Meeting Soft Deadlines in Single- and Multi-Server Systems

Esa Hyytiä
Department of Computer Science
University of Iceland

Rhonda Righter
Department of Industrial Engineering
and Operations Research, UC Berkeley

Jorma Virtamo
Department of Communications
and Networking, Aalto University

***Abstract*—We consider single- and multi-server systems, where jobs have a maximum waiting time (deadline) defined, e.g., by a service level agreement. A fixed cost is associated with deadline violations and the task is to minimize the long-run cumulative costs. Job sizes (service durations) are observed upon arrival, and current queue backlogs are known. For a single FCFS server, the optimization task is to find the optimal admission policy that may reject a job upon arrival if admitting it would cause in future one or more deadlines to be violated (in expectation). For parallel FCFS servers, the policy must (i) either accept or reject a job upon arrival, and if accepted, (ii) assign it to one of the servers. We derive efficient deadline-aware policies in the MDP framework. For a single server, we obtain the optimal admission policy. For dispatching to parallel servers, we develop efficient heuristic admission and dispatching policies, whose performances are evaluated by means of numerical examples. Additionally, we give some exact closed-form results for heavy-traffic limits.**

***Index Terms*—admission control; task assignment; deadline; QoE; parallel processing; cloud computing; non-linear cost**

I. Introduction

Today's Internet is full of services, requiring rapid and timely responses. For example, for interactive applications such as the large-scale online services provided by Google, Facebook and Amazon, subsecond response times are a common objective [1]. In particular, tails of the service time are seen as one of the most crucial performance measures [2] affecting the *quality of experience* (QoE), i.e., how customers perceive the service, which then eventually translates to profits (or losses). Therefore, in this paper we assume jobs have deadlines for waiting times until the start of service. If this deadline is exceeded, a fixed cost is incurred. In particular, we assume a best effort loss system where a job missing its deadline may as well be discarded without any additional cost other than the cost due to the missed deadline.

In general, the possible controls to achieve SLA goals are (i) admission control, (ii) dispatching rules, (iii) scheduling within a server / data-center, (iv) job migration, and (v) control of service rate, e.g., via number of servers or speed scaling. We consider the first two: admission control and dispatching.

We first consider a single-server system modelled as an M/G/1-FCFS queue, where jobs have a maximum waiting time τ that should not be exceeded, or the fixed unit cost is incurred. The dynamic decision problem is to find the optimal admission policy that minimizes long run costs. Then we consider admission and dispatching to a set of parallel heterogeneous FCFS servers. The routing decision must be made upon arrival and is irrevocable, i.e., a job cannot be moved to another queue later. (Job reallocations can incur large overheads.)

Our main contributions are the following: (i) we derive new theoretical results for the M/G/1 queue subject to this non-linear cost structure (the so-called value function with respect to deadline violations) that enable efficient static and dynamic admission policies; (ii) we give exact closed-form expressions for the probability of deadline violations, the steady-state distribution, and the value function (at the heavy-traffic limit) in M/M/1; (iii) we devise an explicit method for solving the optimal admission policy for the M/G/1 queue; and (iv) we derive efficient heuristic admission and dispatching policies for systems of parallel servers. The performance of the obtained deadline-aware policies are further evaluated numerically, giving insight into this important cost structure.

The rest of the paper is organized as follows. Section II formally introduces the model and notation. In Section III, we analyze a single M/G/1 queue with respect to deadlines, and complement this with numerical examples in Section IV. In Section V, we apply our results to derive efficient dynamic dispatching policies for the original multiserver problem. Section VI concludes the paper.

A. Related work

Our work is different from past work in that our policies are dynamic, they minimize costs due to deadline violations, and they take into account the size of arriving jobs. Liu et al. [3] consider service-level-agreements (SLAs) in the context of e-commerce with a general class of SLAs comprising throughput, mean delay and (soft) deadlines. Web servers are modelled as a queueing network with generalized processor sharing (GPS) servers. The decision variables are the probabilities for static routing and the weights for the GPS scheduling. (Note that, e.g., Hadoop uses FCFS by default. This is often also the case in the context of supercomputing where concurrent multitasking may be impractical or even infeasible.) Saovapakhiran et al. [4] consider *average delay SLAs* in the context of cloud computing. The mean delays follow directly from Little's theorem, which simplifies their analysis considerably.

Stidham [5] considers the admission problem. The models closest to ours are the admission control to an GI/M/1 queue and to two parallel GI/M/1 queues. However, the cost structure is different: each admitted job gives a fixed reward, while costs are incurred according to a convex holding cost rate function of number of jobs in (each) server. Lehoczky [6] studies a single GI/M/1 queue with deadlines in heavy-traffic. Glazebrook et al. [7] consider a system of parallel (exponential) servers with abandonments. These works, however, are different as the admission control and routing (if any) are based on the number of jobs. Li and Glazebrook [8] consider a single server with jobs that leave the system if their waiting time exceeds a random class-dependent waiting time. The scheduling problem is to minimize the number of abandonments. In contrast, we know the deadline of each job and obtain a critical-backlog policy that depends also on the (exact) size of the arriving job.

Gupta and Harchol-Balter [9] study a system where a PS server, preceded by a FCFS queue, admits at most k jobs concurrently. The aim is to minimize the mean delay. The minimization of the mean delay, or some other related linear quantity such as slowdown, has been the typical objective also for routing (dispatching) problems (without the option to reject jobs). See, for example, [10], [11], [12], [13], [14], and the references therein. For related scheduling problems see [15], [16], [17]. In contrast, [18] studies a closely related model with deadlines, but *without* the option to reject jobs.

II. Preliminaries

A. Model and notation

We assume a service level agreement (SLA) in the form of the maximum waiting time τ. The maximum tolerated waiting time is simply referred to as the *deadline*. If this deadline is exceeded, a fixed cost of 1 is incurred. We also allow the policy to reject a job upon arrival. For simplicity, we assume the same unit cost for rejected jobs as for deadline violations (or higher cost for deadline violations), even though it would be possible to consider models where discarding a job incurs, e.g., a higher cost than missing a deadline. Consequently, we limit ourselves to admission policies that discard a new job unconditionally when the target deadline cannot be met.

More formally, the cost function is a step-function of waiting time in queue, W, where, if $W = w$,

$$d_\tau(w) = I(w > \tau),$$

so the mean cost rate in terms of SLA violations is

$$r = \lambda\,\mathbb{P}\{W > \tau \text{ or job is rejected}\}.$$

We consider FCFS queues, with Poisson arrival rate λ, and i.i.d. service times denoted by $X_j \sim X$. Job sizes become known upon arrival, and hence, for a single server, u, the backlog in the queue, gives the relevant state information. In the multi-server setting, jobs are assigned (dispatched) to a server immediately upon arrival and the assignment is irrevocable. The service rate at server i is c_i, so the service time of job j if assigned to server i is X_j/c_i. We denote the offered load $\lambda\,\mathbb{E}[X]$ by ρ (in the multi-server setting, $\rho = \lambda\,\mathbb{E}[X]/\sum_i c_i$), and $f(x)$ is the pdf of the job size distribution, $F(x) = \mathbb{P}\{X \le x\}$ the corresponding cdf, and $\bar{F}(x) = 1 - F(x)$.

