Probability in the Engineering and Informational Sciences, **22**, 2008, 301–332. Printed in the U.S.A.
doi:10.1017/S0269964808000272

SCHEDULING IMPATIENT JOBS IN A CLEARING SYSTEM WITH INSIGHTS ON PATIENT TRIAGE IN MASS CASUALTY INCIDENTS

NILAY TANIK ARGON AND SERHAN ZIYA
Department of Statistics and Operations Research
University of North Carolina
Chapel Hill, NC 27599-3180
E-mail: nilay@email.unc.edu; ziya@email.unc.edu

RHONDA RIGHTER
Department of Industrial Engineering and Operations Research
University of California
Berkeley, CA 94720-1777
E-mail: rrighter@ieor.berkeley.edu

Motivated by the patient triage problem in emergency response, we consider a single-server clearing system in which jobs might abandon the system if they are not taken into service within their "lifetime." In this system, jobs are characterized by their lifetime and service time distributions. Our objective is to dynamically determine the optimal or near-optimal order of service for jobs so as to minimize the total number of abandonments. We first show that if the jobs can be ordered in such a way that the job with the shortest lifetime (in the sense of hazard rate ordering) also has the shortest service time (in the sense of likelihood ratio ordering), then the optimal policy gives the highest priority to this "time-critical" job independently of the system state. For the case in which the jobs with shorter lifetimes have longer service times, we observed that the optimal policy generally has a complex structure that might depend on the type and number of jobs available. For this case, we provide partial characterizations of the optimal policy and obtain sufficient conditions under which a state-independent policy is optimal. Furthermore, we develop two state-dependent heuristic policies, and by means of a numerical study, we show that these heuristics perform well, especially when jobs abandon the system at a relatively faster rate when compared to service rates. Based on our analytical and numerical results, we develop several insights on patient triage in the immediate aftermath of a mass casualty event. For example, we conclude that in a worst-case scenario, where medical resources are overwhelmed

0269-9648/08 $25.00

with a large number of casualties who need immediate attention, it is crucial to implement state-dependent policies such as the heuristic policies proposed in this article.

1. INTRODUCTION

Consider a service system in which a finite number of jobs are waiting to receive service from a single server. There will be no additional arrivals to this system and the existing jobs might abandon the system without receiving service if they are not given service within their *lifetime*. For such a system, we are interested in the dynamic scheduling of jobs that are characterized by their lifetime and service time distributions, with the objective of minimizing the expected number of total abandonments. More specifically, our aim is to develop insights on optimal or near-optimal scheduling policies that determine the priorities among different job types dynamically depending on the system state.

Our primary motivation to study this optimal control problem originates from the *patient triage* problem that arises in the immediate aftermath of a mass casualty incident such as a natural disaster or a terrorist attack[1]. After such an event, which may cause a significant number of injuries and overwhelm the local medical resources, emergency responders perform triage on patients and determine the order by which these patients will be admitted to scarce resources. These resources may include ambulances, imaging devices (e.g., X-ray and magnetic resonance imaging machines), or operating rooms. For example, in case of a mass trauma event caused by a bombing, it is estimated that at least half of the casualties will require surgical procedures, and therefore operating rooms are expected to constitute a serious bottleneck (see, e.g., Levi, Michaelson, Admi, Bregman, and Bar-Nahor [18] and Peleg, Aharonson-Daniel, Michael, Shapira, et al. [21]).

Many local emergency response divisions and hospitals have adopted triage procedures that will be used in case of a mass casualty incident. Following these procedures, immediately after the incident, patients are classified into different priority groups. (According to one of the most widely adopted triage procedures, which is called START, patients are classified into four groups: immediate, delayed, minimum, and expectant; see, e.g., Nocera and Garner [19].) Then each patient is given treatment in the order determined by his/her priority class. According to these triage procedures, patients are typically classified into priority classes based solely on their initial health conditions, ignoring other factors such as the size of the event (i.e., the number of patients waiting for care) and treatment times. Furthermore, these initially assigned priorities do not change with any changes in the system state (i.e., the number of patients at different criticality levels and time). However, recent work from the emergency response literature suggests that when determining priorities, the numbers of patients at different criticality levels should also be considered to achieve *the greatest good for the greatest number* (see Arnold, Tsai, Halpern, Smithline, Stok, and Ersoy [1] and Frykberg [12]). Indeed, one of the findings of our study is that making prioritization decisions dynamically over time, based on the system state, can bring significant improvements in the number of patients saved.

We first investigate conditions under which one can safely ignore the system state (i.e., the composition of patients at different criticality levels and time) in making prioritization decisions. For example, we find that if the patient with the shortest service time (in the sense of likelihood ratio ordering) also has the shortest lifetime (in the sense of hazard rate ordering), then priority should be given to this patient independently of the system state. However, in general, we show that there are clear benefits of updating priorities as the system state changes. We prove structural results for our optimal control problem and propose easy-to-implement and insightful heuristic policies that will help make these dynamic prioritization decisions. Our numerical experiments show that these heuristic policies perform quite well.

The outline of the article is as follows. In Section 2 we provide a literature review on scheduling, queuing, and optimal control problems that are most relevant to our problem. In Section 3 we formally define the optimal control problem under consideration and obtain analytical results on the characteristics of this problem and its solution for general service time and lifetime distributions. In Section 4 we consider the Markovian case in which the service times and lifetimes are exponentially distributed. Under certain conditions on the service and abandonment rates, we provide a complete characterization of the optimal policy for the Markovian case. Based on our analytical results, we propose two heuristic policies that are described in Section 5. A numerical study that tests the performance of these policies is presented in Section 6. Finally, Section 7 includes our concluding remarks and discussions.

Before we proceed further, we would like to mention two important points regarding this article. First, it should be noted that patient triage is an extremely complicated decision problem that involves a significant degree of human judgment in classifying patients and determining priorities. Any mathematical model of this problem that is also analytically tractable needs to be a significantly simplified version of the reality. Therefore, our objective here is not to develop policies that can be readily implemented but to provide some general insights on the problem. Second, although our main focus in this article is on the patient prioritization problem, our treatment of the problem is general enough to be relevant to other application areas as well (see Glazebrook, Ansell, Dunn, and Lumley [14] for examples of other applications). Hence, in order to keep the general appeal of our results, throughout the article we use the general queuing terminology (e.g., by referring to patients as "jobs" and operating times as "service times"). However, we discuss most of our results and observations in relation to the patient triage problem when they are especially relevant and significant within that context.

2. LITERATURE REVIEW AND CONTRIBUTION OF THE ARTICLE

Although the problem under consideration has relevance to several different areas in the literature, to our knowledge the only closely related study is conducted by

Glazebrook et al. [14]. In this section we first review the work by Glazebrook et al. [14] and then provide a survey of the relevant literature on stochastic scheduling and optimal control of queuing and clearing systems with reneging (abandonment).

Glazebrook et al. [14] considered scheduling of jobs in a clearing system with impatient jobs that abandon the system unless they receive service before their random due dates. One of the models studied in Glazebrook et al. [14] is concerned with the scheduling of jobs in a clearing system with $N \geq 2$ jobs having exponentially distributed times to abandonment, where jobs are characterized by their abandonment rates, mean service times, and rewards. The objective is to maximize the total expected reward earned. The authors proved that a policy resembling the "$c\mu$ rule" is asymptotically optimal in the class of nonpreemptive policies as the abandonment rates approach zero. The authors also provided numerical results on the performance of the suggested policy. To our knowledge, this heuristic method is the only other policy available in the literature that is alternative to those that we provide in this study. We discuss the policy by Glazebrook et al. [14] in more detail in Section 5, and we compare its performance with the performance of our heuristics by means of a numerical study in Section 6.

There are several other articles in the general context of scheduling in queuing systems with random or predetermined deadlines for jobs and queuing systems with expulsion (where the system controller can eject jobs); see, for example, Bhattacharya and Ephremides [3,4], Doytchinov, Lehoczky, and Shreve [10], Glazebrook [13], Jang and Klein [16], Jiang, Lewis, and Colin [17], Righter [25], Panwar, Towsley, and Wolf [20], Van Mieghem [28], Xu [33], and Zhao, Panwar, and Towsley [34]. Among these articles, we find Bhattacharya and Ephremides [3,4], Panwar et al. [20], and Zhao et al. [34] to be the most relevant to our work mainly because the performance measure of interest in these articles is the (weighted) number of tardy jobs (i.e., jobs whose deadline expires while waiting in the queue). Bhattacharya and Ephremides [4] and Panwar et al. [20] studied the scheduling problem under the assumption that the stochastic due date of a job is announced upon the arrival of the job and showed that a form of the "shortest-time-to-extinction" policy is optimal under certain conditions. Bhattacharya and Ephremides [3] and Zhao et al. [34], on the other hand, assumed that the decision-maker knows only the distribution of the due date of a job, not the exact due dates, at any decision epoch. In particular, Bhattacharya and Ephremides [3] showed that under the assumption of independent and identically distributed (i.i.d.) lifetimes, i.i.d. service times, and i.i.d. interarrival times (that are all mutually independent), the "earliest-arrival" policy is optimal if the lifetime distribution has a nondecreasing failure rate.

There are also several articles on the scheduling of jobs in (stochastic) clearing systems; see, for example, Boxma and Forst [5], Coffman, Flatto, Garey, and Weber [9], Emmons and Pinedo [11], Pinedo [22], Righter [24], Weber, Varaiya, and Walrand [31], and Weiss and Pinedo [32]. Among these articles, Boxma and Forst [5], Emmons and Pinedo [11], and Pinedo [22] are the most relevant to our work since they focus on the objective of minimizing the (weighted) number of tardy jobs. In these three articles, all jobs are available at time zero (except in Pinedo [22]), they have job-dependent stochastic due dates, and processing times are stochastic. The objective is to obtain a job sequence that minimizes the mean number of tardy jobs (i.e., jobs that are not completed by their due date). Two types of policies were considered in these articles: (static) *list scheduling policies* and *dynamic policies*. Under list scheduling policies, the decision-maker arranges all jobs into a list at time zero and is not allowed to change this list thereafter. Hence, when a list scheduling policy is applied, all jobs (even those jobs that are tardy) are processed. Under dynamic policies, on the other hand, the decision-maker is allowed to modify earlier decisions at any time as the new information becomes available. Pinedo [22] and Boxma and Forst [5] considered only list scheduling policies, whereas Emmons and Pinedo [11] considered both list scheduling and dynamic policies. In particular, Pinedo [22] showed that if the processing times of jobs are independent and exponentially distributed, their release dates (i.e., the times that the jobs are available for processing) are random, and their due dates are identically distributed, then the optimal static list policy sequences jobs in increasing order of mean processing times when the system has a single server. Boxma and Forst [5] studied the same problem (except that all jobs are available for processing at time zero) and identified optimal static list policies under several sets of conditions on due date and processing time distributions. One of the results proved by Boxma and Forst [5] stated that if all due dates are i.i.d. and a stochastic ordering exists among the processing time distributions, then the jobs should be sequenced according to the increasing stochastic ordering; that is, jobs with stochastically shortest processing times should be processed first. Emmons and Pinedo [11] studied the same problem but with multiple servers. They provided a set of special cases for which optimal dynamic policies or list policies can be identified. One of their results stated that if the processing times are i.i.d. exponential, and the due dates are independent, and can be ordered according to their failure rates, then the optimal preemptive dynamic policy is to process the jobs in the increasing order of their failure rates.

Although there is a clear connection between our work and the stochastic scheduling literature reviewed earlier, none of these articles considered the problem studied in this article. Almost all of the stochastic scheduling problems for clearing systems with an objective of minimization of number of tardy jobs focus on (static) list policies in contrast to dynamic policies that we consider in this article. In our model, jobs abandon the system once their "deadline" is reached and, thus, they cannot be processed afterward. Because of this property, the system state (i.e., the set of jobs in the system) changes after every job abandonment. Therefore, there are potential benefits of assigning priorities dynamically in time. Indeed, we provide several examples in the article to show that policies that are not dynamic in nature might perform very poorly in our case. Furthermore, the majority of articles on stochastic scheduling aim at identifying conditions that are generally in the form of an ordering condition among lifetimes and service times, under which a list scheduling policy is optimal. These conditions typically require the service times and lifetimes of jobs to be agreeably ordered (i.e., jobs with shorter lifetimes to also have shorter service times), which does not hold in many practical settings. In the first part of the article, we identify conditions under which state-independent policies are optimal. However, the main focus of the article is on obtaining good policies for all possible situations (especially when the service times and lifetimes are not agreeably ordered). Therefore, a significant portion of this article deals with obtaining near-optimal policies that would work well under all possible conditions.

