Dynamic Policies for Uncertain Time-Critical Tasking Problems

Kevin D. Glazebrook,[1] Emma L. Punton[2]

[1] *Department of Mathematics and Statistics and the Management School, Lancaster University, Lancaster LA1 4YX, United Kingdom*

[2] *School of Management, University of Edinburgh, Edinburgh EH8 9JY, United Kingdom*

Received 13 June 2006; revised 7 August 2007; accepted 25 November 2007
DOI 10.1002/nav.20272
Published online 9 January 2008 in Wiley InterScience (www.interscience.wiley.com).

Abstract: A recent paper by Gaver et al. [6] argued the importance of studying service control problems in which the usual assumptions (i) that tasks will wait indefinitely for service and (ii) that successful service completions can be observed instantaneously are relaxed. Military and other applications were cited. They proposed a model in which arriving tasks are available for service for a period whose duration is unknown to the system's controller. The allocation of a large amount of processing to a task may make more likely its own successful completion but may also result in the loss of many unserved tasks from the system. Gaver et al. [6] called for the design of dynamic policies for the allocation of service which maximizes the rate of successful task completions achieved, or which come close to doing so. This is the theme of the paper. We utilize dynamic programming policy improvement approaches to design heuristic dynamic policies for service allocation which may be easily computed. In all cases studied, these policies achieve throughputs close to optimal. © 2008 Wiley Periodicals, Inc. Naval Research Logistics 55: 142–155, 2008

Keywords: dynamic programming; heuristic dynamic policies; service control problems

1. INTRODUCTION

The paper considers a simple scenario in which tasks arrive and seek service which is provided by a single server. In most standard models of such systems, strong assumptions are made about (i) all tasks' limitless availability for service, namely their preparedness to wait in the system until service is delivered, and (ii) the server's capacity to deliver successful processing of all tasks and to know when this has been achieved. In a recent paper, Gaver et al. [6] argued that in many situations in which service is offered, one or both of these assumptions may not hold.

As an example Gaver et al. [7] consider a Homeland Security scenario in which hostile Red vessels arrive at and move through a maritime domain toward a vulnerable target, protected by Blue. The domain is also visited by nonhostile vessels (Whites) of interest. In this context service by Blue consists of an attempted classification (into nonhostile White or hostile Red) of suspicious vessels encountered in the domain by an overhead sensor. The Blue sensor will track any vessel it classifies as Red, until it is relieved by another platform, perhaps a destroyer pair. While correct classification by the sensor is never certain in finite time, the longer the sensor spends on classification the more likely it is to be correct. However, long services by Blue increase the chance that hostile Reds will traverse the domain to the target (i.e. be lost from the system) unclassified. The question naturally arises as to how the sensor should be deployed (i.e., how long allocated times spent on classification should be) to achieve a maximized correct classification rate among all vessels arriving at the domain.

Gaver et al. [6] mention other scenarios in which Red agents seek to penetrate Blue defenses. In one such, a Red will leave once his offensive task has been completed while Blue's service of Red seeks his destruction, with longer services more likely to achieve this. In such a scenario, Blue may not be able to establish with certainty whether any attempt to destroy Red has been successful. Gaver et al. [6] also cite examples from the delivery of emergency medical treatment.

We shall consider a scenario in which tasks arrive at a single server according to a Poisson process. Each task has a limited availability for useful service which is unknown to the controller of the system. Once this time has expired, the task is lost. The case of tasks with *known* deadlines

Correspondence to: Kevin D. Glazebrook (k.glazebrook@lancaster.ac.uk) is important and (relatively) well studied [4, 8, 12, 14–16]. We shall further suppose that the server has limited information concerning the efficacy of each service. In light of this, the server adopts an approach to processing in which a period of allocated time for the service of a given task is determined in advance to its processing. In choosing such *allocated service times* there is a clear trade-off to be considered. Large allocated times are more likely to lead to a successful service, but may also lead to more waiting tasks being lost from the system. How this trade-off is resolved will plainly depend upon the number of currently waiting tasks at risk of loss. We would expect shorter allocated times to be appropriate when the task queue is long. Gaver et al. [6] studied the *static* problem of determining the *constant* allocated service time which, when applied to *all* served tasks, maximizes *throughput* (the time average rate of tasks served successfully) and found that even this simple version of the service allocation problem is extremely challenging. They further made a simple myopic proposal for the *dynamic* problem in which service allocations take account of queue length, but argued that the development of approximately optimal policies remained an important outstanding issue. It is this issue which is the concern of the current paper.

The authors know of no previous work on their model, save that of Gaver et al. [6], cited above. In earlier work, Glazebrook and Punton [10] considered a simpler set up in which the penalty imposed upon waiting tasks for large allocated service times is expressed through a discounted reward structure. Further, Glazebrook et al. [9] and Harrison and Zeevi [11] have explored the optimal dynamic allocation of service in the face of customer loss through impatience in contexts where the successful completion of service is observed immediately. There is also a developing literature concerning how the phenomenon of customer impatience should impact the design and operation of call centers [2, 5].

The paper is structured as follows: our service control model is introduced in Section 2, while in Section 3 a tractable class of Markovian policies is introduced. These were first discussed by Gaver et al. [6] and permit calculation of key performance measures, including throughput, via explicit formulae. In Section 4 we develop a class of dynamic heuristics (Heuristic I) by a dynamic programming (DP) policy improvement approach which applies a single DP step to the value function associated with the optimal Markovian policy. In Section 5, a second class of dynamic heuristics (Heuristic II) is developed by application of a single policy improvement step to a strongly performing static policy. The paper concludes in Section 6 with an account of a numerical investigation which testifies to the very strong performance of Heuristic II throughout. A carefully designed static policy, while certainly inferior to Heuristic II, is seen to perform remarkably well in the dynamic class quite widely, but does less well for problems where the achievable throughput from any policy is low.

2. THE MODEL

Tasks seeking service arrive in a Poisson stream with rate λ. Once in the system, each task has an exponentially distributed amount of time (with mean θ^{-1}) during which it is available for service. These times are independent for distinct tasks. If a task's $\exp(\theta)$ *availability time* expires before successful service has been achieved, it leaves the system unprocessed. We shall suppose throughout that $\theta > 0$.

A single server is available to serve tasks. Decision epochs for the server are the times at which periods of allocated service expire (and the system is nonempty) together with the times at which tasks arrive at an empty system. At each decision epoch, the server observes the number of tasks currently present (n) and chooses a period of allocated service ($t(n)$) for one of them. The server is then committed to this task for this allocated period. Each task is assumed to have a *service requirement*, which is an unobservable positive-valued random variable with distribution function F. Service requirements for distinct tasks are independent and identically distributed. The probability that an allocated service of time t sees a task successfully completed is given by

$$\gamma(t) \equiv \int_0^t e^{-\theta s} dF(s), \tag{1}$$

namely the probability that the task's service requirement is less than $\min(t, S)$, where $S \sim \exp(\theta)$. Moreover, if n tasks are present at the beginning of this allocated service of length t (including the one about to be served) then it is easy to show that the number of tasks remaining at its conclusion is

$$X(t|n) + Y(t), \tag{2}$$

where $X(t|n)$ and $Y(t)$ are independent random variables taking values in the non-negative integers. The random variable $X(t|n)$ has a binomial distribution $\text{Bin}(n-1, e^{-\theta t})$ and is the number of those tasks present in the system at the start of the allocated service (excepting the one chosen for service) which remain at its conclusion. The random variable $Y(t)$ has a Poisson distribution $\text{Poisson}\{\lambda\theta^{-1}(1-e^{-\theta t})\}$ and is the number of tasks which arrive in the system during the allocated service and which remain at its conclusion. The random variable $X(t|n)$ decreases stochastically in t for each n while $Y(t)$ increases stochastically in t. Whether the system is likely to experience a net gain or a net loss of tasks during the allocated service of length t depends critically upon the relative sizes of $n-1$ and $\lambda\theta^{-1}$. Each task is deemed to be unavailable for further service at the conclusion of its allocated service time.

A *stationary policy* is given by a map $t(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$ which determines service times as a function of the number of tasks present. The goal of analysis is the determination of a stationary policy which maximizes throughput, or which comes close to doing so. However, this is a stochastic dynamic optimization problem which is challenging in several respects. First, its state space is countably infinite (the natural numbers $\mathbb{N}$) while its action space is uncountable (the positive reals $\mathbb{R}^+$). Second, actions taken relate directly to the expiration of time which means that many standard approaches require modification. A conventional proposal would apply stochastic DP to a series of finite state/finite action/discrete time approximations to the system. Each would involve truncating the state space (replacing $\mathbb{N}$ by $\{0, 1, \ldots, N\}$ for some suitably large N) and introducing a positive discrete time quantum δ along with a truncated finite action space ($\{\delta, 2\delta, \ldots, M\delta\}$ for some suitably large M). The existence of an optimal stationary policy $t(\cdot; N, M, \delta)$ is guaranteed for any choice of (N, M, δ). However, its determination by conventional DP is computationally expensive for large N, M, and small δ. Moreover, the need to demonstrate *convergence* of the solution (in relation to throughputs achieved), namely that suitably increasing N, M while reducing δ will have negligible benefits, is exacting.