B. Value functions

The value function, well known in the context of Markov decision processes [19], [20], is defined as the expected difference in costs between a system that is initially in a given state $\mathbf{s}$ and a system in equilibrium,

$$v(\mathbf{s}) := \lim_{t\to\infty} \mathbb{E}[V_t(\mathbf{s}) - r\,t],$$

where the random variable $V_t(\mathbf{s})$ denotes the costs the system incurs during $(0, t)$ when initially in state $\mathbf{s}$, and r is the long-run mean cost rate. The value function enables us to quantify how much better or worse initial state $\mathbf{s}_2$ is than $\mathbf{s}_1$ simply by computing $v(\mathbf{s}_2) - v(\mathbf{s}_1)$, which is the important quantity for policy iteration. In this paper, we will derive and utilize value functions related to the M/G/1 queue subject to deadline cost structure.

III. Single-Server Analysis

We start by analyzing a single M/G/1 queue where jobs can be discarded upon arrival in order to minimize the long-run rate of deadline violations.

A. Admission policies $\xi(u)$

We let the random variable U denote the backlog in the queue, and consider admission policies in which a function $\xi(u)$ defines a threshold for the maximum service time a job may have in order to be admitted to the queue in state $U = u$, i.e., jobs of size $x > \xi(u)$ are rejected. We will show that such a policy is optimal, and how it can be efficiently computed.

Because it is optimal to unconditionally discard jobs whose deadline τ would be violated, we start with a basic policy that discards such jobs and accepts the rest:

Definition 1 (Basic admission policy):

$$\xi_0(u) := \begin{cases} \infty, & u \le \tau, \\ 0, & u > \tau. \end{cases} \tag{1}$$

This policy is also the individually optimal policy for arriving jobs to minimize their own costs. The M/G/1 queue with $\xi_0(u)$ is clearly stable for *any* $\lambda \ge 0$, and the mean cost rate (blocking rate) is

$$r := \lambda\,\mathbb{P}\{U > \tau\}. \tag{2}$$

This simple admission policy can be optimal:

Proposition 1: The basic admission policy $\xi_0(u)$ is optimal for M/D/1 queues.

Proof: This follows trivially from the fact that all jobs are identical and therefore there is no reason to wait for a better job to arrive later (waiting actually includes a risk that no job arrives before the server idles). ■

In general $\xi_0(u)$ is not the optimal admission policy (and more jobs will be rejected because of the negative externalities they would impose on future jobs if admitted). The standard approach to find the optimal policy in the MDP framework is based on studying the value function defined Section II-B. Before discussing this in detail, we first demonstrate how the steady-state distribution can be computed for an arbitrary admission policy $\xi(u)$, with $\xi(u) = 0$ for $u > \tau$.

B. Steady-state distribution of backlog

In this section, we study how to determine the steady-state distribution of the backlog U in a single M/G/1 queue subject to an arbitrary admission policy $\xi(u)$. This quantity then allows one to examine, e.g., the service-time distribution of the admitted and rejected jobs with the given $\xi(u)$.

To this end, we adapt the level-crossing methodology developed in [21], where the workload distribution in a finite size buffer was studied. In this system, jobs are admitted as long as they fit in the buffer, i.e., the size (service time) X of the arriving job is less than the free space in the buffer, which in our context would translate to a deadline for job *completion time* (in contrast to waiting time).

For an arbitrary admission policy, $\xi(u)$, such that $\xi(u) = 0$ for $u > \tau$, we let $g(u)$ denote the unknown continuous density function of the steady-state distribution of U for $U > 0$, and let $\pi_0 = \mathbb{P}\{U = 0\}$. The probability flow downwards across a test level at u is simply $g(u)$. The probability flow upwards is due to jobs that arrive when the backlog is below the level u, $U < u$, and the service time of the job X is sufficiently long to cause a jump across the level u, $U + X > u$, and the job is admitted, i.e., $X < \xi(u)$. This results in the Volterra integral equation of the second kind,

$$g(u) = \lambda \left(\pi_0 Q(0,u) + \int_0^u g(v) Q(v,u)\, dv \right), \qquad (3)$$

where $Q(v, u)$ is the probability that a job arriving in state $v < u$ is admitted and the backlog increases beyond u,

$$Q(v,u) = \left(F(\xi(v)) - F(u-v)\right)^+,$$

and where $(x)^+ = \max\{x, 0\}$.

Because $\xi(u) = 0$ for $u > \tau$, $Q(v, u) = 0$ for $v > \tau$, so the Volterra equation (3) reduces to

$$g(u) = \lambda \left(\pi_0 Q(0,u) + \int_0^\tau g(v) Q(v,u)\, dv \right), \quad u > \tau, \quad (4)$$

and it remains to determine $g(u)$ for $0 < u \le \tau$.

In general, the above integral equation can be solved numerically by first setting $\pi_0 = 1$. Then for arbitrary u, (3) is defined in terms of $g(v)$ with $v \le u$, and $g(u)$ can be worked out iteratively *in the forward* direction, and then the whole distribution, including the atom π_0, is normalized, as explained in [21].

Now that the equilibrium distribution of the backlog in the M/G/1 queue with an arbitrary admission policy $\xi(u)$ is available, several other interesting performance quantities can be determined. The job rejection rate is

$$r = \lambda \left(1 - \pi_0\, F(\xi(0)) - \int_0^\tau g(u)\, F(\xi(u))\, du \right). \qquad (5)$$

The rejection probability is $\mathbb{P}\{X > \xi(U)\} = r/\lambda$, and the fraction of time the server is busy (*carried work*) is simply

$$\rho^* = 1 - \pi_0.$$

Let $p(x)$ denote the probability that a job with service time x is admitted to the queue. From PASTA,

$$p(x) = \pi_0\, \mathbf{1}(\xi(0) \ge x) + \int_0^\tau g(u)\, \mathbf{1}(\xi(u) \ge x)\, du.$$

When $\xi(u)$ is a decreasing function of u, the above simplifies significantly. For the special case of the M/M/1 queue with the basic admission policy $\xi_0(u)$, we have a closed-form solution:

Proposition 2: The steady-state distribution of the M/M/1 queue with deadline τ and the basic admission policy $\xi_0(u)$ is given by

$$\begin{aligned} \pi_0 &= \frac{\mu(\mu - \lambda)}{\mu^2 - \lambda^2 e^{(\lambda-\mu)\tau}}, \\ g(u) &= \pi_0 \lambda\, e^{\lambda \min\{u,\tau\} - \mu u}. \end{aligned} \qquad (6)$$

Proof: Substitute the above trial into (4). ■

Corollary 3: The mean cost rate in the M/M/1 queue with the basic admission policy $\xi_0(u)$ is

$$r = \lambda \int_\tau^\infty g(u)\, du = \frac{\lambda^2(\mu - \lambda)}{\mu^2 e^{(\mu-\lambda)\tau} - \lambda^2}. \qquad (7)$$

C. Value function for the M/G/1 queue

For an arbitrary $\xi(u)$ with $\xi(u) = 0$ for $u > \tau$, we obtain the following general result for the value function $v(u)$.

