POMS PRODUCTION AND OPERATIONS MANAGEMENT

PRODUCTION AND OPERATIONS MANAGEMENT POMS

Vol. 29, No. 2, February 2020, pp. 241–262
ISSN 1059-1478 | EISSN 1937-5956 | 20 | 2902 | 0241

DOI 10.1111/poms.13105

When to Switch? Index Policies for Resource Scheduling in Emergency Response

Dong Li*
School of Business and Economics, Loughborough University, Loughborough, LE11 3TU, UK, d.li@lboro.ac.uk

Li Ding
Durham University Business School, Durham University, Durham, DH1 3LB, UK, li.ding@durham.ac.uk

Stephen Connor
Department of Mathematics, University of York, York, YO10 5DD, UK, stephen.connor@york.ac.uk

This study considers the scheduling of limited resources to a large number of jobs (e.g., medical treatment) with uncertain lifetimes and service times, in the aftermath of a mass casualty incident. Jobs are subject to triage at time zero, and placed into a number of classes. Our goal is to maximize the expected number of job completions. We propose an effective yet simple index policy based on Whittle's restless bandits approach. The problem concerned features a finite and uncertain time horizon that is dependent upon the service policy, which also determines the decision epochs. Moreover, the number of job classes still competing for service diminishes over time. To the best of our knowledge, this is the first application of Whittle's index policies to such problems. Two versions of Lagrangian relaxation are proposed in order to decompose the problem. The first is a direct extension of the standard Whittle's restless bandits approach, while in the second the total number of job classes still competing for service is taken into account; the latter is shown to generalize the former. We prove the indexability of all job classes in the Markovian case, and develop closed-form indices. Extensive numerical experiments show that the second proposal outperforms the first one (that fails to capture the dynamics in the number of surviving job classes, or bandits) and produces more robust and consistent results as compared to alternative heuristics suggested from the literature, even in non-Markovian settings.

Key words: dynamic programming; restless bandits; index policy; emergency resource scheduling
History: Received: November 2018; Accepted: August 2019 by Zuo-Jun (Max) Shen after 1 revision.

1. Introduction

In the aftermath of natural or man-made disasters, resource scarcity is potentially the biggest hurdle to a successful emergency response (Green and Kolesar 2004). With the sudden surge of demand and limited resources available to deal with rescue-like missions, jobs varying from medical treatment for casualties to evacuations from affected areas all require prioritization decisions (Sun et al. 2017). A typical process of an emergency response may include: first, an initial assessment of the urgency of jobs (known as *triage* in practice); second, an assignment of jobs to different categories depending on the initial assessment; third, allocation of emergency resources to the category of the most urgent jobs, and then to less urgent ones once all jobs in the preceding categories are completed. This common practice is often known as Simple Triage and Rapid Treatment (START) (see, e.g., Nocera and Garner 1999, Sacco et al. 2007). However, such widely adopted practice has been criticized for being short-sighted by not taking into account the scarcity of emergency resources and/or the dynamics of the remaining jobs. Most notably, the simple policy described in the third step above may not help to save the most lives in the aftermath of a casualty event (Jacobson et al. 2012). Therefore, the primary objective of this study is to contribute to this important and ongoing debate by providing a simple yet near optimal policy to the scheduling of emergency resources following triage.

In a nutshell, we consider a scenario in which a collection of impatient jobs is seeking service which is provided by a single server. There are two major sources of uncertainty related to each job. Firstly, its service is of uncertain duration. Second, the job's *lifetime*, namely the period of time during which it is available for service, is also uncertain. We shall assume that a job abandons the system unserved if its service does not begin before the expiration of its lifetime. Further, any job whose service begins is guaranteed to be served to completion. No preemptions are allowed. Each job is subject to triage at time zero and is placed in one of *J classes*. Jobs in each class are assumed to have independent and identically distributed (i.i.d.) lifetimes and i.i.d. service times. Following triage, the central challenge addressed by the paper concerns how the jobs should be scheduled for service such that the expected number of jobs served to completion is maximized.

Optimal solutions (as illustrated by Argon et al. 2008, Jacobson et al. 2012) to the problem described above may well have a structure of threshold policies. When the switch happens, the prioritized job class is not necessarily emptied yet. Nonetheless, the general optimal policy is hard to elucidate. Optimal solutions obtained using a stochastic dynamic programming (DP) approach can only be obtained for problems of small size. Naturally, the idea of developing an index policy that assigns a state-dependent index value to different job classes that require service is both intuitively and computationally desirable for the task of interest.

Direct precursors to this study under operations literature include Glazebrook et al. (2004), Li and Glazebrook (2010), and Jacobson et al. (2012), which develop heuristic policies to the resource scheduling problem where all impatient jobs are present at the outset (i.e., under the assumption of no further arrivals). Essentially, two types of policies have been proposed: static and state-dependent. Both Glazebrook et al. (2004) and Jacobson et al. (2012) develop simple and static (state-independent) heuristic policies which generate a fixed priority among the job classes over time. Due to the fact that the optimal switch between the job classes might happen before the prioritized job class is completely emptied, the number of jobs remaining in each class provides important information in constructing effective scheduling policies. Therefore, static policies generally yield inferior performances than their counterparts, such as a *two-step* policy and a *threshold* policy proposed also in Jacobson et al. (2012), and a single-step DP policy improvement heuristic developed by Li and Glazebrook (2010), which make explicit use of state information. However, despite the near-optimal performance demonstrated by the deployment of a single-step policy improvement approach in Li and Glazebrook (2010), this is still computationally challenging due to the DP formation employed. The threshold policy generalizes to problems where the payoff for serving a job is not necessarily one and could be different between classes. It is however only applicable to two job classes and thus its usage is restrictive. Among others, Jacobson et al. (2012) provide a similar rationale to our approach in developing heuristic policies to assign priorities among the job classes. That is, given the resource scarcity and a tight time frame, their two-step policy takes explicit account of opportunity costs incurred by not providing the service, as well as the number of jobs remaining. Basically, the two-step policy assigns priority to the classes which lead to the minimal expected number of abandonments from the system during the next service, which is intuitive and is shown to perform strongly in a system featuring “heavy premature job loss” (i.e., very short lifetimes). Nonetheless, the development and evaluation of strongly performing state-dependent policies remains a central challenge in more general cases which allow a wide variation among job classes with different combinations of lifetimes and service times. Our proposed index policy addresses this challenge by applying a suitable theoretical model to capture the “switch.”

Whittle’s restless bandit model seems perfectly applicable to the aforementioned resource scheduling problem, where job classes following triage with random lifetimes evolve with or without being served. Restless bandits (RBs) are known as a considerable extension to the seminal work by Gittins (1979) where index solutions were first proposed for classical multi-armed bandit problems (MABs). This is a class of models concerned with the sequential allocation of a single indivisible resource to a collection of stochastic reward-generating bandits. However, in MABs, bandits are thought to remain frozen while not in receipt of resources. Whittle’s RBs relaxes this constraint by allowing bandits to evolve in both active and passive states. However, this generalization comes at significant cost. In contrast to MABs, RBs are almost certainly intractable having been shown to be PSPACE-hard by Papadimitriou and Tsitsiklis (1999). Whittle (1988) proposes an index policy which emerges from a Lagrangian relaxation of the original resource allocation problem. Weber and Weiss (1991) establish a form of asymptotic optimality for Whittle’s index policy under given conditions. The index derived may be interpreted as a fair charge in receipt of resources to a particular bandit in a particular state. A natural index policy, emerging from the deployment of an adaptive greedy algorithm, will use the fair charges/subsidies to determine the allocation/scheduling of the resources among the job classes. However, those indices derived need to pass an *indexability* test. Indexability is a structural property that is required to establish the existence of the optimal solution to the Lagrangian relaxation problem (Glazebrook et al. 2014). This has become a primary inhibitor for a range of RB problems to which index theory can be applied, leading to a number of theoretical developments to remove this notorious obstacle. Niño-Mora (2001) describes sufficient conditions to prove indexability by the use of polyhedral approaches (as given by Bertsimas and Niño-Mora 1996) when the system concerned can be shown to satisfy so-called *partial conservation laws*. Glazebrook et al. (2011) show full indexability for a small number of dynamic allocation problems for which resources are divisible and evolution of the system is of birth-death type. More recently, several studies have demonstrated the power of Whittle's index theory and approach in a range of application areas. These include the queueing (admission) control (see, e.g., Ding and Glazebrook 2013, Argon et al. 2009), machine maintenance (see, e.g., Glazebrook et al. 2005), asset management (see, e.g., Glazebrook et al. 2006), congestion control (see, e.g., Jacko and Sansò 2012), dynamic assortment (see, e.g., Caro and Gallien 2007) and inventory routing (see, e.g., Archibald et al. 2009).

Further, Whittle (1988) and successive studies developed and evaluated index policies by utilizing a Lagrangian relaxation approach largely based on a time average reward/cost criterion in an infinite time horizon with either discrete or continuous decision epochs; as Whittle comments, indices are often simpler in this case. To the best of our knowledge, the only work that develops index policies for a finite (and deterministic) time horizon is due to Graczová and Jacko (2014), who have studied a knapsack problem for perishable inventories in the context of retail revenue management. Yet they only consider the problem in a discrete time setting. From the extant literature, Whittle's index policy tends to perform more strongly in admission control problems (where decision epochs are either discrete or are exogenously determined by arrivals) than those that require stochastic scheduling of resources/servers (where future decision epochs are also affected by current scheduling/allocation actions). Our problem setting described above is characterized with non-discrete decision epochs in a finite and uncertain time horizon, both of which are dependent upon the adopted service policies. Further, unlike other works in the RB literature which always consider a fixed number of bandits, the problem concerned features a diminishing number of bandits (surviving job classes) over time. Such problems have wide applications in practice and their unique features require different treatments to the canonical Whittle's RB approach. Hence, our study aims to provide an initial attempt at addressing this existing research gap and to further extend the index theory literature.

In summary, our contributions are twofold. To the emergency response literature, we propose a strongly performing yet simple index policy underpinned by Whittle's index theory for the scheduling of emergency resources after mass casualty events. The resulting index policy is state-dependent and can be quickly derived, which is particularly important for emergency responses where time is critical. Our numerical study shows that the index policy performs most strongly compared with other benchmarks for problems featuring large variations between job classes. To the index policy literature we fill the gap for problems with continuous decision epochs and a finite (and uncertain) time horizon, both of which are dependent upon the service policy. To the best of our knowledge, we are also the first to address an RB problem with a changing number of bandits over time. We extend Whittle's Lagrangian relaxation approach to accommodate these features and decompose the original problem into single class problems. We prove indexability and develop closed-form equations for the calculation of the index values in the Markovian case. Furthermore, we argue and demonstrate that for such problems a simple application of the standard Whittle's relaxation approach leads to an index policy with comparatively poor performance. The alternative relaxation that we propose to account for all the competing job classes shows noticeable improvements in maximizing the number of jobs served to completion.

The study proceeds as follows: section 2 presents a description of the problem, which is modeled as a semi-Markov decision process. In section 3, the original problem is relaxed and decomposed into single class problems by extending Whittle's Lagrangian relaxation approach in two seemingly different ways, although it is subsequently demonstrated that one is in fact a generalization of the other. In section 4, we show that the single class problems are indexable in the Markovian case and develop closed-form index values for different scenarios. We further propose a heuristic policy based on the indices following our second decomposition. The proposed heuristic is subject to numerical investigation in section 5 where it is compared to earlier proposals in the literature, including the index policy derived from the standard Whittle's approach (the first decomposition), and (where possible) to the optimal solution. Both Markovian and non-Markovian settings are considered. In section 6 we provide an in-depth analysis on the switching patterns of alternative policies. Section 7 concludes the study.

2. The Model

A collection of impatient jobs emerge at time zero for service by a single server. Each job belongs to one of J classes and the total initial number of jobs in class j is L_j, $1 \leq j \leq J$. All jobs in the same class j are characterized by two positive valued random variables: the lifetime X_j and service time Y_j. All lifetimes and service times have finite expectation and are independent of each other; we denote the lifetime and service time distribution functions for jobs in class j as F_j and G_j respectively. The single server processes individual jobs non-preemptively. A job will abandon the system unserved if its service has not begun before its lifetime. It is assumed that once a job has begun service, it will be served through to completion. The objective is to find an optimal service policy to maximize the expected total number of jobs served before the system is empty.

This problem can be modeled as a semi-Markov decision process as follows:

- Decisions are made at time zero and at all service completion times. Denote the *state* at decision epoch t by $\{\mathbf{n}(t), t\}$ where $\mathbf{n}(t) = \{n_j(t):\ 1 \le j \le J\}$ and $n_j(t)$ is the number of class j jobs which at time t have not yet begun service and have not abandoned the system. Write

$$\Omega = \{(\mathbf{n}, t):\ 0 \le n_j \le L_j,\ 1 \le j \le J, t \ge 0\}$$

for the system's state space.