Finally, we note that our model in general can be viewed as a queuing system with reneging (abandonment), and there is a vast literature on this topic. For some recent work on queuing systems with reneging, see, for example, Bae, Kim, and Lee [2], Brandt and Brandt [6,7], Choi, Kim, and Chung [8], Ward and Glynn [29], and Ward and Kumar [30].

3. GENERAL SERVICE TIME AND LIFETIME DISTRIBUTIONS

In this article, we consider a single-server clearing system initially having $N \geq 2$ jobs that might abandon the system before they receive service. Throughout the article, we refer to the maximum time a job can tolerate waiting in the queue as the *lifetime* of the job. If the lifetime of a job expires before it is taken into service, then the job abandons the system. We assume that a job that is already taken into service does not abandon the system. We also assume that the service is performed in a nonpreemptive manner; that is, once the server starts processing a job, it cannot start working on another job before completing the processing of the job that is already in service. This assumption is reasonable within the context of the patient triage problem, since it is generally not desirable to interrupt a medical procedure.

We let Y_i be the random variable representing the lifetime of job i at time zero and let S_i be the random variable representing the service time of job i for $i = 1, 2, \ldots, N$. We also define $C_\pi(t)$ to be the total number of jobs taken into service by time $t \geq 0$ when policy π is applied, where $\pi \in \Pi$ and Π is the set of all admissible, dynamic, and nonpreemptive scheduling (prioritization) policies. A dynamic prioritization policy is a collection of rules that determines which job the server takes into service at any given decision epoch based on the state of the system (i.e., the time of the decision epoch and the set of jobs in the system). (As we show in Proposition 1, idling is suboptimal and, hence, decision epochs are the time instances when service completions occur.) Our objective is to identify characteristics of policies that maximize $C_\pi(t)$ stochastically and, hence, maximize the expected total number of jobs served when the system is cleared.

In the remainder of this section we provide characterizations of the optimal control problem described earlier without making any distributional assumptions for service times and lifetimes. First, note that a standard coupling argument can be applied to prove that an idling policy (i.e., a policy under which the server might idle in the presence of jobs) can never be optimal.

PROPOSITION 1: *Any idling policy is suboptimal in the sense of maximizing $C_\pi(t)$ along any given sample path.*

Based on Proposition 1, in the rest of the article, we only consider nonidling policies. We next provide a complete characterization of the optimal policy when service times and lifetimes of jobs are "agreeably ordered" according to certain stochastic orders. We first provide definitions of three stochastic orders. Suppose that X and Y are two random variables. If $\Pr\{X > u\} \leq \Pr\{Y > u\}$, for all $u \in (-\infty, \infty)$, then X is said to be smaller than Y in the sense of *usual stochastic orders* (denoted by $X \leq_{\text{st}} Y$). On the other hand, if $\Pr\{X - v > u | X > v\} \leq \Pr\{Y - v > u | Y > v\}$, for all $u \geq 0$ and $v \in (-\infty, \infty)$, then X is said to be smaller than Y in the sense of *hazard rate orders* (denoted by $X \leq_{\text{hr}} Y$). Finally, let $f(t)$ and $g(t)$ be the densities or probability mass functions of X and Y, respectively. If $f(t)/g(t)$ is decreasing in t over the union of the supports of X and Y, then X is said to be smaller than Y in the sense of *likelihood ratio orders* (denoted by $X \leq_{\text{lr}} Y$). We will need the following lemma (see, e.g., Lemma 13.D.1 in Righter [23]) to prove Theorem 1.

LEMMA 1: *Let X and Y be two independent random variables. Then $X \leq_{\text{lr}} Y$ if and only if $(X|\min(X, Y) = m, \max(X, Y) = M) \leq_{\text{st}} (Y|\min(X, Y) = m, \max(X, Y) = M)$ for all $m \leq M$.*

In other words, given $m = \min(X, Y)$ and $M = \max(X, Y)$, we have that $X \leq_{\text{lr}} Y$ if and only if $\Pr\{X = m|m, M\} = \Pr\{Y = M|m, M\} \geq \Pr\{X = M|m, M\} = \Pr\{Y = m| m, M\}$. Note that $X \leq_{\text{lr}} Y \Rightarrow X \leq_{\text{hr}} Y \Rightarrow X \leq_{\text{st}} Y$.

THEOREM 1: *If $Y_1 \leq_{\text{hr}} Y_2 \leq_{\text{hr}} \cdots \leq_{\text{hr}} Y_N$ and $S_1 \leq_{\text{lr}} S_2 \leq_{\text{lr}} \cdots \leq_{\text{lr}} S_N$, then a nonpreemptive and nonidling policy that prioritizes the job with the smallest index at any decision epoch maximizes $\{C_\pi(t)\}_{t=0}^{\infty}$ in the sense of usual stochastic orders.*

PROOF: We use induction on the number of jobs, so suppose the theorem is true when there are $k \leq N - 1$ jobs and consider the case of N jobs. (Note that the theorem holds trivially when $N = 1$.) Suppose policy π does not comply with the smallest index (SI) policy given in Theorem 1 at time 0; that is, policy π takes job j into service at time 0, where $j \in \{2, 3, \ldots, N\}$. We will construct a policy γ, which serves job 1 at time 0 and for which $C_\pi(t) \leq C_\gamma(t)$ for all $t \geq 0$ along any given sample path. The SI policy, which agrees with γ for the first decision and is optimal thereafter from the induction hypothesis, will then have $C_{\text{SI}}(t) \geq C_\gamma(t)$ for all $t \geq 0$.

First, suppose π does not agree with the SI policy for some later decisions, as well as the decision at time 0. Then, from the induction hypothesis, we can let γ serve job j at time 0 and then agree with the SI policy for all later decisions, such that $C_\pi(t) \leq C_\gamma(t)$ for all $t \geq 0$. Therefore, without loss of generality, assume π agrees with the SI policy after the first decision and, in particular, that π serves job 1 after job j if job 1 is still available. Let γ serve job j after job 1 if job j is still available, and let it agree with π (and the SI policy) thereafter. Let Y_i^ρ, $\rho = \pi, \gamma$, denote the remaining lifetime of job i at time 0 under policy ρ, where $i = 1, 2, \ldots, k$. Note that by the stochastic ordering relation among remaining lifetimes of jobs, we can couple the random variables so that $Y_1^\pi = y_1 \leq y_j = Y_j^\gamma$. Because policy π (γ) serves job j (1) at time 0 and the job that is in service will not abandon, we do not need Y_j^π or Y_1^γ. Let $Y_i^\gamma = Y_i^\pi$ for all $i \neq 1, j$. Let S_i^ρ, $\rho = \pi, \gamma$, denote the service time of job i under policy ρ and let $S_i^\gamma = S_i^\pi$ for all $i \neq 1, j$. From Lemma 1, we can couple (S_1^π, S_j^π) with (S_1^γ, S_j^γ) so that $m = \min\{S_1^\pi, S_j^\pi\} = \min\{S_1^\gamma, S_j^\gamma\} \leq M = \max\{S_1^\pi, S_j^\pi\} = \max\{S_1^\gamma, S_j^\gamma\}$ and either $S_j^\pi = S_1^\gamma =: a \in \{m, M\}$ and $S_1^\pi = S_j^\gamma =: b \in \{m, M\} \backslash \{a\}$ (Case I) or $S_1^\pi = S_1^\gamma = m \leq S_j^\pi = S_j^\gamma = M$ (Case II).

Case Ia: We first consider the case where $a < y_1$. Then π will serve job 1 at time a and γ will serve job j. From time $a + b$ on, the states will be the same under both policies, and we have $C_\gamma(t) = C_\pi(t)$ for all $t \geq 0$.

Case Ib: Now, suppose $y_1 \leq a < y_j$. Let $C'(t), t \geq a$, be the number of jobs taken into service under the SI policy starting from time a and assuming the first job completion occurs at time a, and with the state the same as under π at time a and with all random variables coupled to be the same as those under π, except that we assume job 1 is still present at time a, with remaining life from time a of $Y_1(a) =_{\text{st}} (Y_1 - a | Y_1 > a)$. Let π' be the corresponding policy. Then, arguing as in Case Ia, $C_\gamma(t) = C'(t)$ for $t \geq a$, and $C_\gamma(t) = C_\pi(t)$ for $0 \leq t \leq a$.

It remains to show that $C'(t) \geq C_\pi(t)$ for $t \geq a$. Define a new policy π'' that starts in the same state as π' at time a and that follows the SI policy except that it serves job 1 last (at time τ say) if it is still available. Then, for $a \leq t \leq \tau$, $C_{\pi''}(t) = C_\pi(t)$ and for $t \geq \tau$, $C_{\pi''}(t) \geq C_\pi(t)$. Since π' agrees with the SI policy, from the induction hypothesis, $C'(t) \geq C_{\pi''}(t)$ for all $t \geq a$.

Case Ic: Suppose $y_j \leq a$. Then, from time a on, the states will be the same under both policies and we have $C_\gamma(t) = C_\pi(t)$ for all $t \geq 0$.

Case IIa: Suppose $M < y_1$ and, therefore, $m < y_j$. At time $m + M$, the states under π and γ are the same, so $C_\gamma(t) = C_\pi(t)$ for $t \geq m + M$. Before time $m + M$, we have $C_\gamma(t) = C_\pi(t) = 1$ for $0 \leq t < m$, $C_\gamma(t) = 2 > C_\pi(t) = 1$ for $m \leq t < M$, and $C_\gamma(t) = C_\pi(t) = 2$ for $M \leq t < m + M$. Thus, we have $C_\gamma(t) \geq C_\pi(t)$ for all $t \geq 0$.

Case IIb: Suppose $m < y_j$ and $y_1 \leq M$. Then we can argue as in Case Ib that $C_\gamma(t) \geq C_\pi(t), t \geq 0$.

Case IIc: We finally consider the case where $y_j \leq m$ and, therefore, $y_1 < M$. Let $C'(t)$ be the number of jobs taken into service by time t under the SI policy starting from time m with the state the same as under π at time m and with all random variables coupled to be the same as those under π, except that we assume job j finishes at time m instead of time M, and let π' be the corresponding policy. Then $C'(t) = C_\gamma(t)$ for $t \geq m$ and $C_\gamma(t) = C_\pi(t)$ for $0 \leq t < m$, so we need to only show that $C'(t) \geq C_\pi(t)$ for $t \geq m$. Note that $C_\pi(t), t \geq m$, is the same as if we started with the same state as π' but idled from time m to M and then followed the SI policy and did not count the job completion until time M, so $C'(t) \geq C_\pi(t)$ for $t \geq m$ from Proposition 1 and the induction hypothesis. ■

Theorem 1 implies that if jobs can be ordered in such a way that the one with the shortest lifetime in the sense of hazard rate orders also has the shortest service time in the sense of likelihood ratio orders, then giving this job priority for service maximizes the expected total number of jobs served. We next provide a definition that we will frequently use in the remainder of the article.

Definition 1: *If $Y_i \leq_{\mathrm{hr}} Y_j$, then job i is said to be more* time-critical *than job j, for $i,j \in \{1,2,\ldots,N\}$ and $i \neq j$.*

Using this definition, Theorem 1 states that *regardless of the system state, time-critical jobs with shorter service times in the sense of likelihood ratio orders should always be given priority for service.* Theorem 1 provides us with a criterion as to what makes a job a top-priority job. This is especially important in the context of patient triage, since it implies that a patient with a certain injury can be given the highest priority irrespective of the number of other patients if his/her lifetime and service time are shorter than those for any other patient in the sense of hazard rate and likelihood ratio orders, respectively.