The paper presents two classes of effective dynamic heuristics for this service control problem, whose design avoids the aforementioned complex and cumbersome solution procedures. These policies are easy to compute. Moreover, numerical evidence suggests that they achieve throughputs which are close to optimal in all cases studied. Both policy classes emerge from a simple *Markovian* policy class which is discussed in Gaver et al. [6] and which is described briefly in the next section.

2.1. Comment

For simplicity and clarity we shall focus exclusively on the scenario described in the opening two paragraphs of this section. However, the methods of analysis we describe in the upcoming sections are readily extended to situations in which (a) tasks are guaranteed not to depart the system during service, and/or (b) a delivered service of length t achieves service quality or customer satisfaction $s(t) \in \mathbb{R}^+$, with $s(t)$ increasing in t. In the latter case, the goal of analysis is the determination of a stationary policy which maximizes the time average level of customer satisfaction achieved over an infinite horizon, or which comes close to doing so.

Regarding (b) mentioned earlier, please note that Deshmukh and Jain [3] consider a finite capacity queueing model in which the objective is maximization of customer satisfaction net of (linear) waiting costs. There are no losses from queue or from service. In their simple finite state scenario they are able to obtain characterizations of optimal policies. They do not discuss the computation of optimal or near optimal policies.

3. A STATIC MARKOVIAN POLICY CLASS

Suppose that, in the context of the problem described in Section 2, every allocated service is drawn independently from an exponential distribution with mean μ^{-1}. It then follows that, in advance of sampling, each allocated service results in a successful service completion with probability

$$\bar{\gamma}(\mu) \equiv \int_0^\infty e^{-(\theta+\mu)t} dF(t). \tag{3}$$

Moreover, the number of tasks present in the system (namely the task currently being served together with any waiting for service) follows a state-dependent birth–death process with birth (arrival) rate λ in all states and death (completion or loss) rate $\mu + \theta(n - 1)$ in state $n \geq 1$.

Standard theory indicates that the throughput achieved under the aforementioned policy is given by

$$\bar{\omega}(\mu) = \mu\bar{\gamma}(\mu) \left[\sum_{n=1}^{\infty} \left\{ \prod_{k=0}^{n-1} \lambda(\mu + \theta k)^{-1} \right\} \right] \times \left[1 + \sum_{n=1}^{\infty} \left\{ \prod_{k=0}^{n-1} \lambda(\mu + \theta k)^{-1} \right\} \right]^{-1}. \tag{4}$$

Gaver et al. [6] argue that $\bar{\omega}(\mu)$ is maximized by some finite μ, μ^* say. Numerical evidence is given in Gaver et al. [6] that the policy of giving each served task an allocation fixed in size and equal to $(\mu^*)^{-1}$ is close to optimal in the *static class* of policies in which all allocated services are equal.

3.1. Comment

All of the numerical examples considered in the paper suppose that service times are drawn independently from a gamma distribution $\Gamma(r, \nu)$ with mean r/ν and probability density function

$$f(t|r, \nu) = \nu^r t^{r-1} e^{-\nu t} \{\Gamma(r)\}^{-1}, \quad t \geq 0. \tag{5}$$

In this case the quantity $\bar{\gamma}$ in (3) is given by

$$\bar{\gamma}(\mu) = \{\nu/(\nu + \theta + \mu)\}^r. \tag{6}$$

It is standard that such service times are stochastically increasing in r (with ν fixed) and stochastically decreasing in ν (with r fixed). It is straightforward to show that, for any fixed μ, the throughput $\bar{\omega}(\mu)$ is increasing in ν and λ, while decreasing in r and θ. In all cases, the nonspecified parameters are assumed to be held constant. Further, the quantity $\bar{\omega}(\mu)/\lambda$, namely the proportion of arriving tasks which are successfully served, is decreasing in λ for any fixed μ.

To facilitate the analysis of Section 4 we shall require the *bias function* associated with the Markov policy μ, written $b(\cdot,\mu) : \mathbb{N} \to \mathbb{R}$ and which measures the relative transient effect of commencing processing in general state n rather than 0. To develop this, we require the following quantities:

- for T, $\mu \in \mathbb{R}^+$ and $n \in \mathbb{N}$ we write $C(n,T;\mu)$ for the expected number of successful task completions during the period $[0,T)$ under Markov policy μ when n is the system state at time 0;
- for $\mu \in \mathbb{R}^+$ and $n \in \mathbb{Z}^+$ we write $\bar{C}(n,\mu)$ for the expected number of successful task completions under Markov policy μ from time 0 up to the time at which the system empties for the first time when n is the initial system state;
- for $\mu \in \mathbb{R}^+$ and $n \in \mathbb{Z}^+$ we write $\bar{T}(n,\mu)$ for the expected time it takes the system to empty for the first time from initial state n under Markov policy μ.

We pause to note that the system evolving under Markov policy μ is trivially ergodic, from which we deduce that the quantities $\bar{T}(n,\mu)$ and $\bar{C}(n,\mu)$ are guaranteed to be finite for any $n \in \mathbb{N}$. We now define the bias function $b(\cdot,\mu)$, which is also guaranteed to be finite by the aforementioned ergodicity and standard theory [17].

DEFINITION 1: The bias function $b(\cdot,\mu) : \mathbb{N} \to \mathbb{R}$ is defined by

$$b(n,\mu) = \lim_{T\to\infty} \{C(n,T;\mu) - C(0,T;\mu)\}, \qquad (7)$$

where the limit in (7) is guaranteed to exist and be finite for any $n \in \mathbb{N}$.

LEMMA 1: The bias $b(n,\mu)$ is given by

$$b(n,\mu) = \frac{\mu\bar{\gamma}(\mu)}{\lambda}\left[\sum_{m=1}^{n}\sum_{r=1}^{\infty}\left\{\prod_{k=0}^{r-1}\lambda(\mu+\theta(m-1)+\theta k)^{-1}\right\}\right] \times \left[1+\sum_{r=1}^{\infty}\left\{\prod_{k=0}^{r-1}\lambda(\mu+\theta k)^{-1}\right\}\right]^{-1}, \quad n \in \mathbb{Z}^+,$$

and is zero when $n = 0$.

PROOF: By deployment of the ergodicity of the system under Markov policy μ and utilizing the fact that the system regenerates upon entry into the empty state, standard theory [17] yields that

$$b(n,\mu) = \bar{C}(n,\mu) - \bar{\omega}(\mu)\bar{T}(n,\mu), \quad n \in \mathbb{Z}^+. \qquad (8)$$

Plainly from (7) we have that $b(0,\mu) = 0$. The quantity $\bar{\omega}(\mu)$ is given by the expression in (4).

In order to compute $\bar{C}(n,\mu)$ and $\bar{T}(n,\mu)$, we first consider the quantity $\bar{T}(m,\mu) - \bar{T}(m-1,\mu)$ for $m \geq 1$. To compute this, we consider a state-dependent birth–death process defined on the state space $\{m-1, m, m+1, \ldots\}$. The birth rate in all states is λ, while the death rate in state $m-1$ is zero and in all other states is $\mu + \theta(n-1)$, $n \geq m$. By direct comparison with our service system evolving under Markov policy μ we see that the stationary probability that the aforementioned birth–death process is in its lowest state $(m-1)$ may be expressed as

$$\lambda^{-1}\{\lambda^{-1} + \bar{T}(m,\mu) - \bar{T}(m-1,\mu)\}^{-1}. \qquad (9)$$

However, by standard theory, this probability may also be expressed as

$$\left[1+\sum_{r=1}^{\infty}\left\{\prod_{k=0}^{r-1}\lambda(\mu+\theta(m-1)+\theta k)^{-1}\right\}\right]^{-1}. \qquad (10)$$

Equating the expressions in (9) and (10) yields the relation

$$\bar{T}(m,\mu) - \bar{T}(m-1,\mu) = \lambda^{-1} \times \sum_{r=1}^{\infty}\left\{\prod_{k=0}^{r-1}\lambda(\mu+\theta(m-1)+\theta k)^{-1}\right\}, \quad m \in \mathbb{Z}^+. \qquad (11)$$

It is straightforward to establish that

$$\bar{C}(m,\mu) - \bar{C}(m-1,\mu) = \mu\bar{\gamma}(\mu)\{\bar{T}(m,\mu) - \bar{T}(m-1,\mu)\}, \quad m \in \mathbb{Z}^+. \qquad (12)$$

