

Admission Control to M/G/1 Subject to General Class-Specific Admission and Rejection Costs

Esa Hyttiä

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 the M/G/1 queue where job sizes become known upon arrival subject to a general cost structure. More specifically, we are interested in determining the optimal admission policy to the (size-aware) system with multiple job-classes each having its own admission and rejection costs. The cost for admitting a job is a class-specific function of the waiting time. As a special case, we consider a deadline cost structure where admitting a job that will be late has a smaller cost than rejecting it. We analyse the system within the framework of Markov decision processes, and derive expressions that enable us to determine the size-aware value function, and the optimal class-specific admission control, as well as the resulting mean cost. The availability of the value function allows one to develop efficient dispatching policies for a system with heterogeneous parallel servers.

I. INTRODUCTION

Admission and dispatching control are critical issues in large cloud computing systems, and the control is complicated by the fact that jobs are heterogeneous in their sizes and costs, and servers are heterogeneous in their processing speeds. Admission control is especially important when some jobs are time-critical, so it may be less costly to reject a job that will not be completed by a fixed deadline, for example, than to accept it. We first study the admission control problem for heterogeneous jobs arriving to a single server, and then discuss how the resulting value functions can be used to develop efficient dispatching policies for routing admitted jobs to heterogeneous servers.

In this paper, we consider two cost structures. The first cost structure is a direct generalization of the deadline cost structure assumed in our previous work [1], where a job that has to wait longer than a given threshold τ incurs a fixed cost. A job can also be rejected upon arrival with the same fixed cost. Consequently, any reasonable admission control policy rejects unconditionally all jobs that would be late. The remaining question is to determine which jobs should be rejected *proactively*. In this paper, we generalize the cost structure by assuming that rejected jobs incur a cost that is larger than the cost for starting late. In this case, it may be beneficial to admit some jobs even if the maximum tolerated waiting time is exceeded. The second cost structure is even more general, with arbitrary class-specific admission and rejection costs. For each class, the cost of accepted jobs is defined by an arbitrary function of the waiting time. Rejected jobs incur a class-specific rejection cost. The system is illustrated in Figure 1.

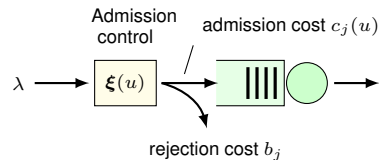


Fig. 1. The size-aware M/G/1 queue subject to class-specific admission and rejection costs. Admission decision depends on job's class j and the current backlog u in the server.

We analyze these M/G/1 queues with admission control in the framework of Markov decision processes (MDPs), and determine the corresponding (size-aware) value functions. In particular, we show how the value function can be computed for a given admission control policy. In fact, it turns out that we can also determine the optimal admission control policy at the same time. The only unknown parameter, the mean cost rate with the optimal (or given) policy, must be determined iteratively.

The multi-class model with arbitrary class-specific admission and rejection costs is very general with wide application. For example, it allows us to analyze scenarios where real-time communication and bulk file transfers share the same link, or equivalently, when real-time and background computations are carried out in the same system. Similarly, the admission or rejection cost can be defined to depend on the size of the job.

These days, parallel server systems are ubiquitous as every popular service in the Internet relies on a huge number of parallel servers. A dispatching model enables performance analysis and optimization of such systems. The availability of the value functions allows us to develop efficient state- and cost-aware dispatching policies for parallel server systems by using the policy improvement step of the Markov decision processes (MDPs).

The rest of the paper is organized as follows. We start by generalizing the deadline-specific cost structure in Section II. In Section III, we consider arbitrary class-specific costs and show how the optimal admission policy and the corresponding value function can be determined for the generalized model without much additional complexity. In Section IV we provide a numerical example, in Section V explain how our results can be applied in the context of parallel server systems, and Section VI concludes the paper.

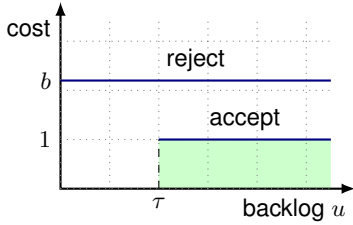


Fig. 2. Deadline cost structure with an option to reject.

