

An Approximate Dynamic Programming Approach to the Development of Heuristics for the Scheduling of Impatient Jobs in a Clearing System

Dong Li, Kevin D. Glazebrook

Department of Management Science, Lancaster University, Lancaster LA1 4YX, United Kingdom

Received 21 October 2008; revised 25 August 2009; accepted 22 November 2009

DOI 10.1002/nav.20395

Published online 12 January 2010 in Wiley InterScience (www.interscience.wiley.com).

Abstract: A single server is faced with a collection of jobs of varying duration and urgency. Each job has a random lifetime during which it is available for nonpreemptive service. Should a job's lifetime expire before its service begins then it is lost from the system unserved. The goal is to schedule the jobs for service to maximize the expected number served to completion. Two heuristics have been proposed in the literature. One (labeled π^S) operates a static priority among the job classes and works well in a "no premature job loss" limit, whereas the second (π^M) is a myopic heuristic which works well when lifetimes are short. Both can exhibit poor performance for problems at some distance from the regimes for which they were designed. We develop a robustly good heuristic by an approximative approach to the application of a policy improvement step to the asymptotically optimal heuristic π^S , in which we use a fluid model to obtain an approximation for the value function of π^S . The performance of the proposed heuristic is investigated in an extensive numerical study. © 2010 Wiley Periodicals, Inc. *Naval Research Logistics* 57: 225–236, 2010

Keywords: approximate dynamic programming, fluid models; heuristic policies; impatient jobs; stochastic scheduling

1. INTRODUCTION

The article considers a scenario in which a collection of jobs is seeking service which is provided by a single server. There are two major sources of uncertainty related to each job. First, its service is of uncertain duration. Second, the jobs *lifetime*, namely, the period of time during which it is available for service, is also uncertain. For most of the article we shall assume that a job abandons the system unserved if its service does not begin before the expiration of its lifetime. Furthermore, all jobs whose service begins are guaranteed to be served to completion. No preemptions are allowed. Each job is subject to *triage* (an assessment of its service requirement and its urgency) 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 (denoted $X_j \sim F_j$ for class j) and i.i.d. service times (denoted $Y_j \sim G_j$ for class j). Following triage, the central challenge discussed by the article concerns how the jobs should be scheduled for service such that the expected number of jobs served to completion is maximized. Intuitively, it seems clear that priority should be given to jobs with small service times (since these delays others less) and/or small lifetimes (since these are most urgent).

Precisely how this should be done to achieve optimality is far from clear, however.

Gaver and Jacobs [8] have argued the importance of incorporating job/customer impatience into service system models. They cite call center where customers will hang up if they are required to wait excessively for service and military scenarios in which enemy targets may move out of reach if not dealt with promptly. There is now an extensive literature tailored to call center applications. See, for example, Garnett et al. [7], Bassamboo et al. [3], and a recent special issue of *Management Science* (Kooze [15]). Contributions related to military applications include those of Gaver et al. [9] and Glazebrook and Punton [12]. Furthermore, Glazebrook et al. [13] develop heuristic policies for the admission control and routing of impatient customers seeking service. These studies concern situations in which a stream of new jobs arrive for service over time. The clearing system model alluded to in the opening paragraph and which is the focus of the article posits an amount (possibly large) of urgent work, arising perhaps as the result of a natural or man-made disaster, which is present *ab initio* and which must be accomplished with limited resources. Direct precursors of this article include the studies of Glazebrook et al. [11] and Argon et al. [2]. The latter emphasizes applications concerning the use of limited medical resources after mass casualty incidents. Before proceeding further, we pause to remark that

Correspondence to: D. Li (d.li@lancaster.ac.uk)

the situation envisaged in the article should be clearly distinguished from cases where jobs are to be scheduled in the face of known hard deadlines. For important examples of the latter, see Glazebrook [10], Jiang et al. [14], Doytchinov et al. [6], and Van Mieghem [24].

In principle, the problems described in the opening paragraph are capable of solution by the methods of stochastic dynamic programming (DP) (Puterman [20] and Tijms [23]). In practice, the scale of the computational effort required means that this is a realistic option only for small problems and special cases. Argon et al. [2] have shown that when lifetimes and service times are suitably agreeable (jobs with the shortest lifetimes also have the shortest service times) then the optimal policy always gives highest priority to the time-critical job regardless of the system state. Such special results notwithstanding, the central challenge concerns the development and evaluation of strongly performing *heuristic policies*. The literature contains discussions of two candidate policy classes. Glazebrook et al. [11] develop a simple static policy (hereafter denoted π^S) which operates a fixed priority among the job classes. The class with the smallest associated value of $E(X_j)E(Y_j)$ is accorded the highest priority by π^S (jobs scheduled first) whereas that with the largest associated value has lowest priority (jobs scheduled last). This simple, intuitive policy is shown to be asymptotically optimal for problems with exponentially distributed lifetimes in a “no premature job loss” limit ($\min_j E(X_j) \rightarrow \infty$). In contrast, Argon et al. [2] develop a myopic “triangular” heuristic which performs well in a “heavy premature job loss” limit ($\max_j E(X_j) \rightarrow 0$). Although these policies perform satisfactorily in the neighborhood of the regimes for which they were designed, they are nonrobust and can exhibit poor performance more generally. We exploit the simplicity (especially the static nature) of the Glazebrook et al. [11] heuristic π^S to develop a class of heuristic policies with robustly strong performance via a two-stage procedure. At Stage 1, we use a fluid model to approximate the value function of the system operating under π^S . We then adopt an approximate DP (Powell [19]) approach and design in Stage 2 a dynamic heuristic by using the approximating value function from Stage 1 in a DP recursion. Expressed differently, motivated by computational considerations, we introduce an approximating fluid model to render feasible the task of applying a single policy improvement step to the asymptotically optimal heuristic π^S . The idea of applying a single policy improvement step to some given heuristic policy is not new. See Ott and Krishnan [18], Tijmas [23], Anseli et al. [1], Bhulai and Koole [5], and Opp et al. [17] for a few examples. All these contributions utilize the value function of the initial policy. This, however, may be very difficult to obtain. A variety of approximation schemes have been proposed to overcome this difficulty. See, for example, Marback et al. [16], Bhulai [4], Powell [19], and Roubos and Bhulai [21, 22]. A primary contribution of the

article is the idea of developing a good approximation to the value function of an initial policy by use of an approximating fluid model.

The article proceeds as follows: our clearing model for the scheduling of impatient jobs is presented in Section 2. In Section 3 we describe an approach to the development of heuristic policies via an approximating fluid model. Our heuristics are subject to extensive numerical investigation in Section 4 where they are compared with earlier proposals in the literature and (where possible) to optimal. They are found to perform extremely well, comfortably outperforming competitors.

2. THE MODEL

A clearing system has a single server and a collection of impatient jobs (or customers) awaiting service which begins at time zero. Each job belongs to one of J classes. We use the pair jk to denote the job which is the k th member of class j , $1 \leq k \leq L_j, 1 \leq j \leq J$. Observe that L_j is the number of class j jobs present at time 0. Associated with each job jk are two positive valued random variables, namely, its *lifetime* X_{jk} and its *service time* Y_{jk} . For each j the collection $\{X_{j1}, X_{j2}, \dots, X_{jL_j}\}$ of class j lifetimes are independent and identically distributed (i.i.d.), having the same distribution as X_j whose distribution function is F_j . Similarly, the collection $\{Y_{j1}, Y_{j2}, \dots, Y_{jL_j}\}$ of class j service times are independent and identically distributed, having the same distribution as Y_j whose distribution function is G_j . All lifetimes and service times have finite expectation and are independent of each other.