Remark 1: In certain applications, serving different jobs might not bring the same amount of benefit. For example, in the context of patient triage, certain operations might be riskier than others in that the chance of survival of a patient after such an operation might be lower than that for other operations. In such a situation, instead of maximizing the number of patients taken into service, it is more sensible to maximize the average number of patients saved by taking into account risk factors associated with the operation of each patient. To formulate such an objective function, let θ_i be the fixed reward earned by serving job i, for $i = 1,2,\ldots,N$, and let $C_\pi(t)$ be the total reward earned by time t under policy π. (For the patient triage problem, θ_i might denote the probability of survival of patient i after he/she is taken into operation.) Proposition 1 trivially holds under this new performance measure. Furthermore, we can also show that Theorem 1 holds if in addition to the stochastic ordering conditions on the service time and lifetimes given in Theorem 1, we also have $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_N$. In other words, *if serving jobs with stochastically smaller service times and lifetimes brings larger rewards, then these jobs should be given higher priority no matter what the state of the system is.*

In most applications, it might not be practically feasible to characterize each job in the system with its own lifetime and service distribution. For example, for the patient triage problem, it is common practice to classify patients into at most three or four categories. Especially when there is a time pressure, as in the case of mass casualty events, patients' conditions are quickly assessed right after the event and then they are classified into a small number of priority classes even though each patient has his/her own unique injuries. Based on this, in the remainder of the article we will consider the case in which jobs are classified into only two types, each type having its own lifetime and service time distribution. Let m_i be the number of type $i \in \{1,2\}$ jobs initially in the system such that $m_1 + m_2 = N$.

For the problem with two types of job in the system, Theorem 1 implies that a job type with shorter lifetimes (in the sense of hazard rate orders) and shorter service times (in the sense of likelihood ratio orders) should always receive priority for service no matter how many jobs from each type are present in the system. Although this result provides us a criterion as to what makes a type a top-priority class, a more interesting (perhaps a more common) case is when time-critical jobs have longer service times. In Sections 4–6, our main focus will be on this case.

4. EXPONENTIAL SERVICE TIME AND LIFETIME DISTRIBUTIONS

In this section we study the same model described in Section 3 except that we now assume that jobs are categorized into two classes and that their service times and lifetimes are exponentially distributed. For $i = 1,2$, let $\mu_i > 0$ and $r_i > 0$ be the service rate and lifetime rate for a type i job, respectively. Let also $D_\pi(m_1,m_2)$ denote the expected total number of jobs taken into service when the system is cleared if scheduling (prioritization) policy $\pi \in \Pi$ is applied and m_i jobs of type $i \in \{1,2\}$ are initially in the system. We use a dynamic programming formulation to find an optimal or near-optimal solution to the optimization problem stated as

$$\max_{\pi \in \Pi} D_\pi(m_1,m_2).$$

We define the state of the system as $(x_1,x_2;Q)$, where x_i is the number of type i jobs in the system and $Q \in \{P_1,P_2,R\}$ is the status of the server. When $Q = P_i$, it means that the server is processing a job of type $i \in \{1,2\}$, and when $Q = R$, it means that the server is ready to process a new job. Let $V(x_1,x_2;Q)$ be the maximum expected number of jobs served starting from state $(x_1,x_2;Q)$. Then, using the convention that $V(0,x_2;P_1) = V(x_1,0;P_2) = 0$, where $x_i = 0,1,\ldots,m_i$ for $i = 1,2$ and the notation that $\mathbb{I}_A$ is the indicator function of event A, the dynamic programming equations are given as follows:

$$\begin{aligned} V(x_1,x_2;R) &= \mathbb{I}_{\{x_1+x_2\geq 1\}} + \max\{V(x_1,x_2;P_1), V(x_1,x_2;P_2)\}, \\ &\quad \forall x_i = 0,1,\ldots,\ m_i, i = 1,2; \\ V(x_1,x_2;P_1) &= (\mu_1 V(x_1-1,x_2;R) + (x_1-1)r_1 V(x_1-1,x_2;P_1) \\ &\quad + x_2 r_2 V(x_1,x_2-1;P_1))/(\mu_1 + (x_1-1)r_1 + x_2 r_2), \\ &\quad \forall x_1 = 1,2,\ldots,m_1, x_2 = 0,1,\ldots,m_2; \end{aligned}$$

$$\begin{aligned} V(x_1,x_2;P_2) = {} & (\mu_2 V(x_1,x_2-1;R) + x_1 r_1 V(x_1-1,x_2;P_2) \\ & + (x_2-1) r_2 V(x_1,x_2-1;P_2))/(\mu_2 + x_1 r_1 + (x_2-1) r_2), \\ & \forall x_1 = 0,1,\ldots,m_1, x_2 = 1,2,\ldots,m_2. \end{aligned}$$

We will use this dynamic programming formulation to identify optimal or near-optimal dynamic prioritization policies that maximize the expected total number of jobs served. First, note that Theorem 1 implies that for this Markovian system, if type 1 jobs have shorter mean service times and lifetimes than type 2 jobs, then type 1 jobs should be given priority for service regardless of the system state. This is in agreement with one's intuition because there seems to be no reason to give priority to less time-critical jobs when serving them takes at least as much time as serving more time-critical jobs. Thus, the more interesting (and realistic) case is when time-critical jobs have longer mean service times (i.e., $\mu_1 < \mu_2$ and $r_1 > r_2$). We focus on this case in the remainder of this section.

From our numerical experiments, we observed that the optimal policy gives higher priority to jobs that are less time-critical and that have shorter service times, when the number of jobs from each type is sufficiently large. As the number of jobs from each type drops, giving priority to more time-critical jobs that require longer service times becomes a better strategy. Figure 1 demonstrates this behavior with $\mu_1 < \mu_2$ and $r_1 > r_2$. In the context of emergency response planning, this suggests that depending on the number of patients at different criticality levels, it might be better to give priority to less time-critical patients. This contradicts the general belief and the common practice that gives priority to time-critical patients at all times, and it strongly supports the argument that *when the number of casualties is high, emergency resources should be allocated to less time-critical patients with shorter treatment times, as the objective is to do the greatest good for the greatest number.* (Indeed, as $m_1 \to \infty$ and $m_2 \to \infty$,

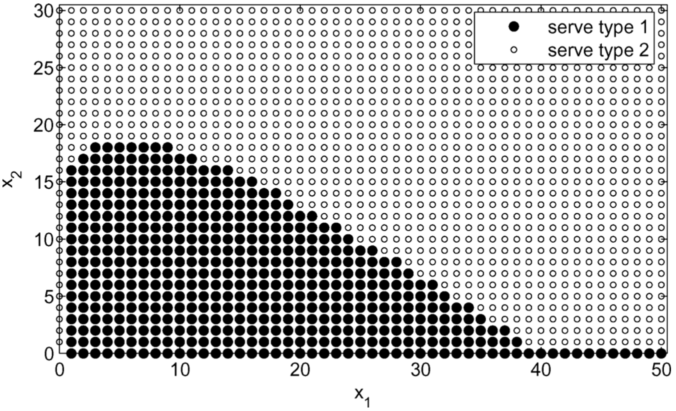


FIGURE 1. A typical structure for the optimal policy when $\mu_1 < \mu_2$ and $r_1 > r_2$.

our objective essentially becomes to maximize the throughput and, hence, shortest-expected-processing-time-first rule (SEPT) becomes the optimal policy.) In order to implement this principle, one needs to first answer the following question: What number of jobs is considered to be sufficiently high to follow this principle? In the remainder of this section, we present some results aiming at answering this question.

We first present a proposition that gives a necessary condition for an index policy to be optimal. An index policy is a state-independent policy that gives priority based only on job type, or index, at any state (x_1, x_2). For example, SEPT is an index policy that always gives priority to the job with the largest value of μ. Similarly, the $r\mu$ rule of Proposition 2 is the index policy that always gives priority to the job with the largest value of $r\mu$. Note that the $r\mu$ rule resembles the well-known $c\mu$ rule in queuing theory since it gives priority to job type i with the highest value of $r_i\mu_i$. (For a comprehensive literature review on the $c\mu$ rule, the interested reader is referred to Van Mieghem [27].)

Proposition 2: *If there is an optimal policy among the set of index policies, it must agree with the $r\mu$ rule.*

Proof: It is easy to check that $V(1,1;P_1) \geq V(1,1;P_2)$ if and only if $r_1\mu_1 \geq r_2\mu_2$. ■

Note that Theorem 1 is consistent with Proposition 2 because $r_1 \geq r_2$ and $\mu_1 \geq \mu_2$, which implies that $r_1\mu_1 \geq r_2\mu_2$. We next present four results that will be used in identifying optimal or near-optimal policies for the case when $\mu_1 < \mu_2$ and $r_1 > r_2$. More specifically, Proposition 3, which is a rather technical result, will be instrumental in the development of two heuristic policies, as we explain in Section 5. Proposition 3 is also used in proving Propositions 4–6, which provide sufficient conditions under which the optimal policy can be characterized.

In the proof of Proposition 3, we use the following lemma, which states that for a fixed number of type 1 and type 2 jobs, x_1 and x_2 in queue, we prefer to have the job in service be a job with a smaller mean service time. This makes sense because the remaining lifetime of the job in service is no longer relevant.

Lemma 2: *If $\mu_1 \leq \mu_2$, then we have $V(x_1+1, x_2; P_1) \leq V(x_1, x_2+1; P_2)$ for all $x_1 = 0, 1, \ldots, m_1$ and $x_2 = 0, 1, \ldots, m_2$.*

Proof: We can couple the processing times of the jobs in service for the two states such that $S_2 \leq S_1$ with probability 1, where S_i denotes the processing time of a type i job. Let $V_0(x_1, x_2+1; P_2)$ be the value function when the starting state is $(x_1, x_2+1; P_2)$ and we idle from time S_2 to S_1 and then follow the optimal policy. Then, from Proposition 1, we have $V(x_1, x_2+1; P_2) \geq V_0(x_1, x_2+1; P_2) = V(x_1+1, x_2; P_1)$. ■

Remark 2: Note that in the proof of Lemma 2, we have actually shown a stronger result. Suppose we have an arbitrary number of types of jobs, with arbitrary lifetime and service time distributions, and let $V_t(\mathbf{x}; P_i)$ be the value function when the numbers of each type of job are given by the vector $\mathbf{x}$, the current time is t, and a job of type i starts service at time t. Let $\mathbf{e_i}$ be a vector with a 1 in the ith position and 0s elsewhere. Then we have shown that if $S_i \geq_{st} S_j$, then $V_t(\mathbf{x} + \mathbf{e_i}; P_i) \leq V_t(\mathbf{x} + \mathbf{e_j}; P_j)$.

We are now ready to prove Proposition 3. Proposition 3 gives us conditions for the monotonicity of the policy in the state. To be more specific, it gives us conditions such that if it is optimal to serve a type i job in state $(x_1, x_2 - 1)$ and $(x_1 - 1, x_2)$, then it is also optimal to serve a type i job in state (x_1, x_2). Note that these conditions require that the preferred job agrees with the $r\mu$ rule.

PROPOSITION 3:

(i) Suppose $\mu_1 < \mu_2 \leq r_2$, $\mu_1 \leq r_1$ *and fix* $x_1 \geq 1$, $x_2 \geq 1$. *If*

$$x_1 r_1 + x_2 r_2 \geq \frac{r_1\mu_2 - r_2\mu_1}{\mu_2 - \mu_1} \tag{1}$$

and if $V(x_1 - 1, x_2; P_2) \geq V(x_1 - 1, x_2; P_1)$ *and, for* $x_2 \neq 1$, *if* $V(x_1, x_2 - 1; P_2) \geq V(x_1, x_2 - 1; P_1)$, *then* $V(x_1, x_2; P_2) \geq V(x_1, x_2; P_1)$.