Proposition 4: The value function $v(u)$ for the M/G/1 queue with arbitrary admission policy $\xi(u)$ with respect to deadline violations is

$$v(u) - v(\tau) = (\lambda - r)(u - \tau), \quad u > \tau, \qquad (8)$$

and, for $0 < u \le \tau$, with $y = \xi(u)$, it satisfies

$$\begin{aligned} v'(u) &= -r + \lambda \bar{F}(y) + \lambda\, F(y) \mathbb{E}[v(u+X) - v(u) \,|\, X \le y] \\ &= -r + \lambda\, \bar{F}(y) + \lambda \int_0^y f(x)\, [v(u+x) - v(u)]\, dx \end{aligned} \qquad (9)$$

with the boundary condition $v'(0) = 0$.

Proof: When $u > \tau$ no new jobs are accepted until the backlog decreases to the level τ, after a deterministic time interval $u - \tau$. Thus, $v(u) - v(\tau)$ represents the difference between the mean number of jobs arriving in this interval and the costs that the system on average incurs in the same interval (without conditioning on the initial state, i.e., starting in equilibrium), yielding (8).

For $0 < u \le \tau$, consider an indefinitely small time interval δ such that $\delta < u \le \tau$, and let $y = \xi(u)$. Then, referring to Figure 1, we have, by definition,

$$\begin{aligned} v(u) = &\left(\lambda\, \bar{F}(y) - r\right) \delta + \lambda\, F(y)\, \delta\, \mathbb{E}[v(u + X) \,|\, X \le y] \\ &+ \left(1 - \lambda\, F(y)\, \delta\right) v(u - \delta), \end{aligned}$$

which leads to (9).

The boundary condition $v'(0) = 0$ follows by considering an initially empty system, $u = 0$, until the arrival of the first accepted job. The mean time to this event is $1/(\lambda\, F(y))$, where $y = \xi(0)$. It follows that

$$v(0) = \frac{\lambda\, \bar{F}(y) - r}{\lambda\, F(y)} + \mathbb{E}[v(X) \,|\, X \le y].$$

Substitution into (9) with $u = 0$ yields $v'(0) = 0$. ■

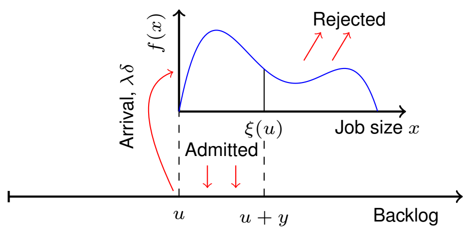


Fig. 1. Transition rates with an arbitrary $\xi(u)$.

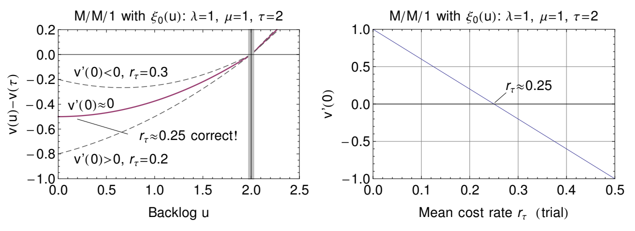


Fig. 2. Determination of the (correct) mean cost rate r for M/M/1 with $\lambda = \mu = 1$ and $\tau = 2$, corresponding to the solution with $v'(0) = 0$.

We also have, with $y = \xi(\tau)$,

$$\begin{cases} v'(\tau^-) = (1+\rho F(y))(\lambda - r) - \lambda F(y) \\ v'(\tau^+) = \lambda - r. \end{cases} \tag{10}$$

D. *On computing $v(u)$ for M/G/1*

Note that, given r and $\xi(u)$, $v'(u)$ in (9) depends only on the values of $v(u')$ with $u' \geq u$, so using (8), it is possible to solve the differential equation backwards from $u = \tau$ to $u = 0$. The problem is that in general we do not know the mean cost rate r even for the basic admission policy $\xi_0(u)$. One option is to solve the Volterra equation (4) and then utilize (5) to determine r. Alternatively, it is also possible to estimate r by process simulation.

However, in the case of a discrete state-space system, solving Howard's equations would yield both the relative values (i.e., the value function apart from an unimportant additive constant) and the mean cost rate. It turns out that this is the case also in our problem when the boundary condition (9) is taken into account. When the differential equation (9) is solved (numerically) backwards from $u = \tau$ to $u = 0$, e.g., using the Runge-Kutta method with a given value of r, the resulting value $v'(0)$ at the origin depends parametrically on r, $v'(0) = v'(0)[r]$. The value of r is then uniquely determined by the condition $v'(0)[r] = 0$. Generally, this is a non-linear equation and has to be solved numerically using some iterative method.

The situation is illustrated in Figure 2, where the correct r is determined for the M/M/1 queue with the basic admission policy $\xi_0(u)$. The left graph shows three solutions for the differential equation obtained with $r \in \{0.2, 0.25, 0.3\}$, and the right graph depicts $v'(0)$ as a function of (trial) r. We observe that the differential equation system behaves systematically and it is straightforward to determine the correct r that satisfies the boundary condition of (9). As a result, we obtain both the value function and the mean cost rate for a given $\xi(u)$.

E. *Exact solutions with $\xi_0(u)$ when $\rho \to 1$ and $\rho \to \infty$*

We first consider the M/M/1 queue when $\rho \to 1$, and then the M/G/1 queue when $\rho \to \infty$. If jobs missing their deadlines were not discarded, these queues would become unstable. However, stability is not an issue in our case. Interestingly, it turns out that in these specific cases, we obtain exact closed-form expressions for several important quantities.

For the M/M/1 queue at the "heavy-traffic" limit where $\rho \to 1$, we have the following result.