The single server processes individual jobs nonpreemptively. Job jk will abandon the system unserved if its service has not begun before X_{jk} . However, once a job has begun service, it will be served through to completion. Let π denote a service policy (a nonanticipative rule for allocating the server to waiting jobs) and $T_{jk}(\pi)$ the random time at which policy π begins to process job jk . If jk is not served by π then we write $T_{jk}(\pi) = \infty$. The number of jobs served to completion under π is denoted $N(\pi)$ and is given by

$$N(\pi) = \sum_{j=1}^J \sum_{k=1}^{L_j} \mathbb{I}\{T_{jk}(\pi) \leq X_{jk}\}. \quad (1)$$

In (1), \mathbb{I} is an indicator. The goal of analysis is the determination of a policy π to maximize $E\{N(\pi)\}$. Argon et al. [2] argue that under the optimal policy the server will never idle, whereas the theory of stochastic dynamic programming (see, for example, Puterman [20]) guarantees the existence of an optimal policy which takes actions which depend only upon the current system state.

We model this problem as a semi-Markov decision process which is as follows:

1. Decision epochs are at time zero and at all service completion times. The *state of the process* at decision epoch $t \geq 0$ is denoted $\{n_j(t), 1 \leq j \leq J; t\} \equiv \{\mathbf{n}(t), t\}$, where $n_j(t)$ is the number of class j jobs which at time t have not yet been served and have not abandoned the system. Generic states of the system are denoted (\mathbf{n}, t) , (\mathbf{n}', s) ;
2. At each decision epoch, one of the jobs remaining in the system is chosen for processing. In any system state (\mathbf{n}, t) , the collection of admissible actions is written $A(\mathbf{n})$ and is given by

$$A(\mathbf{n}) = \{j; n_j \geq 1, 1 \leq j \leq J\}. \quad (2)$$

In (2) the action j is identified with the class of the job chosen for processing.

3. Let t be a decision epoch and (\mathbf{n}, t) the system state then. If action $j \in A(\mathbf{n})$ is taken and results in a service time (realized value of Y_j) equal to s then the system at the next decision epoch $t + s$ will be $(\mathbf{n}', t + s)$ with probability $p(\mathbf{n}'|\mathbf{n}, t, j, s)$ given by

$$\begin{aligned} p(\mathbf{n}'|\mathbf{n}, t, j, s) &= \prod_{i=1}^J \binom{n_i - \delta_{ij}}{n'_i} \{P[X_i \geq t + s | X_i > t]\}^{n'_i} \{P[X_i(t + s | X_i)t]\}^{n_i - \delta_{ij} - n'_i} \\ &= \prod_{i=1}^J \binom{n_i - \delta_{ij}}{n'_i} \left\{ \frac{1 - F_i(t + s)}{1 - F_i(t)} \right\}^{n'_i} \\ &\quad \times \left\{ \frac{F_i(t + s) - F_i(t)}{1 - F_i(t)} \right\}^{n_i - \delta_{ij} - n'_i}, \\ &0 \leq n'_j \leq n_j - \delta_{ij}, 1 \leq i \leq J. \end{aligned} \quad (3)$$

In (3), δ_{ij} is the Kronecker delta which is equal to one when $i = j$ and is otherwise zero.

4. A *policy* π is any nonanticipative rule for choosing admissible actions. Our goal is the determination of a policy to maximize the expected number of jobs served from initial state $(\mathbf{L}, 0)$.

In principle (an ϵ -approximation to) an optimal policy could be developed with the tools of stochastic dynamic programming (DP). Write

$$\Omega = \{(\mathbf{n}, t); 0 \leq n_j \leq L_j, 1 \leq j \leq J, t \in \mathbb{R}^+\}$$

for the system's state space and develop the value function $V : \Omega \rightarrow [0, \sum_{j=1}^J L_j]$, where $V(\mathbf{n}, t)$ is the maximal

expected number of service completions from state (\mathbf{n}, t) . Assuming sufficient regularity, V satisfies the optimality equations

$$\begin{aligned} V(\mathbf{n}, t) &= 1 + \\ &\max_{j \in A(\mathbf{n})} \left\{ \int_0^\infty \sum_{\mathbf{n}'} p(\mathbf{n}'|\mathbf{n}, t, j, s) V(\mathbf{n}', t + s) dG_j(s) \right\}, \mathbf{n} \neq \mathbf{0}, \\ V(\mathbf{0}, t) &= 0. \end{aligned} \quad (4)$$

Computational approaches to the determination of optimal policies built around the recursive scheme in (4) are not practical for problems of realistic size. We seek to develop *heuristic* approaches which are close to reward maximizing.

Two heuristics have been proposed in the literature and both will play a role in our narrative. Glazebrook et al. [11] proposed a *static priority rule* π^S . Suppose that the job classes are numbered in *increasing order* of the quantity $E(X_j)E(Y_j)$, i.e., such that

$$E(X_1)E(Y_1) \leq E(X_2)E(Y_2) \leq \dots \leq E(X_J)E(Y_J). \quad (5)$$

In any state (\mathbf{n}, t) , π^S chooses action $\pi^S(\mathbf{n}, t)$ where

$$\pi^S(\mathbf{n}, t) = \min\{j; n_j \geq 1\}.$$

Hence π^S implements a class priority according to the ordering $1 \rightarrow 2 \rightarrow \dots \rightarrow J$. It favors jobs with small mean service times and/or small mean lifetimes and was shown by Glazebrook et al. [11] to be optimal in a "no premature job loss" limit when job lifetimes are exponentially distributed.

Argon et al. [2] propose a myopic heuristic policy π^M which takes the following form: in state (\mathbf{n}, t) , π^M chooses action $\pi^M(\mathbf{n}, t)$ to be the nonempty class j with smallest associated value of

$$E(Y_j) \left[\sum_{i=1}^J (n_i - \delta_{ij}) \{E(X_i - t | X_i > t)\}^{-1} \right]. \quad (6)$$

As we shall see, both heuristics π^S and π^M perform well on occasion, but exhibit a lack of robustness in performance. Namely, there are problems for which they do not work at all well. In the next section, we propose an approach to the development of a heuristic policy with associated performance which is robust and stronger than either.

REMARK: Minor adjustments to the above are required for simple variants of our model, such as (a) incorporating the possibility of loss during service and/or (b) different returns earned upon completion of services of jobs from different

classes. For example, scenario (a) requires an adjustment of the optimality equations in (4) to

$$\begin{aligned} \bar{V}(\mathbf{n}, t) &= \max_{j \in A(\mathbf{n})} \left\{ P(X_j > t + Y_j | X_j > t) \right. \\ &+ \left. \int_0^\infty \sum_{\mathbf{n}'} p(\mathbf{n}' | \mathbf{n}, t, j, s) \bar{V}(\mathbf{n}', t + s) dG_j(s) \right\}, \mathbf{n} \neq \mathbf{0}, \\ \bar{V}(\mathbf{0}, t) &= 0. \end{aligned} \tag{7}$$

In (7) we make an assumption that service times are delivered in full even when premature loss occurs. There are other modeling possibilities. It is evidently the case that $\bar{V}(\mathbf{n}, t) \leq V(\mathbf{n}, t)$ for all choices of \mathbf{n}, t .