(ii) Suppose $r_1 \leq \mu_1 < \mu_2$, $r_2 \leq \mu_2$ *and fix* $x_1 \geq 1$, $x_2 \geq 1$. *If*

$$x_1 r_1 + x_2 r_2 \leq \frac{r_1\mu_2 - r_2\mu_1}{\mu_2 - \mu_1} \tag{2}$$

and if $V(x_1, x_2 - 1; P_1) \geq V(x_1, x_2 - 1; P_2)$ *and, for* $x_1 \neq 1$, *if* $V(x_1 - 1, x_2; P_1) \geq V(x_1 - 1, x_2; P_2)$, *then* $V(x_1, x_2; P_1) \geq V(x_1, x_2; P_2)$.

PROOF:

(i) We have

$$\begin{aligned} V(x_1, x_2; P_2) &= (\mu_2 V(x_1, x_2 - 1, R) + x_1 r_1 V(x_1 - 1, x_2; P_2) \\ &\qquad + (x_2 - 1) r_2 V(x_1, x_2 - 1; P_2))/(\mu_2 + x_1 r_1 + (x_2 - 1) r_2) \\ &\geq (\mu_2(1 + V(x_1, x_2 - 1, P_1)) + x_1 r_1 V(x_1 - 1, x_2; P_2) \\ &\qquad + (x_2 - 1) r_2 V(x_1, x_2 - 1; P_1))/(\mu_2 + x_1 r_1 + (x_2 - 1) r_2), \end{aligned}$$

where the inequality follows because, for the first term, $V(x_1, x_2 - 1, R) \geq 1 + V(x_1, x_2 - 1, P_1)$, and, for the last term, either $x_2 = 1$, so the inequality is trivial, or $V(x_1,x_2-1;P_2) \geq V(x_1,x_2-1;P_1)$. Similarly,

$$\begin{aligned}V(x_1,x_2;P_1) &= (\mu_1 V(x_1-1,x_2,R) + (x_1-1)r_1 V(x_1-1,x_2;P_1)\\ &\quad + r_2x_2V(x_1,x_2-1;P_1))/(\mu_1+(x_1-1)r_1+x_2r_2)\\ &\leq (\mu_1(1+V(x_1-1,x_2,P_2)) + (x_1-1)r_1V(x_1-1,x_2;P_2)\\ &\quad + r_2x_2V(x_1,x_2-1;P_1))/(\mu_1+(x_1-1)r_1+x_2r_2).\end{aligned}$$

Hence,

$$\begin{aligned}&V(x_1,x_2;P_2) - V(x_1,x_2;P_1)\\ &\geq \left(\frac{x_1r_1}{\mu_2+x_1r_1+(x_2-1)r_2} - \frac{\mu_1+(x_1-1)r_1}{\mu_1+(x_1-1)r_1+x_2r_2}\right)\\ &\quad \times V(x_1-1,x_2;P_2)\\ &\quad + \left(\frac{\mu_2+(x_2-1)r_2}{\mu_2+x_1r_1+(x_2-1)r_2} - \frac{x_2r_2}{\mu_1+(x_1-1)r_1+x_2r_2}\right)\\ &\quad \times V(x_1,x_2-1;P_1)\\ &= \frac{(x_1-1)r_1(r_2-\mu_2)+(x_2-1)r_2(r_1-\mu_1)+r_1r_2-\mu_1\mu_2}{(\mu_1+(x_1-1)r_1+x_2r_2)(\mu_2+x_1r_1+(x_2-1)r_2)}\\ &\quad \times (V(x_1-1,x_2;P_2)-V(x_1,x_2-1;P_1))\\ &\geq 0,\end{aligned}$$

where the first inequality follows from condition (1) of the proposition and the second follows from Lemma 1 and the conditions that $\mu_1 < \mu_2 \leq r_2$ and $\mu_1 \leq r_1$.

(ii) We have

$$\begin{aligned}V(x_1,x_2;P_1) &= (\mu_1 V(x_1-1,x_2;R) + (x_1-1)r_1 V(x_1-1,x_2;P_1)\\ &\quad + x_2r_2V(x_1,x_2-1;P_1))/(\mu_1+(x_1-1)r_1+x_2r_2)\\ &\geq (\mu_1(1+V(x_1-1,x_2;P_2)) + (x_1-1)r_1V(x_1-1,x_2;P_2)\\ &\quad + x_2r_2V(x_1,x_2-1;P_1))/(\mu_1+(x_1-1)r_1+x_2r_2),\end{aligned}$$

where the inequality follows because, for the first term, $V(x_1-1,x_2;R) \geq 1+V(x_1-1,x_2;P_2)$, and, for the second term, either $x_1 = 1$, so the inequality is trivial, or $V(x_1-1,x_2;P_1)\geq V(x_1-1,x_2;P_2)$. Similarly,

$$\begin{aligned} V(x_1,x_2;P_2) &= (\mu_2 V(x_1,x_2-1;R)+x_1r_1V(x_1-1,x_2;P_2)\\ &\qquad + (x_2-1)r_2V(x_1,x_2-1;P_2))/(\mu_2+x_1r_1+(x_2-1)r_2)\\ &\leq (\mu_2(1+V(x_1,x_2-1;P_1))+x_1r_1V(x_1-1,x_2;P_2)\\ &\qquad + (x_2-1)r_2V(x_1,x_2-1;P_1))/(\mu_2+x_1r_1+(x_2-1)r_2). \end{aligned}$$

Hence,

$$\begin{aligned} &V(x_1,x_2;P_1)-V(x_1,x_2;P_2)\\ &\geq \left(\frac{\mu_1+(x_1-1)r_1}{\mu_1+(x_1-1)r_1+x_2r_2}-\frac{x_1r_1}{\mu_2+x_1r_1+(x_2-1)r_2}\right)\\ &\quad \times V(x_1-1,x_2;P_2)\\ &\quad +\left(\frac{x_2r_2}{\mu_1+(x_1-1)r_1+x_2r_2}-\frac{\mu_2+(x_2-1)r_2}{\mu_2+x_1r_1+(x_2-1)r_2}\right)\\ &\quad \times V(x_1,x_2-1;P_1)\\ &=\frac{(x_1-1)r_1(\mu_2-r_2)+(x_2-1)r_2(\mu_1-r_1)+\mu_1\mu_2-r_1r_2}{(\mu_1+(x_1-1)r_1+x_2r_2)(\mu_2+x_1r_1+(x_2-1)r_2)}\\ &\quad \times (V(x_1-1,x_2;P_2)-V(x_1,x_2-1;P_1))\\ &\geq 0, \end{aligned}$$

where the first inequality follows from condition (2) and the second follows from Lemma 1 and the conditions that $r_1\leq\mu_1<\mu_2$ and $r_2\leq\mu_2$. ■

PROPOSITION 4: *If $\mu_1<\mu_2\leq r_2$ and $\mu_1\leq r_1$, then for every $x_1\geq 0$, the optimal policy has a threshold, $t(x_1)$ (which might be infinite), such that for all $x_2\geq t(x_1)$, it is optimal to serve a type 2 job.*

PROOF: We prove the result by induction. Since serving a type 2 job is optimal for $x_1=0$, part (i) of Proposition 3 implies that if there is a state $(1,b)$ such that serving a type 2 job is optimal, then it is also optimal in all states $(1,x_2)$ such that $x_2\geq \max(b,(r_1\mu_2-r_2\mu_1)/[r_2(\mu_2-\mu_1)]-r_1/r_2)$. Hence, there exists a threshold $t(1)$ such that for all $x_2\geq t(1)$, it is optimal serve a type 2 job in states $(1,x_2)$. Now, suppose that serving a type 2 job is optimal in states (a,x_2) for all $x_2\geq t(a)$. Then, if there exists a state $(a+1,b)$ such that $b\geq t(a)-1$, where serving a type 2 job is optimal, by part (i) of Proposition 3 we see that serving a type 2 job is also optimal in all states $(a+1,x_2)$ such that $x_2\geq\max(b,(r_1\mu_2-r_2\mu_1)/[r_2(\mu_2-\mu_1)]-r_1(a+1)/r_2)$. This completes the proof. ■

In Proposition 5, we show that if in addition to the conditions of Proposition 4 we also have that prioritizing type 2 jobs is agreeable with the $r\mu$ rule, then serving type 2 jobs is always optimal; that is, the threshold in Proposition 4, $t(x_1)$, is 1 for all $x_1 \geq 0$.

Proposition 5: *If $\mu_1 < \mu_2 \leq r_2$, $\mu_1 \leq r_1$, and $r_1\mu_1 \leq r_2\mu_2$, then $V(x_1, x_2; P_2) \geq V(x_1, x_2; P_1)$ for all $x_i = 1, 2, \ldots, m_i$, $i = 1, 2$.*

Proof: First, note that condition (1) is satisfied for all $x_1, x_2 \geq 1$ since $\mu_1 < \mu_2$ and $r_1\mu_1 \leq r_2\mu_2$. Second, $r_1\mu_1 \leq r_2\mu_2$ implies that $V(1,1;P_1) \leq V(1,1;P_2)$ and, hence, part (i) of Proposition 3, which holds since $\mu_1 < \mu_2 \leq r_2$ and $\mu_1 \leq r_1$, yields that $V(x_1, 1; P_1) \leq V(x_1, 1; P_2)$ for all $x_1 \geq 1$. Finally, note that, by convention, $V(0, x_2; P_1) = 0 \leq V(0, x_2; P_2)$ for all $x_2 \geq 1$. Combining this result with the fact that $V(x_1, 1; P_1) \leq V(x_1, 1; P_2)$ for all $x_1 \geq 1$ completes the proof once we apply part (i) of Proposition 3. ■

Proposition 6: *If $r_1 \leq \mu_1 < \mu_2$ and $r_2 \leq \mu_2$, then for every $x_1 \geq 1$, the optimal policy has a threshold $t(x_1)$ such that for all $x_2 \leq t(x_1)$, it is optimal to serve a type 1 job, and $t(x_1)$ is given by*

$$t(x_1) = \frac{r_1\mu_2 - r_2\mu_1}{r_2(\mu_2 - \mu_1)} - \frac{x_1 r_1}{r_2}.$$

Proof: First, note that condition (2) is equivalent to $x_2 \leq t(x_1)$. For the condition to be satisfied at all (for some $x_1 \geq 1$, $x_2 \geq 1$), we must have $r_2\mu_2 \leq r_1\mu_1$, which implies that $V(1,1;P_1) \geq V(1,1;P_2)$. Hence part (ii) of Proposition 3 yields that $V(1, x_2; P_1) \geq V(1, x_2; P_2)$ for all $1 \leq x_2 \leq t(1)$. Note also that by convention, $V(x_1, 0; P_2) = 0 \leq V(x_1, 0; P_1)$ for all $x_1 \geq 1$. It is easy to see that $t(x_1)$ is non-increasing in x_1. Hence, for all $x_2 \leq t(x_1)$, we have $V(x_1 - 1, x_2; P_1) \geq V(x_1 - 1, x_2; P_2)$. Then, applying part (ii) of Proposition 3 completes the proof. ■

Suppose now that $\mu_1 < \mu_2$ and $r_1 > r_2$ (i.e., type 1 jobs, which are more time-critical have larger mean service times). For this case, Propositions 5 and 6 yield the following conclusions:

- By Proposition 5, if $r_i \geq \mu_i$ for $i = 1, 2$, and $r_2\mu_2 \geq r_1\mu_1$, then it is always optimal to serve a type 2 job, which is less time-critical and has a smaller mean service time than a type 1 job. The first condition means that both types of jobs abandon the system at higher rates than their service rates. On the other hand, the second condition is the necessary and sufficient condition for the optimality of serving a type 2 job when $x_1 = x_2 = 1$. Hence, *if jobs abandon the system at higher rates (relative to service rates) and if it is better to serve a type 2 job when there is only one of each type, then type 2 jobs should always be given priority regardless of the system state.*
- Proposition 6 implies that if $r_i \leq \mu_i$ for $i = 1, 2$, then it is optimal to serve a type 1 job, which is more time-critical and has a larger mean service time,

> when the number of jobs in the system satisfy condition (2). Note that condition (2) generally holds when x_1 and x_2 are small. Hence, Proposition 6 implies that *when service is fast (relative to abandonments) and the number of jobs in the system are sufficiently small, then the time-critical but slower jobs should be given priority for service.*

We end this section with a conjecture. Based on extensive numerical experiments, we believe that Proposition 5 holds under a set of less restrictive conditions.