Proposition 5: For the M/M/1 queue with $\lambda = \mu$ and $\xi_0(u)$, the value function is initially a quadratic and then a linear function of the backlog u,

$$v(u) - v(\tau) = \begin{cases} \lambda \dfrac{1+\lambda\tau}{2+\lambda\tau}(u-\tau), & u > \tau, \\[2ex] \dfrac{\lambda^2}{4+2\lambda\tau}(u^2 - \tau^2), & 0 \leq u \leq \tau, \end{cases} \tag{11}$$

where

$$v(\tau) = \frac{(\lambda\tau)^2(3+2\lambda\tau)/6 - 1 - \lambda\tau}{(2+\lambda\tau)^2}. \tag{12}$$

Proof: The value function (11) can be shown to satisfy (8) and (9) simply by substitution, and hence it is the correct value function. For the constant term $v(\tau)$, we utilize the identity

$$\mathbb{E}[v(U)] = \pi_0 v(0) + \int_0^\infty g(u)\, v(u)\, du = 0,$$

where π_0 and $g(u)$ define the steady-state distribution of the backlog, given by (6) when $\lambda \to \mu$. Hence,

$$\pi_0(v(0) - v(\tau)) + \int_0^\infty g(u)\, (v(u) - v(\tau))\, du = -v(\tau),$$

and substituting (11) into the left-hand side gives (12). ∎

Even though a queue rejecting jobs when the backlog $U > \tau$ is always stable (for any $\lambda \geq 0$), solving the value function numerically for $u < \tau$ from the differential equation (9) becomes difficult even for M/M/1 when λ is large, as small errors in $v(u')$ for $u' \geq u$ can have a big impact on $v'(u)$, and consequently, on the solution itself. Fortunately, the asymptotic case $\lambda \to \infty$ can be deduced for general service-time distributions:

Lemma 6: When $\lambda \to \infty$, the value function for M/G/1 with $\xi_0(u)$ satisfies

$$v(u) - v(\tau) = \frac{u - \tau}{\mathbb{E}[X]}, \qquad u \geq 0. \tag{13}$$

Proof: Suppose $u \geq \tau$. When λ is large, the backlog in the queue decreases only slightly below τ before a new job arrives. Consequently, the job admission rate, $\lambda - r$, tends to $1/\mathbb{E}[X]$ as $\lambda \to \infty$, and (8) reduces to

$$v(u) - v(\tau) = \frac{u - \tau}{\mathbb{E}[X]}, \qquad u \geq \tau. \tag{14}$$

Suppose next that $u < \tau$. Let N denote the number of (practically instantaneously) admitted jobs until the backlog exceeds τ, i.e., N is the smallest number ("stopping time") such that

$$u + X_1 + \dots X_N > \tau.$$

Focusing on the admitted jobs, we have

$$v(u) = \mathbb{E}[-N + v(u + X_1 + \dots X_N)].$$

Applying (14) on the right-hand side then gives,

$$v(u) = -\mathbb{E}[N] + \frac{\mathbb{E}[u - \tau + X_1 + \dots X_N]}{\mathbb{E}[X]} + v(\tau).$$

For the random sum, we can apply Wald's theorem [19],

$$v(u) - v(\tau) = -\mathbb{E}[N] + \frac{u-\tau}{\mathbb{E}[X]} + \frac{\mathbb{E}[N]\mathbb{E}[X]}{\mathbb{E}[X]} = \frac{u-\tau}{\mathbb{E}[X]},$$

and (14) holds for all $u \geq 0$. ∎

The reference state can be chosen arbitrarily, and, e.g.,

$$v(u) - v(0) = \frac{u}{\mathbb{E}[X]}. \tag{15}$$

Note also that with $\lambda - r = 1/\mathbb{E}[X]$, the derivative $v'(\tau^-)$ of (10) reduces to $\lambda - r$ equalling $v'(\tau^+)$, i.e., in this limit, there is no jump in the derivative at $u = \tau$. This is obvious also from (13) and (15), which are valid for all $u \geq 0$.

Proposition 7: The value function for M/G/1 with $\xi_0(u)$ in the heavy-traffic limit, when $\lambda \to \infty$, is

$$v(u) = \frac{u-\tau}{\mathbb{E}[X]} - \frac{\mathbb{E}[X^2]}{\mathbb{E}[X]^2}. \tag{16}$$

Proof: At the heavy-traffic limit, a job with service time X is admitted to the system each time and immediately after the backlog decreases below $u = \tau$. That is, jobs are admitted after i.i.d. time intervals X. Consequently, the residual time to next arrival has pdf

$$f_r(t) = \frac{t\, f(t)}{\mathbb{E}[X]}.$$

Hence, the steady-state distribution at the heavy-traffic limit is $g(u) = 0$ for $u < \tau$, and for states $u = \tau + t \geq \tau$,

$$g(\tau + t) = \frac{t\, f(t)}{\mathbb{E}[X]}, \qquad t \geq 0.$$

The identity $\int_0^\infty g(u)\, v(u)\, du = 0$ (as $\pi_0 = 0$) with (13) gives

$$v(\tau) = -\int_0^\infty \frac{t\, f(t)}{\mathbb{E}[X]} \cdot \frac{t}{\mathbb{E}[X]}\, dt = -\frac{\mathbb{E}[X^2]}{\mathbb{E}[X]^2}.$$

Substituting this into (13) completes the proof. ∎

Trivially, when $\lambda \to 0$, no job arrives, no deadlines are violated, and therefore both $r \to 0$ and $v(u) \to 0$. Hence, for very small values of λ, the value function is practically constant zero, then at $\rho = 1$ we obtain the quadratic form (for $u < \tau$ and M/M/1), which then transforms to the straight line with slope μ as $\rho \to \infty$ (for a general service-time distribution). This is illustrated in Figure 3 with the standard M/M/1 queue.

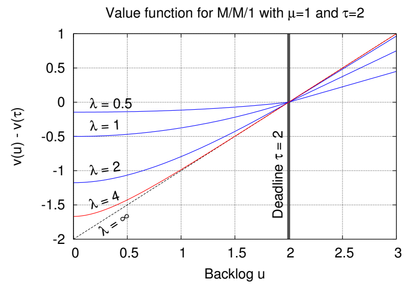


Fig. 3. Value functions for M/M/1 with τ=2 when ρ=$\{0.5, 1, 2, 4\}$, and at the heavy-traffic limit when $\rho \to \infty$ (dashed line).

F. Quadratic approximation for $v(u)$

Motivated by the special case of M/M/1 with $\xi_0(u)$ and $\rho = 1$, we propose a quadratic approximation for $v(u)$ when $0 \leq u \leq \tau$. This approximation can be expected to be a good match when ρ is not too large. (See the previous section. When ρ is large, linear approximation should be used instead.) In particular, our proposal is

$$\hat{v}(u) - \hat{v}(\tau) = \begin{cases} (\lambda - r)(u - \tau), & u > \tau, \\ A(u^2 - \tau^2), & 0 \leq u \leq \tau, \end{cases}$$

where the constant A can be defined in different ways. First, requiring that $\hat{v}(0) - \hat{v}(\tau) = v(0) - v(\tau)$, yields

$$A_1 = \frac{v(\tau) - v(0)}{\tau^2}. \tag{17}$$

Second, we can set $\hat{v}'(\tau^-) = v'(\tau^-)$ (in addition to $\hat{v}'(u) = v'(u) = 0$) and use (10), yielding

$$A_2 = \frac{(1 + \rho F(y))(\lambda - r) - \lambda F(y)}{2\tau}, \tag{18}$$

with $y = \xi(\tau)$. We recall that for the M/M/1 queue with $\xi_0(u)$ and $\rho = 1$, the value function is quadratic, these expressions are exact, and $A_1 = A_2$. However, in general, with an arbitrary $\xi(u)$ and an arbitrary arrival process, this is not the case. (See Figure 5, discussed later.) This approximation will be exploited in Sections IV-A and V-C.