3. HEURISTIC POLICY DEVELOPMENT- APPROXIMATE POLICY IMPROVEMENT FROM π^S VIA FLUID MODELS

Of the heuristic policies described at the conclusion of the proceeding section, π^S enjoys the benefits of a very simple structure. It seems reasonable to explore the possibility of designing effective dynamic heuristics for our problem by strengthening the performance of this *static* policy via the implementation of a single DP policy improvement step. Write $V_{\pi^S} : \Omega \rightarrow [0, \sum_{j=1}^J L_j]$ for the value function for π^S , namely, $V_{\pi^S}(\mathbf{n}, t)$ is the expected number of service completions from state (\mathbf{n}, t) under policy π^S . The function V_{π^S} satisfies the recursion

$$\begin{aligned} V_{\pi^S}(\mathbf{n}, t) &= 1 + \int_0^\infty \sum_{\mathbf{n}'} p(\mathbf{n}' | \mathbf{n}, t, \pi^S(\mathbf{n}, t), s) \\ &\times V_{\pi^S}(\mathbf{n}', t + s) dG_{\pi^S(\mathbf{n}, t)}(s), \mathbf{n} \neq \mathbf{0}, \\ V_{\pi^S}(\mathbf{0}, t) &= 0. \end{aligned} \tag{8}$$

A single DP policy improvement step applied to π^S will result in a new dynamic policy π^{SPI} determined as follows:

$$\begin{aligned} \pi^{SPI}(\mathbf{n}, t) &= \arg \max_j \left\{ \int_0^\infty \sum_{\mathbf{n}'} p(\mathbf{n}' | \mathbf{n}, t, j, s) V_{\pi^S}(\mathbf{n}', t + s) dG_j(s) \right\} \end{aligned} \tag{9}$$

with the argmax in (9) being taken over the admissible set $A(\mathbf{n})$. In words, policy π^{SPI} makes optimal decisions under an assumption that all future decisions are made according to π^S .

Our experience is that policy π^{SPI} performs very strongly when it is available. In support of this claim we make reference to 2000 randomly generated problems in which $J = 2$

and both lifetimes and service times are exponentially distributed. The performance of four heuristics, which includes π^S and π^{SPI} , when applied to these problems is given in Table 1, Section 4, together with the details of the problems themselves. Similar results for 2000 randomly generated problems with $J = 5$ and exponentially distributed lifetimes and service times are given in Table 2. In both tables, the results are presented in four groups (500 problems in each group) labeled *A, B, C* and *D* according to the *relative* lengths of lifetimes and service times in the generated problems. In Table 1, the worst performances of heuristic π^S within each group, as measured by the percentage deviation from optimum, are 23.79% (*A*), 24.33% (*B*), 16.52% (*C*), and 5.17% (*D*). Once a policy improvement step is applied to π^S as in (9) above, the corresponding worst case percentages for π^{SPI} are 0% for all four groups. In Table 2, the worst case percentages for π^S are 15.95% (*A*), 11.86% (*B*), 3.87% (*C*), and 0.33% (*D*) whereas those for π^{SPI} are 0.48% (*A*), 0.88% (*B*), 1.04% (*C*), and 0.05% (*D*). In all groups and for both tables the median percentage suboptimality for π^{SPI} was 0%. Furthermore, for nonexponential cases, results for 2000 randomly generated problems in which $J = 2$, lifetimes are Weibull distributed and service times are deterministic are given in Table 3. This may be found in Section 4 along with full problem details. The design of the study is along the lines of the exponential cases above. In Table 3, the worst case percentage for heuristic π^S are 36.43% (*A'*), 26.16% (*B'*), 20.21% (*C'*), and 4.13% (*D'*). The corresponding percentage for π^{SPI} are 0.00% (*A'*), 0.24% (*B'*), 1.46% (*C'*), and 0.20% (*D'*).

REMARK: A dynamic version of the priority policy π^S recalculates the expected remaining lifetimes at every decision epoch and updates the priority list. We have also implemented a single policy improvement step for this policy in several cases with Weibull lifetimes. The numerical evidence we have suggested is that the resulting policy does not offer any significant improvement on π^{SPI} . In some cases its performance is weaker.

The strong performance of π^{SPI} notwithstanding, its development via (8) and (9) is computationally prohibitive other than for small problems and special cases. In light of this computational intractability we proceed as follows: we shall develop an approximation $V_{\pi^S}^{(1)} : \Omega \rightarrow [0, \sum_{j=1}^J L_j]$ to the value function V_{π^S} . The dynamic heuristic which then results is obtained by using the approximation $V_{\pi^S}^{(1)}$ within (9). Hence we have

$$\begin{aligned} \pi^{SF1}(\mathbf{n}, t) &= \arg \max_j \left\{ \int_0^\infty \sum_{\mathbf{n}'} p(\mathbf{n}' | \mathbf{n}, t, j, s) V_{\pi^S}^{(1)}(\mathbf{n}', t + s) dG_j(s) \right\}. \end{aligned} \tag{10}$$

The approximating value function $V_{\pi^S}^{(1)}$ is obtained by developing a suitable fluid (deterministic) analog of the stochastic system emptying under policy π^S . In this approximating model, the (random) number of jobs remaining is represented by a fluid level which diminishes at a suitable deterministic rate to reflect *both* service completions and losses from the system of unserved jobs under π^S .

3.1. Approximating V_{π^S} : No Losses During Service

We proceed to discuss how to develop $V_{\pi^S}^{(1)}(\mathbf{n}, t)$, an approximation to $V_{\pi^S}(\mathbf{n}, t)$ based on a fluid model which drains fluid in a way which is appropriate given our assumption that jobs in service cannot experience premature loss. We focus initially on the contribution to $V_{\pi^S}^{(1)}(\mathbf{n}, t)$ from a single job class. We drop the class identifier and use X , Y and $\theta(\cdot)$ for the class lifetime, service time (both assumed absolutely continuous in this account) and lifetime hazard, respectively. Note that the lifetime hazard is given by $\theta(t) = F'(t)\{1 - F(t)\}^{-1}$, where F is the distribution function of X and $'$ denotes derivative. We also write $E(Y) = \mu^{-1}$.

We use the pair (m, s) to denote the amount of fluid (number of jobs) present when the processing of the class begins (m) and the time at which this happens (s) when π^S is applied to the fluid model. Because of the way in which π^S imposes static priorities among the classes, this class will be served continually from s until all of the corresponding fluid is drained. We use $N^{(1)}(m, s)$ for the number of services completed (which in the fluid model may be noninteger) during the processing of the class. It will then follow that $\mu^{-1}N^{(1)}(m, s)$ is the time taken to process the class under the fluid model.

The fluid is drained as follows: if $m \geq 1$ then a single unit of fluid is removed instantaneously (to signify the guaranteed service completion of one job) at time s . Loss of fluid is then experienced at a rate determined by the hazard rate $\theta(\tau)$ during the time period $s < \tau \leq s + \mu^{-1}$. Note that this period is the time occupied by the processing of the first job in the fluid model. If the amount of fluid remaining at time $s + \mu^{-1}$ exceeds one then a further single unit of fluid is removed instantaneously at time $s + \mu^{-1}$ and signifies the guaranteed service completion of a second job. Loss of fluid is then expected at a rate determined by the hazard rate $\theta(\tau)$ during the period $s + \mu^{-1} < \tau \leq s + 2\mu^{-1}$, and so on. If we write $R(\tau)$ for the amount of fluid remaining at time τ , we have for $\tau \geq s$

$$\begin{aligned} R'(\tau) &= -\theta(\tau)R(\tau), \tau \neq s + k\mu^{-1}, k \in \mathbb{N}, \\ R(\{s + k\mu^{-1}\}^+) &= \max\{R(s + k\mu^{-1}) - 1, 0\}, k \in \mathbb{N}, \\ R(s) &= m. \end{aligned} \tag{11}$$

If we define $k(m, s)$ by

$$k(m, s) = \min\{k; R(\{s + k\mu^{-1}\}^+) = 0\} \tag{12}$$

we then have

$$N^{(1)}(m, s) = k(m, s) + R(s + k(m, s)\mu^{-1}). \tag{13}$$

Note from (12) that $k(m, s)$ is the (integer) number of fully completed jobs under the fluid model whereas $R(s + k(m, s)\mu^{-1})$ is a fractional amount of fluid remaining after those completions and is deemed to yield a further fractional completion within the approximating fluid model.