II. DEADLINE COSTS

Let us consider the M/G/1 queue with arrival rate λ and general i.i.d. service times (job sizes) denoted by X . For now, as in [1], we assume that jobs have a deadline for waiting time before service: if waiting time exceeds time τ , a unit cost is incurred for all admitted jobs. We will later extend the analysis to general cost functions for admitted jobs. In [1] we also assumed a unit cost is incurred for each rejected job, but now we assume an additional cost for rejection. (Deadlines have been considered also in many other contexts, see, e.g., [2], [3], [4], [5], [6]).

We assume service is FCFS (first-come first-served), so a job's waiting time will equal the backlog, u , observed upon arrival. Formally, our cost structure is as follows (cf. Figure 2):

- 1) **If the job is admitted**, it incurs a deadline cost 1 only if its waiting time u exceeds the set threshold τ , i.e., if $u > \tau$.
- 2) **If the job is rejected**, it incurs a blocking cost of b , where $b > 1$.

We let $\xi(u)$ denote the maximum size of a job that will be accepted in backlog state u . The function $\xi(u)$ defines our admission policy.

Note that if $b \leq 1$, admission control becomes trivial when $u > \tau$ as it is always beneficial to reject jobs in those states. We also note that the cost structure assumed in [7] corresponds to the case $b \rightarrow \infty$, and the cost structure of [1] to $b = 1$.

A. Value Function and Optimal Admission Control

For simplicity, we assume that service time X has a continuous support, $(0, \infty)$. Let $F(x)$ denote the cdf of X , and $\bar{F}(x) = 1 - F(x)$.

The central notion of MDPs is the value function, which, for a given policy, characterizes the expected long-run difference in the costs between a system that starts from the given state, and a system in equilibrium,

$$v(u) := \lim_{t \rightarrow \infty} \mathbb{E}[V(u, t) - rt], \quad (1)$$

where the random variable $V(u, t)$ denotes the total costs incurred in time $(0, t)$ when the system is initially in state u , and r is the long-run mean cost rate (with the given policy),

$$r := \lambda \mathbb{E}[C],$$

and where C denotes the cost incurred for a given policy by a random job in steady state. The value function then enables

policy improvement. It can be also used to show that the current policy is optimal. For more details, see, e.g., [8], [9].

The value function for the optimal admission policy $\xi(u)$ satisfies

$$v(u + \xi(u)) + 1(u > \tau) = v(u) + b, \quad (2)$$

where $1(\cdot)$ is the indicator function. For a given state of the system, u , Eq. (2) defines the critical size $\xi(u)$: smaller jobs are accepted, whereas larger jobs are rejected. Exactly at this point the expected costs of the two possible actions are equal.

In our model, with $1 < b < \infty$, so the maximal cost of admission is less than the cost of rejection, it is easy to show that a policy with $\xi(u) = 0$ for some $u \geq m$ with arbitrary $m \geq \tau$ cannot be optimal because a policy iteration step yields a better policy for those states given sufficiently small jobs exist.

Proposition 1: The optimal admission control for states $u > \tau$, where the deadline will be violated, is a constant threshold, $\xi(u) = h^*$, where h^* is the solution to

$$\frac{h^*}{1 - \rho^*} (\lambda + \lambda(b-1) \mathbb{P}\{X > h^*\} - r) = b - 1, \quad (3)$$

where

$$\rho^* = \lambda \mathbb{P}\{X \leq h^*\} \mathbb{E}[X | X \leq h^*],$$

and r is the long-run cost rate. Moreover, the corresponding value function for the tail $u > \tau$ is,

$$v(u) - v(\tau) = \frac{u - \tau}{1 - \rho^*} (\lambda + \lambda(b-1) \mathbb{P}\{X > h^*\} - r). \quad (4)$$