Conjecture 1: If $\mu_1 < \mu_2$ and $r_1\mu_1 \leq r_2\mu_2$, then $V(x_1, x_2; P_2) \geq V(x_1, x_2; P_1)$ for all $x_i = 1, 2, \ldots, m_i$, $i = 1, 2$.

In other words, we conjecture that as long as the $r\mu$ rule is agreeable with SEPT, the $r\mu$ rule is optimal among all policies in Π.

Although we were able to prove this conjecture for states $(1, x_2)$, where $x_2 = 1, 2, \ldots, m_2$, and states $(x_1, 1)$, where $x_1 = 1, 2, \ldots, m_1$, proving it in its most general form appears to be a significant challenge. We defer the proof of Proposition 7 to the Appendix.

PROPOSITION 7: *The $r\mu$ rule is optimal among all policies within* Π *when it is agreeable with SEPT and either x_1 or x_2 is at most 1.*

5. HEURISTIC POLICIES

In this section, we propose two new heuristic policies—namely the *triangular* and *rectangular heuristics*—for state-dependent job prioritization decisions when there are two types of job in the system. We also describe two other heuristics, both of which are static policies in the sense that under these policies, priorities do not change with the system state. In Section 6, we compare the performances of these four heuristics along with the performance of the optimal policy by means of a numerical study.

When time-critical jobs have shorter service times (i.e., $r_1 \geq r_2$ and $\mu_1 \geq \mu_2$ using the notation of Section 4), Theorem 1 shows that the optimal policy should always give priority to time-critical jobs (type 1 jobs) regardless of the system state. On the other hand, when time-critical jobs have longer service times (i.e., $r_1 > r_2$ and $\mu_1 < \mu_2$), it is generally not clear what the best prioritization policy is (except for the cases identified in Propositions 5 and 6). Therefore, we develop heuristic policies for this particular case and assume that $r_1 > r_2$ and $\mu_1 < \mu_2$ in the following discussion. In the following, we describe how the heuristic policies work when service times and lifetimes are exponentially distributed. However, these heuristics can be also applied in more general settings, as we discuss in Section 6.1.2.

The heuristic procedures that we propose—namely the triangular and rectangular heuristics—are primarily based on the structural results given in Section 4 and our observations on the structure of the optimal policy from numerical experiments. We observed from our experiments that the optimal policy divides the state space into at most two regions, as shown in Figure 1. In general, the optimal policy does not have any obvious monotonic structure, as Figure 1 reveals. However, in many cases, the optimal policy is monotone in the number of type 1 and type 2 jobs. More specifically, if it is optimal to serve a type 1 job in state (x_1, x_2), it is also optimal to serve a type 1 job in state $(x_1 - 1, x_2)$ or in state $(x_1, x_2 - 1)$; similarly, if it is optimal to serve a type 2 job in state (x_1, x_2), it is also optimal to serve a type 2 job in state $(x_1, x_2 + 1)$ or in state $(x_1 + 1, x_2)$. The triangular heuristic mimics this behavior of the optimal policy with some support from our theoretical results. The rectangular heuristic is based on the triangular heuristic but is easier to implement.

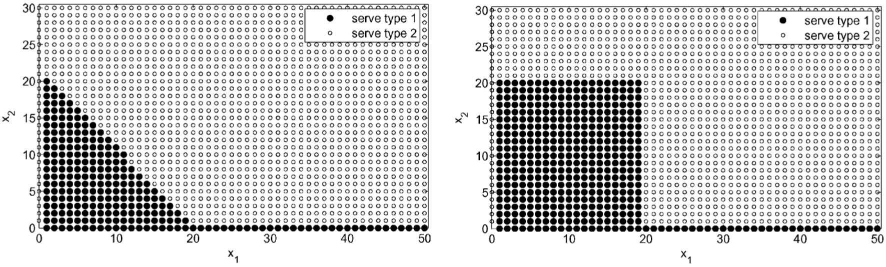
We next describe these heuristic policies in detail.

1. **Triangular heuristic:** This heuristic is primarily based on Proposition 3. Condition (1), or, equivalently, condition (2), of Proposition 3 divides the state space into two regions, one of which roughly has the shape of a triangle. When the system is in state (x_1, x_2), the triangular heuristic gives priority to type 1 jobs if (x_1, x_2) falls inside the triangle (i.e., if (x_1, x_2) satisfies condition (2)) and it gives priority to type 2 jobs otherwise. Note that Proposition 3 does not give a complete characterization of the optimal policy, but whether condition (1) or (2) holds in Proposition 3 can be seen as an indicator of preference toward one type over the other. Figure 2a shows how the triangular heuristic works for the example studied in Figure 1.

   The triangular heuristic is also insightful. To illustrate, we first rewrite condition (2) as

$$\frac{(x_1 - 1)r_1 + x_2 r_2}{\mu_1} \leq \frac{x_1 r_1 + (x_2 - 1)r_2}{\mu_2}. \tag{3}$$

   The left-hand side of condition (3) is the mean number of abandonments during the service of a type 1 job, whereas the right-hand side is the mean number of abandonments during the service of a type 2 job in the same state. Hence, in some sense, the triangular heuristic (myopically) gives priority to jobs with a smaller mean number of abandonments during service.



(a) Triangular heuristic (b) Rectangular heuristic

FIGURE 2. Examples on the structure of the triangular and rectangular heuristics.

Note also that the triangular heuristic is in agreement with all of our analytical results on the characterization of the optimal policy. More specifically, when $r_1\mu_1 \leq r_2\mu_2$ and $\mu_1 < \mu_2$, the heuristic gives priority to type 2 jobs independently of the system state, which is consistent with Propositions 5 and 7 and Conjecture 1. Similarly, the heuristic is also in agreement with Proposition 6. Finally, when $r_1 \geq r_2$ and $\mu_1 \geq \mu_2$, the heuristic gives higher priority to type 1 jobs in all states, which is consistent with Theorem 1.

2. **Rectangular heuristic:** The rectangular heuristic is based on the triangular heuristic and works similarly. The idea is simply to expand the triangular region associated with the triangular heuristic to a rectangle by adding another point to the existing three points and connecting the four points on the state space. Figure 2b shows how the rectangular heuristic works for the case studied in Figure 1. One advantage of the rectangular heuristic is its simple structure. It is completely characterized by two threshold values (i.e., the length and the width of the rectangle). More precisely, in a given state (x_1, x_2), if $r_1 > r_2$ and $\mu_1 < \mu_2$, then the rectangular heuristic gives priority to type 1 jobs if and only if $1 \leq x_1 \leq \min\{m_1, T_1\}$ and $1 \leq x_2 \leq \min\{m_2, T_2\}$, where

$$T_1 = \frac{\mu_2(r_1 - r_2)}{r_1(\mu_2 - \mu_1)}$$

and

$$T_2 = \frac{\mu_1(r_1 - r_2)}{r_2(\mu_2 - \mu_1)}.$$

T_1 is obtained by plugging $x_1 = T_1$ and $x_2 = 1$ in condition (2) and solving it as an equality, and T_2 is obtained similarly by plugging $x_1 = 1$ and $x_2 = T_2$ in the same inequality. Since we assume that $r_1 > r_2$ and $\mu_1 < \mu_2$, T_1 and T_2 will always be nonnegative. Note that when $r_1\mu_1 \leq r_2\mu_2$ and $\mu_1 < \mu_2$, the rectangular heuristic prioritizes type 2 jobs in all states for which $x_2 \geq 1$, which is consistent with Propositions 5 and 7 and Conjecture 1. On the other hand, if $r_1\mu_1 > r_2\mu_2$ and $\mu_1 < \mu_2$, the rectangular heuristic gives a state-dependent policy. In this case, the threshold for type 2 jobs is higher than the threshold for type 1 jobs (i.e., $1 < T_1 < T_2$).

3. ***rμ*-Heuristic:** This heuristic is studied by Glazebrook et al. [14] for a slightly different version of our problem. More specifically, the authors considered scheduling of jobs in a clearing system with $N \geq 2$ jobs initially. Each job is characterized by its service rate μ, abandonment rate r, and the positive reward θ that it brings after service completion. (See Section 2 for more details on the work by Glazebrook et al. [14].) The authors proposed a heuristic that schedules the jobs in a nonincreasing order of the index $\theta r\mu$. For our problem, their heuristic is equivalent to the $r\mu$ rule.

Glazebrook et al. [14] proved that the $r\mu$-heuristic, which is a state-independent policy, is asymptotically optimal as the abandonment rates approach zero, when the times to abandonment are exponentially distributed.

Note that the $r\mu$-heuristic is identical to the triangular and rectangular heuristics and is consistent with Conjecture 1 when it is also agreeable with SEPT. Also, when $r_1 \geq r_2$ and $\mu_1 \geq \mu_2$, it gives higher priority to type 1 jobs, which is consistent with Theorem 1.

4. **Time-critical first (TCF) heuristic:** This heuristic simply gives priority to jobs with higher abandonment rates, ignoring the service rates as well as the system state. To be more precise, it gives priority to type 1 jobs if and only if $r_1 > r_2$. This heuristic is not expected to perform well in most cases. However, it is still included as a benchmark policy since it is commonly applied in daily operations, especially in situations where the pressure of making quick decisions with lack of prior planning leads to giving priorities to more urgent jobs with possibly long processing times.

6. NUMERICAL RESULTS

In this section, we provide numerical results on the performance of the heuristic policies described in Section 5 for the case in which time-critical jobs have longer service times. In Section 6.1 we compare the performance of the heuristic policies under various randomly generated scenarios, and in Section 6.2 we take a closer look at the effects of certain input parameters on the performance of the heuristics.

6.1. Comparison of Heuristic Policies

We performed two sets of numerical experiments. In the first set, we considered jobs with exponential service times and lifetimes; in the second set, we considered jobs with deterministic service times and lifetimes that have Weibull distribution. For both sets of experiments, we can obtain the optimal policy by solving the backward dynamic programming recursions; hence, we can compare the performance of the heuristic policies with the optimal performance.

6.1.1. Exponential service times and lifetimes. We consider systems for which the service times and lifetimes for type i jobs are exponentially distributed with respective rates $\mu_i > 0$ and $r_i > 0$ for $i = 1, 2$. Since we would like to study many different scenarios with a wide range of system parameters, we have sampled the system parameters randomly. More specifically, the initial number of jobs m_i for each type $i \in \{1, 2\}$ is drawn independently from a discrete uniform distribution over the set $\{1, 2, \ldots, 100\}$. We have also generated the service rate μ_i of each job type i from a (continuous) uniform distribution with range $[0.5, 2.0]$. Similarly, we have generated the abandonment rate r_i of each job type i from a uniform distribution for $i = 1, 2$. We have considered five subsets of experiments depending on the range of the abandonment rates, namely $[2.0, 5.0]$, $[0.5, 2.0]$, $[0.1, 0.5]$, $[0.01, 0.1]$, and $[0.005, 0.001]$. (Note that in the first two subsets, jobs are very critical since the abandonment rates are either larger than or close to the service rates. On the other hand, in the last three subsets, jobs are not very critical since the abandonment rates are smaller than service rates.) For each subset, we generated 5000 random scenarios in which $r_1 > r_2$ and $\mu_1 < \mu_2$. We excluded the cases for which we already know what the optimal policy is, based on Proposition 5. For each scenario, we calculated the expected number of jobs taken into service under each of the four heuristic policies and the optimal policy. Then we computed the percentage deviation of the performance of each heuristic from that of the optimal policy. Based on these 5000 percentage deviations, we constructed a 95% confidence interval (C.I.) on the mean, estimated the median, lower, and upper quartiles, and determined the maximum percentage deviation (i.e., the worst performance). For each heuristic, we also calculated the number of times that the heuristic provided the best performance among the four heuristics. These results are presented in Table 1. (Note that the values in the last column of Table 1 do not add up to 5000 for each subset of experiments due to ties among heuristics.)