In fact, the system (11) is straightforward to solve explicitly. To state the solution with a minimum of notation we develop the sequence

$$m_r(s) = \begin{cases} 1, & r = 0 \\ 1 + \sum_{u=0}^{r-1} \exp\left\{-\int_0^{(u+1)\mu^{-1}} \theta(s+v)dv\right\}, & r \in \mathbb{Z}^+. \end{cases}$$

Please note that $m_r(s)$ may be understood as the amount of fluid at time s to achieve $r + 1$ service completions. The proof of Proposition 1 can be found in the Appendix. Before its statement please note that the assumption that $E(X) < \infty$ implies that the hazard rate θ has an infinite integral over \mathbb{R}^+ and hence that $m_r(s) \rightarrow \infty, r \rightarrow \infty$ for all choices of s .

PROPOSITION 1:

- (a) If $m_{r-1}(s) < m \leq m_r(s)$ for some $r \in \mathbb{Z}^+$ then

$$\begin{aligned} N^{(1)}(m, s) &= r + \{m - m_{r-1}(s)\} \exp\left\{-\int_0^{r\mu^{-1}} \theta(s+v)dv\right\}; \end{aligned} \tag{14}$$

- (b) If $m \leq m_0(s) = 1$ then

$$N^{(1)}(m, s) = m.$$

To obtain $V_{\pi^S}^{(1)}(\mathbf{n}, t)$, we need to restore the class identifier to the notation and write $N_j^{(1)}(m_j, s_j)$ for the above fluid approximation for the number of Class j services completed from an initial state (m_j, s_j) . We now suppose that the classes are numbered according to their ordering by π^S , with Class 1 processed first and Class J last.

For fixed system state (\mathbf{n}, t) and $1 \leq j \leq J$, we inductively develop the quantities $v_j(\mathbf{n}, t)$ which records the number of class j services completed under the fluid model when static policy π^S is applied from this state, as follows:

$$v_1(\mathbf{n}, t) = N_1^{(1)}(n_1, t),$$

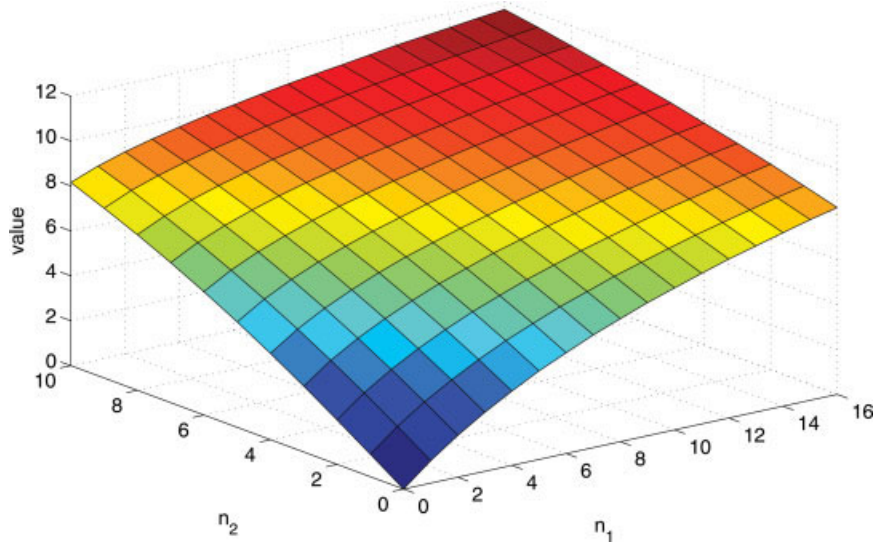


Figure 1. Values of $V_{\pi^S}^{(1)}(n_1, n_2, 0)$ where $0 \leq n_1 \leq 16, 0 \leq n_2 \leq 10$. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com]

and

$$v_j(\mathbf{n}, t) = N_j^{(1)} \left(n_j \exp \left\{ - \int_0^{\sum_{i=1}^{j-1} \mu_i^{-1} v_i(\mathbf{n}, t)} \theta_j(t+v) dv \right\}, t + \sum_{i=1}^{j-1} \mu_i^{-1} v_i(\mathbf{n}, t) \right), 1 \leq j \leq J. \quad (15)$$

The first argument of $N_j^{(1)}$ on the right-hand side of (15) is the number of class j jobs present when the processing of that class begins. The original number n_j (present at t) is diminished by losses occurring over the time period $[t, t + \sum_{i=1}^{j-1} \mu_i^{-1} v_i(\mathbf{n}, t)]$ during which the first $j - 1$ classes are processed. We now use the quantities in (15) to develop the needed approximating value function as

$$V_{\pi^S}^{(1)}(\mathbf{n}, t) = \sum_{j=1}^J v_j(\mathbf{n}, t).$$

Dynamic heuristic π^{SF1} is then developed from (10).

REMARK: It is straightforward to establish from the expression on the right-hand side of (14), together with the preceding expressions for the $m_r(s)$, that the quantity $N^{(1)}(m, s)$, regarded as a function of m only (fixed s) is continuous, increasing, piecewise linear and concave. It will then follow that the derived approximating value $V_{\pi^S}^{(1)}(\mathbf{n}, t)$ is increasing and concave componentwise in \mathbf{n} , for fixed t . This is exemplified in Fig. 1 below where values of $V_{\pi^S}^{(1)}(n_1, n_2, 0)$ are plotted for a two class problem whose details are as in the following Example.

EXAMPLE: Consider a two class example in which the lifetimes X_j are i.i.d. Weibull with hazard rate given by

$$\theta_j(t) = \alpha_j \beta_j^{-\alpha_j} t^{\alpha_j-1}, t \in \mathbb{R}^+, j = 1, 2. \quad (16)$$

In this illustrative example we set the parameter values to be $\alpha_1 = 1.06, \beta_1 = 56.77, \alpha_2 = 1.81, \beta_2 = 81.22$. We further assume that the service times are deterministic with rates $\mu_1 = 0.11, \mu_2 = 0.13$. In Fig. 2 find a summary of the decisions (which job class to process) taken by dynamic heuristic π^{SF1} at time points 0, 18.15, 30.85, and 45.39, should decisions be required then. These time points are chosen to be representative of likely decision times.

In Fig. 2 each filled circle indicates a decision in favor of Class 1, and each diamond a decision in favor of Class 2. Please note that $\theta_1(t) \geq \theta_2(t), 0 \leq t \leq 66.02$, whereas $\theta_2(t)/\theta_1(t)$ increases with t . Hence at time $t = 0$, Class 1 jobs will appear more urgent and in most states this is reflected in a decision to process this class. As time increases all residual lifetimes of jobs decrease, but those of Class 2 jobs decrease more rapidly. Hence decisions taken by π^{SF1} increasingly favor Class 2 jobs as time elapses.

3.2. Approximating \bar{V}_{π^S} : Losses During Service

We now give a brief account of how the above discussion should be modified for the variant of our model in which losses are allowed during service. See (7) mentioned earlier. It is clear that our initial policy π^S will continue to be asymptotically optimal in the sense discussed in Glazebrook et al. [11] for this variant of the basic model discussed earlier.