Proof: Suppose first, for a given h , that $\xi(u) = h$ for $u > \tau$, and let ρ^* denote the admitted load, $\rho^* = \lambda \mathbb{P}\{X < h\} \mathbb{E}[X | X < h]$. With the constant control while $u > \tau$, we can utilize the results for the (remaining partial) busy period in the M/G/1 queue to get the value function using threshold h , which is the same as (4) with the arbitrary h replacing h^* . In particular, the first factor corresponds to the mean time for the system to move from state u to state τ . Then $\lambda \mathbb{P}\{X < h\} + \lambda \mathbb{P}\{X > h\} \cdot b$ is the cost rate (while $u > \tau$), and r is the long-run cost rate, by definition. From PASTA, we immediately obtain (4) for our arbitrary h . Because (4) is a strictly increasing linear function of u , according to a policy iteration step, we will accept a job of size x if $v(u+x) - v(u) + 1 < b$. That is, we have that $\xi(u)$ is some constant for $u > \tau$. In order for $\xi(u) = h^*$ to be the optimal admission control, we must have

$$v(u + h^*) - v(u) = b - 1, \quad \forall u > \tau, \quad (5)$$

so that a policy iteration step gives the same policy: no job larger than h should be admitted when $u > \tau$, and conversely, it is less expensive to admit jobs shorter than h than to reject them. That is, $\xi(u) = h^*$, for h^* satisfying (5), is the optimal policy when $u > \tau$.

Substitution of the value function (4) into (5) gives (3), which defines h^* as a function of the mean cost rate r .

Consequently, we can determine both the optimal admission policy, as well as the corresponding value function, for all

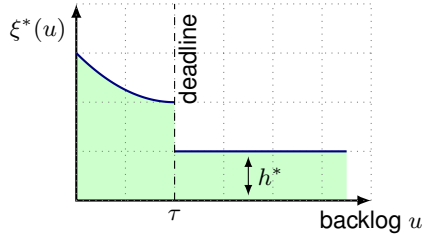


Fig. 3. Optimal admission control to M/G/1 queue.

states $u \geq \tau$ (given we know r). Consider next states where $u < \tau$.

Proposition 2: For state $u < \tau$, the value function $v(u)$ satisfies

$$v'(u) = \lambda \left[\bar{F}(\xi(u))b - \bar{c} + F(\xi(u)) \left(E[v(u+X) - v(u) | X < \xi(u)] \right) \right], \quad (6)$$

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

Proposition 2 is a special case of Proposition 4 below, so its proof is omitted.

The key observation is that given the value function $v(u)$ for $u > \tau$, (4), the differential equation (6) can be solved backwards from $u = \tau$ to $u = 0$ for any given value of r . This leads to some value of the derivative at the origin $v'(0)$ which depends on the chosen value of r , call it $T(r) = v'(0)$. The unknown mean cost rate r can then be determined from the condition $T(r) = 0$. The numerical solution leads to a similar iterative procedure as in [1]. Usually only a few iteration steps (in each of which (6) is integrated backwards) are needed since, as demonstrated in [1] by an example, $T(r)$ is a smooth, almost linear function of r .

More specifically, we have the same two options as in [1]:

- 1) We can determine $v(u)$ and r for any given $\xi(u)$.
- 2) We can determine the optimal admission policy $\xi^*(u)$, together with the corresponding value function $v(u)$ and mean cost rate r , at the same time.

The latter follows from the observation that at every step, for given u , when $v(t)$ is known for $t \geq u$, we can first determine the optimal $\xi(u)$ from (2) and then the corresponding $v'(u)$. Both $\xi(u)$ and $v(u)$ can be solved backwards from $u = \tau$ to $u = 0$.

The optimal admission control is illustrated in Figure 3.

Example 1: Consider an M/M/1 queue with $\lambda = \mu = 1$, $\tau = 2$ and $b = 2$. Then the optimal admission policy has $h = 0.715$ and $r = 0.316$. For comparison, the optimal admission policy with $h = 0$ (admitting jobs only when $u \leq \tau$) gives $r = 0.336$. Hence, in this case, serving short jobs, $x < 0.715$, even if they will be late decreases costs by about 6%. The benefit of serving late jobs will be even higher with higher rejection costs.