- At each decision epoch, one of the jobs remaining in the system is chosen for processing. The collection of admissible *actions* is denoted $\mathcal{U}$ and is given by

$$\mathcal{U} = \{(u_1, u_2, \ldots, u_J):\ u_j \in \{0,1\},\ \textstyle\sum_{j=1}^{J} u_j \le 1\}, \quad (1)$$

where $u_j = 1$ means that a job from class j is chosen to be served. In words, this says that each permissible action chooses at most one of the J classes to process next. For clearing systems such as this, it has been shown by Argon et al. (2008) that idling is always suboptimal and thus the server would always be busy under an optimal policy. In other words, any optimal policy will choose the action in which $u_j = 0$ for all j if and only if the system is already empty.

- Let ($\mathbf{n}$, t) be the system state at some decision epoch t. If the action is to process a class j job ($u_j = 1$) with a service time equal to s, the next decision epoch will be t+s and with probability $p(\mathbf{n}'|\mathbf{n}, j, t, s)$ the system will transit to state $(\mathbf{n}', t + s)$. The probability is given by

$$p(\mathbf{n}'|\mathbf{n}, j, t, s) = \prod_{i=1}^{J} \binom{n_i - \delta_{ij}}{n_i'} \left\{\frac{1 - F_i(t+s)}{1 - F_i(t)}\right\}^{n_i'} \left\{\frac{F_i(t+s) - F_i(t)}{1 - F_i(t)}\right\}^{n_i - \delta_{ij} - n_i'}, \quad (2)$$
$$0 \le n_i' \le n_i - \delta_{ij}, 1 \le i \le J,$$

where recall that F_i is the distribution function for lifetimes in class i, and δ_{ij} is the Kronecker delta which is equal to one when $i = j$ and is otherwise zero.

- A *policy* π is any non-anticipative rule to choose the next job for processing after observing the system state at each decision epoch. Let Π denote the set of policies $\pi : \Omega \to \mathcal{U}$ whose actions are prescribed by $\mathcal{U}$. We aim to find such a policy that maximizes the expected number of job completions from initial state ($\mathbf{L}$,0).

The problem described above can be solved in principle by standard dynamic programming approaches; however, this is only tractable for small scale problems due to the curse of dimensionality. Therefore previous research has focused on the development of heuristic policies. See for example Li and Glazebrook (2010) and references therein.

In this work, we follow Whittle (1996) to develop index policies, which are obtained by firstly decomposing the original problem into J single job class problems, and then developing a mapping from states (of the single class) to numerical indices for each class. The computational complexity is significantly reduced due to the latter step only concerning a single class at a time.

3. Relaxation and Decomposition

Given a policy $\pi \in \Pi$, let Z^π denote the total number of service completions (across all J classes) until the time at which the system empties. Then we write

$$V^\pi(\mathbf{L}) = \mathbb{E}[Z^\pi | \mathbf{n}(0) = \mathbf{L}]$$

for the expected total number of jobs served from the initial state ($\mathbf{L}$,0) under policy π. (Here, and throughout, the expectation is taken with respect to the underlying distributions of lifetimes and service times.)

The original problem can be represented in the following form.

$$\text{(P)} \qquad V(\mathbf{L}) = \sup_{\pi \in \Pi} V^\pi(\mathbf{L}) \quad (3)$$

$$\text{subject to}:\ \pi(\mathbf{n}, t) \in \mathcal{U}, \forall (\mathbf{n}, t) \in \Omega, \quad (4)$$

where the constraint in (4) requires that at each decision epoch at most one class is selected for service.

We now propose a different set of policies in which more than one class can be chosen to receive service at each decision epoch. The action space becomes

$$\tilde{\mathcal{U}} = \{(u_1, u_2, \ldots, u_J):\ u_j \in \{0,1\}\}. \quad (5)$$

Clearly we have $\mathcal{U} \subset \tilde{\mathcal{U}}$. We denote by $\tilde{\Pi}$ the set of policies $\tilde{\pi} : \Omega \to \tilde{\mathcal{U}}$ whose action space is given by $\tilde{\mathcal{U}}$, and thus $\Pi \subset \tilde{\Pi}$. We may also extend the definition of $\tilde{\pi}(\mathbf{n}, t)$ to all time points t rather than only the service completion times. We still require that $\tilde{\pi}$ is non-preemptive, and so if $\tilde{\pi}(\mathbf{n},\, t)_j = 1$ for some class j at state ($\mathbf{n}$,t) then this coordinate of the action vector will remain unchanged until completion of the class j service.

Under policies $\tilde{\pi}$, it is possible to choose more than one class to serve simultaneously, and thus the total resource consumed until the system is cleared in general might exceed that which could be offered by a single server. It is therefore natural to penalize policies $\tilde{\pi}$ which lead to many classes being served at once (or, equivalently, to reward those policies which do the opposite): two versions of this idea are considered in the remainder of this section. In section 3.1, we attempt to penalize policies which on average consume more total resource than that used when only one class may be served at any time. This almost allows us to decompose the optimization problem into J independent single-class problems, but to do so requires a further relaxation in which we bound the final emptying time of the entire system by the *sum* of emptying times of each class.

In section 3.2, we suggest an alternative strategy in which the penalty applied varies over time, depending upon how many classes are currently non-empty. This approach circumvents the problem of having to (somewhat crudely) approximate the final emptying time, but introduces its own complication: decomposing our optimization into single-class problems now requires us to assume that the number of non-empty classes is in fact some constant, $M \geq 1$. However, using this additional parameter will prove to be useful when developing our index policies in section 4. Furthermore, it will transpire that even though the two decompositions are obtained from seemingly different penalizations, the second one is in fact a generalization of the first. (Setting M=1 gets us back to the first decomposition.)

3.1. First Decomposition

Our first approach to decomposing the multi-class problem is to impose the requirement that *on average* the total resource consumed by $\tilde{\pi}$ is the same as that which would be consumed if only one class were being served at any time while there are jobs remaining in the system. To be more explicit, firstly define $T^{\tilde{\pi}} = \inf\{t : \; n_j(t) = 0, \; \forall 1 \leq j \leq J\}$ to be the time when the last job leaves the system. Due to the uncertainty of jobs' lifetimes and service times, $T^{\tilde{\pi}}$ is a positive valued random variable which is dependent upon the policy that has been implemented. In addition, define $\|\tilde{\pi}(\mathbf{n},\, t)\|_1$ to be the *1-norm* of the action vector. (Recall that this is simply equal to the sum of the vector's coordinates.) It is clear that any policy $\pi \in \Pi$ solving problem (P) satisfies $\|\pi(\mathbf{n},\, t)\|_1 = 1$ for all $t \leq T^{\pi}$. The first constraint that we consider applying when working with the larger set of policies $\tilde{\Pi}$ is the following:

$$\mathbb{E}\left[\int_0^{T^{\tilde{\pi}}} (1 - \|\tilde{\pi}(\mathbf{n}, t)\|_1)dt \middle| \mathbf{n}(0) = \mathbf{L}\right] = 0, \tag{6}$$

where we shall always work under the natural convention that $\tilde{\pi}(\mathbf{n},\, t)_j = 0$ if $n_j(t) = 0$. (That is, once class j has emptied, it can no longer be selected for service.)

Replacing the constraint (4) in problem (P) by (6), we obtain a relaxed problem:

$$\text{(P1)} \qquad \tilde{V}(\mathbf{L}) = \sup_{\tilde{\pi} \in \tilde{\Pi}} V^{\tilde{\pi}}(\mathbf{L})$$
$$\text{subject to: } \mathbb{E}\left[\int_0^{T^{\tilde{\pi}}} (1 - \|\tilde{\pi}(\mathbf{n}, t)\|_1)dt \middle| \mathbf{n}(0) = \mathbf{L}\right] = 0. \tag{7}$$

Associating a non-negative Lagrangian multiplier W to constraint (6) and adding it to (7), we obtain the following problem. It is a relaxation of (P1) and thus of the original problem (P).

$$\text{(P2)} \quad \sup_{\tilde{\pi} \in \tilde{\Pi}} \mathbb{E}\left[Z^{\tilde{\pi}} + W \int_0^{T^{\tilde{\pi}}} (1 - \|\tilde{\pi}(\mathbf{n}, t)\|_1)dt \middle| \mathbf{n}(0) = \mathbf{L}\right]. \tag{8}$$

Now observe that since

$$\|\tilde{\pi}(\mathbf{n}, t)\|_1 = \sum_{j=1}^{J} \tilde{\pi}(\mathbf{n}, t)_j,$$

we may write

$$\mathbb{E}\left[\int_0^{T^{\tilde{\pi}}} (1 - \|\tilde{\pi}(\mathbf{n}, t)\|_1)dt \middle| \mathbf{n}(0) = \mathbf{L}\right]$$
$$= \mathbb{E}\left[T^{\tilde{\pi}} \middle| \mathbf{n}(0) = \mathbf{L}\right] - \sum_{j=1}^{J} \mathbb{E}\left[\int_0^{T_j^{\tilde{\pi}}} \tilde{\pi}(\mathbf{n}, t)_j dt \middle| \mathbf{n}(0) = \mathbf{L}\right],$$

where we have written $T_j^{\tilde{\pi}}$ for the (random) time at which class j empties, and once again made use of the convention that $\tilde{\pi}(\mathbf{n},\, t)_j = 0$ for $t \geq T_j^{\tilde{\pi}}$. We may also write

$$Z^{\tilde{\pi}} = \sum_{j=1}^{J} Z_j^{\tilde{\pi}},$$

with $Z_j^{\tilde{\pi}}$ being the number of service completions in class j. This allows us to rewrite (P2) as

$$\text{(P2)} \quad \sup_{\tilde{\pi}\in\tilde{\Pi}} \mathbb{E}\left[\sum_{j=1}^{J}\left(Z_j^{\tilde{\pi}} - W\int_0^{T_j^{\tilde{\pi}}} \tilde{\pi}(\mathbf{n},t)_j dt\right) + WT^{\tilde{\pi}} \middle| \mathbf{n}(0)=\mathbf{L}\right]. \tag{9}$$

This problem is still difficult to solve analytically, but note that Equation (9) almost decomposes into J independent single-class optimization problems. The one thing preventing such a decomposition is the $T^{\tilde{\pi}}$ term, since the time at which the entire system empties depends upon all coordinates of $\tilde{\pi}$. However, we may further relax the problem by bounding $T^{\tilde{\pi}}$ above by the sum of emptying times $\sum_{j=1}^{J} T_j^{\tilde{\pi}}$. Writing $T_j^{\tilde{\pi}}$ as $\int_0^{T_j^{\tilde{\pi}}} 1dt$ then leads to our final relaxation of the original optimization problem:

$$(\text{P}^{\text{W}}) \qquad V^W(\mathbf{L}) = \sup_{\tilde{\pi}\in\tilde{\Pi}} \sum_{j=1}^{J} \mathbb{E}\left[Z_j^{\tilde{\pi}} + W\int_0^{T_j^{\tilde{\pi}}}\left(1-\tilde{\pi}(\mathbf{n},t)_j\right)dt \middle| \mathbf{n}(0)=\mathbf{L}\right]. \tag{10}$$

As we have observed, problem (P^{W}) can be decomposed into J single class problems as follows:

$$(\text{p}_\text{j}^{\text{W}}) \; v_j^W(L_j) = \sup_{\pi_j} \mathbb{E}\left[Z^{\pi_j} + W\int_0^{T_j^{\pi}} \left(1-\pi_j(n_j(t),t)\right) dt \middle| n_j(0)=L_j\right], \; 1\le j\le J. \tag{11}$$

Each single class problem can be understood as that the class faces a dedicated server, and the action is to either accept or reject service at each decision epoch in light of the current state of that class. In a slight abuse of notation we still use π_j to denote such single class policies for class j. The Lagrangian multiplier W can be viewed as a subsidy rate to a job class if it rejects the service. In other words, if a job class chooses not to accept service it receives a subsidy of W per unit time. The first term in the right-hand side of Equation (11) is the total expected service completions for class j under a policy π_j (since Z^{π_j} increases by one each time that a job is served), while the second term can be viewed as the total expected subsidy received. It is reasonable to expect that if W is high, the class would prefer to be subsidized for job losses rather than accept service ($\pi_j(t) = 0$). On the other hand, for small values of W the reward from job completion will exceed the expected subsidy received, and so we would expect the class to prefer to accept service.

3.2. Second Decomposition

A weakness of the first decomposition is the way in which we bounded $T^{\tilde{\pi}}$ by $\sum_{j=1}^{J} T_j^{\tilde{\pi}}$; bounding the maximum of a set of random variables by their sum allowed us to decompose the problem, but is somewhat unsatisfactory. In addition, the resulting single class problems take no account whatsoever of the existence of the other classes, whereas it seems reasonable that the selected job class should have to compensate for occupying the single server when there are other job classes competing for service. It may be possible to obtain a better policy by allowing the decision of whether to accept or reject service in the single class problem to use readily available information about how many other classes currently have jobs waiting.