Table 1. Performance of the Heuristics for Exponential Service Times and Lifetimes (in Terms of the Percentage Deviation From the Optimal Performance) When $\mu_i \sim$Uniform[0.5, 2.0] and $m_i \sim$Uniform$\{1, 2, \ldots, 100\}$, for $i = 1, 2$

Heuristic	95% C.I. on the mean	Lower quartile	Median	Upper quartile	Maximum	Best heuristic in
$r_i \sim$Uniform[2.0, 5.0]						
Triangular	0.011 ± 0.001	0.000	0.001	0.009	2.064	4297 scenarios
Rectangular	0.013 ± 0.001	0.000	0.001	0.011	1.946	3275 scenarios
$r\mu$	7.600 ± 0.179	2.340	5.950	11.512	33.7632	146 scenarios
TCF	7.600 ± 0.179	2.340	5.950	11.512	33.7632	146 scenarios
$r_i \sim$Uniform[0.5, 2.0]						
Triangular	0.053 ± 0.007	0.000	0.000	0.011	6.304	3990 scenarios
Rectangular	0.042 ± 0.006	0.000	0.000	0.007	6.218	4628 scenarios
$r\mu$	4.873 ± 0.207	0.000	0.000	7.928	37.255	2554 scenarios
TCF	18.214 ± 0.386	6.189	15.656	28.153	57.445	179 scenarios
$r_i \sim$Uniform[0.1, 0.5]						
Triangular	0.425 ± 0.031	0.000	0.000	0.221	11.264	3678 scenarios
Rectangular	0.372 ± 0.028	0.000	0.000	0.200	11.134	4325 scenarios
$r\mu$	3.072 ± 0.151	0.000	0.000	4.111	30.303	3084 scenarios
TCF	13.049 ± 0.330	2.668	10.055	20.529	52.741	737 scenarios
$r_i \sim$Uniform[0.01, 0.1]						
Triangular	2.162 ± 0.102	0.000	0.047	2.978	21.659	2893 scenarios
Rectangular	2.077 ± 0.099	0.000	0.033	2.790	21.636	2923 scenarios
$r\mu$	0.581 ± 0.047	0.000	0.000	0.001	20.678	4134 scenarios
TCF	4.117 ± 0.167	0.000	1.161	6.026	38.170	2160 scenarios
$r_i \sim$Uniform[0.005, 0.01]						
Triangular	0.340 ± 0.025	0.000	0.000	0.016	7.939	3822 scenarios
Rectangular	0.335 ± 0.025	0.000	0.000	0.014	7.939	3831 scenarios
$r\mu$	0.043 ± 0.006	0.000	0.000	0.000	3.573	4681 scenarios
TCF	3.189 ± 0.106	0.096	1.773	4.989	22.323	1199 scenarios

From Table 1, it can be seen that the performances of the heuristics depend on how fast the jobs abandon the system. When jobs are very critical (i.e., when r_i's are larger than or similar to μ_i's), then the triangular and rectangular heuristics clearly provide the best performance. On the other hand, when jobs are not very critical (i.e., when r_i's are small compared to μ_i's), then the $r\mu$-heuristic has the best performance. This is an expected result since the $r\mu$-heuristic was shown to be asymptotically optimal as the abandonment rates approach zero; see Glazebrook et al. [14]. However, Table 1 shows that all heuristics (except for the TCF heuristic, which gives the worst overall performance in all cases) perform reasonably well when abandonment rates are very small. This is not surprising because when abandonment rates are very small, jobs are likely to stay in the system for a long time, which makes the difference between the performance of any two nonidling policy less significant.

Table 1 also reveals that the worst performance for the triangular and rectangular heuristics is ~22%, whereas the worst performance is 37% and 57% for the $r\mu$-heuristic and the TCF heuristic, respectively. This suggests that, overall, the triangular and rectangular heuristics seem to be more robust since they do not perform very poorly even over the parameter regions where the $r\mu$-heuristic gives a better performance.

6.1.2. Weibull lifetimes and deterministic service times. In this subsection, our objective is to test the performance of the heuristics under a nonexponential setting. More specifically, we consider systems for which the service time of a type i job is deterministic and equal to $1/\mu_i$ and its lifetime has a Weibull distribution with shape parameter $\alpha_i > 0$ and scale parameter $\beta_i > 0$. Then the abandonment rates are given by $r_i = \alpha_i/(\beta_i\Gamma(1/\alpha_i))$ for $i = 1, 2$, where $\Gamma(i)$ is the gamma function. The Weibull distribution is commonly used in modeling lifetimes of humans and possesses some nice properties such as the possibility of an increasing failure rate (see, e.g., Hougaard [15, Sect. 2.2.2]). (We choose service times to be deterministic since this allows us to compute the performance of the optimal policy.)

In Section 5, we described the heuristics under the assumption that the service times and lifetimes are exponentially distributed. Generalization of these heuristics to systems with nonexponential lifetimes is not immediate due to the lack of the memoryless property. Thus, we propose and test the following generalization to all four heuristics: At each decision epoch (i.e., at the end of each service completion), we calculate the *updated abandonment rate*, which is the reciprocal of the mean remaining lifetime, for each job type. Let $r_i(t)$ denote the updated abandonment rate for job type $i \in \{1, 2\}$ at time $t \geq 0$. (Note that $r_i(0) = r_i$.) Then, at each decision epoch, the heuristics use the same decision rules as before except that these $r_i(t)$ values are used instead of r_i's. (If the order of $r_i(t)$'s switch at a decision epoch such that the type with the faster service becomes time-critical, then we apply the optimal policy characterized by Theorem 1.) For the Weibull distribution with shape parameter α_i and scale parameter β_i, $i \in \{1, 2\}$, we obtain that

$$r_i(t) = \frac{\alpha_i}{\beta_i\Gamma(1/\alpha_i, (t/\beta_i)^{\alpha_i})} e^{-(t/\beta_i)^{\alpha_i}},$$

where $\Gamma(a, b)$, which is the incomplete gamma function, is defined as

$$\Gamma(a, b) = \int_b^{\infty} u^{a-1} e^{-u}\, du,$$

for $a > 0$ and $b \geq 0$.

In our experiments presented in this section, the initial number of jobs m_i for each type i is drawn independently from a discrete uniform distribution over the set $\{1, 2, \ldots, 20\}$. We have generated the service rate μ_i of each job type i from a (continuous) uniform distribution with range $[0.5, 2.0]$. For each job type $i \in \{1, 2\}$, we let $\alpha_i = 1.5$ and then generate the initial abandonment rate $r_i(0)$ from a uniform distribution. We considered five subsets of experiments depending on the range of the initial abandonment rate, namely $[2.0, 5.0]$, $[0.5, 2.0]$, $[0.1, 0.5]$, $[0.01, 0.1]$, and $[0.005, 0.01]$. For each subset, we generated 5000 random scenarios where $r_1(0) > r_2(0)$ and $\mu_1 < \mu_2$. (Since in this example the lifetime distributions for both types of jobs are Weibull with the same shape parameter α, having $r_1(0) > r_2(0)$ implies that the lifetime of a type 1 job is smaller than the lifetime of a type 2 job in the sense of hazard rate ordering; i.e., $r_1(t) \geq r_2(t)$ for all $t \geq 0$.) We computed the performance of each heuristic as in the exponential case; see Section 6.1.1. The results are summarized in Table 2.

Perhaps the most important observation from Table 2 is that the heuristics that we developed for the exponential case also perform well in a nonexponential setting. Furthermore, the general behavior of the heuristics does not appear to be much affected by the distributional assumption. More specifically, as in the case with exponential distributions, triangular and rectangular heuristics still provide the best performance when jobs abandon the system with high rates and the $r\mu$-heuristic is the best when abandonment rates are small. Interestingly, the TCF heuristic yields a more pronounced performance in extreme cases when compared with its performance under exponential distributions. To be more specific, when abandonment rates are large, TCF's performance is worse than its performance for the exponential case, whereas when abandonment rates are small, its performance is better than its performance for the exponential case.

6.2. Effects of Some System Parameters on the Performance of Heuristics

In this subsection, we investigate the effects of two system parameters on the performances of the four heuristic policies under exponential lifetime and service time distributions. These two parameters are the initial total number of jobs, $m_1 + m_2$, and the ratio $\phi = r_2\mu_2/(r_1\mu_1)$, which can be considered as a measure of similarity between the two job types. To observe the effect of $m_1 + m_2$, we first computed $m_1 + m_2$ for each of the 5000 random scenarios (generated for the experiments presented in Section 6.1.1) and sorted the scenarios in an ascending order of their $m_1 + m_2$ values. Then, we computed the moving average (with a window size of 1500) of the percentage deviation of each heuristic from the optimal policy. To observe the effect of ϕ, we

TABLE 2. Performance of the Heuristics for Deterministic Service Times and Weibull Lifetimes (in Terms of the Percentage Deviation From the Optimal Performance) When $\alpha_i = 1.5$, $\mu_i \sim \text{Uniform}[0.5, 2.0]$, and $m_i \sim \text{Uniform}\{1, 2, \ldots, 20\}$, for $i = 1, 2$

Heuristic	95% C.I. on the mean	Lower quartile	Median	Upper quartile	Maximum	Best heuristic in
$r_i \sim$Uniform[2.0, 5.0]						
Triangular	0.015 ± 0.004	0.000	0.000	0.000	3.490	4943 scenarios
Rectangular	0.027 ± 0.007	0.000	0.000	0.000	8.688	4796 scenarios
$r\mu$	3.089 ± 0.203	0.000	0.000	1.258	46.922	3032 scenarios
TCF	19.609 ± 0.488	2.755	15.289	34.561	63.145	207 scenarios
$r_i \sim$Uniform[0.5, 2.0]						
Triangular	0.163 ± 0.019	0.000	0.000	0.051	14.010	4609 scenarios
Rectangular	0.175 ± 0.019	0.000	0.000	0.056	14.010	4024 scenarios
$r\mu$	4.534 ± 0.191	0.000	0.002	7.609	34.212	2505 scenarios
TCF	17.004 ± 0.384	5.033	14.440	26.546	62.378	428 scenarios
$r_i \sim$Uniform[0.1, 0.5]						
Triangular	1.129 ± 0.066	0.000	0.007	1.057	19.745	3320 scenarios
Rectangular	1.006 ± 0.063	0.000	0.003	0.813	19.745	3639 scenarios
$r\mu$	1.242 ± 0.077	0.000	0.000	0.738	21.399	3626 scenarios
TCF	7.502 ± 0.261	0.000	3.613	11.815	49.918	1643 scenarios
$r_i \sim$Uniform[0.01, 0.1]						
Triangular	2.398 ± 0.112	0.000	0.405	3.030	25.039	1997 scenarios
Rectangular	2.313 ± 0.111	0.000	0.326	2.918	25.039	2076 scenarios
$r\mu$	0.017 ± 0.003	0.000	0.000	0.000	3.982	4662 scenarios
TCF	0.614 ± 0.047	0.000	0.000	0.309	23.985	3297 scenarios
$r_i \sim$Uniform[0.005, 0.01]						
Triangular	0.208 ± 0.012	0.000	0.001	0.203	5.266	2502 scenarios
Rectangular	0.201 ± 0.012	0.000	0.000	0.188	5.266	2555 scenarios
$r\mu$	0.008 ± 0.001	0.000	0.000	0.000	0.514	4446 scenarios
TCF	0.135 ± 0.008	0.000	0.000	0.131	3.259	2673 scenarios

followed the same procedure except that the window size for the moving average was set to 500. The moving average plots for $m_1 + m_2$ and ϕ are given in Figures 3 and 4, respectively. In the interest of space, we present the plots for only three subsets of our experiments, namely the ones where the ranges for the abandonment rates are $[0.5, 2.0]$, $[0.1, 0.5]$, and $[0.01, 0.1]$.

