffs entailed by new policies.

Unlike previous ED simulation work, our 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 MD until the end of the MD's shift). Further, this paper considers the interactions and interferences between MDs and RNs in patient care. Our staffing demands algorithm is unique in creating MD and RN requirements for each hour based on prespecified target utilization ranges.

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

We continue to study ED staffing approaches by exploring: 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 and monthly staff scheduling; and 3) further validating our approach via more detailed comparisons between our results and empirical measurements of actual EDs.

In the longer term, we will apply this approach to processes and systems in other domains such as elections, where our simulations could to facilitate and expedite such processes as tabulation and recounts, and software development, where various staffing profiles and resource assignment constraints could affect quality and productivity in agile methods such as Scrum.

REFERENCES

[1] V. A. Augusto and X. Xie, "A modeling and simulation framework for health care systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 1, pp. 30–46, Jan. 2014.

[2] E. Beck, "A discrete event simulation approach to resource management, process changes and task prioritization in emergency departments," M.S. thesis, Dept. Mech. Ind. Eng., Univ. Massachusetts, Amherst, MA, USA, 2009.

[3] S. Brenner *et al.*, "Modeling and analysis of the emergency department at University of Kentucky Chandler Hospital using simulations," *J. Emergency Nursing*, vol. 36, no. 4, pp. 303–310, 2010.

[4] P. D. Bruecker, J. V. den Bergh, J. Beliën, and E. Demeulemeester, "Workforce planning incorporating skills: State of the art," *Eur. J. Oper. Res.*, vol. 243, no. 1, pp. 1–16, May 2015.

[5] J. O. Brunner, J. F. Bard, and R. Kolisch, "Midterm scheduling of physicians with flexible shifts using branch and price," *IIE Trans.*, vol. 43, no. 2, pp. 84–109, 2010.

[6] M. W. Carter and S. D. Lapierre, "Scheduling emergency room physicians," *Health Care Manag. Sci.*, vol. 4, no. 4, pp. 347–360, 2001.

[7] L. A. Clarke *et al.*, "Process programming to support medical safety: A case study on blood transfusion," in *Proc. Int. Conf. Unifying Softw. Process Spectr. (SPW)*, 2005, pp. 347–359.

[8] J. K. Cochran and K. T. Roche, "A multi-class queuing network analysis methodology for improving hospital emergency department performance," *Comput. Oper. Res.*, vol. 36, no. 5, pp. 1497–1512, 2009.

[9] J. Van den Bergh, J. Beliën, P. D. Bruecker, E. Demeulemeester, and L. D. Boeck, "Personnel scheduling: A literature review," *Eur. J. Oper. Res.*, vol. 226, no. 3, pp. 367–385, 2013.

[10] C. Duguay and F. Chetouane, "Modeling and improving emergency department systems using discrete event simulation," *Simulation*, vol. 83, no. 4, pp. 311–320, 2007.

[11] M. P. Fanti, A. M. Mangini, M. Dotoli, and W. Ukovich, "A three-level strategy for the design and performance evaluation of hospital departments," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 4, pp. 742–756, Jul. 2013.

[12] Y. Ferrand, M. Magazine, U. S. Rao, and T. F. Glass, "Building cyclic schedules for emergency department physicians," *Interfaces*, vol. 41, no. 6, pp. 521–533, 2011.

[13] L. V. Green, J. Soares, J. F. Gjulio, and R. A. Green, "Using queuing theory to increase the effectiveness of physician staffing in the emergency department," *Acad. Emergency Med.*, vol. 13, pp. 61–68, Jan. 2006.

[14] P. L. Henneman *et al.*, "Using computer simulation to study nurse-to-patient ratios in an emergency department," *J. Nursing Admin.*, vol. 45, no. 11, pp. 551–556, Nov. 2015.

[15] N. Izady and D. Worthington, "Setting staffing requirements for time dependent queueing networks: The case of accident and emergency departments," *Eur. J. Oper. Res.*, vol. 219, no. 3, pp. 531–540, 2012.