To that end, we now propose an alternative decomposition which, rather than using the constraint in (6) (which required bounding the maximum emptying time), instead uses a constraint based directly on the sum of emptying times. We begin by observing that if $M^{\tilde{\pi}}(t)$ denotes the number of classes which are *non-empty* at time t, when using policy $\tilde{\pi}$, then any policy $\pi \in \Pi$ solving problem (P) satisfies the following:

$$\sum_{j=1}^{J} T_j^{\pi} = \int_0^{\infty} M^{\pi}(t)dt.$$

It follows that any policy $\pi \in \Pi$ also satisfies:

$$\begin{aligned}\sum_{j=1}^{J} T_j^{\pi} &= \int_0^{\infty} M^{\pi}(t)\|\pi(\mathbf{n},t)\|_1 dt \\ &= \sum_{j=1}^{J}\int_0^{\infty} M^{\pi}(t)\pi(\mathbf{n},t)_j dt \\ &= \sum_{j=1}^{J}\int_0^{T_j^{\pi}} M^{\pi}(t)\pi(\mathbf{n},t)_j dt\end{aligned}$$

where we still work under the natural convention that $\pi(\mathbf{n}, t)_j = 0$ if $n_j(t) = 0$. That is, any policy $\pi \in \Pi$ satisfies the constraint:

$$\sum_{j=1}^{J}\int_{0}^{T_j^{\pi}}\Big(1-M^{\pi}(t)\pi(\mathbf{n},t)_j\Big)dt=0.$$

When considering the larger set of policies $\tilde{\Pi}$, it is therefore reasonable to ask that this constraint is satisfied on average:

$$\mathbb{E}\left[\sum_{j=1}^{J}\int_{0}^{T_j^{\tilde{\pi}}}\Big(1-M^{\tilde{\pi}}(t)\tilde{\pi}(\mathbf{n},t)_j\Big)dt\right]=0. \qquad (12)$$

Proceeding as with our first decomposition, we associate a non-negative Lagrangian multiplier W to this new constraint, yielding the following problem:

$$\sup_{\tilde{\pi}\in\tilde{\Pi}}\sum_{j=1}^{J}\mathbb{E}\left[Z_j^{\tilde{\pi}}+W\int_{0}^{T_j^{\tilde{\pi}}}\Big(1-M^{\tilde{\pi}}(t)\tilde{\pi}(\mathbf{n},t)_j\Big)dt\,\Big|\,\mathbf{n}(0)=\mathbf{L}\right]. \qquad (13)$$

This would now decompose into J separate single class problems if they were they not still linked together by $M^{\tilde{\pi}}(t)$, but since this process depends upon all coordinates of $\tilde{\pi}$, it is infeasible to solve analytically the problem in (13). So for the time being, we instead propose to consider a related single class problem in which we replace $M^{\tilde{\pi}}(t)$ by an arbitrary constant $M \geq 1$:

$$(\mathrm{p}_{\mathrm{j}}^{\mathrm{W,M}})\quad v_j^{W,M}(L_j)=\sup_{\pi_j}\mathbb{E}\left[Z^{\pi_j}+W\int_{0}^{T_j^{\pi}}\big(1-M\pi_j(n_j(t),t)\big)dt\,\Big|\,n_j(0)=L_j\right],\quad 1\leq j\leq J. \qquad (14)$$

This new single-class problem has a different interpretation to that in our first decomposition. Once again the job class has the choice of accepting or rejecting a service and receives a constant subsidy of W per unit time until the next decision epoch if it rejects a service (i.e., if $\pi_j(t) = 0$); however, if it accepts the service ($\pi_j(t) = 1$) then it receives an instantaneous payoff of 1 unit (reflected by a unit increase in Z^{π_j}) but also incurs a *penalty* at rate $W(M - 1)$ while completing this service. This penalty reflects the fact that there are $(M - 1)$ other non-empty classes (not including class j) which might possibly want to be accepting service.

Note that if $M = 1$ in Equation (14) then problem $(\mathrm{p}_{\mathrm{j}}^{\mathrm{W,M}})$ reduces to $(\mathrm{p}_{\mathrm{j}}^{\mathrm{W}})$ in Equation (11), and so it suffices to study the more general problem $(\mathrm{p}_{\mathrm{j}}^{\mathrm{W,M}})$ in what follows. If there exists a critical value of W at which the optimal policy is indifferent between accepting or rejecting service, then we shall follow the literature and call this critical subsidy rate *Whittle's index*. In the following section, we show the existence of Whittle's index for exponentially distributed lifetimes and service times in problem (14), for all fixed values of $M \geq 1$, and develop closed form equations to calculate the corresponding Whittle's index values. We then use these values to propose modified Whittle's index policies for the original problem (P), one of which takes into account the fact that $M(t)$ (the number of non-empty classes at time t) is of course a decreasing stochastic process, rather than a constant.

4. Indexability and Index Policies for the Markovian Case

Even though, in this section, we study the indexability in the Markovian case, by no means do we claim that the lifetime and service time are exponentially distributed in reality. However, this assumption facilitates our analysis on indexability and the calculation of index values. It also helps us develop insights into the switching patterns of the index policies. As shown in section 5, the proposed index policy works well even in non-Markovian settings where the lifetime and service time are no longer exponentially distributed. The insights we obtain are therefore applicable to more general conditions beyond the Markovian case.

4.1. Indexability for A Single Job Class

We now focus on a single class problem and thus drop the subscript j in all notation. The problem $(\mathrm{p}^{\mathrm{W,M}})$ now concerns a single job class that is served by a single server dedicated to this class. At each decision epoch, two actions may be selected; either to accept the service or reject it. If the action is to accept service, one job is chosen for service that carries on until the service completion; this brings an immediate reward of one, but at a penalty rate of $W(M - 1)$ while completing this service. If the decision is to reject service, subsidy W is received for each time unit that the server remains idle.

In the Markovian case, each job's lifetime and service time are exponentially distributed, with rates θ and μ respectively. Define $\rho = \theta/\mu$ to be the mean service time divided by the mean lifetime. We name this rather useful parameter *opportunity loss*, which essentially measures the ratio of the mean number of jobs having lost relative to the mean number of jobs still waiting when the current service completes. Indeed, given that there are currently $n - 1$ jobs waiting and one being served, the probability that k jobs are still waiting when the current service completes is given by

$$
\begin{aligned}
p(k|n) &= \int_0^\infty \mu e^{-\mu s} \binom{n-1}{k} e^{-\theta s k}(1-e^{-\theta s})^{n-1-k} ds \\
&= \binom{n-1}{k} \sum_{j=0}^{n-1-k} \binom{n-1-k}{j} \frac{(-1)^j}{1+(k+j)\rho} \\
&= \frac{\Gamma(n)}{\rho\Gamma(n+1/\rho)} \frac{\Gamma(k+1/\rho)}{\Gamma(k+1)}, \quad k = 0, , n-1.
\end{aligned} \tag{15}
$$

The second equality here is obtained by applying the binomial theorem to $(1 - e^{-\theta s})^{n-1-k}$, and in Equation (15) we write Γ for the gamma function. Thus, the mean number of jobs still waiting when this service completes is given by $(n - 1)/(1 + \rho)$, and accordingly the mean number of jobs having lost is $(n - 1)\rho/(1 + \rho)$. The former/latter is clearly a decreasing/increasing function of ρ, and when $\rho = 1$ we expect exactly half of the remaining jobs to expire before the current service completes and the other half to survive. When $\rho > 1$ (i.e., more jobs are expected to expire than those expected to survive during the service completion), we might expect the server to choose to reject service in favor of subsidies when n is large, then to elect to accept service when n has decreased sufficiently for the payoff from completing a service to exceed the expected gain from further subsidies. The opposite strategy may similarly be expected to be optimal when $\rho < 1$. We shall show below that this intuition is correct, and that the optimal service policy is profoundly dependent upon the opportunity loss ρ.

Let N_t denote the number of jobs in the system at time t, *including* the job being served (if there is one). Thus we start with $N_0 = L$, and N_t decreases by one whenever a job completes service or a waiting job departs the system due to their lifetime being reached. (Note that our assumptions on the lifetime and service time distributions mean that the probability of N_t instantaneously decreasing by more than one is precisely zero.) A policy π for the single-class problem can be viewed as a function π: $\{0, 1, 2, \ldots\} \to \{0, 1\}$, where

$$
\pi(n) = \begin{cases} 0 & \text{if policy } \pi \textit{ rejects} \text{ service when } N_t = n \\ 1 & \text{if policy } \pi \textit{ accepts} \text{ service when } N_t = n. \end{cases}
$$

Given a policy π, the payoff is equal to the number of jobs served over the period $[0, T^\pi]$, minus the penalty incurred while serving, plus the total subsidy received while the server is idle during this time. Note that we stop receiving any benefit when N_t hits zero, which happens in finite time with probability one, whatever policy we use. (But note also that the trajectory of N_t depends on π, since the time at which jobs depart the system will depend on whether or not they start service before expiring, and so we shall henceforth write N_t^π to make this dependence explicit.) Given π, let Z_t^π denote the number of service initiations by time t. Then the total payoff when using policy π over the interval $[0, t]$, with $t \leq T^\pi$, can be expressed as

$$
Z_t^\pi + W \int_0^t (1 - M\pi(N_s^\pi))ds. \tag{16}
$$

It is convenient for the single-class problem to set $\pi(0) = 1/M$: this ensures that the integrand in Equation (16) is zero for $s \geq T^\pi$, and implies that the payoff received over $[0, t]$ converges almost surely as $t \to \infty$ to the random variable

$$
Z^\pi + W \int_0^{T^\pi} (1 - M\pi(N_s^\pi))ds.
$$

The value function of policy π is defined as

$$
v^\pi(n) = \mathbb{E}\left[Z^\pi + W \int_0^{T^\pi} (1 - M\pi(N_s^\pi))ds \middle| N_0 = n \right]. \tag{17}
$$

Our aim is to determine a policy π^* which maximizes this function.

THEOREM 1. *Given values for θ, μ, M and W, define the set A^* as follows:*

$$
A^* = \begin{cases} \left\{ n \in \mathbb{N} : n\left(\frac{WM}{\mu} - 1\right) \leq \frac{W}{\theta}(\rho - 1) \right\} & \text{if } \rho \geq 1; \\ \left\{ n \in \mathbb{N} : n\left(1 - \frac{W}{\mu}(M - p(0|n))\right) \geq \frac{W}{\theta} \right\} & \text{if } \rho \leq 1. \end{cases} \tag{18}
$$

Then the policy π^* defined by

$$
\pi^*(n) = \begin{cases} 1 & n \in A^* \\ 0 & \text{otherwise} \end{cases}
$$

is the unique optimal policy which maximizes the value function (17).

That is, A^* determines the optimal *acceptance set*: it is optimal to accept service if and only if the number of jobs in the system belongs to A^*. We observe that if $\rho = 1$ (meaning that lifetimes and service times have the same mean), both formulas in (18) simplify to

$$A^* = \left\{ n \in \mathbb{N} : n\left(\frac{WM}{\mu} - 1\right) \le 0 \right\}.$$

In other words, when $\rho = 1$ the optimal policy is to *always* accept service if $WM \le \mu$, and otherwise *always* to reject.

Corollary 1. *The policy π^* is monotonic in n: $\pi^*(n) \ge \pi^*(n+1)$ when $\rho \ge 1$, and $\pi^*(n) \le \pi^*(n+1)$ when $\rho \le 1$.*

The proofs of Theorem 1 and Corollary 1 can be found in the Appendix. An important consequence of Corollary 1 is that when $\rho \ge 1$, there exists some critical value n^* with the property that the acceptance set is given by $\{1, 2, , n^*\}$, where we write $n^* = 0$ if $A^* = \emptyset$ and $n^* = \infty$ if $A^* = \mathbb{N}$. Similarly, if $\rho \le 1$ there exists n^* with the property that $A^* = \{n^*+1, n^*+2, \}$. We shall write $A^*(W)$ and $n^*(W)$ when we wish to emphasize the dependence upon the subsidy rate W.

We now formally define indexability of a job class as follows.

Definition 1. *A job class is indexable if $W \le W' \Rightarrow A^*(W') \subseteq A^*(W)$.*

In other words, when a class is indexable, the set of states at which it is optimal to accept service decreases as the subsidy rate increases. Therefore when the subsidy rate is larger it becomes more attractive to reject the service and instead receive subsidy, and vice versa. This motivates the following definition.

Definition 2. *For an indexable job class, the Whittle's index is defined as*

$$w(n) = \sup\{W : n \in A^*(W)\}.$$

Thus, $w(n)$ is the maximum subsidy rate under which it is optimal to accept service when there are n jobs left in the class. Proposition 1 shows that the single class problem is indexable in the Markovian case, for all values of ρ and M. Therefore, as $W \to 0$ the acceptance set grows until $A^*(W) = \{1, 2, \ldots, L\}$, and at the other extreme, $A^*(W) \to \emptyset$ as $W \to \infty$.