We now obtain the expression for $b(n,\mu)$ in the lemma by writing

$$\begin{aligned} b(n,\mu) &= \bar{C}(n,\mu) - \bar{\omega}(\mu)\bar{T}(n,\mu) \\ &= \sum_{m=1}^{n}[\bar{C}(m,\mu) - \bar{C}(m-1,\mu) \\ &\quad - \bar{\omega}(\mu)\{\bar{T}(m,\mu) - \bar{T}(m-1,\mu)\}] \\ &= \{\mu\bar{\gamma}(\mu) - \bar{\omega}(\mu)\}\sum_{m=1}^{n}\{\bar{T}(m,\mu) - \bar{T}(m-1,\mu)\} \end{aligned}$$

and using (4) and (11). This concludes the proof. □

4. HEURISTIC I—DYNAMIC POLICY DEVELOPMENT FROM THE MARKOVIAN CLASS

We use the notation $\{t(\cdot), \mu^*\}$ to denote a policy which chooses its *first* allocated service according to the map $t(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$ and which makes all subsequent decisions according to optimal Markovian policy μ^*. In a sense to be made precise below our first heuristic is given by a map $\bar{t}(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$ enjoying the property that $\{\bar{t}(\cdot), \mu^*\}$ achieves the maximum expected number of successful task completions over an infinite horizon among the class of policies $\{t(\cdot), \mu^*\}$, uniformly over all initial states of the system. We may assume without loss of generality in what follows that the system is nonempty at time zero.

Developing the notation of Section 3 we write $C(n, T; t, \mu)$ for the expected number of successful task completions during the period $[0, T)$ under the policy (t, μ) whose first allocated service is t and which makes all subsequent decisions according to Markov policy μ when n is the system state at time 0. For definiteness, we suppose here and hereafter that $T \geq t$.

LEMMA 2: The difference between the expected number of successful task completions achieved over an infinite horizon by policies (t, μ) and μ from initial state $n \in \mathbb{Z}^+$ is given by

$$\lim_{T\to\infty} \{C(n, T; t, \mu) - C(n, T; \mu)\} = \gamma(t) + \mathbb{E}[b\{X(t|n) + Y(t), \mu\}] - b(n, \mu) - t\bar{\omega}(\mu), \quad (13)$$

where in (13), $X(t|n) \sim \text{Bin}(n-1, e^{-\theta t})$ and $Y(t) \sim \text{Poisson}\{\lambda\theta^{-1}(1 - e^{-\theta t})\}$ are independent random variables.

PROOF: If the first allocated service is t and the initial system state is $n \in \mathbb{Z}^+$ then by conditioning on the system state at time t we obtain

$$C(n, T; t, \mu) = \gamma(t) + \mathbb{E}[C\{X(t|n) + Y(t), T - t; \mu\}], \quad (14)$$

where $X(t|n)$ and $Y(t)$ are as in the statement of the lemma. We now expand (14) by writing

$$\begin{aligned} C(n, T; t, \mu) - C(n, T; \mu) &= \gamma(t) \\ &+ \mathbb{E}[C\{X(t|n) + Y(t), T - t; \mu\} - C(0, T - t; \mu)] \\ &- \{C(0, T; \mu) - C(0, T - t; \mu)\} \\ &- \{C(n, T; \mu) - C(0, T; \mu)\}. \end{aligned} \quad (15)$$

We now take the limit $T \to \infty$ on both sides of (15). First note that it is easily deduced from Lemma 1 that the bias function satisfies

$$0 \leq b(n, \mu) \leq n\bar{\gamma}(\mu), \quad n \in \mathbb{N}. \quad (16)$$

Further, it is straightforward that

$$C(n, T; \mu) - C(0, T; \mu) \leq \bar{C}(n, \mu) = \left\{1 - \frac{\bar{\omega}(\mu)}{\mu\bar{\gamma}(\mu)}\right\}^{-1} \times b(n, \mu), \quad T \in \mathbb{R}^+, \ n \in \mathbb{N}. \quad (17)$$

By utilizing (16) and (17) together with Definition 1, we infer that

$$\lim_{T\to\infty} \mathbb{E}[C\{X(t|n) + Y(t), T - t; \mu\} - C(0, T - t; \mu)] = \mathbb{E}[b\{X(t|n) + Y(t), \mu\}]. \quad (18)$$

By application of Blackwell's Theorem to the renewal process consisting of successive entries of the system (evolving under Markov policy μ) into the empty state, we conclude that

$$\lim_{T\to\infty} \{C(0, T; \mu) - C(0, T - t; \mu)\} = t\bar{\omega}(\mu). \quad (19)$$

The result now follows from (14), (15), (18), (19) and Definition 1. This concludes the proof. □

We now develop the map $\bar{t}(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$ by choosing

$$\bar{t}(n) = \arg\max_{t\geq 0}(\gamma(t) + \mathbb{E}[b\{X(t|n) + Y(t); \mu^*\}] - t\bar{\omega}(\mu^*)), \quad (20)$$

where $X(t|n)$ and $Y(t)$ are as in the statement of Lemma 2. If we suppose that the service requirement is absolutely continuous then the expression on the r.h.s. of (20) is certainly continuous. Moreover, since both $\gamma(t)$ and $\mathbb{E}[b\{X(t|n) + Y(t); \mu^*\}]$ can easily be shown to converge to finite limits as $t \to \infty$, this expression must be decreasing in t beyond some finite value T_n, say. Hence maxima must occur in $[0, T_n]$. It now follows from standard real analysis that the maximum in (20) must be achieved. For definiteness choose $\bar{t}(n)$ to be the largest value achieving the maximum. The following is an immediate consequence of Lemma 2 and the construction of $\bar{t}(\cdot)$.

THEOREM 1: The map $\bar{t}(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$ is such that

$$\lim_{T\to\infty} [C\{n, T; \bar{t}(n), \mu^*\} - C\{n, T; t(n), \mu^*\}] \geq 0, \quad n \in \mathbb{Z}^+,$$

for any choice of $t(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$.

Theorem 1 substantiates the claim made in the opening paragraph of this section, namely that $\{\bar{t}(\cdot), \mu^*\}$ maximizes the expected number of successful task completions among the policy class $\{t(\cdot), \mu^*\}$ uniformly over all initial states. Heuristic I is given by the map $\bar{t}(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$, namely it chooses allocated service $\bar{t}(n)$ when the queue length is n.

4.1. Numerical Examples

Table 1 shows values of the allocated service times under Heuristic I, namely $\bar{t}(n)$, for a range of problems in which service times are assumed to be drawn independently from a gamma distribution $\Gamma(r, \nu)$ with probability density function given in (5) earlier. For the time being, expand the notation to $\bar{t}(n, \lambda, \theta, r, \nu)$ to express dependence upon the key model parameters. We would expect these allocated service times to *decrease* in (i) n, the number of waiting tasks (more tasks at risk of loss from the system); (ii) λ, the rate of arriving tasks (more alternative tasks to service); and (iii) θ, the loss rate (long allocated services risk more severe losses). However, we would expect the allocated service times to *increase* as the times for successful service *increase* stochastically. Reassuringly, the results in Table 1 confirm these expectations.

Table 1. Values of the allocated service $\bar{t}(n)$ determined by Heuristic I for $n = 1$–6, 8, 10 for a range of problems with gamma $\Gamma(r, \nu)$ service times.