G. Policy iteration in a single M/G/1 queue

Let us now consider the single M/G/1 queue, starting with using $\xi_0(u)$. Let $v(u)$ and r denote the value function and mean cost rate with $\xi_0(u)$.

a) Policy iteration: Suppose $u \leq \tau$ when a job with service time x arrives, so $\xi_0(u)$ would accept the job, but that may not be the optimal action. The first policy iteration (FPI) step rejects the job if the expected increase in future costs would be higher than the cost of discarding the job immediately, i.e., if

$$v(u + x) - v(u) > 1.$$

As $v(u)$ is an increasing function of u for $\xi_0(u)$ (and for any other reasonable admission policy) and it has a linear tail with positive slope $\lambda - r$, the above defines a new admission policy when $u \leq \tau$, $\xi_{\text{FPI}}(u) = y$, where y is such that $v(u + y) - v(u) = 1$. Note that $\xi_{\text{FPI}}(u)$ is finite for every $u \leq \tau$.

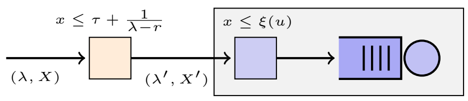


Fig. 4. Static and dynamic admission policies for an M/G/1 queue.

b) Dynamic rule from bound: Determining $\xi_{\text{FPI}}(u)$ requires either an exact (numerical) solution for $v(u)$ or at least an approximation such as those discussed in Section III-F. However, we can obtain an improved policy also by using a lower bound for the admission cost.

Lemma 8: For $u \leq \tau$, we have the lower bound

$$v(u+x) - v(u) > (\lambda - r)(u + x - \tau), \quad 0 \leq u < \tau. \quad (19)$$

Proof: When $u + x < \tau$, the right-hand side of (19) is negative and the inequality holds trivially as $v(u)$ is increasing. Noting that (8) holds for $u + x \geq \tau$,

$$v(u+x) - v(u) > v(u+x) - v(\tau) = (\lambda - r)(u + x - \tau),$$

which completes the proof. ■

Note that Lemma 8 holds for r and $v(u)$ corresponding to an arbitrary non-increasing $\xi(u)$ such that $\xi(u) = 0$ for $u > \tau$, which are properties of a good admission policy.

Consequently, if $(\lambda - r)(u + x - \tau) \geq 1$, with r corresponding, e.g., to the basic admission policy $\xi_0(u)$, then $v(u + x) - v(u) > 1$, and, according to policy iteration, discarding the job reduces the expected long term costs (when compared to $\xi_0(u)$). That is, we can refine (1),

$$\xi_{\text{b}}(u) = \begin{cases} \tau - u + \frac{1}{\lambda - r}, & u \leq \tau, \\ 0, & \text{otherwise.} \end{cases} \quad (20)$$

The obtained rule corresponds to *partial policy iteration*. Note that a lower r obtained with a better $\xi(u)$ than $\xi_0(u)$, would give an even better admission policy than $\xi_{\text{b}}(u)$.

c) Static early reject: In particular, it makes sense to discard a job with service time x at every state $u \geq 0$ if $x \geq \tau + 1/(\lambda - r)$, yielding a *static rejection rule*,

$$\xi_{\text{st}} = \tau + \frac{1}{\lambda - r}, \quad (21)$$

which is then followed by the basic dynamic policy $\xi_0(u)$ that also rejects a job if $u \geq \tau$.

Jobs with service time $x > \xi_{\text{st}}$ are detrimental for the system. Note that in the online setting both λ and r can be estimated; for the latter one needs to feed the arriving jobs also to a virtual duplicated system operating under the basic policy (1) and record the accrued costs.

The jobs passing the static rule constitute a Poisson process with rate $\lambda' = \lambda \mathbb{P}\{X < \xi_{\text{st}}\}$ with a truncated service-time distribution $X' \sim (X \mid X < \xi_{\text{st}})$. Hence, we can consider a two-stage arrangement as illustrated in Figure 4. Such an arrangement can be useful if the first static stage is located near the source, and the second dynamic stage is near the server, saving us from unnecessary job transfers in a scalable fashion. (There is no need to have exact state information at the first stage decision point(s).)

H. Optimal admission policy for M/G/1

In this section, we determine the optimal admission policy for an arbitrary M/G/1 queue with respect to deadline violations. The results in Section III-C hold for an arbitrary admission rule $\xi(u)$. The optimal admission rule is obviously also some threshold policy $\xi_{\text{opt}}(u)$, with value function $v_{\text{opt}}(u)$.

Corollary 9: For $u \leq \tau$, and for $y = \xi_{\text{opt}}(u)$ such that

$$v_{\text{opt}}(u + y) - v_{\text{opt}}(u) = 1, \quad (22)$$

$$v'_{\text{opt}}(u) = \lambda \bar{F}(y) - r + \lambda F(y)\, \mathbb{E}[v_{\text{opt}}(u+X) - v_{\text{opt}}(u) \mid X < y]. \quad (23)$$

The key observation is that $y = \xi_{\text{opt}}(u)$, so $v'_{\text{opt}}(u)$ depends solely on $v_{\text{opt}}(u')$ with $u' \geq u$. As we know $v_{\text{opt}}(u)$ for $u \geq \tau$, it is again possible to solve the differential equation backwards, eventually giving us the value function $v_{\text{opt}}(u)$, the mean cost rate r_{opt}, and the optimal admission policy $\xi_{\text{opt}}(u)$.

As explained in Section III-D, for a fixed $\xi(u)$ and unknown r, we can either (i) solve (9) multiple times iteratively in order to determine the correct r, or (ii) solve the Volterra equation (4) first that gives the correct r, and then solve (9) once. For determining the optimal admission policy and the corresponding mean cost rate r_{opt}, however, the only possibility is to use method (i), since ξ_{opt} is unknown as long as r_{opt} is unknown, and hence the Volterra equation cannot be used to determine r_{opt}. In contrast, it is easier to use the approximations discussed in Section III-F, and they yield near-optimal solutions.

IV. Single-Server Experiments

In this section, we give several numerical examples with single-server systems illustrating both the theoretical results derived in the earlier section, and some interesting phenomena caused by the deadline cost structure itself.