Proposition 1. (Whittle's Index Values). *For any fixed $M \ge 1$, the job class with loss rate θ and service rate μ is indexable, with Whittle's index given as follows:*

$$w(n) = \begin{cases} \frac{n\theta}{1+(nM-1)\rho} & \text{if } \rho \ge 1, \\ \frac{n\theta}{1+n(M-p(0|n))\rho} & \text{if } \rho \le 1. \end{cases} \qquad (19)$$

Proof. When $\rho \ge 1$, rearranging Equation (18) we see that $n \in A^*$ if and only if

$$W \le w(n) := \frac{n\theta}{1+(nM-1)\rho}.$$

For any fixed n this inequality will hold (giving $\pi^*(n) = 1$) for all values of W less than the critical value $w(n)$, and fail otherwise (giving $\pi^*(n) = 0$); thus $\pi^*(n)$ is a decreasing function of W. We conclude that as W increases the size of the acceptance set can only decrease, and hence the class is indexable.
Rearranging Equation (18) when $\rho \le 1$ shows that in this case $n \in A^*$ if and only if

$$W \le w(n) := \frac{n\theta}{1+n(M-p(0|n))\rho},$$

and so the same conclusion applies. □

Corollary 2. *The index value $w(n)$ is a decreasing (respectively, increasing) function of n when $\rho \ge 1$ (respectively, $\rho \le 1$). Furthermore, $w(n)$ is a decreasing function of M for all values of ρ.*

Proof. It is immediate from Equation (19) that $w(n)$ is a decreasing function of M.

When $\rho \ge 1$ we see that

$$\begin{aligned} w(n+1) - w(n) &\propto (n+1)(1+(nM-1)\rho) \\ &\quad - n(1+((n+1)M-1)\rho) \\ &= 1-\rho \le 0, \end{aligned}$$

and so $w(n)$ is decreasing in n. Similarly, when $\rho < 1$, we obtain

$$\begin{aligned} w(n+1) - w(n) &\propto (n+1)(1+n(M-p(0|n))\rho) \\ &\quad - n(1+(n+1)(M-p(0|n+1))\rho) \\ &= 1 - n(n+1)\rho(p(0|n) - p(0|n+1)) \\ &= 1 - n(n+1)\rho\frac{p(0|n)}{1+n\rho} \end{aligned}$$

where this final equality follows from Equation (A.4) in the Appendix. We now use the inequality $p(0|n) \le 1/n$ (which holds for any $\rho \le 1$) to see that

$$\begin{aligned} 1 - n(n+1)\rho\frac{p(0|n)}{1+n\rho} &\ge 1 - \frac{(n+1)\rho}{1+n\rho} \\ &= \frac{1-\rho}{1+n\rho} \ge 0. \end{aligned}$$

So in this case $w(n)$ is an increasing function of n, as claimed. □

REMARK 1. Throughout this work, we always consider the payoff for serving a job to be identically one. Changing this to some other (positive) value is a trivial exercise, and it is easy to see that the resulting Whittle's index values will scale linearly with the payoff value.

4.2. Index Policies for the Original Problem

Having shown that all job classes are indexable and obtained their Whittle's index values, we are ready to develop service policies for the original problem. We relax the assumption that M is a constant and restore its original definition as the number of non-empty classes in state $\mathbf{n}$, written as $M(\mathbf{n}) = \sum_{j=1}^{J} \mathbb{1}\{n_j > 0\}$, where $\mathbb{1}$ is an indicator. The Whittle's index values in Equation (19) now take the following form for class j:

$$w_j(\mathbf{n}) = \begin{cases} \frac{n_j\theta_j}{1+(n_jM(\mathbf{n})-1)\rho_j} & \text{if } \rho_j \geq 1, \\ \frac{n_j\theta_j}{1+n_j(M(\mathbf{n})-p(0|n_j))\rho_j} & \text{if } \rho_j \leq 1. \end{cases} \tag{20}$$

We follow the seminal work of Whittle (1988) and define the following index policy for the original problem.

Dynamic Whittle's Index Policy (DWI): Suppose that the system occupies state $\mathbf{n}$ at a decision epoch. The index policy DWI always allocates the server to the job class j^* which satisfies

$$w_{j*}(\mathbf{n}) = \max_{\substack{1 \leq j \leq J \\ n_j \geq 1}} w_j(\mathbf{n}).$$

We call this policy "*Dynamic Whittle's Index policy*"; here "dynamic" emphasizes that the index values are dependent upon the (dynamic) number of job classes that still compete for service, in addition to the single class state n.

We also derive an index policy that follows the first decomposition.

Whittle's Index Policy (WI): Suppose that the system occupies state $\mathbf{n}$ at a decision epoch. The index policy WI always allocates the server to the job class j^* which satisfies

$$w'_{j*}(n_{j*}) = \max_{\substack{1 \leq j \leq J \\ n_j \geq 1}} w'_j(n_j),$$

where the index values are calculated by letting $M(\mathbf{n}) \equiv 1$ in Equation (20), as shown below.

$$w'_j(n_j) = \begin{cases} \frac{n_j\theta_j}{1+(n_j-1)\rho_j} & \text{if } \rho_j \geq 1, \\ \frac{n_j\theta_j}{1+n_j(1-p(0|n_j))\rho_j} & \text{if } \rho_j \leq 1. \end{cases}$$

Note that w'_j values are solely determined by the class j themselves.

4.3. An Illustrative Example

We consider a $J = 2$ class problem, with key parameters as follows.

$$L_1 = 20,\ \theta_1 = 0.15,\ \mu_1 = 0.14,$$
$$L_2 = 20,\ \theta_2 = 0.05,\ \mu_2 = 0.20.$$

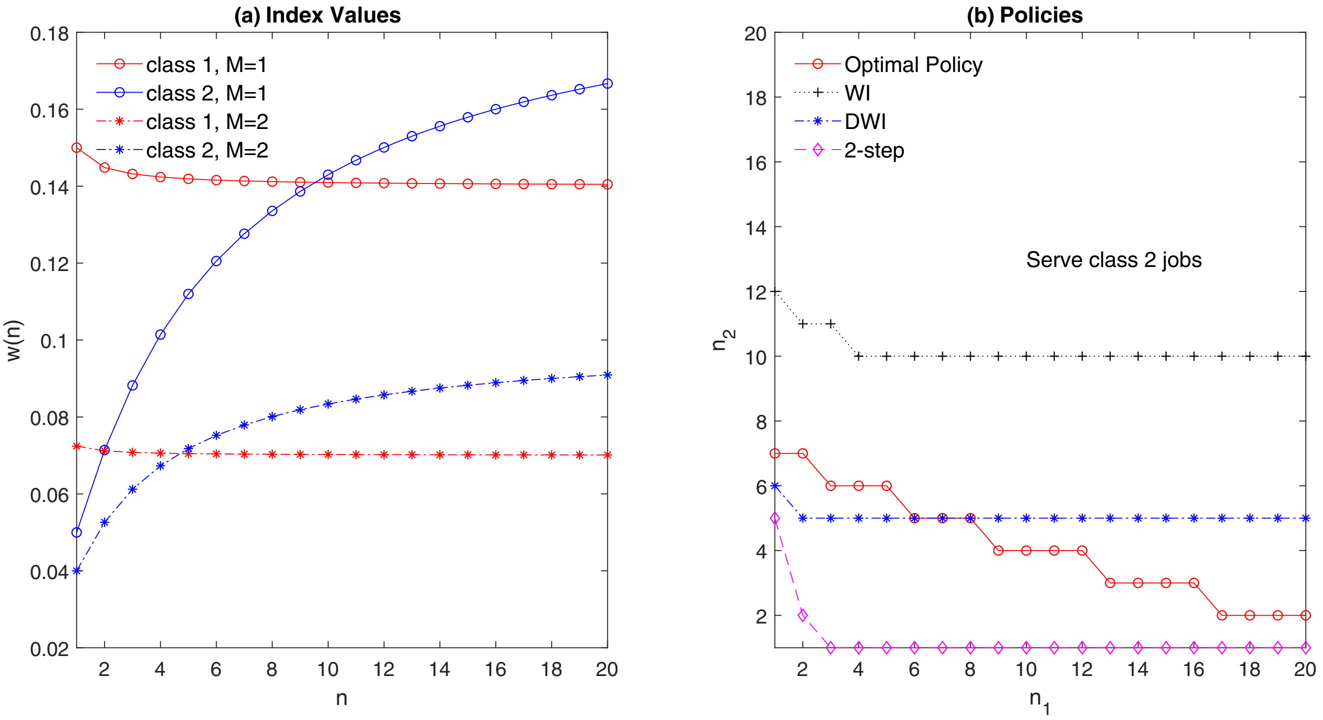
Therefore class 1 has shorter lifetime but longer service time than class 2. Moreover we have $\rho_1 > 1 > \rho_2$. We calculated the Whittle's index values for different M and obtained both WI and DWI policies. Furthermore, for this small example, we calculated the optimal policy by solving the corresponding Bellman equations. Also included is the two-step policy proposed by Jacobson et al. (2012). The results are plotted in Figure 1. The area above each policy curve, including the curve itself, is where class 2 jobs are prioritized, while that below is where class 1 jobs are given priority.

It is first of all evident that, for fixed M, the index values decrease with n for class 1 and increase for class 2, and the index values decrease with M for both classes; this is of course to be expected following Corollary 2. The policy plot shows that all four policies (including the optimal policy) take the form of threshold policies in this example. They all begin with prioritizing class 2 jobs, due to their shorter service times, and switch to prioritizing class 1 jobs after a certain threshold. However, the switching points differ between policies, with WI switching much earlier than the other two. Indeed, pretty much as soon as n_2 drops below 10 and $w'_2(n_2)$ falls below $w'_1(n_1)$ (as shown by the points corresponding to $M = 1$ in Figure 1a), the service is always allocated to class 1 jobs under policy WI. In clear contrast, DWI's switching points come much later and are closer to those of the optimum. The consideration of the other non-empty classes helps delay the switching. From Figure 1a, it is observed that the crossing point of the bottom two index value curves (for $M = 2$) comes at $n = 5$, much later than $n = 10$ for the top two curves ($M = 1$). The two-step policy switches rather too late. We also calculated the suboptimality for each heuristic (as defined by Equation (24) in section 5.1). The suboptimality for DWI is only 0.06%, which is significantly lower than 0.50% for WI and 0.37% for two-step. The enhancement due to the second decomposition is therefore substantial.

5. Numerical Experiments

In the following discussion, we study extensively the performance of the proposed policies. Section 5.1 lists

Figure 1 The Whittle's Index Values (a) and the Index Policies WI, DWI, Two-Step and An Optimal Policy (b) for the Illustrative Example [Color figure can be viewed at wileyonlinelibrary.com]



a few alternative heuristic policies from the literature that are used as the benchmarks in all the experiments. In section 5.2, we test their performances in the Markovian case as described in section 4. In section 5.3, we extend to non-exponential cases with Weibull distributed lifetimes. A number of sensitivity analyses are reported in section 5.4 and in section 5.5 we provide further discussions of our numerical experiments.

5.1. Heuristic Policies

In addition to our proposed index policies WI and DWI, the following five heuristics from the literature are included. For brevity, we describe all these heuristics for the Markovian problems. As we shall see in section 5.3, they can be also applied in settings with non-exponential distributions.

- **two-step**: A state-dependent policy was proposed by Jacobson et al. (2012) for problems with two classes. We extend this policy to multiple job classes. Given that the system occupies state **n** at a decision epoch, the two-step policy chooses to serve the non-empty class j with the largest value of

$$\frac{\mu_j}{\mu_j + \sum_{i=1}^{J}(n_i - \delta_{ij})\theta_i}. \tag{21}$$

Essentially this policy gives priority to the classes which would result in the minimum mean number of abandonments during the next service.

- **Threshold**: In Jacobson et al (2012) a threshold policy was also proposed. For $J = 2$ problems, the threshold $\mathcal{T}$ is obtained as follows.

$$\mathcal{T} = \frac{\theta_1 - \theta_2}{\mu_2 - \mu_1} \max\left\{\frac{\mu_2}{\theta_1}, \frac{\mu_1}{\theta_2}\right\}. \tag{22}$$

For a state **n**, the threshold policy always chooses to serve the class with larger loss rate if $n_1 + n_2 \le \mathcal{T}$; otherwise the jobs from the other class are served. Note that this policy is not applicable to problems with three or more job classes.

- **ThetaMu**: Renumber the job classes in descending order of the quantity $\theta_j\mu_j$, such that

$$\theta_1\mu_1 \ge \theta_2\mu_2 \ge \cdots \ge \theta_J\mu_J.$$

In any state **n**, the policy ThetaMu always chooses to serve the non-empty class with the largest $\theta_j\mu_j$ value. Essentially this is a static priority policy that favors job classes with large service rates and/or larger loss rates. It has been shown by Glazebrook et al. (2004) that the ThetaMu policy has optimal performance in a "no premature job loss" limit.

- **TCF**: The *time critical first* rule resembles the static START policy in practice and always prioritizes jobs with shorter expected lifetimes. Specifically, it always chooses to serve the non-empty class j with the largest value of θ_j.
- **SEPT**: The *shortest expected processing time* rule is another static policy commonly used in many applications. SEPT favors jobs with shorter expected service times. Specifically, it always chooses to serve the non-empty class j with the largest value of μ_j.