		$\bar{t}(n)$							
(λ, ν)	θ	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 8$	$n = 10$
					$r = 2$				
(0.25, 0.3)	0.1	7.98	7.37	6.95	6.68	6.51	6.39	6.26	6.19
	0.2	6.59	6.10	5.71	5.44	5.26	5.15	5.03	4.99
	0.3	5.67	5.26	4.90	4.63	4.43	4.31	4.19	4.15
(0.9, 0.3)	0.1	5.78	5.67	5.59	5.54	5.50	5.48	5.45	5.43
	0.2	4.86	4.74	4.65	4.58	4.53	4.50	4.45	4.43
	0.3	4.21	4.10	4.00	3.93	3.88	3.83	3.78	3.75
(0.9, 0.8)	0.1	3.23	3.03	2.89	2.79	2.73	2.68	2.65	2.62
	0.2	2.93	2.74	2.62	2.53	2.48	2.44	2.41	2.39
	0.3	2.70	2.53	2.41	2.33	2.28	2.24	2.19	2.17
					$r = 1$				
(0.25, 0.3)	0.1	4.73	3.78	3.23	2.92	2.73	2.61	2.47	2.39
	0.2	3.85	2.97	2.47	2.23	2.10	2.03	1.96	1.92
	0.3	3.32	2.51	2.03	1.82	1.73	1.69	1.65	1.63
(0.9, 0.3)	0.1	2.72	2.33	2.04	1.84	1.69	1.58	1.45	1.37
	0.2	2.30	1.94	1.68	1.51	1.40	1.32	1.23	1.18
	0.3	2.03	1.70	1.46	1.31	1.21	1.15	1.08	1.05
(0.9, 0.8)	0.1	1.95	1.64	1.44	1.30	1.20	1.14	1.04	0.99
	0.2	1.70	1.40	1.21	1.09	1.02	0.97	0.90	0.86
	0.3	1.54	1.25	1.07	0.97	0.90	0.86	0.81	0.78
					$r = 0.5$				
(0.25, 0.3)	0.1	3.13	2.09	1.60	1.36	1.22	1.13	1.03	0.97
	0.2	2.49	1.49	1.11	0.95	0.87	0.82	0.77	0.74
	0.3	2.13	1.18	0.87	0.76	0.70	0.68	0.64	0.63
(0.9, 0.3)	0.1	1.68	1.27	0.99	0.82	0.71	0.64	0.55	0.50
	0.2	1.38	0.97	0.74	0.61	0.53	0.48	0.43	0.40
	0.3	1.20	0.81	0.61	0.50	0.44	0.41	0.37	0.34
(0.9, 0.8)	0.1	1.33	1.01	0.81	0.69	0.61	0.56	0.49	0.44
	0.2	1.12	0.80	0.62	0.53	0.47	0.43	0.38	0.35
	0.3	1.00	0.68	0.52	0.44	0.39	0.36	0.32	0.30
					$r = 0.25$				
(0.25, 0.3)	0.1	2.22	1.21	0.85	0.68	0.59	0.53	0.47	0.43
	0.2	1.73	0.79	0.54	0.44	0.39	0.36	0.33	0.31
	0.3	1.47	0.60	0.41	0.34	0.31	0.29	0.27	0.26
(0.9, 0.3)	0.1	1.17	0.78	0.56	0.44	0.37	0.32	0.27	0.24
	0.2	0.93	0.55	0.38	0.30	0.25	0.22	0.19	0.17
	0.3	0.80	0.44	0.30	0.23	0.20	0.18	0.16	0.14
(0.9, 0.8)	0.1	0.97	0.66	0.49	0.40	0.34	0.30	0.25	0.22
	0.2	0.80	0.49	0.35	0.28	0.24	0.21	0.18	0.16
	0.3	0.70	0.39	0.27	0.22	0.19	0.17	0.15	0.13

Formally, we conjecture that $\bar{t}(n, \lambda, \theta, r, \nu)$ is *decreasing* in n, λ, θ and ν but is *increasing* in r. In all cases, the nonspecified parameters are assumed to be held constant. Sadly, the complexity of the r.h.s. of (20) has rendered it impossible to confirm any of these conjectures mathematically.

One case where we can get a partial result and some insight concerns the nature of the n-dependence of $\bar{t}(n, \lambda, \theta, r, \nu)$ when the loss rate θ is small. The only term on the r.h.s. of (20) which depends upon n is the second one involving the bias function. It is straightforward to show that for $n \geq 1$,

$$\frac{\partial}{\partial t}\mathbb{E}[b\{X(t|n) + Y(t); \mu^*\}] = \lambda\mathbb{E}[b\{n + \bar{Y}(t); \mu^*\} - b\{n - 1 + \bar{Y}(t); \mu^*\}] + O(\theta), \quad (21)$$

where $\bar{Y}(t) \sim \text{Poisson}(\lambda t)$ and $O(\theta)$ denotes a quantity which, when divided by θ, remains bounded in the limit $\theta \to 0$. From the expression for the bias given in Lemma 1, it is straightforward to show that $b(n, \mu)$ is increasing and concave in n, for fixed μ. It follows that the first term on the r.h.s. of (21) is both decreasing in t (for fixed n) and decreasing in n (for fixed t). This in turn suggests via straightforward analysis that we have $\bar{t}(n + 1, \lambda, \theta, r, \nu) \leq \bar{t}(n, \lambda, \theta, r, \nu)$ when θ is small enough.

Contracting the notation again, we finally note from Table 1 that $\bar{t}(n)$ appears to be convex in n, with significant reductions in allocated service as the queue length increases through small values (1,2,3) but with fairly rapid subsequent convergence to a limit.

5. HEURISTIC II—POLICY IMPROVEMENT FROM THE STATIC CLASS

By extension of the notation established in Section 4, we use $\{t(\cdot), [\tau]\}$ to denote a policy which chooses its *first* allocated service according to the map $t(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$, with all subsequent allocated services equal to $\tau \in \mathbb{R}^+$. In a sense to be made precise, our second heuristic will be given by a map $\hat{t}(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$ enjoying the property that $\{\hat{t}(\cdot), [(\mu^*)^{-1}]\}$ achieves the maximum expected number of successful task completions over an infinite horizon among the class of policies $\{t(\cdot), [(\mu^*)^{-1}]\}$.

5.1. Comments

1. Gaver et al. [6] gave numerical evidence that a policy which gives an allocated service of $(\mu^*)^{-1}$ to each task comes close to achieving maximum throughput in the static class of policies.
2. What we are in fact achieving in this section is the development of a strongly performing dynamic policy for service allocation by the application of a single DP policy improvement step applied to a strongly performing static policy. This is broadly in the spirit of Krishnan [12] who discussed the optimal routing of incoming customers to parallel queues.

To develop our analysis, we shall require the following notations:

- for T, t, $\tau \in \mathbb{R}^+$ and $n \in \mathbb{N}$ we write $C(n, T; t, [\tau])$ for the expected number of successful task completions during the period $[0, T)$ under the policy $(t, [\tau])$ when n is the system state at time 0;
- for T, $\tau \in \mathbb{R}^+$ and $n \in \mathbb{N}$ we write $C(n, T; [\tau])$ for the expected number of successful task completions during $[0, T)$ under the static policy $[\tau]$ which gives allocated service τ to all tasks when n is the system state at time 0;
- for $\tau \in \mathbb{R}^+$ and $n \in \mathbb{Z}^+$ we write $\bar{C}(n, [\tau])$ for the expected number of successful task completions under static policy $[\tau]$ from 0 up to the time at which the system empties for the first time when n is the initial system state;
- for $\tau \in \mathbb{R}^+$ and $n \in \mathbb{Z}^+$ we write $\bar{T}(n, [\tau])$ for the expected time it takes the system to empty for the first time from initial state n under static policy $[\tau]$;
- for $\tau \in \mathbb{R}^+$ we write $\bar{\omega}([\tau])$ for the throughput achieved under static policy $[\tau]$. Utilizing results of Baccelli et al. [1] we have that

$$\bar{\omega}([\tau]) = \lambda\gamma(\tau)\left(\sum_{n=0}^{\infty}(\lambda\tau)^n \times \left[\prod_{k=1}^{n}\{(1 - e^{-k\theta\tau})(k\theta\tau)^{-1}\}\right]\right)\left(1 + \sum_{n=0}^{\infty}(\lambda\tau)^n \times \left[\prod_{k=1}^{n}\{(1 - e^{-k\theta\tau})(k\theta\tau)^{-1}\}\right]\right)^{-1}. \quad (22)$$

5.2. Comment

For the gamma service time examples of (5), we find that, for any fixed τ, the throughput $\bar{\omega}([\tau])$ is increasing in ν, λ and decreasing in r, θ. Further, the quantity $\bar{\omega}([\tau])/\lambda$ is decreasing in λ. In all cases, the nonspecified parameters are assumed to be held constant. Note that these properties correspond exactly with those of throughput $\bar{\omega}(\mu)$ described in Section 2.

As with the class of Markov policies, we are able to define a bias function $b(\cdot, [\tau])$ for the static class, guaranteed finite whenever $\theta > 0$.

DEFINITION 2: The bias function $b(\cdot, [\tau]) : \mathbb{N} \to \mathbb{R}$ is defined by

$$b(n, [\tau]) = \lim_{T\to\infty} \{C(n, T; [\tau]) - C(0, T; [\tau])\}, \tag{23}$$

where the limit in (23) is guaranteed to exist and be finite for any $n \in \mathbb{N}$.

The following result makes use of the fact that the system regenerates upon every entry into the empty state [17].

LEMMA 3: The bias $b(n, [\tau])$ is given by

$$b(n, [\tau]) = \bar{C}(n, [\tau]) - \bar{\omega}([\tau])\bar{T}(n, [\tau]), \quad n \in \mathbb{Z}^+,$$

and is zero when $n = 0$.

In contrast to the account in Section 3, a closed form expression is not available for the bias function of a static policy. We shall now develop a numerical approach to its computation. First, we state a preparatory lemma.