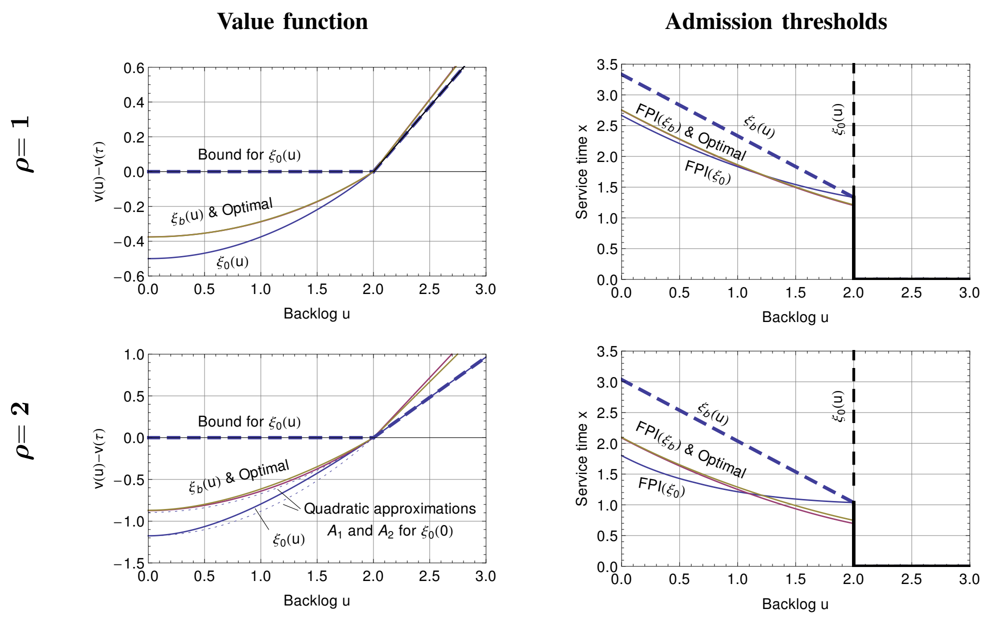
A. M/M/1

In Figure 5, we have depicted several value functions and the corresponding admission policies resulting from one policy iteration round for the M/M/1 queue with $\mu = 1$, $\tau = 2$, and $\rho = 1$ and $\rho = 2$. By FPI(ξ) we mean one policy iteration round from ξ, and Optimal is the optimal admission policy. The value function corresponding to $\xi_{\text{b}}(u)$ is computed backwards from the admission rule. Interestingly, FPI(ξ_{b}) and Optimal are practically equivalent.

With $\rho = 1$ (upper row), the two quadratic approximations for the value function of $\xi_0(u)$ are exact and not shown. However, with $\rho = 2$ the corresponding value function is not a quadratic function, and therefore the approximations using A_1 and A_2 indeed deviate from the exact value function. Also, A_2, defined by the derivative $v'(\tau^-)$, yields a value function that is similar to that of the optimal policy.

B. Steady-state distribution

The steady-state distribution of U for the M/M/1 queue with $\xi_0(u)$ and $\tau = 2$ is given by (6) and depicted in Figure 6 (left) for $\rho = 0.75,\ 1,\ 1.5$. With $\rho = 1$, the steady-state distribution is flat for $0 \leq u \leq \tau$, whereas with $\rho \neq 1$ it is either exponentially decreasing ($\rho < 1$) or increasing ($\rho > 1$). For



$\rho = 1$	
Opt	0.168
FPI(ξ_b)	0.168
FPI(ξ_0)	0.168
ξ_b	0.175
ξ_0	0.25

$\rho = 2$	
Opt	0.282
FPI(ξ_b)	0.282
FPI(ξ_0)	0.293
ξ_b	0.331
ξ_0	0.518

Fig. 5. Value functions and the corresponding admission thresholds for the M/M/1 queue with $\rho = 1$ (upper) and $\rho = 2$ (lower) when $\mu = 1$ and $\tau = 2$. The dashed lines in the bottom left graph correspond to quadratic approximations with A_1 and A_2. With $\rho = 1$, the approximations are exact (see Section III-E).

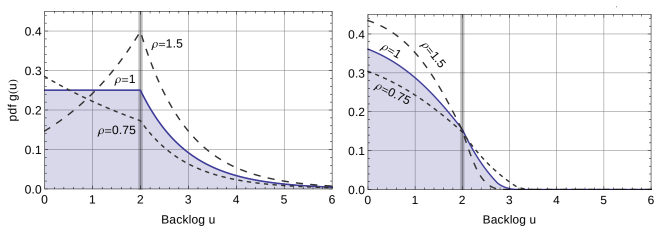


Fig. 6. Steady-state distribution of backlog in M/M/1 with the basic (left) and the optimal (right) admission policies, $\xi_0(u)$ and $\xi_{\rm opt}(u)$.

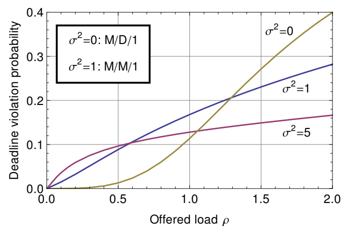


Fig. 7. Deadline violation probability in different M/G/1 queues under the optimal admission policy $\xi_{\rm opt}(u)$. (X ∼Weibull, $\tau = 2$.)

the tail, when $u \geq \tau$, the steady-state distribution decreases exponentially to zero in all cases.

The steady-state distribution for the optimal policy must be computed numerically by first solving for the optimal policy, and then solving the corresponding Volterra equations. The resulting distribution is depicted in Figure 6 (right), where we can see that the backlog rarely exceeds the deadline τ by more than $1/\lambda$.

C. M/G/1

Let us next study the effect of the service-time distribution on the deadline violation rate in the M/G/1 queue with the optimal admission policy. We consider three Weibull distributions, $X_i \sim \text{Weibull}(\alpha_i, \beta_i)$, $i = 1, 2, 3$, with parameters (α_i, β_i) such that the mean service time is $\mathbb{E}[X_i] = 1$, while the variance is varied, $\sigma_i^2 \in \{0,\ 1,\ 5\}$. With $\sigma^2 = 0$, we obtain the M/D/1 queue, $\sigma^2 = 1$ corresponds to the M/M/1 queue, and $\sigma^2 = 5$ then represents an M/G/1 queue with a more variable service-time distribution. For Proposition 1, $\xi_0(u)$ is optimal for the M/D/1 queue, and the mean cost rate can be obtained by solving (9) (or by means of simulation). For the other two queues, we need to solve (23) in order to determine the optimal admission policy and the mean cost rate.

Figure 7 depicts the deadline violation probability as a function of the offered load ρ for these three service-time distributions with the optimal admission policies. When ρ is low, the fixed service time of M/D/1 yields the lowest deadline violation rate. However, as the offered load increases, it becomes obvious that a high variance can actually be advantageous. The reason for this is that when service times vary and jobs are plenty, it is possible to save many mice by discarding a few elephants.