Moreover, we can obtain the optimal policy (for modest values of J and L) by solving the Bellman equations. For problems in the Markovian case, we have

$$V(\mathbf{n})=1+\max_{\substack{1\leq j\leq J\\ n_j\geq 1}}\left\{\int\limits_0^{\infty}\sum_{\mathbf{n}'}p(\mathbf{n}'|\mathbf{n},j,s)V(\mathbf{n}')\mu_j e^{-\mu_j s}ds\right\},\ \mathbf{n}\neq\mathbf{0},$$
$$V(\mathbf{0})=0, \tag{23}$$

where

$$p(\mathbf{n}'|\mathbf{n},j,s)=\prod_{i=1}^{J}\binom{n_i-\delta_{ij}}{n'_i}e^{-\theta_i s n'_i}\left(1-e^{-\theta_i s}\right)^{n_i-\delta_{ij}-n'_i}.$$

Similarly, the value functions for a specific policy π can be obtained from solving the equations below.

$$V^{\pi}(\mathbf{n})=1+\left\{\int_0^{\infty}\sum_{\mathbf{n}'}p(\mathbf{n}'|\mathbf{n},\pi(\mathbf{n}),s)V^{\pi}(\mathbf{n}')\mu_{\pi(\mathbf{n})}e^{-\mu_{\pi(\mathbf{n})}s}ds\right\},\quad \mathbf{n}\neq\mathbf{0},$$
$$V^{\pi}(\mathbf{0})=0.$$

The performance of policy π can then be measured by its suboptimality, defined as

$$\Delta^{\pi}(\mathbf{L})=100(V(\mathbf{L})-V^{\pi}(\mathbf{L}))/V(\mathbf{L}). \tag{24}$$

5.2. The Markovian Case

As indicated in section 4, the opportunity loss $\rho_j=\theta_j/\mu_j$ has a significant impact on the resulting heuristic policies and their performance. We first study the scenario in which all job classes have a similar level of opportunity loss. In light of real emergency response situations where jobs may well have different degrees of urgency and service requirements, we move on to investigate the performance of alternative policies when job classes have clearly different levels of opportunity losses. To cover a wide range of different problem scenarios, we randomly generate the problem parameters, as specified in each section below.

5.2.1. Similar Level of Opportunity Losses Between Job Classes. The key parameters in the problem considered here are the service rates μ_j, loss rates θ_j, and the initial numbers of jobs L_j, which are randomly sampled as below.

$$\mu_j\sim U[0.1,1.0],\ \text{for all cases} \tag{25a}$$

$$\rho_j\sim\begin{cases}U[0.1,0.5], & \text{Low opportunity loss}\\ U[0.5,2.0], & \text{Medium opportunity loss}\\ U[2.0,10.0], & \text{High opportunity loss}\end{cases} \tag{25b}$$

$$L_j\sim\begin{cases}DU[10,20], & J=2\\ DU[5,10], & J=3\\ DU[2,5], & J=4\end{cases} \tag{25c}$$

We consider three different levels of opportunity loss as shown in Equation (25b), which are arranged in ascending order of ρ_j values. For each of these levels and for each $J\in\{2, 3, 4\}$, we randomly generate 500 testing instances according to Equation (25a–25c). Specifically, for each level of opportunity loss and each $j\leq J$, we firstly sample the service rates μ_j from the distribution $U[0.1, 1.0]$, and ρ_j from the corresponding uniform distributions. These determine the loss rates $\theta_j=\mu_j\rho_j$. The initial number of jobs L_j are then sampled from corresponding discrete uniform distributions. Note that their values are limited and reduced with J such that the optimal policies can be obtained within reasonable time. Moreover, we are only interested in situations where the jobs with longer expected lifetimes have shorter expected service times, and vice versa. These conditions are common in practice and the optimal service policies in such situations are far from obvious. Without loss of generality, we require that $\theta_j<\theta_{j+1},\ \mu_j>\mu_{j+1}$. Therefore, instances which do not meet these conditions are re-sampled.

For every problem instance, value iteration is employed to compute the mean number of service completions achieved under alternative heuristics and an optimal policy. Then the suboptimality for each heuristic is computed. For each opportunity loss category, the average performance of every heuristic across 500 instances is calculated and summarized in Table 1.

The results in Table 1 confirm that ThetaMu and two-step policies perform well when jobs have low and high opportunity losses, respectively, which agree with the previous results in the literature. Their performances clearly worsen in other scenarios, however. The index policy WI produces rather weak results across the board. Even though it is better than ThetaMu in the

Table 1 Suboptimality (in %) for Problems with the Similar Level of Opportunity Loss in the Markovian Case (The strongest performances are highlighted in bold)

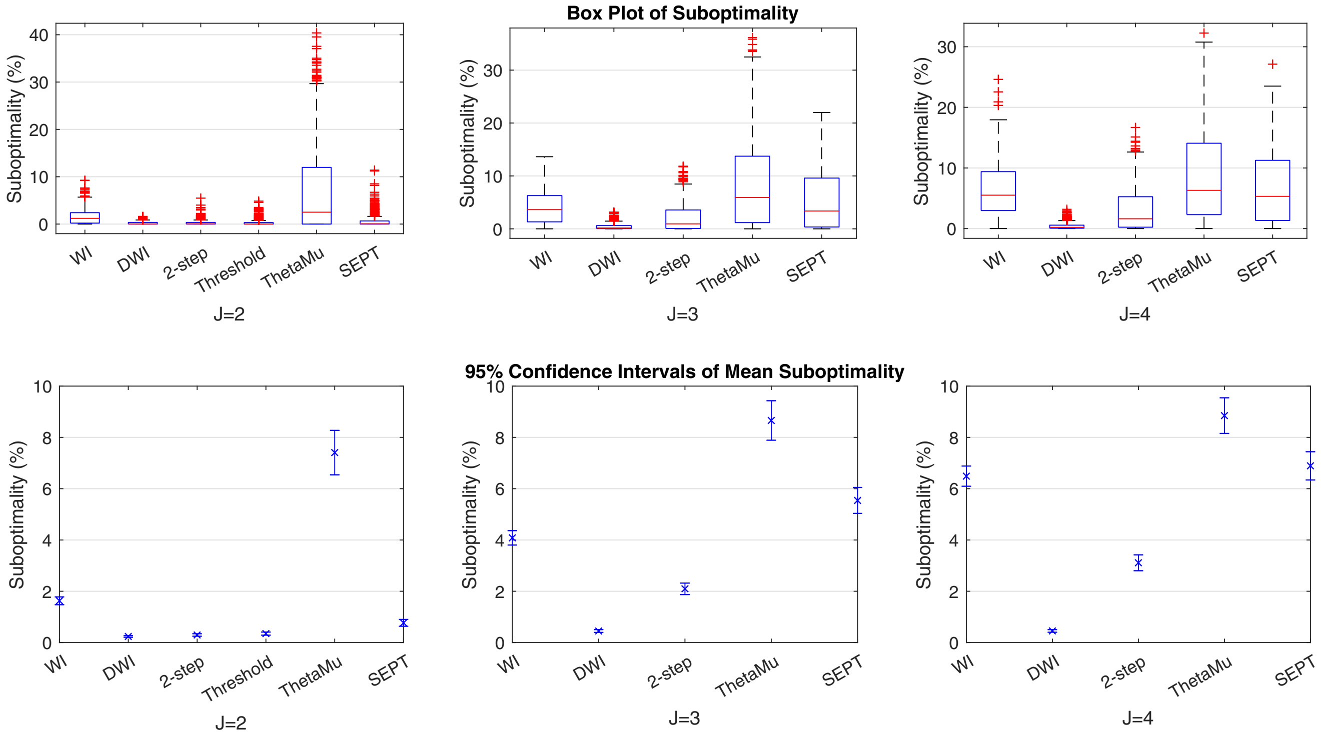
	$J = 2$			$J = 3$			$J = 4$		
Opportunity loss	Mean	Max	SD	Mean	Max	SD	Mean	Max	SD
Low									
WI	2.14	8.26	1.61	3.10	9.54	2.09	3.20	12.89	2.54
DWI	1.53	**4.81**	**1.04**	1.79	**4.34**	**0.84**	1.62	**5.47**	**0.87**
Two-step	1.18	8.67	1.82	1.41	9.33	1.85	1.42	7.20	1.61
Threshold	0.99	5.70	1.44	–	–	–	–	–	–
ThetaMu	**0.88**	17.41	2.04	**1.02**	10.50	1.79	**0.94**	7.49	1.41
TCF	6.58	43.85	8.95	7.62	36.53	7.78	5.48	25.89	5.46
SEPT	2.15	17.23	3.34	2.68	17.06	3.33	2.74	14.25	3.08
Medium									
WI	0.79	2.92	0.62	2.23	6.07	1.34	4.11	12.52	2.42
DWI	**0.09**	**0.97**	**0.16**	**0.16**	**1.10**	**0.19**	**0.18**	**1.76**	**0.22**
Two-step	0.12	1.57	0.23	0.27	2.61	0.49	0.37	4.41	0.64
Threshold	0.10	1.02	0.16	–	–	–	–	–	–
ThetaMu	2.60	20.79	4.54	3.55	20.43	4.39	2.89	18.10	3.43
TCF	10.32	44.46	9.91	11.83	41.81	9.22	11.47	38.25	8.16
SEPT	0.74	8.13	1.36	1.25	10.04	1.89	1.45	11.14	1.92
High									
WI	0.48	2.62	0.40	1.07	4.81	0.76	2.31	10.72	2.05
DWI	0.20	1.87	0.22	0.25	2.39	0.30	0.33	4.16	0.51
Two-step	**0.01**	**0.23**	**0.03**	**0.05**	**1.75**	**0.13**	**0.08**	**2.49**	**0.22**
Threshold	0.03	1.40	0.09	–	–	–	–	–	–
ThetaMu	3.35	21.14	4.69	4.35	18.74	4.66	5.20	19.66	4.67
TCF	9.90	41.30	8.58	12.89	40.75	8.73	12.76	35.27	7.13
SEPT	0.20	3.39	0.42	0.40	3.74	0.64	0.49	6.49	0.84

medium to high scenarios, it is always outperformed by two-step and Threshold (where applicable). In sharp contrast, the other index policy DWI achieves significant improvement over WI, regardless of the opportunity loss level. The suboptimality of DWI is the lowest on average among all heuristics in the medium category. In the other two categories, the average suboptimality gap between DWI and ThetaMu(two-step) is always less than 0.7%(0.3%). Therefore, DWI produces comparable average performance to the best performing policies across all scenarios. However, what makes DWI standing out is its consistency and robustness in performance. In fact, its worst-case performance and standard deviation are always the best except for the high scenarios, which makes DWI the most robust policy. Its overall worst case performance is at most 5.47%, while the value for the other alternatives is 12.89%(WI), 9.33%(2-step), 5.70%(Threshold), 21.14% (ThetaMu), 43.85%(TCF) and 17.23% (SEPT). We also observe that Threshold is strong when it is applicable; it is always the second best policy on average. Its worst performance is quite good as well. However, as we shall see, its performance reduces significantly for the Weibull lifetime problems. Among the three static policies, TCF is always the weakest, far worse than the others, while SEPT shows much stronger performance over ThetaMu in medium to high scenarios.

5.2.2. Different Level of Opportunity Losses between Job Classes. The numerical experiments so far are restricted to situations where all job classes have a similar level of opportunity loss, either being low, medium, or high. In real life, however, this may not always be the case. It is very likely that there exist a combination of jobs with various degrees of urgency and treatment needs. Indeed in some mass casualty incident situations, it is reasonable to expect that some jobs will have a low opportunity loss, while others have medium or high opportunity losses. In this section we consider the situation of this kind, by requiring that ρ_j values are sampled from different levels. Again we consider problems with J = 2, 3, 4 classes, with their ρ_j values now sampled as below.

$$\begin{cases} \rho_1 \sim U[0.1,1.0], \rho_2 \sim U[1.0,10.0] & J=2 \\ \rho_1 \sim U[0.1,0.5], \rho_2 \sim U[0.5,2.0], \rho_3 \sim U[2.0,10.0] & J=3 \\ \rho_1 \sim U[0.1,0.5], \rho_2 \sim U[0.5,1.0], \rho_3 \sim U[1.0,2.0], \rho_4 \sim U[2.0,10.0] & J=4 \end{cases} \tag{26}$$

Figure 2 Suboptimality (in %) for Problems with Different Level of the Opportunity Loss in the Markovian Case [Color figure can be viewed at wileyonlinelibrary.com]



The other two parameters for each class are still sampled independently according to Equations (25a) and (25c). Again 500 instances are randomly generated for each *J* and the results are presented in Table 2. The distributions of the suboptimality and the confidence intervals are plotted in Figure 2, in which TCF is excluded in order to show clearer contrast between the other policies.

It can be observed that DWI performs exceptionally well in these problems, with the strongest performance among all heuristics. Compared to the results in Table 1, its mean suboptimality is much smaller, while all the other heuristics perform significantly worse. The box plots indicate that DWI has the lowest suboptimality on all measures (mean, median, and all quartiles) in every scenario. Indeed, the mean suboptimality is at most 0.45% and the worst case at most 3.18% across all instances. These are far better than the second best heuristic, two-step, whose mean suboptimality could be as large as 3.11% and worst case 16.68%. Threshold shows comparable performance, followed by WI. SEPT comes close and outperforms ThetaMu. It also delivers better results than WI when *J* = 2, but this relationship is reversed for larger *J*. Still TCF provides the weakest performance with mean of 19.79% and maximum of 65.28%. The dominant performance of DWI is further evidenced by its 95% confidence interval of the mean suboptimality, which is tight and significantly below all the other alternatives, especially when there are more job classes.