LEMMA 4: The quantities $\bar{C}(n, [\tau])$ and $\bar{T}(n, [\tau])$ are such that

$$\bar{C}(n, [\tau]) = \tau^{-1}\gamma(\tau)\bar{T}(n, [\tau]) \le \gamma(\tau)\{n + \lambda\theta^{-1}e^{\lambda\theta^{-1}}\}, \quad n \in \mathbb{Z}^+. \tag{24}$$

PROOF: The equality in (24) is trivial. Hence it is enough to show that

$$\bar{T}(n, [\tau]) \le \tau\{n + \lambda\theta^{-1}e^{\lambda\theta^{-1}}\}, \quad n \in \mathbb{Z}^+. \tag{25}$$

Consider the system evolving under policy $[\tau]$ from initial state $n \in \mathbb{Z}^+$. Write $\tilde{T}(n, [\tau])$ for the time at which the system empties for the first time. This is a random variable whose expectation is $\bar{T}(n, [\tau])$. It is not difficult to see that

$$\tilde{T}(n, [\tau]) \le_{\mathrm{ST}} \tau\left(n + \sum_{m=1}^{\infty} Y_m I\left[\bigcap_{r=1}^{m-1}\{Y_r > 0\}\right]\right), \quad n \in \mathbb{Z}^+, \tag{26}$$

where in (26), $\le_{\mathrm{ST}}$ denotes stochastic ordering, $I[\cdot]$ is the indicator function and $\{Y_m, m \in \mathbb{Z}^+\}$ is a collection of independent and identically distributed Poisson random variables with mean $\lambda\theta^{-1}$.

To understand (26), we develop stochastic bounds on the times taken for successive attempts to empty the system under policy $[\tau]$. The first attempt processes the n tasks present initially, and is of duration no greater than $n\tau$. This attempt will be successful if the number of tasks present at its conclusion is zero. However, this number is stochastically bounded above by $Y_1 \sim \text{Poisson}(\lambda\theta^{-1})$ as is clear from comments in the paragraph following (2). Repeat the above with Y_1 the number of tasks present initially, and so on. The interpretation of Y_m is as a stochastic bound on the number of tasks present following the mth attempt to empty the system. Plainly

$$\tilde{T}(n, [\tau]) \le_{\mathrm{ST}} \tau\left(n + \sum_{m=1}^{N} Y_m\right), \quad n \in \mathbb{Z}^+, \tag{27}$$

where

$$N = \min(r; Y_r = 0). \tag{28}$$

Clearly (26) and (27) are equivalent. It is trivial to show that (25) is obtained by taking expectations on both sides of (26) while exploiting the independence of the Y_ms. This concludes the proof. □

In order to develop our numerical approach to the determination of key quantities, we write

$$\begin{aligned} P_{nm}(t) &= P\{X(t|n) + Y(t) = m\} \\ &= \sum_{r=0}^{\min(n-1,m)} \frac{(n-1)!}{r!(n-1-r)!(m-r)!}\lambda^{m-r}\theta^{r-m}e^{-r\theta t} \\ &\quad \times (1 - e^{-\theta t})^{n+m-1-2r}\exp\{\lambda\theta^{-1}(1 - e^{-\theta t})\}, \end{aligned} \tag{29}$$

where $t \in \mathbb{R}^+$, $n \in \mathbb{Z}^+$ and $m \in \mathbb{N}$ in (27). Utilization of a simple conditioning argument yields the recursion

$$\bar{C}(n, [\tau]) = \gamma(\tau) + \sum_{m=1}^{\infty} P_{nm}(\tau)\bar{C}(m, [\tau]), \quad n \in \mathbb{Z}^+. \tag{30}$$

To compute the function $\bar{C}(\cdot, [\tau]) : \mathbb{Z}^+ \to \mathbb{R}^+$, we develop the sequence of functions $\{f^{(r)}(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+, r \in \mathbb{Z}^+\}$ defined recursively by

$$f^{(1)}(n) = 0, \quad n \in \mathbb{Z}^+, \tag{31}$$

and

$$f^{(r+1)}(n) = \gamma(\tau) + \sum_{m=1}^{\infty} P_{nm}(\tau)f^{(r)}(m), \quad n, r \in \mathbb{Z}^+. \tag{32}$$

LEMMA 5: The function sequence $\{f^{(r)}(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+, r \in \mathbb{Z}^+\}$ satisfies

$$\lim_{r\to\infty} f^{(r)}(n) = \bar{C}(n, [\tau]), \quad n \in \mathbb{Z}^+. \tag{33}$$

PROOF: A simple proof by induction uses (31) and (32) to establish that the sequence $\{f^{(r)}(n), r \in \mathbb{Z}^+\}$ is nondecreasing for each $n \in \mathbb{Z}^+$. A further induction, together with (30) and (31), establishes that

$$f^{(r)}(n) \le \bar{C}(n, [\tau]), \quad n \in \mathbb{Z}^+. \tag{34}$$

It must then follow that the limit on the l.h.s. of (33) must always exist.

We now define

$$\Delta^{(r)}(n) = \bar{C}(n, [\tau]) - f^{(r)}(n), \quad n, r \in \mathbb{Z}^+. \tag{35}$$

By (34) the $\Delta^{(r)}(n)$ are all non-negative, and by (30)–(32) satisfy the recursion

$$\Delta^{(1)}(n) = \bar{C}(n, [\tau]), \quad n \in \mathbb{Z}^+, \tag{36}$$

and

$$\Delta^{(r+1)}(n) = \sum_{m=1}^{\infty} P_{nm}(\tau)\Delta^{(r)}(n), \quad n, r \in \mathbb{Z}^+. \tag{37}$$

Denote by $\{X(t), t \in \mathbb{N}\}$ a Markov chain with state space $\mathbb{N}$ evolving according to the transition matrix $\mathbf{P}'(\tau)$ where

$$P'_{nm}(\tau) = \begin{cases} \delta_{0m}, & n = 0, \ m \in \mathbb{N}, \\ P_{nm}(\tau), & n \in \mathbb{Z}^+, \ m \in \mathbb{N}, \end{cases} \tag{38}$$

with δ in (38) the Kronecker delta. Plainly, 0 is an absorbing state for the chain, with all other states transient. It is trivial to show that (36)–(38) together with Lemma 4 yields

$$\begin{aligned} \Delta^{(r+1)}(n) &= \mathbb{E}(C\{X(r), [\tau]\}|X(0) = n) \\ &\le \gamma(\tau)\mathbb{E}\{X(r) + \lambda\theta^{-1}e^{\lambda\theta^{-1}} \\ &\quad \times I[X(r) > 0]|X(0) = n\} \\ &\to 0, \quad r \to \infty, \ n \in \mathbb{Z}^+, \end{aligned} \tag{39}$$

where the limit in (39) is established by a simple modification of the argument of the proof of Lemma 4. The statement in (33) is an immediate consequence of (39). This concludes the proof. □

From Lemmas 3 and 4 we observe that

$$b(n, [\tau]) = [1 - \bar{\omega}([\tau])\tau\{\gamma(\tau)\}^{-1}]\bar{C}(n, [\tau]), \quad n \in \mathbb{Z}^+. \tag{40}$$

Hence the recursive scheme of Lemma 5 together with the expression in (22) enable the bias function $b(\cdot, [\tau])$ to be computed.

We now describe how to deploy the bias function to affect the DP policy improvement step which is the basis for the design of Heuristic II. The proof of Lemma 6 is similar to that of Lemma 2 and is omitted.

LEMMA 6: The difference between the expected number of successful task completions achieved over an infinite horizon by policies $(t, [\tau])$ and $[\tau]$ from initial state $n \in \mathbb{Z}^+$ is given by

$$\begin{aligned} \lim_{T\to\infty} \{C(n, T; t, [\tau]) - C(n, T; [\tau])\} &= \gamma(t) \\ + \mathbb{E}[b\{X(t|n) + Y(t)\}, [\tau]] &- b(n, [\tau]) - t\bar{\omega}([\tau]), \end{aligned} \tag{41}$$

where in (41), $X(t|n) \sim \text{Bin}(n-1, e^{-\theta t})$ and $Y(t) \sim \text{Poisson}\{\lambda\theta^{-1}(1 - e^{-\theta t})\}$ are independent random variables.

Following Lemma 6, we now develop the map $\hat{t}(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$ by choosing

$$\begin{aligned} \hat{t}(n) = \arg\max_{t\ge 0}(\gamma(t) + \mathbb{E}[b\{X(t|n) + Y(t), [(\mu^*)^{-1}]] \\ - t\bar{\omega}([(\mu^*)^{-1}])), \end{aligned} \tag{42}$$

with $X(t|n)$ and $Y(t)$ as mentioned earlier. For reasons similar to those following (20), the maximum in (42) must be achieved when service requirements are absolutely continuous. Theorem 2 is an immediate consequence of Lemma 6 and the construction of the map $\hat{t}(\cdot)$.