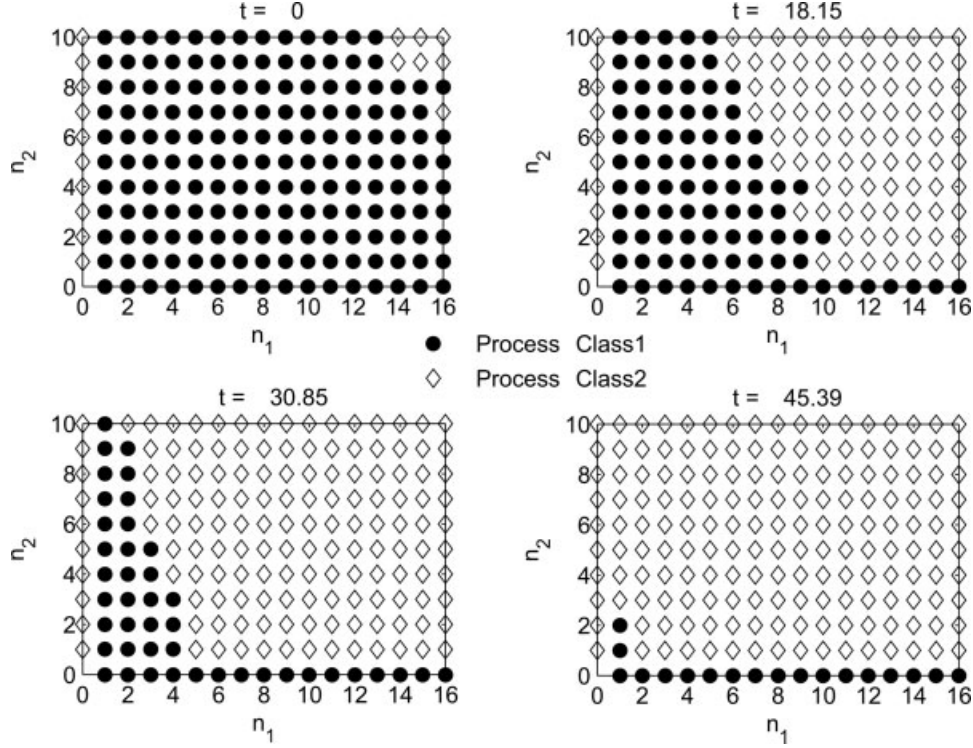


Figure 2. Decisions taken by policy π^{SF1} in states (n_1, n_2, t) where $0 \leq n_1 \leq 16, 0 \leq n_2 \leq 10$, and $t = 0, 18.15, 30.85, 45.39$.

We write $\bar{V}_{\pi^S} : \Omega \rightarrow [0, \sum_{j=1}^J L_j]$ for the value function for π^S now given by

$$\begin{aligned} \bar{V}_{\pi^S}(\mathbf{n}, t) &= P\{X_{\pi^S(\mathbf{n}, t)} > t + Y_{\pi^S(\mathbf{n}, t)} | X_{\pi^S(\mathbf{n}, t)} > t\} \\ &+ \int_0^\infty \sum_{\mathbf{n}'} p(\mathbf{n}' | \mathbf{n}, t, \pi^S(\mathbf{n}, t), s) \bar{V}_{\pi^S}(\mathbf{n}', t + s) dG_{\pi^S(\mathbf{n}, t)}(s), \\ \bar{V}_{\pi^S}(\mathbf{0}, t) &= 0. \end{aligned} \quad (17)$$

Direct computation of \bar{V}_{π^S} from (17) is computationally prohibitive other than for small problems and special cases. Hence we again deploy a fluid approximation to develop an approximating value function $V_{\pi^S}^{(2)} : \Omega \rightarrow [0, \sum_{j=1}^J L_j]$. The dynamic heuristic which results is given by

$$\begin{aligned} \pi^{\text{SF2}}(\mathbf{n}, t) &= \arg \max_j \left\{ P(X_j > t + Y_j | X_j > t) \right. \\ &\left. + \int_0^\infty \sum_{\mathbf{n}'} p(\mathbf{n}' | \mathbf{n}, t, j, s) V_{\pi^S}^{(2)}(\mathbf{n}', t + s) dG_j(s) \right\}. \end{aligned} \quad (18)$$

Consider now a single class with X, Y , and θ as in subsection (3.1) above. Under the fluid model suppose that π^S begins processing the class when in state (m, s) . If we write

$\bar{R}(\tau)$ for the amount of fluid remaining at time τ , we have that for $\tau \geq s$

$$\begin{aligned} \bar{R}'(\tau) &= -\theta(\tau)\bar{R}(\tau) - \mu \mathbb{I}\{\bar{R}(\tau) > 0\}, \\ \bar{R}(s) &= m. \end{aligned} \quad (19)$$

Hence, according to (19) fluid is now drained continuously under the impact of both losses from the system and service effort. If we define $\bar{i}(m, s)$ by

$$\bar{i}(m, s) = \inf\{\tau - s; \tau \geq s \text{ and } \bar{R}(\tau) = 0\}$$

then it is not difficult to show from (19) that $\bar{i}(m, s)$ satisfies the integral equation

$$\int_0^{\bar{i}(m, s)} \left\{ \exp\left[\int_0^u \theta(s+v)dv\right] \right\} du = m\mu^{-1}, \quad (20)$$

which has the solution

$$\bar{i}(m, s) = \theta^{-1} \ln(1 + m\theta\mu^{-1})$$

in the special exponential lifetime case in which the hazard rate is constant, namely, $\theta(\cdot) \equiv \theta$. Since $\bar{i}(m, s)$ is the time for the fluid to be drained, the corresponding number of service completions is given by

$$\bar{N}^{(2)}(m, s) = \mu \bar{i}(m, s).$$

To obtain $\bar{V}_{\pi^S}^{(2)}(\mathbf{n}, t)$ we develop quantities $\bar{v}_j(\mathbf{n}, t), 1 \leq j \leq J$, inductively as follows:

$$\bar{v}_1(\mathbf{n}, t) = \bar{N}_1^{(2)}(n_1, t),$$

and

$$\bar{v}_j(\mathbf{n}, t) = \bar{N}_j^{(2)} \left(n_j \exp \left\{ - \int_0^{\sum_{i=1}^{j-1} \mu_i^{-1} \bar{v}_i(\mathbf{n}, t)} \theta_j(t+v) dv \right\}, t + \sum_{i=1}^{j-1} \mu_i^{-1} \bar{v}_i(\mathbf{n}, t) \right), 1 \leq j \leq J. \tag{21}$$

Please note that in (21) the job classes are numbered in order of their processing by π^S . We now use the quantities in (21) to develop the approximating value function as

$$V_{\pi^S}^{(2)}(\mathbf{n}, t) = \sum_{j=1}^J \bar{v}_j(\mathbf{n}, t).$$

Dynamic heuristic π^{SF2} is then developed from (18).

4. NUMERICAL STUDY

In the following discussion we consider three general scenarios. These are (I) lifetimes and service times are both exponentially distributed, (II) Weibull lifetimes and deterministic service times, and (III) Weibull lifetimes and exponential service times. In both scenarios (I) and (II) it is possible (though expensive) to compute optimal policies for problems of modest size using dynamic programming (DP) by exploiting special features of the structure of the value iteration algorithms concerned. It is thus possible to assess the quality of heuristic π^{SF1} by direct comparison of the expected number of service completions achieved with the optimum. For scenario (III), it has not proved possible to develop optimal policies for problems of even modest size in reasonable time. In sharp contrast, it is a straightforward matter to perform the computations necessary to implement heuristic π^{SF1} *on-line*; namely, to obtain those $\pi^{SF1}(\mathbf{n}, t)$ which are required in any realization of the system. Hence, in scenario (III) it is natural to assess the *relative performance* of π^{SF1} and the competitor heuristics π^S [see (5)] and π^M [see (6)] by means of Monte Carlo simulation. In the numerical results we have also included the heuristic π^{SPI} for comparison when it has proved possible to do so.