Table 2 Suboptimality (in %) for Problems with Different Level of the Opportunity Loss in the Markovian Case (The strongest performances are highlighted in bold)

	J = 2			*J* = 3			*J* = 4		
	Mean	Max	SD	Mean	Max	SD	Mean	Max	SD
WI	1.63	9.24	1.74	4.08	13.63	3.19	6.49	24.61	4.49
DWI	**0.24**	**1.61**	**0.36**	**0.45**	**3.16**	**0.63**	**0.45**	**3.18**	**0.59**
Two-step	0.30	5.48	0.56	2.10	11.83	2.54	3.11	16.68	3.53
Threshold	0.35	4.85	0.69	–	–	–	–	–	–
ThetaMu	7.41	40.40	9.87	8.66	36.16	8.76	8.85	32.26	7.93
TCF	15.30	65.28	16.99	19.12	62.46	14.87	19.79	58.5	12.88
SEPT	0.77	11.40	1.51	5.54	21.98	5.75	6.89	27.12	6.27

5.3. Weibull Lifetime and Deterministic Service time

In this section, we depart from the assumptions of exponentially distributed lifetime and service times. Although this makes questions of indexability etc. rather intractable, it is very much of interest to investigate numerically how well our heuristic policy performs in other scenarios. Here we shall consider deterministic service times and Weibull distributed lifetimes.

For Weibull distributed lifetimes, the loss rate is no longer constant but changes with time. We shall use Weibull(α_j, β_j) to denote the distribution function for class j:

$$F_j(s) = 1 - e^{-(s/\beta_j)^{\alpha_j}}.$$

The mean remaining lifetime $\mathbb{E}(X_j - t|X_j > t)$ can be calculated as follows, where for presentation purposes we temporarily discard the subscript j:

$$\mathbb{E}(X - t|X > t) = \frac{\beta\Gamma(1/\alpha, (t/\beta)^{\alpha})}{\alpha} e^{(t/\beta)^{\alpha}},$$

where $\Gamma(1/\alpha, (t/\beta)^{\alpha})$ is the upper incomplete Gamma function which takes the form

$$\Gamma(1/\alpha, (t/\beta)^{\alpha}) = \int_{(t/\beta)^{\alpha}}^{\infty} x^{1/\alpha - 1} e^{-x} dx.$$

The loss rate at time t is given by the reciprocal of $\mathbb{E}(X - t|X > t)$, as written below.

$$\theta(t) = \frac{\alpha}{\beta\Gamma(1/\alpha, (t/\beta)^{\alpha})} e^{-(t/\beta)^{\alpha}},$$

and this allows us to define $\rho(t) = \theta(t)\mu^{-1}$, where μ^{-1} is the deterministic service time. In particular, we have

$$\theta(0) = \frac{1}{\beta\Gamma(1+\alpha^{-1})}; \quad \rho(0) = \frac{1}{\beta\Gamma(1+\alpha^{-1})\mu}.$$

We are only interested here in cases for which α>1, as these would seem to correspond more closely to more practical situations where jobs' loss rates increase over time if not being served.

We are now ready to extend the service policies described in section 5.1 to the Weibull lifetime setting, following the same approach as Jacobson et al. (2012). For the two index policies, the loss rate is replaced by the updated value $\theta(t)$ to calculate the Whittle's index value for each class at decision epoch t, which are then used in the same manner to derive the service policies. The same approach applies to the two-step and Threshold. For the static policy ThetaMu and TCF, the initial value of $\theta(0)$ is used to determine the service order, which then remains unchanged over time.

The optimal policy can be derived as follows. For deterministic service times, the value iteration procedure needs to only compute V(or V^{π}) at states ($\mathbf{n}$, t) for t-values of the form $t = \sum_{j=1}^{J} m_j \mu_j^{-1}$ where the m_j are non-negative integers. The optimality equation takes the form

$$\begin{aligned} V(\mathbf{n},t) &= 1 + \max_{\substack{1\le j\le J \\ n_j \ge 1}} \left\{ \sum_{\mathbf{n}'} p(\mathbf{n}'|\mathbf{n},t,j,\mu_j^{-1}) V(\mathbf{n}', t+\mu_j^{-1}) \right\}, \\ &\quad \mathbf{n} \neq \mathbf{0}, \\ V(\mathbf{0},t) &= 0, \end{aligned} \tag{27}$$

with the transition probability

$$p(\mathbf{n}'|\mathbf{n},t,j,\mu_j^{-1}) = \prod_{i=1}^{J} \binom{n_i - \delta_{ij}}{n'_i} \left(1 - F_i^t(\mu_j^{-1})\right)^{n'_i} \left(F_i^t(\mu_j^{-1})\right)^{n_i - \delta_{ij} - n'_i},$$

in which $F_j^t(s)$ is given by

$$F_j^t(s) = 1 - \exp\left[\left(\frac{t}{\beta_j}\right)^{\alpha_j} - \left(\frac{t+s}{\beta_j}\right)^{\alpha_j}\right].$$

5.3.1. Similar Level of Opportunity Losses Between Job Classes.

As in the Markovian case, we first of all sample the key problem parameters as follows.

$$\mu_j \sim U[0.1, 1.0] \text{ for all cases} \tag{28a}$$

$$\alpha_j \sim U(1.0, 2.0] \text{ for all cases} \tag{28b}$$

$$\rho_j(0) \sim \begin{cases} U[0.1, 0.5], & \text{Low opportunity loss} \\ U[0.5, 2.0], & \text{Medium opportunity loss} \\ U[2.0, 10.0], & \text{High opportunity loss} \end{cases} \tag{28c}$$

$$L_j \sim \begin{cases} DU[10, 15], & J = 2 \\ DU[5, 10], & J = 3 \\ DU[2, 5], & J = 4 \end{cases} \tag{28d}$$

Note that values of β_j are derived from the values of μ_j and α_j obtained from the draws in Equations (28a) and (28b), and the value of $\rho_j(0)$ obtained from whichever is appropriate of the draws in Equation (28c). As before, we require that

Table 3 Suboptimality (in %) for Problems with Similar Level of Opportunity Loss in the Weibull Lifetime Case (The strongest performances are highlighted in bold)

	J = 2			*J* = 3			*J* = 4		
Opportunity loss	Mean	Max	SD	Mean	Max	SD	Mean	Max	SD
Low									
WI	1.88	7.62	1.60	3.10	11.64	2.37	3.50	17.00	2.97
DWI	1.31	**5.35**	**1.12**	1.61	**5.24**	**1.00**	1.67	**5.45**	**0.97**
Two-step	1.52	9.45	2.22	2.31	12.79	2.88	2.58	12.96	2.59
Threshold	2.17	37.79	4.21	–	–	–	–	–	–
ThetaMu	**0.88**	11.94	1.99	**1.50**	12.37	2.32	**0.96**	6.81	1.25
TCF	7.56	53.79	9.83	8.89	41.20	9.00	6.52	36.06	6.71
SEPT	3.78	19.75	5.21	4.55	21.92	4.95	4.94	19.60	4.63
Medium									
WI	1.73	8.28	1.32	3.85	12.76	2.47	6.77	21.09	4.03
DWI	0.31	2.65	0.45	**0.40**	**2.76**	**0.46**	**0.33**	**1.95**	**0.35**
Two-step	**0.11**	**1.38**	**0.23**	0.41	3.83	0.67	0.43	4.23	0.65
Threshold	2.26	40.24	6.22	–	–	–	–	–	–
ThetaMu	3.81	27.92	5.95	5.02	24.12	5.43	4.98	20.57	4.70
TCF	15.62	54.08	13.38	16.06	47.92	10.65	15.95	50.23	9.64
SEPT	0.81	7.36	1.43	1.89	14.69	2.65	2.35	15.79	2.82
WI	0.05	2.03	0.19	0.26	5.89	0.83	2.64	23.77	4.68
High									
DWI	**0.02**	**1.14**	**0.10**	0.08	**2.28**	**0.28**	0.42	7.61	1.10
Two-step	**0.02**	2.54	0.17	**0.06**	3.89	0.31	**0.25**	**5.12**	**0.79**
Threshold	0.12	6.20	0.55	–	–	–	–	–	–
ThetaMu	4.73	31.87	7.39	8.28	39.97	9.15	7.42	32.09	7.54
TCF	13.60	48.80	12.46	18.47	42.66	11.04	15.64	36.03	8.80
SEPT	0.03	2.29	0.17	0.11	2.60	0.35	0.46	7.61	1.14

$\theta_j(0) < \theta_{j+1}(0)$, $\mu_j > \mu_{j+1}$; instances which do not meet these conditions are resampled. We are forced to impose a lower limit on the maximum number of jobs when J=2 due to the added complexity of the recursion (27) in comparison with Equation (23).

As with the Markovian case, we randomly generate 500 instances for each J and each level of opportunity loss. The comparison results of the suboptimality between the policies are reported in Table 3.

The evidence provided in Table 3 yield similar conclusions as those obtained from Table 1. The policies ThetaMu and two-step still have poor worst-case performance in settings for which they are not designed. The index policy WI is stronger than ThetaMu in medium to high categories, but is always outperformed by two-step. DWI continues to show strong and robust performance across all scenarios. Indeed, it is always one of the top 2 policies on all three measures, with no other heuristic having such consistent performance. Its mean suboptimality is always less than 1.67%, while the values for the others are 6.77% (WI), 2.58%(two-step), 2.26%(Threshold, J=2 only), 8.28%(ThetaMu), 18.47%(TCF) and 4.94%(SEPT). DWI is also the best policy on average for half of the medium to high scenarios, and its worse case performance and standard deviation are the strongest in most scenarios. Unlike in the Markovian case, Threshold does not work particularly well any more in the Weibull case. TCF continues to deliver the weakest performance among static policies, and SEPT still beats ThetaMu in medium to high scenarios.

5.3.2. Different Level of Opportunity Losses between Job Classes.

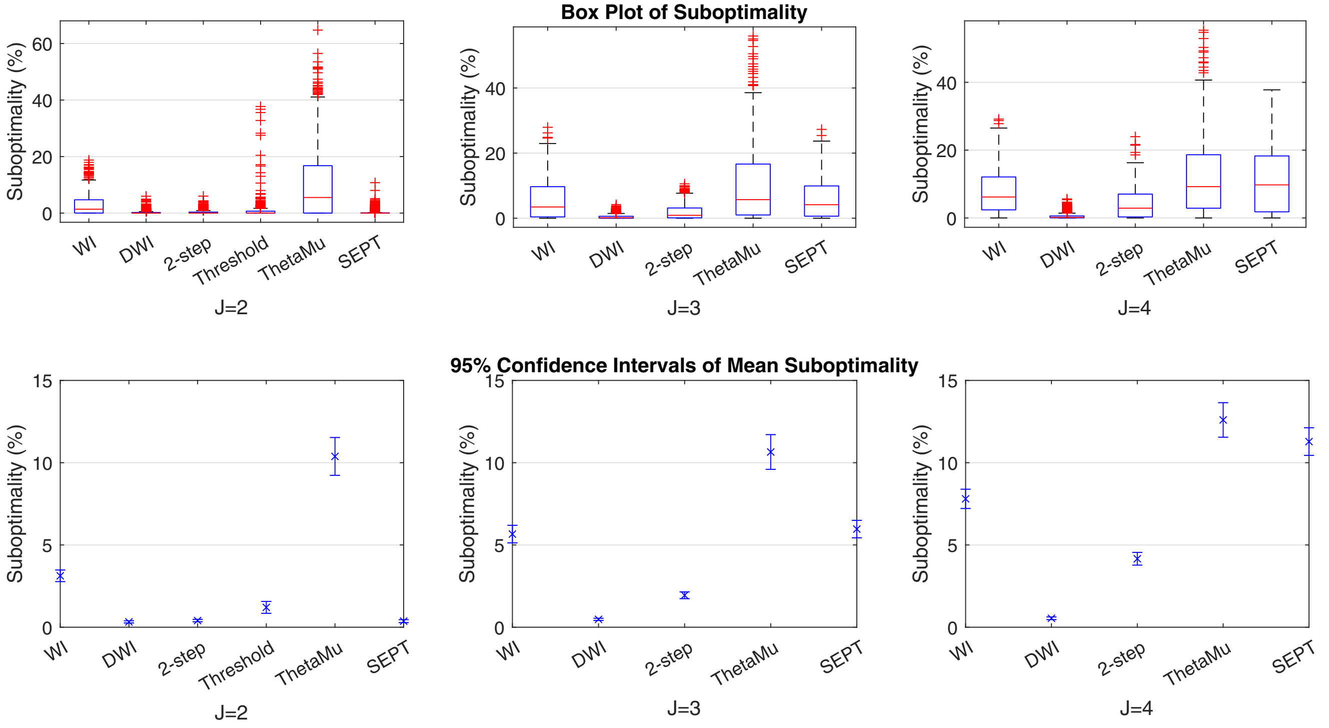
The values of $\rho_j(0)$ are sampled in the same way as Equation (26) in the Markovian case, while all the other parameters are still sampled from Equations (28a–28c). Comparisons between heuristic policies are still based on 500 randomly generated instances. The results are reported in Table 4 and Figure 3. Again TCF is excluded from the plots.