THEOREM 2: The map $\hat{t}(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$ is such that

$$\begin{aligned} \lim_{T\to\infty} [C\{n, T; \hat{t}(n), [(\mu^*)^{-1}]\} \\ - C\{n, T; t(n), [(\mu^*)^{-1}]\}] \ge 0, \quad n \in \mathbb{Z}^+, \end{aligned}$$

for any choice of $t(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$.

Theorem 2 substantiates the claim made in the opening paragraph of this section, namely that $\{\hat{t}(\cdot), [(\mu^*)^{-1}]\}$ maximizes the expected number of successful task completions among the policy class $\{t(\cdot), [(\mu^*)^{-1}]\}$ uniformly over all initial states. Corollary 1 now follows. In its statement we use $C\{n, T; \hat{t}(\cdot)\}$ to denote the expected number of successful task completions during the period $[0, T)$ under the stationary policy $\hat{t}(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$ when n is the system state at time 0.

COROLLARY 1: The map $\hat{t}(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$ is such that

$$\lim_{T\to\infty} [C\{n, T; \hat{t}(\cdot)\} - C\{n, T; [(\mu^*)^{-1}]\}] \ge 0, \quad n \in \mathbb{Z}^+.$$

PROOF: First, by substitution of the map $t(n) = (\mu^*)^{-1}$, $n \in \mathbb{Z}^+$, into the statement of Theorem 2, we have that

$$\lim_{T\to\infty} [C\{n, T; \hat{t}(n), [(\mu^*)^{-1}]\} - C\{n, T; [(\mu^*)^{-1}]\}] \ge 0, \quad n \in \mathbb{Z}^+,$$

and hence that the policy $\{\hat{t}(\cdot), [(\mu^*)^{-1}]\}$ achieves a higher expected number of successful task completions than does the static policy $[(\mu^*)^{-1}]$. We now use standard arguments to infer that a policy whose first r decisions are made according to the map $\hat{t}(\cdot) : \mathbb{Z}^+ \rightarrow \mathbb{R}^+$, with all remaining decisions made according to $[(\mu^*)^{-1}]$ will outperform $[(\mu^*)^{-1}]$ itself. To obtain the result we now take the limit $r \rightarrow \infty$. This concludes the proof. □

The map $\hat{t}(\cdot) : \mathbb{Z}^+ \rightarrow \mathbb{R}^+$ determines Heuristic II, namely that allocated service $\hat{t}(n)$ is chosen when the queue length is n. We see from Corollary 1 that Heuristic II's throughput is guaranteed to be at least as large as that of the static policy $[(\mu^*)^{-1}]$ from which it was developed.

5.3. Numerical Examples

In Table 2 you find values of the map $\hat{t}(n)$ for the same range of problems as is considered in Table 1 at the conclusion

Table 2. Values of the allocated service $\hat{t}(n)$ determined by Heuristic II for $n = 1$–6, 8, 10 for a range of problems with gamma $\Gamma(r, \nu)$ service times.

		$\hat{t}(n)$							
(λ, ν)	θ	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 8$	$n = 10$
					$r = 2$				
(0.25, 0.3)	0.1	6.79	5.99	5.52	5.26	5.10	5.00	4.87	4.79
	0.2	5.52	4.82	4.38	4.16	4.05	4.00	3.93	3.89
	0.3	4.79	4.18	3.73	3.50	3.40	3.37	3.34	3.32
(0.9, 0.3)	0.1	5.07	5.03	5.01	5.00	4.99	4.98	4.98	4.97
	0.2	4.04	3.96	3.91	3.88	3.86	3.85	3.83	3.83
	0.3	3.40	3.31	3.24	3.20	3.18	3.16	3.14	3.13
(0.9, 0.8)	0.1	2.67	2.41	2.28	2.20	2.16	2.13	2.10	2.08
	0.2	2.41	2.18	2.05	1.98	1.94	1.91	1.88	1.86
	0.3	2.23	2.00	1.88	1.81	1.77	1.74	1.71	1.70
					$r = 1$				
(0.25, 0.3)	0.1	4.01	2.89	2.40	2.15	2.01	1.92	1.81	1.74
	0.2	3.29	2.23	1.83	1.68	1.60	1.54	1.47	1.44
	0.3	2.86	1.86	1.51	1.42	1.37	1.33	1.28	1.25
(0.9, 0.3)	0.1	2.10	1.62	1.32	1.14	1.03	0.96	0.87	0.82
	0.2	1.81	1.36	1.10	0.97	0.89	0.84	0.78	0.75
	0.3	1.62	1.19	0.97	0.86	0.80	0.76	0.71	0.68
(0.9, 0.8)	0.1	1.60	1.25	1.04	0.92	0.84	0.79	0.72	0.68
	0.2	1.40	1.05	0.87	0.78	0.72	0.68	0.63	0.60
	0.3	1.28	0.94	0.77	0.69	0.64	0.61	0.57	0.55
					$r = 0.5$				
(0.25, 0.3)	0.1	2.79	1.69	1.28	1.09	0.98	0.92	0.83	0.79
	0.2	2.23	1.18	0.90	0.79	0.73	0.69	0.64	0.61
	0.3	1.92	0.92	0.72	0.64	0.60	0.57	0.54	0.53
(0.9, 0.3)	0.1	1.41	0.97	0.72	0.59	0.51	0.46	0.40	0.37
	0.2	1.17	0.74	0.54	0.45	0.40	0.36	0.32	0.30
	0.3	1.03	0.61	0.45	0.38	0.34	0.31	0.28	0.27
(0.9, 0.8)	0.1	1.17	0.83	0.65	0.54	0.48	0.43	0.38	0.35
	0.2	0.98	0.64	0.49	0.41	0.36	0.33	0.30	0.28
	0.3	0.88	0.54	0.41	0.34	0.31	0.29	0.26	0.24
					$r = 0.25$				
(0.25, 0.3)	0.1	2.07	1.06	0.74	0.60	0.52	0.47	0.42	0.38
	0.2	1.61	0.68	0.48	0.40	0.35	0.33	0.30	0.28
	0.3	1.37	0.51	0.36	0.31	0.28	0.27	0.24	0.23
(0.9, 0.3)	0.1	1.06	0.66	0.47	0.36	0.31	0.27	0.23	0.20
	0.2	0.84	0.47	0.32	0.25	0.21	0.19	0.16	0.15
	0.3	0.73	0.37	0.25	0.20	0.17	0.16	0.14	0.13
(0.9, 0.8)	0.1	0.90	0.58	0.43	0.34	0.29	0.26	0.22	0.19
	0.2	0.74	0.42	0.30	0.24	0.20	0.18	0.16	0.14
	0.3	0.65	0.34	0.24	0.19	0.16	0.15	0.13	0.12

of Section 4. The map $\hat{t}(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$ has the same monotone characteristics as the map $\bar{t}(\cdot) : \mathbb{Z}^+ \to \mathbb{R}^+$ described earlier. In all cases we have $\hat{t}(n) \leq \bar{t}(n)$, namely that Heuristic II allocates smaller service times than does Heuristic I for any given queue length. To see why this might be so, compare the expressions on the r.h.s. of (20) and (42) for some fixed $n \geq 1$. The first term is $\gamma(t)$ in both cases. Suppose now that the bias terms can be assumed to be uniformly close to each other and hence that the comparison is dominated by the final linear terms. We have found that in all cases studied the throughput $\bar{\omega}([(\mu^*)^{-1}])$ which is to be found in (42) is considerably larger than the quantity $\bar{\omega}(\mu^*)$ found in (20). Hence a larger negative penalty is paid via the final term for increasing t in (42) than in (20). It is therefore not surprising that the (largest) maximum is to be found at a smaller value in (42) than in (20) and hence that $\hat{t}(n) \leq \bar{t}(n)$.

Table 3. Estimates of the percentage of arriving tasks successfully served under five heuristics for a range of problems with gamma $\Gamma(r, \nu)$ service times.