4.1. Scenario (I): Lifetimes and Service Times Exponentially Distributed

In this scenario, crucially, the lifetime hazard rates θ_j are all constant functions. Hence the time dependence in the value

function $V_{\pi}(\mathbf{n}, t)$ for any stationary policy (i.e., one which makes decisions in any state (\mathbf{n}, t) which depend upon \mathbf{n} but not on t) disappears. In such cases, the optimality equation in (3) reduces to

$$V(\mathbf{n}) = 1 + \max_{j \in A(\mathbf{n})} \left\{ \int_0^{\infty} \sum_{\mathbf{n}'} p(\mathbf{n}'|\mathbf{n}, j, s) V(\mathbf{n}') \mu_j e^{-\mu_j s} ds \right\}, \mathbf{n} \neq \mathbf{0},$$

$$V(\mathbf{0}) = 0. \tag{22}$$

Equation (22) enables the development of optimal policies in this case along with the value $V(\mathbf{L})$ which is the expected number of service completions achieved from initial state \mathbf{L} . It is a straightforward matter to check that in this case heuristics $\pi^{SPI}, \pi^{SF1}, \pi^S$, and π^M are all stationary. Here, the appropriate value iteration algorithm for the computation of expected services achieved under any stationary policy π are developed from recursions of the form

$$V_{\pi}(\mathbf{n}) = 1 + \int_0^{\infty} \sum_{\mathbf{n}'} p(\mathbf{n}'|\mathbf{n}, \pi(\mathbf{n}), s) V_{\pi}(\mathbf{n}') \mu_j e^{-\mu_j s} ds, \mathbf{n} \neq \mathbf{0},$$

$$V_{\pi}(\mathbf{0}) = 0. \tag{23}$$

Problems were generated at random for two class ($J = 2$) and five class ($J = 5$) cases for each of four distinct assumptions (A, B, C, D) about the relative lengths of lifetimes and service times. We sample the key problem features $\mu_j^{-1}, \theta_j^{-1}, L_j$ as follows:

- $\mu_j^{-1} \sim U[1, 10]$ (all cases); (24a)
- $\theta_j^{-1} \mu_j | \mu_j^{-1} \sim U[0.1, 0.5]$ (very short lifetimes, A); (24b)
- $\theta_j^{-1} \mu_j | \mu_j^{-1} \sim U[0.5, 2.0]$ (short lifetimes, B); (24c)
- $\theta_j^{-1} \mu_j | \mu_j^{-1} \sim U[2.0, 10.0]$ (moderate lifetimes, C); (24d)
- $\theta_j^{-1} \mu_j | \mu_j^{-1} \sim U[10.0, 100.0]$ (long lifetimes, D); (24e)
- $L_j \sim DU[1, 50]$ ($J = 2$ cases); (24f)
- $L_j \sim DU[1, 6]$ ($J = 5$ cases). (24g)

In (24a)–(24g), $U[a, b]$ is a continuous uniform distribution on the range $[a, b]$ whereas $DU[a, b]$ is a discrete uniform distribution on $[a, b]$. Note that in Case A , (24b) indicates that the ratio of mean lifetime to mean service time lies in the range $[0.1, 0.5]$ and hence few service completions are likely. In Case D this ratio lies in the range $[10.0, 100.0]$ and hence many service completions are likely. We observe that the myopic nature of heuristic π^M suggests that it is likely

to perform well in Case *A* where the current decision in any state need take little account of the future. Furthermore, the asymptotic optimality of π^S in a “no premature job loss” limit suggests that it should perform well in case *D*.

For each of $J = 2$ and $J = 5$ and each of the cases *A*, *B*, *C*, and *D*, 500 problems were generated at random according to (24a)–(24g). For each problem value iteration was deployed to compute the mean number of service completions achieved under the heuristics $\pi^{S^{PI}}$, $\pi^{S^{F1}}$, π^S and π^M and under an optimal policy. In every problem, the percentage suboptimality $\Delta(\pi, \text{opt})$ of each heuristic $\pi = \pi^{S^{PI}}, \pi^{S^{F1}}, \pi^S, \pi^M$ was computed. Further, for each collection of 500 problems, the minimum, mean and maximum values of $\Delta(\pi, \text{opt})$ were recorded for each heuristic. These values may be found in Table 1 ($J = 2$) and Table 2 ($J = 5$).

From Table 1, and as indicated earlier, the performance of the asymptotically optimal heuristic π^S improves steadily from Case *A* to Case *D* while for myopic heuristic π^M the reverse is the case. Serious suboptimalities can occur especially in those problem instances for which these heuristics are not designed. The position is similar in Table 2 though the fact that in the $J = 5$ cases the values of the generated L_j are much smaller [see (24g)] means that the maximum suboptimalities for the myopic heuristic are substantially reduced. In sharp contrast, the heuristic $\pi^{S^{F1}}$ is robust; it performs well in all scenarios. It outperforms π^S and π^M in all cases with the single exception of Case *D*, $J = 5$ where the asymptotic optimality of π^S confers on the latter a slight advantage. As reported in Section 3, $\pi^{S^{PI}}$ performs outstandingly well and may be readily computed in exponential cases of modest size.

Table 1. Percentage deviation from optimal performance for exponential distributed lifetimes and service times when $J = 2$.

Case	$\Delta(\pi^{S^{PI}}, \text{opt})$	$\Delta(\pi^{S^{F1}}, \text{opt})$	$\Delta(\pi^S, \text{opt})$	$\Delta(\pi^M, \text{opt})$
<i>A</i> (very short lifetimes)				
Min	0.00	0.00	0.00	0.00
Mean	0.00	0.01	3.01	0.02
Max	0.00	0.55	23.79	1.94
<i>B</i> (short lifetimes)				
Min	0.00	0.00	0.00	0.00
Mean	0.00	0.02	2.52	0.14
Max	0.00	0.18	24.33	4.29
<i>C</i> (moderate lifetimes)				
Min	0.00	0.00	0.00	0.00
Mean	0.00	0.00	1.23	0.90
Max	0.00	0.12	16.52	9.32
<i>D</i> (long lifetimes)				
Min	0.00	0.00	0.00	0.00
Mean	0.00	0.02	0.12	2.23
Max	0.00	0.71	5.17	18.86

Table 2. Percentage deviation from optimal performance for exponential distributed life times and service times when $J = 5$.

Case	$\Delta(\pi^{S^{PI}}, \text{opt})$	$\Delta(\pi^{S^{F1}}, \text{opt})$	$\Delta(\pi^S, \text{opt})$	$\Delta(\pi^M, \text{opt})$
<i>A</i> (very short lifetimes)				
Min	0.00	0.00	0.00	0.00
Mean	0.00	0.01	1.36	0.04
Max	0.48	1.10	15.95	2.42
<i>B</i> (short lifetimes)				
Min	0.00	0.00	0.00	0.00
Mean	0.01	0.02	0.84	0.03
Max	0.88	1.02	11.86	1.80
<i>C</i> (moderate lifetimes)				
Min	0.00	0.00	0.00	0.00
Mean	0.01	0.02	0.13	0.34
Max	1.04	1.27	3.87	4.39
<i>D</i> (long lifetimes)				
Min	0.00	0.00	0.00	0.00
Mean	0.00	0.02	0.00	0.49
Max	0.05	0.46	0.33	4.36

4.2. Scenario (II): Weibull Lifetimes and Deterministic Service Times

In cases in which class j service times are deterministic of value μ_j^{-1} the value iteration procedure need only compute value functions V, 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 (4) now takes the form

$$V(\mathbf{n}, t) = 1 + \max_{j \in A(\mathbf{n})} \left\{ \sum_{\mathbf{n}'} p(\mathbf{n}' | \mathbf{n}, t, j, \mu_j^{-1}) V(\mathbf{n}', t + \mu_j^{-1}) \right\}, \mathbf{n} \neq \mathbf{0},$$

$$V(\mathbf{0}, t) = 0. \quad (25)$$

The effective discretisation of the time axis which results radically simplifies value iteration procedures.

The Weibull family of distributions yields a flexible way of modeling lifetimes. We adopt the parameterization to be found in (16) where the form of the Weibull hazard rate is given. In order for the class j hazard to be increasing, which is a natural assumption in many applications, we require that $\alpha_j > 1$. The mean of the Weibull random variable with hazard rate (16) is $\beta_j \Gamma(1 + \alpha_j^{-1})$.