B. Steady-state distribution

In [1], it was shown that the steady-state distribution of backlog, U , $g(u)$ satisfies a Volterra integral equation of the second kind,

$$g(u) = \lambda \left(\pi_0 Q(0, u) + \int_0^u g(v) Q(v, u) dv \right), \quad (7)$$

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

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

and where $(x)^+ = \max\{x, 0\}$. In [1] $\xi(u) = 0$ for $u > \tau$, so $Q(v, u) = 0$ for $v > \tau$, and the Volterra equation (7) reduced to

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

In our case, where jobs of size at most h are admitted in any state $u > \tau$, we need to determine the tail behavior of $g(u)$ for $u > \tau$. It is easy to see that, for states $u \gg \tau$ where $Q(0, u) \approx 0$, we have

$$g(u) = \lambda \int_{u-h}^u g(v) P\{X > u - v\} dv.$$

1) *Numerical solution:* Numerically, we can proceed as follows. Let $n \geq 3$ be some odd number of endpoints of sub-intervals of $(u - h, u)$ and let $\Delta = h/(n - 1)$ denote the length of each sub-interval. Then let (g_1, \dots, g_n) denote the values of $g(u)$ at points $(u - h, u - h + \Delta, \dots, u)$. Suppose we have some numerical values for (g_1, \dots, g_{n-1}) . With numerical integration, the integral on the right-hand side becomes a weighted sum of the g_i , and we obtain, using Simpson's composite formula,

$$g_n = \frac{\Delta}{3} \sum_{i=1}^n a_i g_i P_i,$$

where the a_i are the coefficient of the Simpson's integration rule $(1, 4, 2, 4, \dots, 1)$ and the P_i correspond to the CCDF of the service time distribution at points $(h, h - \Delta, \dots, 0)$. Solving for g_n then gives¹

$$g_n = \frac{\Delta}{3 - \lambda \Delta} \sum_{i=1}^{n-1} a_i g_i P_i. \quad (8)$$

In the next step, we first drop g_1 and get $(g'_1, g'_2, \dots, g'_n)$, where the last g'_n , corresponding to $u + \Delta$, is now the unknown. Then we can normalize the g'_i so that, e.g., $g'_1 = 1$, and utilize (8) again to find out g'_n . This iteration tends to converge quickly, and the ratio g_1/g_n gives the rate at which the tail decays (with the number of sub-intervals of h).

In summary, by choosing the intervals appropriately, it is possible to numerically compute the steady-state distribution $g(u)$ for a given $\xi(u)$, including the tail.

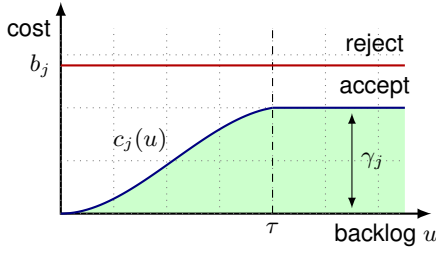


Fig. 4. An example admission and rejection cost for class j .

TABLE I
NOTATION.

j	job class, $j \in \{1, \dots, n_c\}$
λ_j	arrival rate of class j , and $\Lambda = \lambda_1 + \dots + \lambda_{n_c}$
X_j	job size of class j , with CDF $F_j(x)$ and CCDF $\bar{F}_j(x)$
$c_j(u)$	admission cost of class j job with backlog u (waiting time)
b_j	rejection cost of class j jobs, $b_j \geq c_j(u)$
r	the long-run cost rate
\bar{c}	the overall long-run mean cost per job, $\bar{c} = r/\Lambda$
$\xi_j(u)$	admission policy of class j (the largest job size admitted)
r	the long-run cost rate under a given admission policy

III. GENERAL COSTS WITH MULTIPLE JOB CLASSES

Let us next consider the general case with n_c job classes and general admission costs. More specifically, let λ_j and X_j denote the arrival rate and job size of class j jobs, $j = 1, \dots, n_c$, with cdf's $F_j(x)$ and $\bar{F}_j(x) = 1 - F_j(x)$, and $\Lambda = \lambda_1 + \dots + \lambda_{n_c}$. Let $c_j(u)$ and b_j be the admission and rejection costs of class j jobs, where $b_j \geq c_j(u)$, let r be the long-run cost rate, and let $\bar{c} = r/\Lambda$ be the overall long-run mean cost per job. For simplicity, we assume a constant tail behavior for admission costs in the sense that, for all j ,

$$c_j(u) = \gamma_j \quad \text{for } u \geq \tau,$$

where τ is some (possibly large) value. Examples of possible admission and rejection costs are illustrated in Figure 4 (cf. the deadline cost structure illustrated in Figure 2). Finally, we let the $\xi_j(u)$ denote the class-specific admission control. The notation is summarized in Table I.

A. Characterizing the Value Function

It turns out that the solution for the value function has the same structure as for the single-class case. Consider first the tail when $u \geq \tau$.