From Figure 3, it can be seen that the performances of the $r\mu$-heuristic and TCF heuristic worsen with the total number of jobs in the system in all three cases considered. On the other hand, the performances of the triangular and rectangular heuristics have a tendency to improve as the number of jobs in the system increases when jobs abandon the system at a relatively high rate. Together with our conclusions from Section 6.1, this suggests that *the triangular and rectangular heuristics are well suited for worst-case scenarios, where the system is overwhelmed with a large number of jobs that abandon the system quickly*. Such scenarios can be typically realized in the

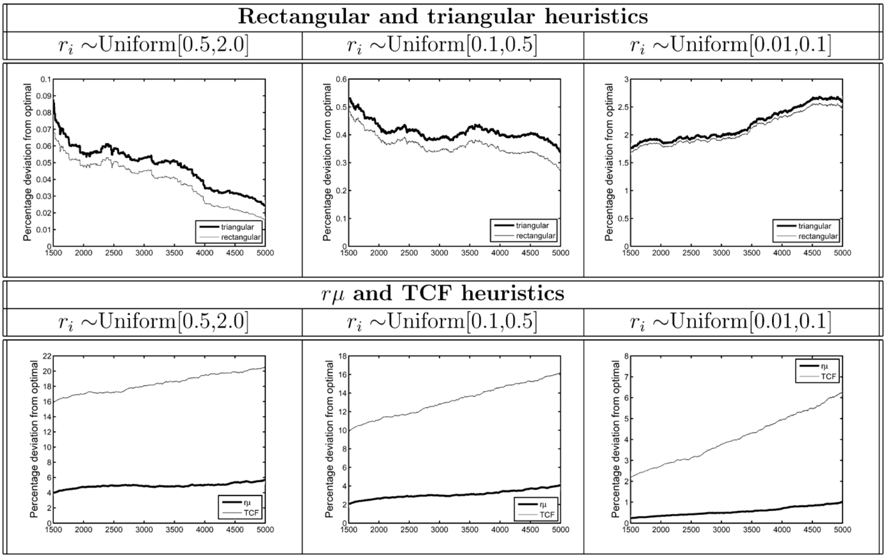


FIGURE 3. Moving-average plots of percentage deviations of heuristics from the optimal with respect to the number of scenarios that are ordered according to the increasing initial total number of jobs $m_1 + m_2$.

wake of mass casualty incidents, which cause significant number of casualties who are in need of immediate care.

Figure 4 shows the relationship between the performance of the heuristics and the parameter ϕ. First, note that the moving-average plots for the triangular, rectangular, and $r\mu$-heuristics hit zero after a certain point (when ϕ is around 1). This is due to the fact that these three heuristics are in fact equivalent and, furthermore, optimal (which is a numerical observation) when $r_2\mu_2 \geq r_1\mu_1$ and $\mu_1 < \mu_2$ under exponential distributions; see Conjecture 1.

As we have discussed in Section 6.1, the triangular and rectangular heuristics perform very well when jobs have high abandonment rates. Figure 4 supports this observation but also suggests that in some cases, these heuristics should be preferred even when the abandonment rates are small. To see this, first note that regardless of whether abandonment rates are small or large, the triangular and rectangular heuristics provide their worst performances when $r_2\mu_2$ is small; however, they perform increasingly well as $r_2\mu_2$ gets closer to $r_1\mu_1$. (Note that $r_2\mu_2$ is close to $r_1\mu_1$ around the middle of the x-axis.) On the other hand, the $r\mu$-heuristic provides its worst performance when $r_2\mu_2/(r_1\mu_1)$ approaches 1. Now, comparing the two plots for which $r_i \in [0.01, 0.1]$ in Figure 4, we see that even when the abandonment rates are small but $r_2\mu_2/(r_1\mu_1)$ is close to 1, the triangular and rectangular heuristics yield a better average performance than the $r\mu$-heuristic and TCF heuristic and thus are preferable.

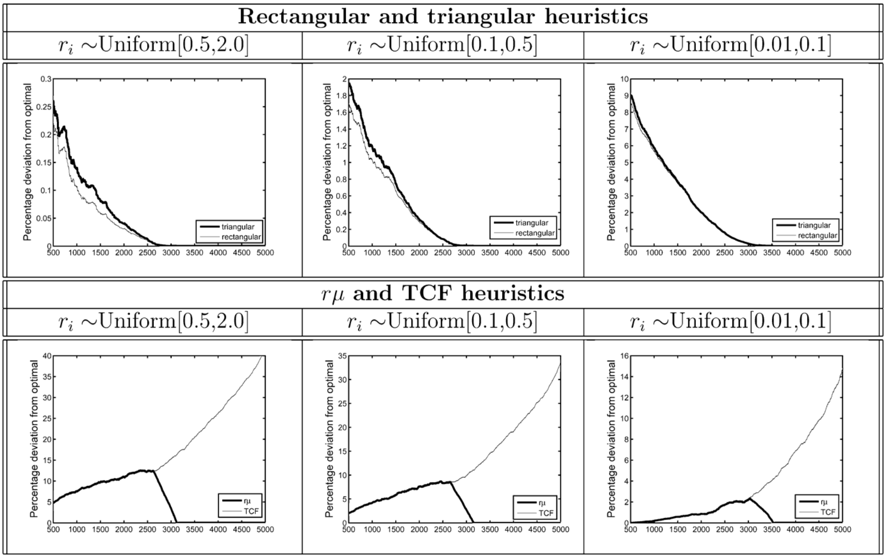


FIGURE 4. Moving-average plots of percentage deviations of heuristics from the optimal with respect to the number of scenarios that are ordered according to the increasing value of $r_2\mu_2/(r_1\mu_1)$.

7. CONCLUSIONS

We considered a *clearing system* with a single server and a finite number of jobs that might abandon the system before receiving service. For such a system, we studied the optimal and near-optimal scheduling of jobs, which are characterized by their service time and lifetime distributions, with the objective of minimizing the total number of abandonments. We are mainly motivated by the *patient triage* problem, which arises in the aftermath of mass casualty events. Our question is: Given the operating/treatment time and lifetime distributions of different patients with different injuries and also given the number of patients, how should the patients be admitted to a scarce resource (e.g., an operating room) so as to maximize the total number of patients saved? In practice, there is not a simple answer to this question since each mass casualty event is unique with challenges that typically cannot be anticipated, and such events require prompt decisions from human beings working in chaotic environments. Thus, in this work, we provide some general insights into the problem by identifying the characteristics of effective prioritization decision rules under different operating conditions.

Using sample-path arguments for general service time and lifetime distributions and a stochastic dynamic programming approach for exponential service time and lifetime distributions, we identified characteristics of the optimal policy analytically for several cases. For example, we showed that if jobs can be ordered in such a way that the job with the shortest lifetime (in the sense of hazard rate orders) also has the shortest service time (in the sense of likelihood ratio orders), then it should always be given priority for service, regardless of the state of the system. This result makes sense intuitively, but, more importantly, it provides us with a criterion as to what makes a job a top-priority job. For the patient triage problem, this implies that regardless of how many patients are in need of treatment, a patient with a certain injury can be given the highest priority if we know that, without any medical intervention, his/her lifetime will be shorter than any other patient in the sense of hazard rate ordering and that the operation for that specific injury takes a shorter time than all the other patients' injuries in the sense of likelihood ratio orders. Nevertheless, the case where time-critical patients have longer service times appears to be more interesting and realistic; hence, we devoted a significant portion of the article to identifying optimal or near-optimal policies for this case.

When time-critical jobs have stochastically longer service times, the optimal policy is not easy to characterize except for certain cases. For example, when service and lifetimes are exponentially distributed and the jobs can be categorized into two classes based on their mean service time and lifetimes, we were able to identify conditions under which it is always optimal to give priority to faster jobs, even when they are less time-critical. For cases in which we cannot characterize the optimal policy, we developed two state-dependent heuristic policies and compared them with two benchmark policies (that are not state dependent) by means of a numerical study. From our numerical experiments, we gained several important insights. When the job abandonment rates are small compared to the service rates, as one would expect, all policies perform reasonably well (with one of the state-independent policies providing the best performance). On the other hand, when jobs abandon the system at a faster rate, the state-independent policies perform very poorly; hence, it is extremely important in this case to employ state-dependent policies such as those proposed in this article. For the patient triage problem, these observations imply that when a major emergency event causes injuries that need to be taken care of very quickly, then it is crucial for a triage policy to take into account the number of patients who need help. Especially for worst-case scenarios, where the event causes a large number of patients who need help immediately, our state-dependent heuristics appear to provide substantially better performance than the state-independent policies and have the potential to achieve the greatest good for the greatest number of people.

Acknowledgments

The work of the first author was supported by the National Science Foundation under Grant No. CMMI-0715020. The work of the second author was supported by the National Science Foundation under Grant No. CMMI-0620737.

Notes

1. Triage is a brief clinical assessment that determines the order in which patients should be seen in the Emergency Department or, if in the field, the speed of transport and choice of hospital destination [26].

References

1. Arnold, J.L., Tsai, M.-C., Halpern, P., Smithline, H., Stok, E., & Ersoy, G. (2004). Mass-casualty, terrorist bombings: Epidemiological outcomes, resource utilization, and time course of emergency needs (Part I). *Prehospital and Disaster Medicine* 18(3): 220–234.
2. Bae, J., Kim, S., & Lee, E.Y. (2001). The virtual waiting time of M/G/1 queue with impatient customers. *Queueing Systems* 38: 485–494.
3. Bhattacharya, P.P. & Ephremides, A. (1989). Optimal scheduling with strict deadlines. *IEEE Transactions on Automatic Control* 34(7): 721–728.
4. Bhattacharya, P.P. & Ephremides, A. (1991). Optimal allocation of a server between two queues with due times. *IEEE Transactions on Automatic Control* 36(12): 1417–1423.
5. Boxma, O.J. & Forst, F.G. (1986). Minimizing the expected weighted number of tardy jobs in stochastic flow shops. *Operations Research Letters* 5(3): 119–126.
6. Brandt, A. & Brandt, M. (2002). Asymptotic results and a Markovian appproximation for the $\mathrm{M}(n)/\mathrm{M}(n)/s + \mathrm{GI}$ system. *Queueing Systems* 41: 73–94.
7. Brandt, A. & Brandt, M. (2004). On the two-class M/M/1 system under preemptive resume and impatience of the prioritized customers. *Queueing Systems* 47: 147–168.
8. Choi, B.D., Kim, B., & Chung, J. (2001). M/M/1 queue with impatient customers of higher priority. *Queueing Systems* 38: 49–66.
9. Coffman, E.G., Flatto, L., Garey, M.R., & Weber, R.R. (1987). Minimizing expected makespan on uniform processor systems. *Advances in Applied Probability* 19: 177–201.
10. Doytchinov, B., Lehoczky, J., & Shreve, S. (2001). Real-time queues in heavy traffic with earliest-deadline-first queue discipline. *Annals of Applied Probability* 11(2): 332–378.
11. Emmons, H. & Pinedo, M. (1990). Scheduling stochastic jobs with due dates on parallel machines. *European Journal of Operational Research* 47(1): 49–55.
12. Frykberg, E.R. (2002). Medical management of disasters and mass casualties from terrorist bombings: How can we cope? *The Journal of Trauma* 53(2): 201–212.
13. Glazebrook, K.D. (1983). Stochastic scheduling with due dates. *International Journal of Systems Science* 14: 1259–1271.
14. Glazebrook, K.D., Ansell, P.S., Dunn, R.T., & Lumley, R.R. (2004). On the optimal allocation of service to impatient tasks. *Journal of Applied Probability* 41(1): 51–72.
15. Hougaard, P. (2000). *Analysis of multivariate survival data*. New York: Springer-Verlag.
16. Jang, W. & Klein, C.M. (2002). Minimizing the expected number of tardy jobs when processing times are normally distributed. *Operations Research Letters* 30: 100–106.
17. Jiang, Z., Lewis, T.G., & Colin, J.-Y. (1996). Scheduling hard real-time constrained periodic tasks on multiple processors. *Journal of Systems and Software* 19: 102–118.
18. Levi, L., Michaelson, M., Admi, H., Bregman, D., & Bar-Nahor, R. (2003). National strategy for mass casualty situations and its effects on the hospital. *Prehospital and Disaster Medicine* 17(1): 12–17.
19. Nocera, A. & Garner, A. (1999). An Australian Mass Casualty Incident triage system for the future based upon triage mistakes of the past: The Homebush Triage Standard. *Australian and New Zealand Journal of Surgery* 69: 603–608.
20. Panwar, S.S., Towsley, D., & Wolf, J.K. (1988). Optimal scheduling policies for a class of queues with customer deadlines to the beginning of service. *Journal of the Association for Computing Machinery* 35(4): 832–844.
21. Peleg, K., Aharonson-Daniel, L., Michael, M., Shapira, S.C., & the Israel Trauma Group. (2003). Patterns of injury in hospitalized terrorist victims. *American Journal of Emergency Medicine* 21(4): 258–262.
22. Pinedo, M. (1983). Stochastic Scheduling with release dates and due dates. *Operations Research* 31(3): 559–572.
23. Righter, R. (1994). Scheduling. I.M. Shaked & J.G. Shanthikumar (eds.), *Stochastic orders*. New York: Academic Press, pp. 381–432.