The conclusions are similar to those drawn from the Markovian case. DWI continues to provide uniformly excellent performance. It is significantly stronger than WI and performs consistently with different numbers of job classes. ThetaMu and TCF perform much worse in the Weibull lifetime case than in the Markovian case. In contrast, the performance of SEPT is stronger when J = 2, even though it deteriorates quickly for larger J. The policy two-step delivers respectable performances, but is still significantly worse than DWI.

5.4. Sensitivity Analysis

We further study the performance sensitivity to the distribution of initial jobs among classes. We focus our attention on Weibull lifetime cases for J = 2 with different levels of opportunity loss. The problem parameters for this and the subsequent experiments

Figure 3 Suboptimality (in %) for Problems with Different Level of the Opportunity Loss in the Weibull Case [Color figure can be viewed at wileyonlinelibrary.com]



are sampled as before unless specified otherwise. Therefore we have $\rho_1(0) \leq 1 \leq \rho_2(0)$. The total number of jobs is fixed at 50 and we sample L_1 from three different uniform distributions. Since the optimal policy cannot be obtained for problems of this size, we calculate the percentage improvement of DWI over the other alternatives via simulation. For each scenario, 500 instances are solved and the average improvements are reported in Table 5.

The results are interesting; clearly the relative performances are not the same across the three scenarios. The improvements of DWI over two-step and SEPT reduce when L_2 (the number of jobs with higher opportunity loss) decreases. In contrast, the improvement of DWI over WI increases with smaller L_2, while for the others the biggest improvements are seen for the balanced scenario.

Table 4 Suboptimality (in %) for Problems with Different Level of the Opportunity Loss in the Weibull Case (The strongest performances are highlighted in bold)

	$J = 2$			$J = 3$			$J = 4$		
	Mean	Max	SD	Mean	Max	SD	Mean	Max	SD
WI	3.12	18.79	4.01	5.66	27.94	6.09	7.80	29.17	6.70
DWI	**0.32**	**5.97**	**0.77**	**0.48**	**4.20**	**0.76**	**0.54**	**5.61**	**0.93**
Two-step	0.41	6.00	0.83	1.94	10.57	2.35	4.16	23.97	4.41
Threshold	1.21	37.76	4.12	–	–	–	–	–	–
ThetaMu	10.38	64.79	13.10	10.65	55.96	12.05	12.60	55.37	11.97
TCF	19.67	78.63	20.31	23.13	77.68	19.41	25.48	72.50	17.31
SEPT	0.37	10.78	1.03	5.97	27.28	6.06	11.28	37.79	9.57

Table 5 Average Performance Improvement (in %) of DWI over the Alternatives with Different Initial Settings, $J = 2$, Weibull Lifetime

Initial setting	L_1	L_2	Two-step	WI	Threshold	ThetaMu	TCF	SEPT
Class 2 dominant	DU[5,7]	$50 - L_1$	1.39	1.00	1.48	11.63	41.57	1.80
Balanced	DU[24,26]	$50 - L_1$	−0.23	3.51	1.69	12.67	47.37	−0.31
Class 1 dominant	DU[43,45]	$50 - L_1$	−0.26	5.50	1.38	11.58	44.21	−0.30

Table 6 Average Performance Improvement (in %) of DWI over the Alternatives with Different Initial Settings, $J = 2$, Weibull Lifetime

(a) $L_j \sim [10, 15]$

J	Two-step	WI	ThetaMu	TCF	SEPT
2	0.09	3.08	14.50	37.75	0.06
3	0.45	6.42	15.10	40.44	2.24
4	0.78	8.70	18.12	45.65	3.99

(b) $J = 4$

L_j	Two-step	WI	ThetaMu	TCF	SEPT
$\sim DU[2, 5]$	4.02	8.70	16.29	43.31	13.56
$\sim DU[10, 15]$	0.78	8.70	18.12	45.65	3.99
$\sim DU[20, 25]$	0.11	8.48	18.77	45.60	1.63

We also study the performance with different values of J (with $L_j \sim [10, 15], \forall j$) and L_j (with $J = 4$) for Weibull lifetime scenarios with different levels of opportunity loss. For each scenario 500 problems are generated as before. The average improvements are reported in Table 6.

Table 6(a) reports the results with increasing number of classes, and Table 6(b) with increasing number of initial jobs. It is shown that the improvements of DWI over the others are clearly bigger with more job classes. The improvements, however, do not necessarily increase with larger initial number of jobs in each class. Indeed, the improvements over two-step and SEPT reduce with more jobs in each class, while the improvements do not change much for the other policies. It is worth mentioning that in these scenarios jobs are evenly distributed into different classes. The performance will be very different if one or few job classes dominate the others, as shown in Table 5. In particular, the DWI's performance is expected to be much stronger when the higher opportunity loss classes make up most of the jobs. Therefore, more jobs do not always result in better (or worse) relative performance between alternative policies. The number of job classes, the initial number of jobs in each class, and the distribution of jobs between classes all make a difference.

At last, we undertake an experiment to study the performance when the same number of jobs are composed of more and more classes (J = 2, 4, 10, 20). Jobs are evenly distributed between classes. We sample ρ (0) values as follows. For each J, the interval [0.1, 1.0] is divided into $J/2$ consecutive subintervals with equal length, and so is the interval [1.0, 10.0]. The $\rho_j(0)$ value is then sampled uniformly from the jth subinterval. Again 500 instances are solved for each J and the results are shown in Table 7.

It is shown that the advantage of DWI over two-step clearly increases when jobs are composed of more classes. The average improvement is only 0.21% for J=2, but it increases to 9.58% when J becomes 20. Similar results are observed for the other policies.

Table 7 Average Performance Improvement (in %) of DWI over the Alternatives with the Same Number of Initial Jobs But Different Number of Classes, Weibull Lifetime

J	L_j	Two-step	WI	ThetaMu	TCF	SEPT
2	10	0.21	2.93	14.74	35.95	0.34
4	5	2.88	8.35	16.87	44.68	11.11
10	2	7.76	10.29	18.73	43.79	15.51
20	1	9.58	44.28	23.01	44.97	16.32

5.5. Further Discussions

The numerical results show the clear and significant enhancement of the index policy derived from the second decomposition over that from the first one. Not only is the policy DWI much stronger on average than WI, but also its performance is consistent and robust across all opportunity loss scenarios and number of job classes. These results highlight the limitation of the classical Whittle's index policies in situations where the number of bandits competing for resource varies over time. (We use bandits and classes interchangeably in this discussion.) In a standard Whittle's RB approach, the index values are obtained solely based on the dynamics of each bandit themselves, while all the information outside is ignored. Such locally focused approaches might still work well when the number of bandits stay the same throughout the time horizon, as shown by previous works in the RB literature. However, as our results have indicated, they are not good enough for problems with a changing number of bandits. In such cases, it becomes essential to take into account other information apart from each single bandit itself, such as the number of bandits still competing for service as we have suggested. Indeed, if there was just one bandit left, there would be no need to have the penalty for receiving service. On the other hand, if there were many others competing for service, a high penalty would need to be imposed to reflect the severity of resource scarcity.

The approach that we have developed provides a simple yet effective way to include the number of job classes when deriving the class specific index values. This is captured by a penalty term for receiving service, which is proportional to the total number of job classes. More attractively, such information is available at no extra cost, as it can be trivially obtained from the system state. This approach can be readily applied to other restless bandit problems where the number of bandits varies over time (not necessarily decreasing), such as situations where new/existing projects are being started/terminated or customer queues are dynamically opened/closed. We strongly believe that it has the potential to produce much stronger index policies compared to those from the classical RB approaches for such problems.

6. When to Switch

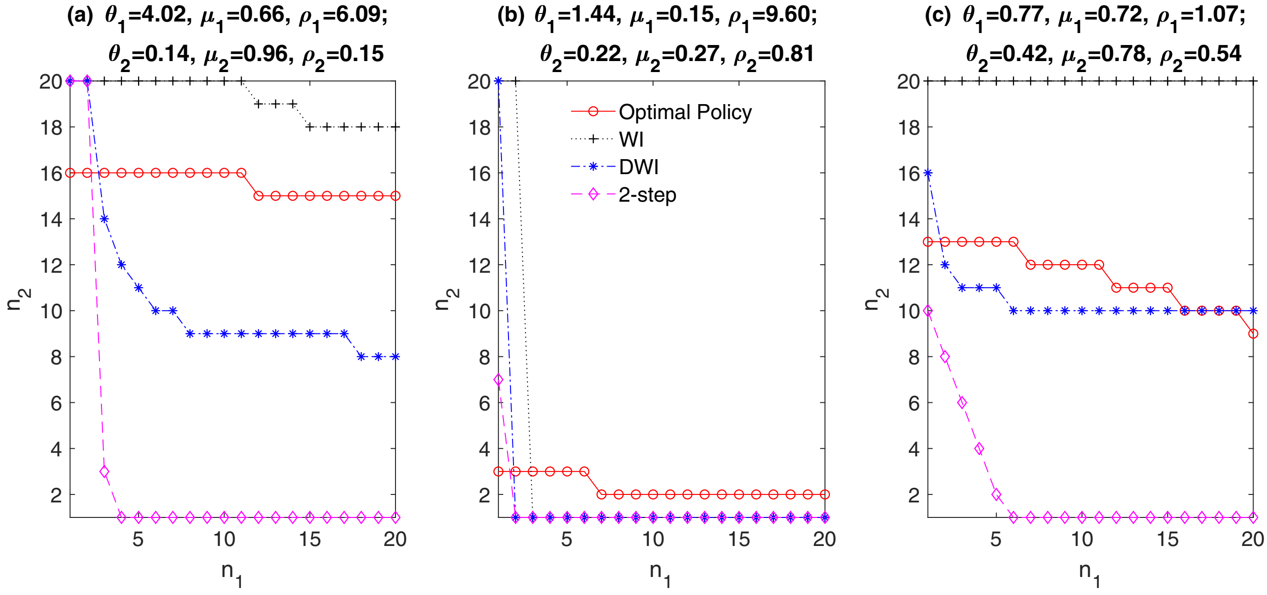
Figure 1 in section 4.3 suggests that the three alternative heuristic policies, that is, two-step, WI, and DWI have very different switching patterns. In this section, we explore further the switching time for each of them and how this compares to the optimal policy. Since DWI works particularly well in the scenarios where job classes have mixed levels of opportunity losses (as shown in Table 2), we focus our attention first on such problems. A number of problem instances that were generated in section 5.2 for the Markovian case and $J = 2$ have been studied. Figure 4 shows the typical switching curves identified in these problems. Once again, the area above/below each curve corresponds to states in which class 2/1 jobs are prioritized.

In most scenarios, the four policies start with serving class 2 jobs due to their shorter service times. With jobs leaving the system (either via service completion or abandonment), the service is gradually switched over to class 1 jobs. In all three examples, the two-step policy is the last to switch while WI always switches first; DWI sits in the middle in all scenarios. (The only exception to this is scenario (c), in which WI always prioritizes class 1 jobs. However, this can simply be viewed as a situation in which the switch from class 2 to class 1 for WI occurs at a larger number of jobs than is plotted here.)

Such switching patterns can be explained by the index values of each policy. According to Equation (21), the index values for two-step are largely determined by the service rate at the beginning (note that the total number of remaining jobs is large and thus the second term in the denominator is similar between classes). With jobs leaving the system the index values for both classes increase. Even though the loss rate begins to make more effect, the catching up takes time and thus the switching points always come quite late, usually when only few jobs still remaining. For DWI, in sharp contrast, the index value for class 2 decreases while that for class 1 increases with smaller n_j (see Corollary 2). Thanks to such a property, the switching takes place much earlier than two-step. Figure 5 plots the index values for both two-step and DWI policies for the illustrative example in section 4.3, clearly demonstrating the difference between the policies. For WI, however, because its index values for both classes are larger than their counterparts in DWI (as illustrated in Figure 1a), the switching points are pulled even earlier and in most cases too early, as shown in Figures 1b and 4.