[16] O. B. Jennings, A. Mandelbaum, W. A. Massey, and W. Whitt, "Server staffing to meet time-varying demand," *Manag. Sci.*, vol. 42, no. 10, pp. 1383–1394, 1996.

[17] S. W. Kang and H. S. Park, "Emergency department visit volume variability," *Clin. Exp. Emergency Med.*, vol. 2, no. 3, pp. 150–154, Sep. 2015.

[18] P. Kazemian, Y. Dong, T. R. Rohleder, J. E. Helm, and M. P. V. Oyen, "An IP-based healthcare provider shift design approach to minimize patient handoffs," *Health Care Manag. Sci.*, vol. 17, no. 1, pp. 1–14, 2014.

[19] B. S. Lerner *et al.*, "Exception handling patterns for process modeling," *IEEE Trans. Softw. Eng.*, vol. 36, no. 2, pp. 162–183, Mar./Apr. 2010.

[20] J. Li and P. K. Howard, "Modeling and analysis of hospital emergency department: An analytical framework and problem formulation," in *Proc. 6th Annu. IEEE Conf. Autom. Sci. Eng.*, Toronto, ON, Canada, Aug. 2010, pp. 897–902.

[21] Y. Liu and W. Whitt, "A network of time-varying many-server fluid queues with customer abandonment," *Oper. Res.*, vol. 59, no. 4, pp. 835–846, Jul./Aug. 2011.

[22] W. A. Massey and W. Whitt, "Networks of infinite-server queues with nonstationary Poisson input," *Queueing Syst.*, vol. 13, no. 1, pp. 183–250, 1993.

[23] 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," *Acad. Emergency Med.*, vol. 18, no. 12, pp. 1269–1277, 2011.

[24] S. A. Paul, M. C. Reddy, and C. J. Deflitch, "A systematic review of simulation studies investigating emergency department overcrowding," *Simulation*, vol. 86, nos. 8–9, pp. 559–571, Aug. 2010.

[25] J. Pennic, *Study: Emergency Medicine Tops Most Complex Physician Schedules*, HIT Consultant, Jun. 2016. [Online]. Available: http://hitconsultant.net/2016/06/17/study-complex-physician-schedules/

[26] M. S. Raunak and L. J. Osterweil, "Resource management for complex, dynamic environments," *IEEE Trans. Softw. Eng.*, vol. 39, no. 3, pp. 384–402, Mar. 2013.

[27] M. S. Raunak, B. Chen, A. Elssamadisy, L. A. Clarke, and L. J. Osterweil, "Definition and analysis of election processes," in *Proc. SPW/ProSim*, vol. 3966. 2006, pp. 178–185.

[28] S. Y. Shin, H. Balasubramanian, Y. Brun, P. L. Henneman, and L. J. Osterweil, "Resource scheduling through resource-aware simulation of emergency departments," in *Proc. 5th Int. Workshop Softw. Eng. Health Care (SEHC)*, San Francisco, CA, USA, May 2013, pp. 64–70.

[29] S. Y. Shin, H. Balasubramanian, Y. Brun, P. L. Henneman, and L. J. Osterweil, "Discrete-event simulation and integer linear programming for constraint-aware resource scheduling," School Comput. Sci., Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. UM-CS-2014-009, 2014.

[30] S. Y. Shin, Y. Brun, and L. J. Osterweil, "Specification and analysis of human-intensive system resource-utilization policies," in *Proc. 8th Int. Workshop Softw. Eng. Healthcare Syst. (SEHS)*, Austin, TX, USA, May 2016, pp. 8–14.

[31] S. Y. Shin, Y. Brun, L. J. Osterweil, H. Balasubramanian, and P. L. Henneman, "Resource specification for prototyping human-intensive systems," in *Proc. 18th Int. Conf. Fundam. Approaches Softw. Eng. (FASE)*, London, U.K., Apr. 2015, pp. 332–346.