24. Righter, R. (1988). Job scheduling to minimize weighted flowtime on uniform processors. *Systems and Control Letters* 10: 211–216.
25. Righter, R. (2000). Expulsion and scheduling control for multiclass queues with heterogenous servers. *Queueing Systems* 34: 289–300.
26. Rund, D.A. & Rausch, T.S. (1981). *Triage*. St. Louis, MO: C.V. Mosby.
27. Van Mieghem, J. (1995). Dynamic scheduling with convex delay costs: the generalized $c\mu$ rule. *Annals of Applied Probability* 5(3): 808–833.
28. Van Mieghem, J. (2003). Due date scheduling: asymptotic optimality of generalized longest queue and generalized largest delay rules. *Operations Research* 51(1): 113–122.
29. Ward, A.R. & Glynn, P.W. (2003). A diffusion approximation for a Markovian queue with reneging. *Queueing Systems* 43: 103–128.
30. Ward, A.R. & Kumar, S. (2008). Asymptotically optimal admission control of a queue with impatient customers. *Mathematics of Operations Research* 33: 167–202.
31. Weber, R.R., Varaiya, P., & Walrand, J. (1986). Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flow time. *Journal of Applied Probability* 23: 841–847.
32. Weiss, G. & Pinedo, M. (1980). Scheduling tasks with exponential service times on non-identical processors to minimize various cost functions. *Journal of Applied Probability* 17: 187–202.
33. Xu, S.H. (1994). A duality approach to admission and scheduling control of queues. *Queueing Systems* 18: 273–300.
34. Zhao, Z.-X., Panwar, S.S., & Towsley, D. (1991). Queueing performance with impatient customers. *Proceedings of IEEE INFOCOM'91* 1: 400–409.

APPENDIX

In this appendix, we prove Proposition 7. We need the following lemma.

LEMMA A1: *Suppose that $\mu_1 < \mu_2$.*

(i) For all $x_2 \geq 1$, we have

$$V(0,x_2;P_2) - V(1,x_2-1;P_1) \leq \frac{(x_2-1)r_2(\mu_2-\mu_1)}{\mu_1(\mu_2+(x_2-1)r_2)}.$$

(ii) For all $x_1 \geq 1$, we have

$$V(x_1-1,1;P_2) - V(x_1,0;P_1) \leq \frac{(x_1-1)r_1(\mu_2-\mu_1)}{\mu_2(\mu_1+(x_1-1)r_1)}.$$

Proof of Lemma A1:

(i) We prove the result by induction on x_2. Since $V(0,1;P_2) - V(1,0;P_1) = 0$, the result holds trivially for $x_2 = 1$. Next, suppose that the result holds for $x_2 - 1$. Then we have

$$\begin{aligned} &V(0,x_2;P_2) - V(1,x_2-1;P_1) \\ &\quad = \frac{(x_2-1)r_2(\mu_2-\mu_1)}{(\mu_1+(x_2-1)r_2)(\mu_2+(x_2-1)r_2)} \\ &\qquad + \frac{(x_2-1)r_2}{\mu_1+(x_2-1)r_2}(V(0,x_2-1;P_2) - V(1,x_2-2;P_1)) \end{aligned}$$

$$\leq \frac{(x_2-1)r_2(\mu_2-\mu_1)}{(\mu_1+(x_2-1)r_2)(\mu_2+(x_2-1)r_2)} + \frac{(x_2-1)(x_2-2)r_2^2(\mu_2-\mu_1)}{\mu_1(\mu_1+(x_2-1)r_2)(\mu_2+(x_2-2)r_2)}$$

(by the inductive hypothesis)

$$= \frac{(x_2-1)r_2(\mu_2-\mu_1)}{\mu_1(\mu_2+(x_2-1)r_2)}\left\{\frac{\mu_1}{\mu_1+(x_2-1)r_2} + \frac{(x_2-2)r_2(\mu_2+(x_2-1)r_2)}{(\mu_1+(x_2-1)r_2)(\mu_2+(x_2-2)r_2)}\right\}$$

$$\leq \frac{(x_2-1)r_2(\mu_2-\mu_1)}{\mu_1(\mu_2+(x_2-1)r_2)},$$

where the last inequality follows from the fact that $(x_2-2)r_2(\mu_2+(x_2-1)r_2) \leq (x_2-1)r_2(\mu_2+(x_2-2)r_2)$.

(ii) We prove the result by induction on x_1. Since $V(0,1;P_2) - V(1,0;P_1) = 0$, the result holds trivially for $x_1 = 1$. Next, suppose that the result holds for $x_1 - 1$. Then we have

$$V(x_1-1,1;P_2) - V(x_1,0;P_1)$$

$$= \frac{(x_1-1)r_1(\mu_2-\mu_1)}{(\mu_1+(x_1-1)r_1)(\mu_2+(x_1-1)r_1)} + \frac{(x_1-1)r_1}{\mu_2+(x_1-1)r_1}(V(x_1-2,1;P_2) - V(x_1-1,0;P_1))$$

$$\leq \frac{(x_1-1)r_1(\mu_2-\mu_1)}{(\mu_1+(x_1-1)r_1)(\mu_2+(x_1-1)r_1)} + \frac{(x_1-1)(x_1-2)r_1^2(\mu_2-\mu_1)}{\mu_2(\mu_2+(x_1-1)r_1)(\mu_1+(x_1-2)r_1)}$$

(by the inductive hypothesis)

$$\leq \frac{(x_1-1)r_1(\mu_2-\mu_1)}{\mu_2(\mu_1+(x_1-1)r_1)}.$$ ■

Proof of Proposition 7:

(i) We prove the result by induction on x_2. For $x_2 = 1$, we have $V(1,1;P_2) - V(1,1;P_1) = (r_2\mu_2 - r_1\mu_1)/(\mu_1+r_2)(\mu_2+r_1) \geq 0$. Next, suppose that $V(1,x_2-1;P_2) \geq V(1,x_2-1;P_1)$. Then we have

$$V(1,x_2;P_2) - V(1,x_2;P_1)$$

$$= \frac{r_2\mu_2 - r_1\mu_1}{(\mu_1+x_2r_2)(\mu_2+r_1+(x_2-1)r_2)} + \frac{(x_2-1)r_2(\mu_2-\mu_1)}{(\mu_1+x_2r_2)(\mu_2+r_1+(x_2-1)r_2)} + \left(\frac{\mu_1}{\mu_1+x_2r_2} - \frac{r_1}{\mu_2+r_1+(x_2-1)r_2}\right) \times (V(1,x_2-1;P_1) - V(0,x_2;P_2))$$

$$
\begin{aligned}
&+ \frac{\mu_2 + (x_2-1)r_2}{\mu_2 + r_1 + (x_2-1)r_2}(V(1,x_2-1;P_2) - V(1,x_2-1;P_1))\\
&\geq \frac{(x_2-1)r_2(\mu_2-\mu_1)}{(\mu_1+x_2r_2)(\mu_2+r_1+(x_2-1)r_2)}\\
&+ \left(\frac{\mu_1}{\mu_1+x_2r_2} - \frac{r_1}{\mu_2+r_1+(x_2-1)r_2}\right)\\
&\times (V(1,x_2-1;P_1) - V(0,x_2;P_2))\\
&= \Bigg((x_2-1)r_2 + \frac{x_2r_1r_2 - \mu_1(\mu_2+(x_2-1)r_2)}{\mu_2-\mu_1}\\
&\times (V(0,x_2;P_2) - V(1,x_2-1;P_1))\Bigg)\\
&\times \frac{(\mu_2-\mu_1)}{(\mu_1+x_2r_2)(\mu_2+r_1+(x_2-1)r_2)}\\
&\geq \left((x_2-1)r_2 - \frac{\mu_1(\mu_2+(x_2-1)r_2)}{\mu_2-\mu_1}(V(0,x_2;P_2) - V(1,x_2-1;P_1))\right)\\
&\times \frac{(\mu_2-\mu_1)}{(\mu_1+x_2r_2)(\mu_2+r_1+(x_2-1)r_2)},
\end{aligned}
$$

since $V(1,x_2-1;P_1) \leq V(0,x_2;P_2)$ when $\mu_1 < \mu_2$ by Lemma 2. Now, the condition that $\mu_1 < \mu_2$ and part (i) of Lemma A1 complete the proof.

(ii) We prove the result by induction on x_1. The case with $x_1 = 1$ is already covered in part (i). Now, suppose that $V(x_1-1,1;P_2) \geq V(x_1-1,1;P_1)$. Then we have

$$
\begin{aligned}
&V(x_1,1;P_2) - V(x_1,1;P_1)\\
&\quad= \frac{r_2\mu_2 - r_1\mu_1}{(\mu_2+x_1r_1)(\mu_1+(x_1-1)r_1+r_2)}\\
&\quad+ \frac{(x_1-1)r_1(\mu_2-\mu_1)}{(\mu_2+x_1r_1)(\mu_1+(x_1-1)r_1+r_2)}\\
&\quad+ \left(\frac{\mu_2}{\mu_2+x_1r_1} - \frac{r_2}{\mu_1+(x_1-1)r_1+r_2}\right)\\
&\quad\times (V(x_1,0;P_1) - V(x_1-1,1;P_2))\\
&\quad+ \frac{(x_1-1)r_1}{\mu_1+(x_1-1)r_1+r_2}(V(x_1-1,1;P_2) - V(x_1-1,1;P_1))\\
&\quad\geq \frac{(x_1-1)r_1(\mu_2-\mu_1)}{(\mu_2+x_1r_1)(\mu_1+(x_1-1)r_1+r_2)}\\
&\quad+ \left(\frac{\mu_2}{\mu_2+x_1r_1} - \frac{r_2}{\mu_1+(x_1-1)r_1+r_2}\right)\\
&\quad\times (V(x_1,0;P_1) - V(x_1-1,1;P_2))
\end{aligned}
$$

$$
\begin{aligned}
&= \Bigg((x_1 - 1)r_1 + \frac{x_1 r_1 r_2 - \mu_2(\mu_1 + (x_1 - 1)r_1)}{\mu_2 - \mu_1} \\
&\quad \times (V(x_1 - 1, 1; P_2) - V(x_1, 0; P_1)) \Bigg) \\
&\quad \times \frac{(\mu_2 - \mu_1)}{(\mu_2 + x_1 r_1)(\mu_1 + (x_1 - 1)r_1 + r_2)} \\
&\geq \Bigg((x_1 - 1)r_1 - \frac{\mu_2(\mu_1 + (x_1 - 1)r_1)}{\mu_2 - \mu_1} (V(x_1 - 1, 1; P_2) - V(x_1, 0; P_1)) \Bigg) \\
&\quad \times \frac{(\mu_2 - \mu_1)}{(\mu_2 + x_1 r_1)(\mu_1 + (x_1 - 1)r_1 + r_2)},
\end{aligned}
$$

since $V(x_1 - 1, 1; P_2) \geq V(x_1, 0; P_1)$ when $\mu_1 < \mu_2$ by Lemma 2. Now, the condition that $\mu_1 < \mu_2$ and part (ii) of Lemma A1 complete the proof. ■