V. Parallel Servers

We now use the value functions for single M/G/1 queues to develop efficient dispatching and admission rules for the parallel server system, as in [22] and [20, Section 11.5].

A. Model and heuristic policies

We let $\mathbf{s}$ denote the state of the system, $\mathbf{s} = (u_1, \ldots, u_n)$, where u_i is the backlog in server i. The admission and routing decision for a job with size x is denoted by $\alpha = \alpha(\mathbf{s}, x)$,

$$\alpha(\mathbf{s}, x) = \begin{cases} 0, & \text{if the job is rejected,} \\ i, & \text{if the job is routed to server } i. \end{cases} \qquad (24)$$

The basic static dispatching policy is the Bernoulli split (RND), which assigns a job to server i with probability p_i, $i = 1, \dots, n$, and where p_i is chosen to balance the load. Given the actual service times and backlogs are available, the commonly used dynamic heuristic dispatching policy is least-work-left (LWL). This policy assigns the new job to the queue with the shortest backlog (in time). Hence, it is the optimal myopic policy with respect to both waiting time and deadline violations. In addition to dispatching, the control policy α should exercise some admission control such as the basic admission rule (1).

The multi-server setting is fundamentally more complicated than a single-server system because there are both admission and dispatching decisions to be made, and the state-space is multi-dimensional. Not surprisingly, only a few optimality results are available (even without the option to reject jobs), and these typically assume identical servers, exponentially distributed service times, and minimizing mean latency as the objective, and they do not use arriving job size information. In this paper, we give an optimality result in the multi-server setting only for the trivial case when all jobs and all servers are identical. Its proof is the same as that of Proposition 1.

Proposition 10: When all jobs are identical, $x_i = x$, and all servers are identical, $c_i = c$, then LWL-ξ_0, choosing the queue with the shortest backlog, and rejecting jobs whose deadline would be violated, is the optimal policy.

B. Admission control with policy iteration

With Poisson arrivals, whenever jobs are assigned in i.i.d. fashion, then each server i also receives jobs according to a Poisson process, and the whole system *decomposes into n independent M/G/1 queues*. Given value functions for the individual M/G/1 queues, the value function of the whole system is

$$v(\mathbf{s}) = \sum_i v_i(u_i),$$

where u_i denotes the backlog in queue i. Then $d_i(u_i) = I(u_i > \tau)$, for $i = 1, \dots, n$, corresponds to the so-called immediate cost for queue i (whether the deadline violation occurs or not). We let $i = 0$ denote the action of rejecting a job, and correspondingly, we define $d_0(u) = 1$ and $v_0(u) = 0$. Consequently, we can carry out FPI, which gives an improved job admission and dispatching rule

$$\alpha(\mathbf{s}, x) = \underset{i \in \{0,\dots,n\}}{\arg\min} \; d_i(u_i) + v_i(u_i + x/c_i) - v_i(u_i), \quad (25)$$

where the difference in the value function corresponds to the expected increase in future costs.

Remark 11: Consider a system of identical servers with (uniform) RND as the dispatching policy and a common arbitrary $\xi(u)$ as the admission rule so that the queue-specific value functions are identical convex functions. Then the admission cost $a(u, x)$ is an increasing function of u, and applying FPI gives LWL-ξ_{FPI}. That is, jobs are routed according to LWL, and then rejected if $x > \xi_{\text{FPI}}(u)$. In particular, starting from RND-ξ_{opt} yields LWL-ξ_{opt}.

There are two observations to be made at this point. First, each policy iteration round improves both the admission and dispatching rules. Second, starting from RND yields "only" LWL, which, though reasonable, is not the optimal dispatching policy in general, because it ignores arriving job sizes (see later numerical examples).

In the ideal case, we would carry out the second policy iteration round. Unfortunately, computing the value function for a dynamic policy such as LWL is hard, and one has to resort to more *adhoc* solutions such as the so-called *Lookahead policy improvement* (LPI) [23], where the idea is to consider also the assignment of the next arriving job, after which the (static) basic policy takes over, i.e., the basic action is (i, j) assigning the current job to queue i and the next (tentatively) to queue j.

In the following two sections, we will evaluate the FPI and LPI policies and compare their performance to that of RND and LWL, augmented with different admission rules.

C. Two identical servers

Let us first consider a small system of $n = 2$ identical exponential servers with $\mu = 1$ and $\tau = 2$. We consider two heuristic reference dispatching policies: static RND and dynamic LWL. Both dispatching policies are complemented with admission control at each server. In particular, we consider two admission policies, $\xi_0(u)$, and the optimal one, $\xi_{\text{opt}}(u)$ (assuming RND), and denote the corresponding complete control policies as RND-ξ_0, RND-ξ_{opt}, LWL-ξ_0 and LWL-ξ_{opt}, respectively. We note that LWL is a dynamic dispatching policy for which we do not know the optimal admission control, and therefore we resort to the admission policy that is optimal for RND.

Additionally, we have FPI and LPI based on the value function derived in Section III. We use the quadratic approximation A_2 for $0 \le u < \tau$ and RND-ξ_{opt} as the starting point. We also experimented with RND-ξ_0. As the policy iteration step also improves the admission rule, the results were only marginally worse than with ξ_{opt}.

The numerical results are depicted in Figure 8 (left). We can see that FPI-ξ_{opt} (i.e., LWL-ξ_{opt}, see Remark 11) is strong, and LPI-ξ_{opt} is even better. Moreover, LWL-ξ_0 is initially good (it is optimal when ρ is very small), but as ρ approaches 1 the basic admission rule is simply inadequate and the performance falls behind the better policies, including even the elementary RND-ξ_{opt}. In general we note that admission control matters more when ρ is high, whereas routing is more important when ρ is low.