[32] D. Sinreich, O. Jabali, and N. P. Dellaert, "Reducing emergency department waiting times by adjusting work shifts considering patient visits to multiple care providers," *IIE Trans.*, vol. 44, no. 3, pp. 163–180, 2012.

[33] R. Stolletz and J. O. Brunner, "Fair optimization of fortnightly physician schedules with flexible shifts," *Eur. J. Oper. Res.*, vol. 219, no. 3, pp. 622–629, 2012.
[34] 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). Homewood, IL, USA: Irwin, 1992.
[35] J. Wang, J. Li, K. Tussey, and K. Ross, "Reducing length of stay in emergency department: A simulation study at a community hospital," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 6, pp. 1314–1322, Nov. 2012.
[36] A. Wise, "Little-JIL 1.5 language report," Dept. Comput. Sci., Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. 2006–051, 2006.
[37] S. Zeltyn *et al.*, "Simulation-based models of emergency departments: Operational, tactical, and strategic staffing," *ACM Trans. Model. Comput. Simulat.*, vol. 21, no. 4, pp. 1–25, Aug. 2011.
[38] 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," *J. Emergency Nursing*, vol. 38, no. 4, pp. 322–328, 2012.
[39] X. Zhao and L. J. Osterweil, "An approach to modeling and supporting the rework process in refactoring," in *Proc. Int. Conf. Softw. Syst. Process*, Zürich, Switzerland, 2012, pp. 110–119.

Seung Yeob Shin received the Ph.D. degree from the Laboratory for Advanced Software Engineering Research, College of Information and Computer Science, University of Massachusetts at Amherst, Amherst, MA, USA, in 2016.

He is a Research Associate with the University of Luxembourg, Luxembourg City, Luxembourg. His current research interests include software engineering and healthcare systems.

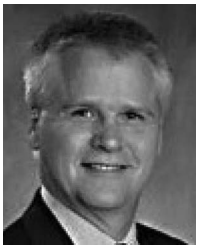
Yuriy Brun (M'12) received the Ph.D. degree from the University of Southern California, Los Angeles, CA, USA, in 2008.

He is an Assistant Professor with the College of Information and Computer Science, University of Massachusetts at Amherst, Amherst, MA, USA. His current research interests include software engineering.

Dr. Brun was a recipient of the NSF CAREER Award in 2015, and the IEEE TCSC Young Achiever in Scalable Computing Award in 2013.

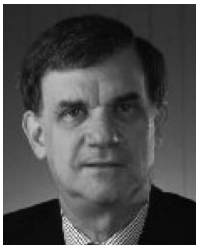
Hari Balasubramanian received the Ph.D. degree from Arizona State University, Tempe, AZ, USA, in 2006, with a focus on scheduling theory, computational complexity, and the design of heuristics.

He is an Associate Professor of Industrial Engineering, University of Massachusetts at Amherst, Amherst, MA, USA. He completed his postdoctoral research at the Department of Health Sciences Research, Mayo Clinic, Rochester, MN, USA, in 2008.



Philip L. Henneman received the medical degree from Harvard Medical School, Boston, MA, USA, in 1980.

He is a Professor of emergency medicine with the University of Massachusetts School of Medicine, Worcester, MA, USA. He was with the Emergency Department, Baystate Medical Center, Springfield, MA, USA.



Leon J. Osterweil (M'84) received the Ph.D. degree from the University of Maryland, College Park, MD, USA, in 1971.

He is a Professor Emeritus with the College of Information and Computer Science, University of Massachusetts at Amherst, Amherst, MA, USA.

Dr. Osterweil was a recipient of the Outstanding Research, Influential Educator, and Distinguished Service Lifetime Achievement Awards from ACM SIGSOFT. He has served on several editorial boards, including the IEEE TRANSACTIONS ON SOFTWARE ENGINEERING and *ACM Transactions on Software Engineering and Methodology*. He is a Fellow of ACM.