(λ, ν)	θ	μ^*	HeurI	$[(\mu^*)^{-1}]$	HeurII	DP
$r = 2$						
(0.25, 0.3)	0.1	0.1554	0.2157 (0.0008)	0.2182 (0.0009)	0.2307 (0.0018)	0.2271
	0.2	0.1050	0.1505 (0.0016)	0.1508 (0.0016)	0.1552 (0.0017)	0.1560
	0.3	0.0772	0.1118 (0.0013)	0.1142 (0.0015)	0.1148 (0.0015)	0.1153
(0.9, 0.3)	0.1	0.0617	0.0735 (0.0006)	0.0678 (0.0006)	0.0738 (0.0006)	0.0753
	0.2	0.0469	0.0580 (0.0006)	0.0564 (0.0006)	0.0590 (0.0005)	0.0594
	0.3	0.0368	0.0466 (0.0005)	0.0464 (0.0005)	0.0472 (0.0005)	0.0483
(0.9, 0.8)	0.1	0.1775	0.2256 (0.0008)	0.2279 (0.0010)	0.2338 (0.0008)	0.2337
	0.2	0.1467	0.1942 (0.0008)	0.1964 (0.0009)	0.2019 (0.0009)	0.2024
	0.3	0.1244	0.1687 (0.0008)	0.1703 (0.0009)	0.1748 (0.0009)	0.1757
$r = 1$						
(0.25, 0.3)	0.1	0.3292	0.4493 (0.0025)	0.4521 (0.0024)	0.4582 (0.0028)	0.4584
	0.2	0.2679	0.3656 (0.0024)	0.3610 (0.0024)	0.3703 (0.0026)	0.3705
	0.3	0.2298	0.3123 (0.0022)	0.3128 (0.0025)	0.3173 (0.0025)	0.3163
(0.9, 0.3)	0.1	0.1904	0.2309 (0.0009)	0.2391 (0.0010)	0.2432 (0.0010)	0.2432
	0.2	0.1635	0.2055 (0.0009)	0.2096 (0.0009)	0.2139 (0.0009)	0.2146
	0.3	0.1448	0.1873 (0.0009)	0.1891 (0.0010)	0.1924 (0.0009)	0.1930
(0.9, 0.8)	0.1	0.3580	0.4691 (0.0012)	0.4763 (0.0015)	0.4879 (0.0013)	0.4876
	0.2	0.3171	0.4254 (0.0011)	0.4267 (0.0014)	0.4344 (0.0012)	0.4361
	0.3	0.2887	0.3884 (0.0011)	0.3911 (0.0013)	0.3973 (0.0012)	0.3982
$r = 0.5$						
(0.25, 0.3)	0.1	0.5214	0.6530 (0.0032)	0.6472 (0.0034)	0.6556 (0.0035)	0.6598
	0.2	0.4667	0.5823 (0.0034)	0.5794 (0.0034)	0.5842 (0.0034)	0.5869
	0.3	0.4315	0.5396 (0.0031)	0.5363 (0.0033)	0.5406 (0.0032)	0.5403
(0.9, 0.3)	0.1	0.3901	0.4848 (0.0014)	0.4806 (0.0015)	0.4924 (0.0015)	0.4924
	0.2	0.3558	0.4429 (0.0013)	0.4381 (0.0014)	0.4469 (0.0014)	0.4512
	0.3	0.3331	0.4194 (0.0014)	0.4142 (0.0015)	0.4235 (0.0015)	0.4228
(0.9, 0.8)	0.1	0.5515	0.6912 (0.0016)	0.6831 (0.0017)	0.6960 (0.0017)	0.6966
	0.2	0.5116	0.6441 (0.0017)	0.6359 (0.0018)	0.6474 (0.0017)	0.6482
	0.3	0.4846	0.6112 (0.0018)	0.6037 (0.0019)	0.6136 (0.0018)	0.6139
$r = 0.25$						
(0.25, 0.3)	0.1	0.6859	0.7850 (0.0038)	0.7792 (0.0038)	0.7860 (0.0038)	0.7951
	0.2	0.6460	0.7463 (0.0038)	0.7421 (0.0038)	0.7476 (0.0038)	0.7463
	0.3	0.6204	0.7123 (0.0039)	0.7079 (0.0039)	0.7120 (0.0039)	0.7155
(0.9, 0.3)	0.1	0.5909	0.6859 (0.0018)	0.6765 (0.0019)	0.6885 (0.0019)	0.6904
	0.2	0.5601	0.6543 (0.0020)	0.6448 (0.0021)	0.6557 (0.0021)	0.6555
	0.3	0.5402	0.6297 (0.0019)	0.6216 (0.0019)	0.6299 (0.0019)	0.6320
(0.9, 0.8)	0.1	0.7111	0.8200 (0.0019)	0.8108 (0.0020)	0.8208 (0.0019)	0.8242
	0.2	0.6800	0.7879 (0.0020)	0.7783 (0.0020)	0.7878 (0.0021)	0.7891
	0.3	0.6593	0.7644 (0.0019)	0.7564 (0.0019)	0.7647 (0.0019)	0.7656

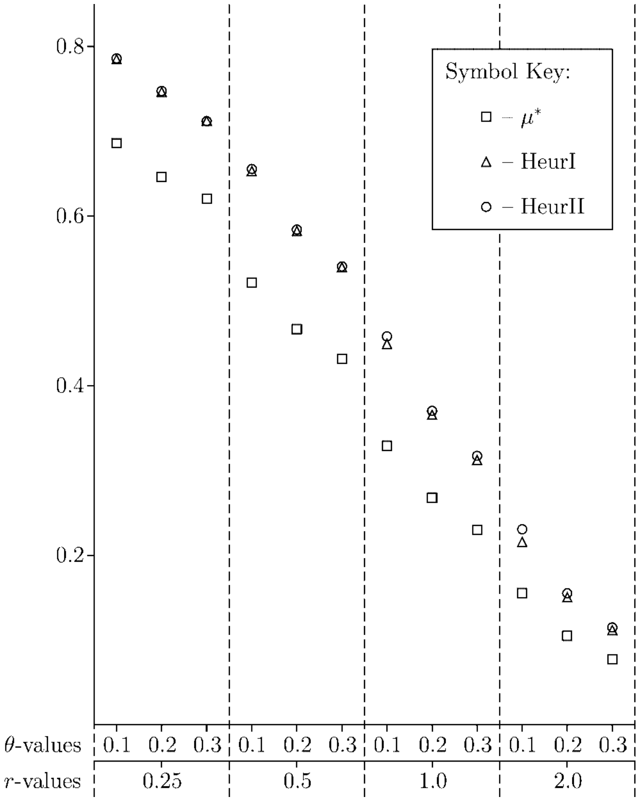

Figure 1. The percentage of arriving tasks successfully served under three heuristics for the case $(\lambda, \nu) = (0.25, 0.3)$.

6. NUMERICAL STUDY

Table 3 shows estimates of the percentage of arriving tasks successfully served (namely, the throughput divided by the arrival rate) under five heuristics for a range of problems with gamma $\Gamma(r, \nu)$ service times. The problems studied are the same as those considered in Sections 4 and 5 and are such that there is a wide range (from under 5% to more than 80%) in the percentage of successfully served tasks. The five heuristics studied are (reading from left to right in the tables):

(i) the optimal Markovian policy μ^* discussed in Section 3;
(ii) Heuristic I as developed in Section 4;
(iii) the static policy $[(\mu^*)^{-1}]$ which allocates time $(\mu^*)^{-1}$ to all served tasks;
(iv) Heuristic II as developed in Section 5;
(v) a heuristic developed by the application of stochastic DP to a finite state/finite action/discrete time approximation of the problem, with a discrete time quantum set equal to $\delta = 0.002$.

In the case of (i) the figures quoted in the table were obtained by application of the formula in (4), while in (v) the methodology deployed was DP value iteration [17]. Please note that for the DP policy, a halving of the discrete time quantum to 0.001 did not improve the achieved throughput (to the accuracy reported in the tables) but greatly increased the computing time required. The results for (ii)–(iv) were obtained by Monte Carlo simulation. For the heuristics concerned, all estimates of the percentage of tasks successfully served are accompanied in the table by their standard errors (in brackets). The simulation study was designed to achieve sufficiently small standard errors to facilitate meaningful performance comparisons between the heuristics. To further assist the reader, the results for the optimal Markovian policy μ^* and for Heuristics I and II are displayed in Figures 1–3. The policy $[(\mu^*)^{-1}]$ and the DP heuristic were omitted from the figures because of their closeness in performance to Heuristics I and II, respectively. See comments later.

Before preceding to discussion of the results, please note that (arbitrarily good approximations to) Heuristics I and II may be computed efficiently. We observe that under these (and, indeed, any other stationary) service heuristics, the number of tasks present in the system and *not* in receipt of service is stochastically bounded above in steady state by the number present in an equivalent system, but for which no service at all is offered. However, the steady state distribution of

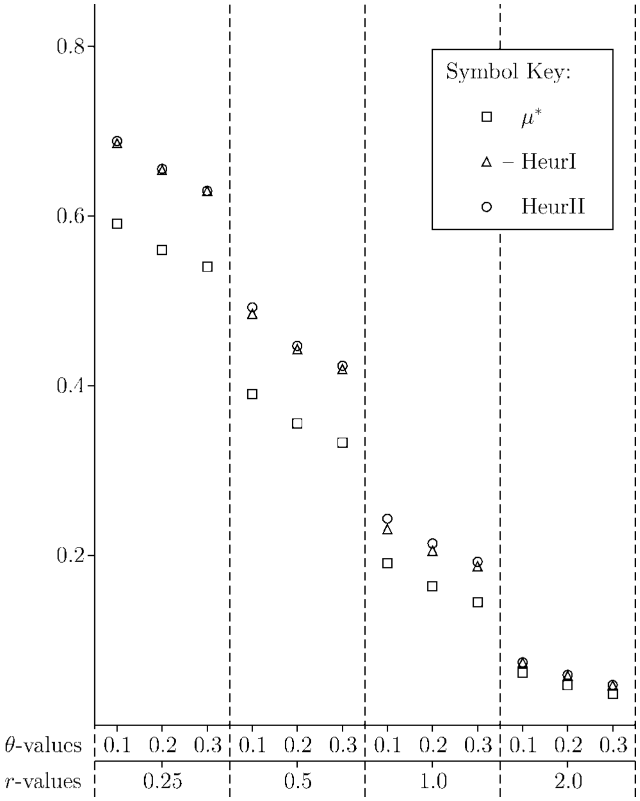

Figure 2. The percentage of arriving tasks successfully served under three heuristics for the case $(\lambda, \nu) = (0.9, 0.3)$.