As a result of such switching patterns, DWI is the closest to the optimal policy in most cases where job classes have different levels of opportunity losses, as illustrated in scenario (a) and (c) in Figure 4. In scenario (b), the loss rate of class 1 is not large enough to compensate the much smaller service rate and justify an earlier switch, and thus two-step works better than DWI with a marginal advantage; indeed, they only differ in states when $n_1 = 1$. But in cases where the

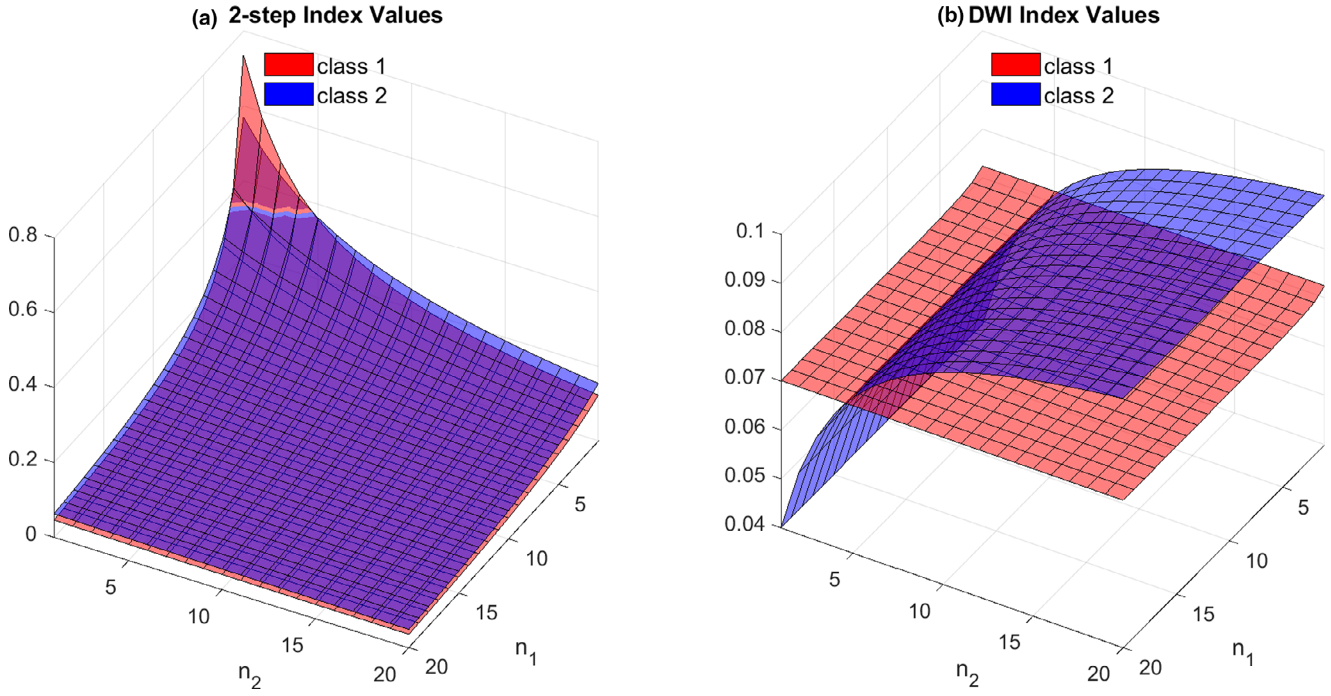
Figure 4 Heuristic Policies Two-Step, WI, DWI and An Optimal Policy for Problems with Different Values of ρ [Color figure can be viewed at wileyonlinelibrary.com]


(a) θ_1=4.02, μ_1=0.66, ρ_1=6.09; θ_2=0.14, μ_2=0.96, ρ_2=0.15

(b) θ_1=1.44, μ_1=0.15, ρ_1=9.60; θ_2=0.22, μ_2=0.27, ρ_2=0.81

(c) θ_1=0.77, μ_1=0.72, ρ_1=1.07; θ_2=0.42, μ_2=0.78, ρ_2=0.54

Figure 5 Index Values for Two-Step and DWI for the Illustrative Example [Color figure can be viewed at wileyonlinelibrary.com]



loss rate of class 1 is obviously larger and/or the service rate is closer to class 2, and thus should be prioritized much earlier in the process, two-step still switches very late. This behavior explains why the worst-case performance could be very poor for two-step. It is also not surprising why DWI is the most robust (and the strongest) policy across all scenarios. Furthermore, the advantage margin of DWI increases with the number of job classes J, as shown in the results in section 5.

The same switching order between the three policies is also observed in both low opportunity loss ($\rho_j \leq 0.5$ for all j) and high opportunity loss ($\rho_j \geq 2$ for all j) problems. In either case the average performance of DWI is not necessarily the strongest. Indeed, due to the relatively smaller difference of loss rates, it is often optimal to switch late (as explained in the previous paragraph), or “so late” that no switching actually takes place. We also found that in some of the low opportunity loss problems DWI and two-step start with the “wrong” class, as it is actually optimal to clear the faster leaving class before switching to the other class that has a much smaller loss rate (and thus barely any abandonment). In such cases the same switching order still holds, and WI delivers the best results as it is the first one to switch to the “right” class (even though not as early as ThetaMu). Therefore ThetaMu and two-step often have the best average performance in the low and high opportunity loss scenarios, respectively. Nevertheless, being able to switch at “right” timing (i.e., neither too early nor too late) warrants DWI still the most robust policy in these problems.

7. Conclusions

In the aftermath of mass casualty incidents, a critical decision problem is how to optimally allocate limited resources to a large number of jobs with different and uncertain lifetimes and service times. In this study, we have addressed such an important and ongoing debate and proposed an effective yet simple service policy, based on the celebrated Whittle’s index policies for restless bandits. Unlike most of the literature which develops indices based on a time average reward/cost criterion with an infinite horizon, the problem concerned features a finite and uncertain time horizon. Moreover, the decision epochs are completely determined by the endogenous actions defined by the service policy, which differs significantly from previous works where the decision epochs are either discrete or largely exogenously determined. Furthermore, in our setting, the number of bandits (job classes) diminishes over time. We have proved the indexability of all job classes in the Markovian case, and developed closed-form equations to compute the corresponding index values. Our numerical experiments demonstrate that our proposal has stronger and more consistent performance over preexisting heuristics in the literature, even in the non-Markovian settings where jobs’ lifetimes follow Weibull distributions and service times are deterministic.

From the methodological point of view, our work extends Whittle's index policies to problems with a finite and uncertain time horizon which depends upon the service policy, and in which the number of bandits changes over time. Such problems cannot be decomposed by a straightforward application of the classical Whittle's Lagrangian relaxation approach. To accommodate the complexity we have developed two versions of Lagrangian relaxation that allow the decomposition of the original problem into single class problems. The first one is a direct extension of the standard Whittle's restless bandits approach. In the second one the total number of job classes still competing for service is taken into account, and this approach reduces to the first decomposition when assuming the total number of job classes to be always one. The second decomposition leads to a strong performing (and consistent) index policy, compared to that from the first. We show that the relatively simple act of taking into consideration a piece of system-wide information that is readily available can make a substantial improvement to performance. This intuitive and effective idea is not restricted to the problem concerned, and should be straightforward to apply to other restless bandit problems where the number of bandits varies over time.

Both approaches proposed in this study entail index policies calculated in two different forms depending on the level of opportunity loss (i.e., the value of ρ). As mentioned above, when $\rho \geq 1$ (respectively, $\rho \leq 1$), the index values decrease (respectively, increase) with the number of remaining jobs n. This property derived from the application of Whittle's Restless Bandits helps to explain the different levels of performance of our policies among scenarios featuring "heavy premature job loss" versus "no premature job loss". It also explains why the proposed index policy performs particularly well where competing job classes have different levels of opportunity losses. In sum, what makes our proposed DWI stand out is the way that the index values are derived: 1) it differentiates at the level of opportunity loss; 2) it incorporates other system-wide information (the number of other classes competing for service).

For emergency response researchers and practitioners, our study generates a number of useful insights. First, prioritization decisions solely dependent on urgency of treatment (i.e., length of remaining lifetimes) may not work well in the aftermath of mass casualty events due to the severity of resource scarcity. The level of resources required to complete the service (i.e., the service time) of each job should also be given sufficient weight in determining priority level and resource scheduling decisions. It may lead to more survivals to give treatment/priority to less urgent casualties with lower resource requirement. Second, common practices (such as START), which direct resources to the next casualty category only if the preceding categories are completely emptied, fail to be efficient. We demonstrate that the optimal "switch" between casualty categories is state-dependent. Therefore, the choice of an effective state-dependent scheduling policy is important for the overall success of emergency response effort. Our numerical study shows that our proposed policy performs better and more consistently, across a variety of parameter combinations, than a number of well-established benchmarks in the field. Last but not least, by having other-regarding nature embedded in our index policy DWI, we attempt to provide a response to the call (Jacobson et al. 2012 and many cited within) that prioritization decisions shall take explicit account of resource scarcity constraint in the modeling of emergency resource allocation policies in the aftermath of mass casualty incidents.

In this work, we have restricted attention to the situation in which all jobs are to be served by a single server; it would clearly be desirable to understand how well our index policy works when this assumption is relaxed. Interesting directions for future work include tackling the question of indexability, and a practical investigation into the performance of the proposed index policy, in more general scenarios where the number of bandits changes over time.

Acknowledgment

The authors thank the two anonymous referees and the senior editor for their helpful comments and insights that have improved the presentation and exposition of this work.

References

Archibald, T. W., D. P. Black, K. D. Glazebrook. 2009. Indexability and index heuristics for a simple class of inventory routing problems. *Oper. Res.* **57**(2): 314–326.

Argon, N. T., S. Ziya, R. Righter. 2008. Scheduling impatient jobs in a clearing system with insights on patients triage in mass casualty incidents. *Probab. Eng. Inform. Sci.* **22**(3): 301–332.

Argon, N. T., L. Ding, K. D. Glazebrook, S. Ziya. 2009. Dynamic routing of customers with general delay costs in a multiserver queuing system. *Probab. Eng. Inform. Sci.* **23**(2): 175–203.

Bertsimas, D., J. Niño-Mora. 1996. Conservation laws, extended polymatroids and multiarmed bandit problems; a polyhedral approach to indexable systems. *Math. Oper. Res.* **21** (2): 257–306.

Caro, F., J. Gallien. 2007. Dynamic assortment with demand learning for seasonal consumer goods. *Management Sci.* **53**(2): 276–292.

Ding, L., K. D. Glazebrook. 2013. Dynamic routing in distinguishable parallel queues: an application of product returns for remanufacturing. *OR Spectrum* **35**(3): 585–608.

Gittins, J. C. 1979. Bandit processes and dynamic allocation indices (with discussion). *J. Roy. Stat. Soc.* **41** (2): 148–177.

Glazebrook, K. D., P. S. Ansell, R. T. Dunn, R. R. Lumley. 2004. On the optimal allocation of service to impatient tasks. *J. Appl. Probab.* **41** (1): 51–72.

Glazebrook, K. D., H. M. Mitchell, P. S. Ansell. 2005. Index policies for the maintenance of a collection of machines by a set of repairmen. *Eur. J. Oper. Res.* **165** (1): 267–284.

Glazebrook, K. D., C. Kirkbride, D. Ruiz-Hernandez. 2006. Spinning plates and squad systems: Policies for bi-directional restless bandits. *Adv. Appl. Probab.* **38** (1): 95–115.

Glazebrook, K. D., D. J. Hodge, C. Kirkbride. 2011. General notions of indexability for queueing control and asset management. *Ann. Appl. Probab.* **21** (3): 876–907.

Glazebrook, K. D., D. J. Hodge, C. Kirkbride, R. Minty. 2014. Stochastic scheduling: A short history of index policies and new approaches to index generation for dynamic resource allocation. *J. Sched.* **17** (5): 407–425.

Graczová, D., P. Jacko. 2014. Generalized restless bandits and the knapsack problem for perishable inventories. *Oper. Res.* **62** (3): 696–711.

Green, L. V., P. J. Kolesar. 2004. Anniversary article, Improving emergency responsiveness with management science. *Management Sci.* **50** (8): 1001–1014.

Jacko, P., B. Sansò. 2012. Optimal anticipative congestion control of flows with time-varying input stream. *Perform. Eval.* **69** (2): 86–101.

Jacobson, E. U., N. T. Argon, S. Ziya. 2012. Priority assignment in emergency response. *Oper. Res.* **60** (4): 813–832.

Li, D., K. D. Glazebrook. 2010. An approximate dynamic programing approach to the development of heuristics for the scheduling of impatient jobs in a clearing system. *Nav. Res. Log.* **57** (3): 225–236.

Niño-Mora, J. 2001. Restless bandits, partial conservation laws and indexability. *Adv. Appl. Probab.* **33** (1): 76–98.

Nocera, A., A. Garner. 1999. An australian mass casualty incident triage system for the future based upon triage mistakes of the past: the homebush triage standard. *ANZ J. Surg.* **69** (8): 603–608.

Papadimitriou, C. H., J. N. Tsitsiklis. 1999. The complexity of optimal queuing network control. *Math. Oper. Res.* **24** (2): 293–305.

Sacco, W. J., D. M. Navin, R. K. Waddell, K. E. Fiedler, W. B. Long, R. F. Buckman, Jr. 2007. A new resource-constrained triage method applied to victims of penetrating injury. *J. Trauma Acute Care Surg.* **63** (2): 316–325.

Sun, Z., N. T. Argon, S. Ziya. 2017. Patient triage and prioritization under austere conditions. *Management Sci.* **64** (10): 4471–4489.

Weber, R. R., G. Weiss. 1991. Addendum to "On an index policy for restless bandits". *Adv. Appl. Probab.* **23** (2): 429–430.

Whittle, P. 1988. Restless bandits: Activity allocation in a changing world. *J. Appl. Probab.* **25** (A): 287–298.

Whittle, P. 1996. *Optimal Control: Basics and Beyond*. Wiley, New York.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Appendix A: Mathematical Proofs.