Proposition 3: The optimal class-specific admission policies are constant for the tail, $\xi_j(u) = h_j^*$ for $u \geq \tau$. Then the value function for the tail is a linear function of u satisfying

$$v(u) - v(\tau) = \frac{u - \tau}{1 - \rho^*} \left(\sum_j \lambda_j (F_j \gamma_j + \bar{F}_j b_j) - r \right), \quad (9)$$

where $F_j = F_j(h_j^*)$ and $\bar{F}_j = \bar{F}_j(h_j^*)$, and

$$\rho^* = \sum_j \lambda_j F_j E[X_j | X_j < h_j^*].$$

¹Given n is sufficiently large, $\lambda \Delta < 3$.

Proof: The argument is similar to the proof of Proposition 1, and the summation corresponds to additional costs due to arrivals, which are γ_j or b_j for each arrival in states $u \geq \tau$.

Note that in (9), we assumed that r is known. Let us consider next the whole interval with some fixed admission policies $\xi_j(u)$.

Proposition 4: The value function $v(u)$ satisfies the differential equation

$$v'(u) = -r + \sum_j \lambda_j \left[\bar{F}_j(\xi_j(u)) b_j + F_j(\xi_j(u)) \left(c_j(u) + E[v(u+X_j) - v(u) | X_j < \xi_j(u)] \right) \right], \quad (10)$$

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

Proof: Let δ denote a small time-interval. Then for $\delta < u < \tau$, dynamic programming gives

$$v(u) = \sum_j \lambda_j \delta \bar{F}_j \cdot (b_j + v(u - \delta)) + \sum_j \lambda_j \delta F_j \cdot (c_j(u) + E[v(u+X_j) | X_j < \xi_j(u)]) + (1 - \Lambda \delta) v(u - \delta) - \Lambda \bar{c}.$$

where $F_j = F_j(\xi_j(u))$ and $\bar{F}_j = \bar{F}_j(\xi_j(u))$. Note that $r = \Lambda \bar{c}$. In the limit $\delta \rightarrow 0$, the above equation yields (10).

Consider then the empty system in state $u = 0$.

$$v(0) = \frac{1}{\sum_j \lambda_j F_j} \cdot \left(\sum_j \lambda_j \bar{F}_j b_j - r + \sum_j \lambda_j F_j \cdot (c_j(0) + E[v(X_j) | X_j < \xi_j(0)]) \right).$$

Thus,

$$\sum_j \lambda_j (\bar{c} - \bar{F}_j b_j) = \sum_j \lambda_j F_j \cdot (c_j(0) + E[v(X_j) - v(0) | X_j < \xi_j(0)]),$$

which upon substitution into (10) yields $v'(0) = 0$.

Note that (10) holds also for the tail when $u \geq \tau$. It follows that the mean long-run cost rate r is also determined when we find a solution to the differential equation (10) that also satisfies the boundary condition. Conversely, r can be seen as a free curve parameter for (10), and we are interested in determining one specific solution.

B. Computing the Value Function and Optimal Control

Finding the correct mean cost rate r and the corresponding value function is a relatively straightforward task also in this case. In fact, as before, we have two options: we can determine i) the value function for a given policy $\xi_j(u)$ or ii) the optimal admission policy $\xi_j^*(u)$ and the corresponding value function $v(u)$.

For a fixed admission policy, the procedure is as follows:

- 1) Pick r such that $0 < r < \sum_j \lambda_j b_j$.
- 2) Solve the $v(u)$ using (9) and (10).
- 3) If $|v'(0)| < \epsilon$, accept the numerical solution.

- 4) If $v'(0) < 0$, the correct mean cost rate is larger, and vice versa. Adjust r accordingly and go to Step 2.

By iteration, we eventually find the correct r and the corresponding value function $v(u)$.

When we are interested in determining (also) the optimal admission control policy, $\xi_j(u)$, we proceed similarly:

- 1) Choose a candidate for r .
- 2) Determine the h_i defining the optimal policy for the tail by solving the set of n_c non-linear equations

$$\frac{h_i}{1 - \rho^*} \left(-r + \sum_j \lambda_j (F_j \gamma_j + \bar{F}_j b_j) \right) = b_i - \gamma_i,$$

where $i = 1, \dots, n_c$, and ρ^* , F_j and \bar{F}_j depend on $\{h_1, \dots, h_{n_c}\}$. Consequently, $\xi_j(u) = h_j$ for $u \geq \tau$, and also $v(u)$ becomes fixed from (9) for $u \geq \tau$.