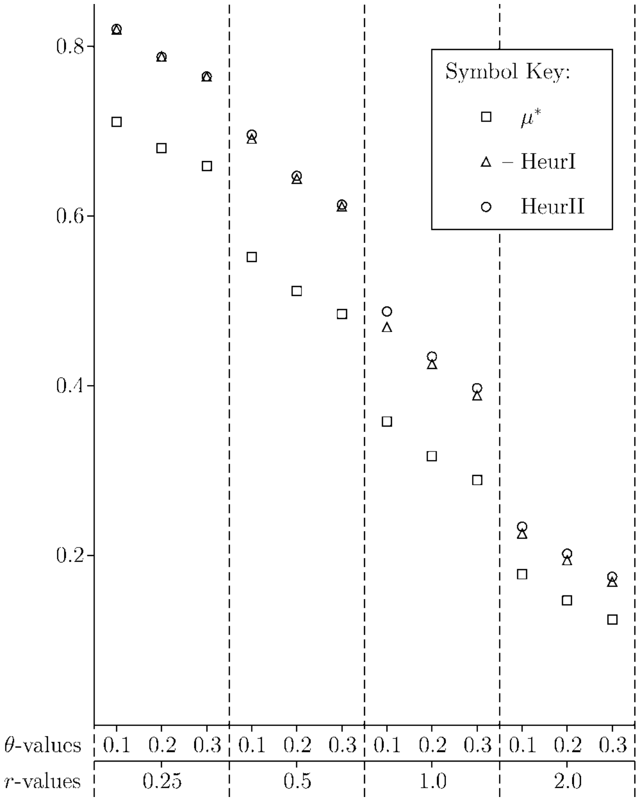

Figure 3. The percentage of arriving tasks successfully served under three heuristics for the case $(\lambda, \nu) = (0.9, 0.8)$.

the latter is well known to be Poisson $(\lambda\theta^{-1})$. We infer that the long run proportion of time for which the queue length is outside the range $0 \le n \le \bar{N}$, where

$$\bar{N} = \max(50, \lambda\theta^{-1}) + 3\sqrt{\max(50, \lambda\theta^{-1})}$$

is small (at most around 10^{-3}). Hence implementation of the maximizations in (20) and (42) over the range $1 \le n \le \bar{N}$ is perfectly adequate. For values of n above $\bar{N}$ we set the values of $\bar{t}(n)$ and $\hat{t}(n)$ equal to $\bar{t}(\bar{N})$ and $\hat{t}(\bar{N})$, respectively. In practice, both $\bar{t}(n)$ and $\hat{t}(n)$ achieve limiting values before n reaches $\bar{N}$. The computational ease with which the dynamic heuristics I and II may be developed (in practice, this takes just a few seconds on a standard PC) stands in sharp contrast to the computational burden of a full DP implementation as outlined in Section 2.

The evidence of Table 3 is that under all heuristics the percentage of successfully served tasks is decreasing in the exogenous arrival rate λ, the loss rate θ, and the mean (actual) service time $r\nu^{-1}$. That for given (λ, ν), the percentage of successfully served tasks is decreasing in r and θ is particularly clear from Figures 1–3. The optimal Markovian policy μ^* performs poorly. An approach to the allocation of service times which is both static (i.e., state independent) and, more crucially, random with a high degree of variability does not work well. Overall, using the performance measures in Table 3, Heuristic I effects an improvement of over 26% on the Markovian policy μ^*. This improvement is most marked in the low throughput/high mean service time cases. A notable feature of Table 3 is the strong performance of the static policy $[(\mu^*)^{-1}]$, reflecting the fact that while the allocated services chosen by strongly performing dynamic policies do vary with the queue length, the degree of variability is modest. Overall, the performances of Heuristic I and the static policy $[(\mu^*)^{-1}]$ are similar, with the dynamic heuristic slightly outperforming the static policy in high throughput instances. Heuristic II outperforms all of the above. That it should outperform static policy $[(\mu^*)^{-1}]$ is guaranteed by Corollary 1. While the performances of Heuristics I and II are fairly comparable in high throughput instances, Heuristic II is plainly the stronger in cases (e.g., when $r = 2$) for which the percentage of successfully served tasks is low. Overall, Heuristic II offers an improvement in performance of around 2% on average over the static policy $[(\mu^*)^{-1}]$, but the degree of improvement is much greater than this in low throughput instances where it rises to 9%. In all cases studied, a comparison of the performance of Heuristic II with that of the policy developed by stochastic DP yields the conclusion that the former is likely very close to optimal. In the cases considered, DP offers an improvement in performance over Heuristic II of just 0.3% on average.

ACKNOWLEDGMENT

Both authors gratefully acknowledge support received from the Engineering and Physical Science Research Council. For the first author, the source of funding was the grant GR/S45188/01, while for the second it was through a research studentship. They also acknowledge the thoughtful comments of Professors Don Gaver and Pat Jacobs of the Naval Postgraduate School and of an associate editor and two referees who led to a range of improvements to the paper. Finally, the authors thank Dr. Chris Kirkbride for his assistance in producing the figures.

REFERENCES

[1] F. Baccelli, P. Boyer, and G. Hebuterne, Single-server queues with impatient customers, Adv Appl Prob 16 (1984), 887–905.

[2] 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, Queueing Syst 51 (2005), 249–285.

[3] S.D. Deshmukh and S. Jain, Capacity design and service quality control in a queueing system, Oper Res 25 (1977), 651–661.

[4] B. Doytchinov, J.P. Lehoczky, and S. Shreve, Real-time queues in heavy traffic with earliest-deadline-first queue discipline, Ann Appl Probab 11 (2001), 332–378.

[5] O. Garnett, A. Mandelbaum, and M. Reiman, Designing a call center with impatient customers, Manuf Service Oper Mgmt 4 (2002), 208–227.

[6] D.P. Gaver, P.A. Jacobs, G. Samorodnitsky, and K.D. Glazebrook, Modeling and analysis of uncertain time-critical tasking problems, Nav Res Logist 53 (2006), 588–599.

[7] D.P. Gaver, P.A. Jacobs, and H. Sato, Assessing resource requirements for maritime domain awareness and protection (security). Working Paper, Department of Operations Research, Naval Postgraduate School, Monterey, CA, 2006.

[8] K.D. Glazebrook, Stochastic scheduling with due dates, Int J Syst Sci 14 (1983), 1259–1271.

[9] K.D. Glazebrook, P.S. Ansell, R.T. Dunn, and R.R. Lumley, On the optimal allocation of service to impatient tasks, J Appl Prob 41 (2004), 51–72.

[10] K.D. Glazebrook and E.L. Punton, Fixed-time schedules for the processing of jobs when service completions are not observable, Math Meth Oper Res 62 (2005), 77–97.

[11] J.M. Harrison and A. Zeevi, Dynamic scheduling of a multiclass queue in the Halfin-Whitt heavy traffic regime, Oper Res 52 (2004), 243–257.

[12] Z. Jiang, T.G. Lewis, and J.Y. Colin, Scheduling hard real-time constrained periodic tasks on multiple processors, J Syst Software 19 (1996), 102–118.

[13] K.R. Krishnan, "Joining the right queue: A Markov decision rule," in: Proceedings of the 28th IEEE Conference on Decision Control, 1987, pp. 1863–1868.

[14] J.P. Lehoczky, "Real-time queueing theory," in: Proceedings of the 17th IEEE Real-Time Systems Symposium, 1996, pp. 186–195.

[15] J.P. Lehoczky, "Real-time queueing network theory," in: Proceedings of the 18th IEEE Real-Time Systems Symposium, 1997, pp. 58–67.

[16] J.P. Lehoczky, Using real-time queueing theory to control lateness in real-time systems, Perform Eval Rev 25 (1997), 158–168.

[17] H.C. Tijms, Stochastic models: An algorithmic approach, Wiley, Chichester, 1994.