In our numerical study, problems were generated at random for two class ($J = 2$) and five class problems ($J = 5$) under each of four distinct assumptions (*A'*, *B'*, *C'*, *D'*) about the relative lengths of lifetimes and service times. We sample the key problem features $\mu_j^{-1}, \alpha_j, \beta_j, L_j$ as follows:

$$\mu_j^{-1} \sim U[1, 10] \quad (\text{all cases}); \quad (26a)$$

$$\alpha_j \sim U[1.0, 2.0] \quad (\text{all cases}); \quad (26b)$$

$$\beta_j \Gamma(1 + \alpha_j^{-1}) \mu_j | \mu_j^{-1}, \quad (\text{very short lifetimes, } A'); \quad (26c)$$

$$\alpha_j \sim U[0.1, 0.5] \quad (26d)$$

$$\beta_j \Gamma(1 + \alpha_j^{-1}) \mu_j | \mu_j^{-1}, \quad (\text{short lifetimes, } B'); \quad (26d)$$

$$\alpha_j \sim U[0.5, 2.0] \quad (26e)$$

$$\beta_j \Gamma(1 + \alpha_j^{-1}) \mu_j | \mu_j^{-1}, \quad (\text{moderate lifetimes, } C'); \quad (26e)$$

$$\alpha_j \sim U[2.0, 10.0] \quad (26f)$$

$$\beta_j \Gamma(1 + \alpha_j^{-1}) \mu_j | \mu_j^{-1}, \quad (\text{long lifetimes, } D'); \quad (26f)$$

$$\alpha_j \sim U[10.0, 100.0] \quad (26g)$$

$$L_j \sim DU[1, 20] \quad (J = 2 \text{ cases}); \quad (26g)$$

$$L_j \sim DU[1, 5] \quad (J = 5 \text{ cases}). \quad (26h)$$

Note that values of β_j are derived from the values of μ_j^{-1} and α_j obtained from the draws in (26a) and (26b) and the value of $\beta_j \Gamma(1 + \alpha_j^{-1})$ obtained from whichever is appropriate of the draws in (26c)–(26f). Note also that the values of the L_j drawn from (26g)–(26h) will tend to be smaller than those in (24f)–(24g). This is forced upon us by the added complexity of the recursion (25) in comparison with (22). As with scenario (I) comparisons between the heuristics are based on 500 problems randomly generated as in (26a)–(26h) above for each of $J = 2$ and $J = 5$ and each of the cases A' , B' , C' , and D' . Note that development of policy π^{SP1} is possible for the $J = 2$ cases, though computationally expensive. Hence this heuristic has been included in the $J = 2$ study but not

Table 3. Percentage deviation from optimal performance for Weibull lifetimes and deterministic service times when $J = 2$.

Case	$\Delta(\pi^{SP1}, \text{opt})$	$\Delta(\pi^{SF1}, \text{opt})$	$\Delta(\pi^S, \text{opt})$	$\Delta(\pi^M, \text{opt})$
<i>A'</i> (very short lifetimes)				
Min	0.00	0.00	0.00	0.00
Mean	0.00	0.00	4.27	0.06
Max	0.00	0.00	36.34	2.80
<i>B'</i> (short lifetimes)				
Min	0.00	0.00	0.00	0.00
Mean	0.00	0.01	2.53	0.20
Max	0.24	0.94	26.16	6.14
<i>C'</i> (moderate lifetimes)				
Min	0.00	0.00	0.00	0.00
Mean	0.02	0.03	0.71	1.30
Max	1.46	1.81	20.21	16.29
<i>D'</i> (long lifetimes)				
Min	0.00	0.00	0.00	0.00
Mean	0.00	0.00	0.22	2.31
Max	0.20	0.38	4.13	21.36

Table 4. Percentage deviation from optimal performance for Weibull lifetimes and deterministic service times when $J = 5$.

Case	$\Delta(\pi^{SF1}, \text{opt})$	$\Delta(\pi^S, \text{opt})$	$\Delta(\pi^M, \text{opt})$
<i>A'</i> (very short lifetimes)			
Min	0.00	0.00	0.00
Mean	0.00	2.63	0.11
Max	0.02	33.42	5.39
<i>B'</i> (short lifetimes)			
Min	0.00	0.00	0.00
Mean	0.02	0.88	0.07
Max	1.79	10.57	2.13
<i>C'</i> (moderate lifetimes)			
Min	0.00	0.00	0.00
Mean	0.02	0.17	0.65
Max	0.84	4.53	6.25
<i>D'</i> (long lifetimes)			
Min	0.00	0.00	0.00
Mean	0.03	0.22	0.67
Max	0.63	2.48	5.66

in the $J = 5$ study since it was not possible to get results for the latter in reasonable computational time. The results are presented in Tables 3 and 4.

The evidence provided by Tables 3 and 4 yields similar conclusions to those drawn from Tables 1 and 2. The heuristics π^S and π^M continue to have poor worst case performance in settings for which they were not designed. The uniform excellence of the performance of π^{SF1} is in clear contrast.

4.3. Scenario (III): Weibull Lifetimes and Exponential Service Times

In both scenarios (I) and (II) it was possible to exploit model features to develop exact analyses based on DP value iteration for problems of modest size. In this way, it was possible (though expensive) to develop optimal policies and calculate the expected number of service completions for the heuristics of interest and for the optimal policy. While this is no longer possible in scenario (III), it is still straightforward to develop on-line applications of the three heuristics π^{SF1} , π^S , and π^M (though not of π^{SP1}). By this we mean that it is straightforward to perform the computations required to determine the action prescribed by each heuristic in any given state.

With this in mind, the natural approach here is to use Monte Carlo simulation to conduct a comparative study of the performance of π^{SF1} , π^S , and π^M . Problem parameters were chosen as in scenario (I) (service times) and scenario (II) (lifetimes) though time constraints have limited the study to the $J = 2$ case, with 100 problems generated in each category. In Table 5 find information on the estimated values of $\Delta(\pi^S, \pi^{SF1})$ and $\Delta(\pi^M, \pi^{SF1})$, the percentage excess expected number of successes achieved by π^{SF1} over π^S and

Table 5. Percentage difference between the performance of heuristics π^S, π^M , and that of heuristic π^{SF1} for Weibull lifetimes and exponential service times when $J = 2$.

Case	$\Delta(\pi^S, \pi^{SF1})$	$\Delta(\pi^M, \pi^{SF1})$
<i>A'</i> (very short lifetimes)		
Min	-0.57	-0.57
Mean	2.18	0.07
Max	14.47	1.16
<i>B'</i> (short lifetimes)		
Min	-0.60	-2.07
Mean	1.43	0.15
Max	19.63	8.86
<i>C'</i> (moderate lifetimes)		
Min	-0.41	-0.56
Mean	0.25	1.02
Max	5.74	10.47
<i>D'</i> (long lifetimes)		
Min	-0.01	0.00
Mean	0.16	0.94
Max	3.39	8.24

π^M , respectively. Hence positive values indicate stronger performance by π^{SF1} while negative values indicate stronger performance by the competitor heuristic.

While we encountered occasional problem instances in which the estimated expected number of successes achieved by the competitor heuristic exceeded that of π^{SF1} these were rare and the differences usually very small. In all categories the average performance of π^{SF1} was superior. There continued to be problem instances in which π^S, π^M performed very poorly in comparison with π^{SF1} .