D. Four heterogeneous servers

Figure 8 (right) shows the results for $n = 4$ servers with service rates $c = \{2, 2, 1, 1\}$ and $\tau = 2$. We use the same dispatching policies as in the previous examples. LWL-ξ_0 is strong when ρ is small (again, it is optimal when $\rho \to 0$), but becomes weak as the load approaches $\rho = 1$ and (longer) jobs need to be rejected. With heterogeneous servers, LWL-ξ_{opt} is no longer the same as FPI-ξ_{opt}, and in fact the latter turns out

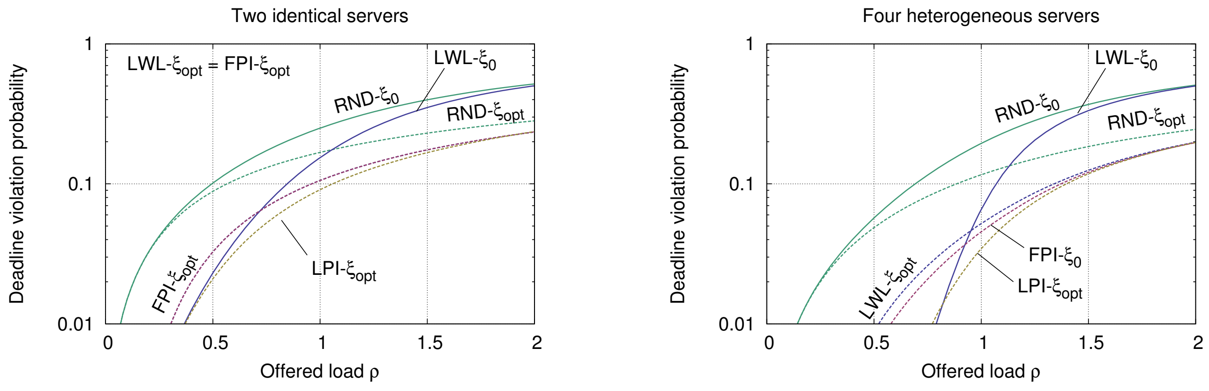


Fig. 8. Results with $n{=}2$ identical servers (left), and with $n{=}4$ heterogeneous servers, $c{=}\{2,2,1,1\}$ (right).

to be a clearly better policy. Finally, LPI is the best control policy especially when ρ is moderate or high, and large jobs should be rejected proactively.

VI. Conclusions

Our deadline-based cost structure is motivated by the need to provide fast responses in today's large-scale cloud based services; *"respond promptly or not at all"*. We derived value functions and the optimal admissions policy for single-server systems. These results were complemented with expressions for the steady-state distribution of backlog and other important performance metrics (e.g., deadline violation rate, carried load), including exact closed-form results in specific (heavy-traffic) limits. We applied our single-server results to systems of parallel servers to develop efficient deadline-aware job admission and dispatching policies. Numerical examples show that our heuristics are superior to standard policies, especially with heterogeneous servers under heavy load. Finally, we note that our results are also useful for more complicated systems of parallel servers, including power-of-two type approaches, where admission and dispatching decisions are based on a small (random) subset of servers. A more detailed investigation of this research direction is left as future work.

Acknowledgements

This work was supported by the Academy of Finland in the FQ4BD and TOP-Energy projects (grant nos. 296206 and 268992).

References

[1] A. J. Thadhani, "Factors affecting programmer productivity during application development," *IBM Systems Journal*, vol. 23, no. 1, pp. 19–35, 1984.

[2] J. Dean and L. A. Barroso, "The tail at scale," *Commun. ACM*, vol. 56, no. 2, pp. 74–80, Feb. 2013.

[3] Z. Liu, M. S. Squillante, and J. L. Wolf, "On maximizing service-level-agreement profits," in *Proceedings of the 3rd ACM Conference on Electronic Commerce*, ser. EC '01. New York, NY, USA: ACM, 2001, pp. 213–223.

[4] B. Saovapakhiran, M. Devetsikiotis, G. Michailidis, and Y. Viniotis, "Average delay slas in cloud computing," in *IEEE International Conference on Communications (ICC)*, Jun. 2012, pp. 1302–1308.

[5] J. Stidham, S., "Optimal control of admission to a queueing system," *IEEE Transactions on Automatic Control*, vol. 30, no. 8, pp. 705–713, Aug. 1985.

[6] J. Lehoczky, "Using real-time queueing theory to control lateness in real-time systems," in *SIGMETRICS*, 1997.

[7] K. D. Glazebrook, C. Kirkbride, and J. Ouenniche, "Index policies for the admission control and routing of impatient customers to heterogeneous service stations," *Operations Research*, vol. 57, no. 4, pp. 975–989, 2009.

[8] D. Li and K. D. Glazebrook, "An approximate dynamic programing approach to the development of heuristics for the scheduling of impatient jobs in a clearing system," *Naval Research Logistics (NRL)*, vol. 57, no. 3, pp. 225–236, 2010.

[9] V. Gupta and M. Harchol-Balter, "Self-adaptive admission control policies for resource-sharing systems," *SIGMETRICS Perform. Eval. Rev.*, vol. 37, no. 1, pp. 311–322, Jun. 2009.

[10] A. Ephremides, P. Varaiya, and J. Walrand, "A simple dynamic routing problem," *IEEE Transactions on Automatic Control*, vol. 25, no. 4, pp. 690–693, Aug. 1980.

[11] Z. Liu and R. Righter, "Optimal load balancing on distributed homogeneous unreliable processors," *Operations Research*, vol. 46, no. 4, pp. 563–573, 1998.

[12] M. Harchol-Balter, M. E. Crovella, and C. D. Murta, "On choosing a task assignment policy for a distributed server system," *Journal of Parallel and Distributed Computing*, vol. 59, pp. 204–228, 1999.

[13] S. Yang and G. de Veciana, "Size-based adaptive bandwidth allocation: optimizing the average QoS for elastic flows," in *IEEE INFOCOM*, vol. 2, 2002, pp. 657–666.

[14] E. Hyytiä, A. Penttinen, S. Aalto, and J. Virtamo, "Dispatching problem with fixed size jobs and processor sharing discipline," in *23rd International Teletraffic Congress (ITC'23)*, San Fransisco, USA, Sep. 2011, pp. 190–197.

[15] L. Schrage, "A proof of the optimality of the shortest remaining processing time discipline," *Operations Research*, vol. 16, no. 3, 1968.

[16] J. G. Shanthikumar and U. Sumita, "Convex ordering of sojourn times in single server queues: Extremal properties of FIFO and LIFO service disciplines," *Journal of Applied Probability*, vol. 24, pp. 737–748, Sep. 1987.

[17] R. Righter, J. G. Shanthikumar, and G. Yamazaki, "On extremal service disciplines in single-stage queueing systems," *Journal of Applied Probability*, vol. 27, no. 2, pp. 409–416, Jun. 1990.

[18] E. Hyytiä and R. Righter, "Routing jobs with deadlines to heterogeneous parallel servers," *Operations Research Letters*, 2016, (to appear).

[19] S. M. Ross, *Applied Probability Models with Optimization Applications*. Holden-Day Inc., 1970.

[20] P. Whittle, *Optimal Control: Basics and Beyond*. Wiley, 1996.

[21] V. Sharma and J. Virtamo, "A finite buffer queue," in *Proceedings of Globecom*, Rio de Janeiro, Brazil, December 1999, pp. 1053–1065.

[22] K. R. Krishnan, "Joining the right queue: a state-dependent decision rule," *IEEE Transactions on Automatic Control*, vol. 35, no. 1, pp. 104–108, Jan. 1990.

[23] E. Hyytiä, "Lookahead actions in dispatching to parallel queues," *Performance Evaluation*, vol. 70, no. 10, pp. 859–872, 2013.