APPENDIX

PROOF OF PROPOSITION 1: From (11), we have that when $R(\{s + k\mu^{-1}\}^+) > 0$ it then follows that

$$R'(\tau) = -\theta(\tau)R(\tau), s + k\mu^{-1} < \tau < s + (k + 1)\mu^{-1},$$

and hence that

$$R(s + (k + 1)\mu^{-1}) = R(\{s + k\mu^{-1}\}^+) \exp \left\{ - \int_{k\mu^{-1}}^{(k+1)\mu^{-1}} \theta(s + v)dv \right\}. \tag{A1}$$

We now develop the sequence $\{\hat{R}(k), k \in \mathbb{N}\}$ as follows:

$$\hat{R}(0) = m,$$

$$\hat{R}(k + 1) = \{\hat{R}(k) - 1\} \exp \left\{ - \int_{k\mu^{-1}}^{(k+1)\mu^{-1}} \theta(s + v)dv \right\}, k \in \mathbb{N}. \tag{A2}$$

Substituting m into the sequence we have for $k \in \mathbb{N}$ that

$$\hat{R}(k) = m \exp \left\{ - \int_0^{k\mu^{-1}} \theta(s + v)dv \right\} - \sum_{u=0}^{k-1} \exp \left\{ - \int_{u\mu^{-1}}^{k\mu^{-1}} \theta(s + v)dv \right\}. \tag{A3}$$

In (A3) and elsewhere we use the convention that an empty sum is zero. From (11) and (A1) it is straightforward that if $\hat{R}(l) > 0, 0 \leq l \leq k$, then

$$\hat{R}(k) = R(s + k\mu^{-1}). \tag{A4}$$

Now consider m in range $m_{r-1}(s) < m \leq m_r(s)$ where $r \in \mathbb{Z}^+$. We write m in the form

$$m = 1 + \sum_{u=0}^{r-2} \exp \left\{ \int_0^{(u+1)\mu^{-1}} \theta(s + v)dv \right\} + \gamma \exp \left\{ \int_0^{r\mu^{-1}} \theta(s + v)dv \right\} \tag{A5}$$

where $\gamma \in (0, 1]$. It is straightforward from (A3), (A5) and an induction argument that $\hat{R}(k)$ decreases as k increases from 0 to r with

$$\hat{R}(k) = 1 + \sum_{u=k}^{r-2} \exp \left\{ \int_{k\mu^{-1}}^{(u+1)\mu^{-1}} \theta(s + v)dv \right\} + \gamma \exp \left\{ \int_{k\mu^{-1}}^{r\mu^{-1}} \theta(s + v)dv \right\}, 0 \leq k \leq r - 1. \tag{A6}$$

It now follows from (A4) and (A6) that

$$\hat{R}(k) = R(s + k\mu^{-1}), 0 \leq k \leq r - 1,$$

and in particular that

$$\hat{R}(r - 1) = R(s + (r - 1)\mu^{-1}) = 1 + \gamma \exp \left\{ \int_{(r-1)\mu^{-1}}^{r\mu^{-1}} \theta(s + v)dv \right\}. \tag{A7}$$

It follows from (A2), (A4) and (A7) that

$$\hat{R}(r) = R(s + r\mu^{-1}) = \gamma. \tag{A8}$$

From (11) to (13) we now infer that

$$N^{(1)}(m, s) = r + \gamma = r + \{m - m_{r-1}(s)\} \exp \left\{ - \int_0^{r\mu^{-1}} \theta(s + v)dv \right\}. \tag{A9}$$

Equation (A9) is recovered from (A5) by solving for γ and using the expression for $m_r(s)$ in the article. This completes the proof of Proposition 1(a). Proposition 1(b) follows trivially from the definition of the quantities concerned. \square

ACKNOWLEDGMENTS

The first author would like to express his gratitude to Lancaster University Management School for the support received during this work.

REFERENCES

[1] P.S. Ansell, K.D. Glazebrook, and C. Kirkbride, Generalised join the shortest queue policies for the dynamic routing of jobs to multi-class queues, *J Oper Res Soc* 54 (2003), 379–389.

- [2] N.T. Argon, S. Ziya, and R. Richter, Scheduling impatient jobs in a clearing system with insights on patients triage in mass casualty incidents, *Probabil Eng Information Sci* 22 (2008), 301–332.
- [3] A. Bassamboo, J.M. Harrison, and A. Zeevi, Dynamic routing and admission control in high-volume service systems: Asymptotic analysis via multi-scale fluid limits, *Queue Systems* 51 (2005), 249–285.
- [4] S. Bhulai, Dynamic routing policies for multiskill call centers, *Probabil Eng Information Sci* 23 (2009), 101–119.
- [5] S. Bhulai and G. Koole, On the structure of value functions for threshold policies in queueing models, *J Appl Probabil* 40 (2003), 613–622.
- [6] B. Doytchinov, J.P. Lehoczyk, and S. Shreve, Real-time queues in heavy traffic with earliest-deadline-first queue discipline, *The Annals of Applied Probabil* 11 (2001), 332–378.
- [7] O. Garnett, A. Mandelbaum, and M.I. Reiman, Designing a call center with impatient customers, *Manufactur Service Oper Manage* 4 (2002), 208–227.
- [8] D.P. Gaver and P.A. Jacobs, Servicing impatient tasks that have uncertain outcomes, Technical Report, Naval Postgraduate School, Monterey, CA, 2000.
- [9] D.P. Gaver, P.A. Jacobs, G. Samorodnitsky, and K.D. Glazebrook, Modeling and analysis of uncertain time-critical tasking problems, *Naval Res Logistics* 53 (2006), 588–599.
- [10] K.D. Glazebrook, On stochastic scheduling problems with due dates, *Int J Systems Sci* 14 (1983), 1259–1271.
- [11] K.D. Glazebrook, P.S. Ansell, R.T. Dunn, and R.R. Lumley, On the optimal allocation of service to impatient tasks, *J Appl Probabil* 41 (2004), 51–72.
- [12] K.D. Glazebrook and E.L. Punton, Dynamic policies for uncertain time-critical tasking problems, *Naval Res Logistics* 55 (2008), 142–155.
- [13] K.D. Glazebrook, C. Kirkbride, and J. Ouenniche, Index policies for the admission control and routing of impatient customers to heterogeneous service stations, *Oper Res* 57 (2009), 975–989.
- [14] Z. Jiang, T.G. Lewis, and J.Y. Colin, Scheduling hard real-time constrained periodic tasks on multiple processors, *J Systems Software* 19 (1996), 102–118.
- [15] G. Koole, Introduction to the special issue on call center management, *Manage Sci* 54 (2008), 237.
- [16] P. Marbach, O. Mihatsch, and J.N. Tsitsiklis, Call admission control and routing in integrated services networks using neuro-dynamic programming, *IEEE J Select Areas Commun* 18 (2000), 197–208.
- [17] M. Opp, K. Glazebrook, and V.G. Kulkarni, Outsourcing warranty repairs: Dynamic allocation, *Naval Res Logistics* 52 (2005), 381–396.
- [18] T.J. Ott and K.R. Krishnan, Separable routing: A scheme for state-dependent routing of circuit switched telephone traffic, *Ann Oper Res* 35 (1992), 43–68.
- [19] W.B. Powell, *Approximate dynamic programming: Solving the curses of dimensionality*, Wiley, Hoboken, New Jersey, 2007.
- [20] M.L. Puterman, *Markov decision processes: Discrete stochastic dynamic programming*, Wiley, New York, 1994.
- [21] D. Roubos and S. Bhulai, Average-cost approximate dynamic programming for the control of birth-death processes, Technical Report, VU University, Amsterdam, NL, 2007.
- [22] D. Roubos and S. Bhulai, Approximate dynamic programming techniques for the control of time-varying queueing systems applied to call centers with abandonments and retrials, Technical Report, VU University, Amsterdam, NL, 2009.
- [23] H.C. Tijms, *Stochastic models: An algorithmic approach*, Wiley, Chichester, 1994.
- [24] J. Van Mieghem, Due-date scheduling: asymptotic optimality of generalized longest queue and generalized largest delay rules, *Oper Res* 51 (2003), 113–122.