- 3) The optimal threshold $\xi_j(u)$ for $u \leq \tau$ depends on $v(t)$ for $t \geq u$, and can be determined from

$$v(u + \xi_j(u)) - v(u) = b_j - c_j(u), \quad \forall j.$$

Consequently, we can determine both $v(u)$ and $\xi_j(u)$ for $u \leq \tau$ and for all j by solving the differential equation (10) backwards from $u = \tau$ to $u = 0$.

- 4) If $|v'(0)| < \epsilon$, accept the numerical solution and stop.
- 5) Update r according to the boundary condition $v'(0) = 0$, and go to Step 2.

Note that without loss of generality, we can subtract b_j from class j admission and rejection costs. By doing this for each class, the expressions simplify somewhat, and, e.g., (10) can be written as

$$v'(u) = \tilde{r} + \sum_j \lambda_j \left[F_j(\xi_j(u)) \left(\tilde{c}_j(u) + E[v(u+X_j) - v(u) | X_j < \xi_j(u)] \right) \right], \quad (11)$$

where $\tilde{c}_j(u) = c_j(u) - b_j \leq 0$ is the admission cost and $\tilde{r} > 0$ corresponds to the decrease in the cost rate when compared to the policy that rejects all jobs, $\tilde{r} = \sum_j \lambda_j b_j - r$, i.e., \tilde{r} represents a profit (or savings) rate. The boundary condition remains the same, $v'(0) = 0$.

Example 2: Consider a system with two job classes. Class 1 represents real-time traffic with target deadline τ for some small τ . The corresponding admission cost is $c_1(u) = 1(u > \tau)$ and rejection cost is $b_1 = 1$. Class 2 represents bulk traffic with parameters $c_2(u) = 0$ and the constant $b_2 > 0$ defines the relative importance of the file transfer. It is straightforward to determine the optimal class-specific admission control policies based on Proposition 4.

IV. NUMERICAL EXAMPLE

Let us continue with Example 2 in the context of data networks. Suppose we have two types of packets (job classes):

- 1) Classes have identical size distributions,

$$X_1 \sim X_2 \sim \text{Exp}(1).$$

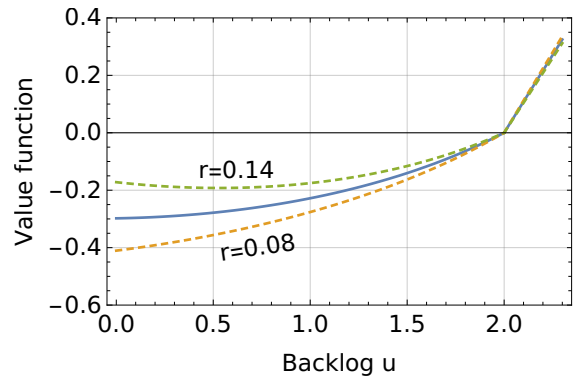


Fig. 5. Determination of the value function for the optimal admission control.

- 2) Classes have also identical arrival rates, $\lambda_1 = \lambda_2 = 1/2$, so that the offered load is $\rho = 1$ (which is not an issue as jobs can be rejected).
- 3) Class 1 packets represent real-time traffic with a relatively short deadline at $\tau = 2$:

$$c_1(u) = \begin{cases} 0, & \text{for } u \leq \tau, \\ 1, & \text{for } u > \tau, \end{cases}$$

$$b_1 = 1.$$

- 4) Class 2 jobs represent bulk file transfer:

$$c_2(u) = 0 \quad \text{for all } u,$$

$$b_2 = 1/4.$$

Thus, when $u > \tau$, Class 1 packets will be discarded as the blocking cost is the same as the cost of deadline violation (i.e., a packet has become obsolete). Class 2 packets have the same value even if delivered late. However, we can still reject them in order to make room for Class 1 packets.

The optimal admission policy for this system can be determined as described in Section III-B. Figure 5 depicts the corresponding value function, obtained as a result of iteration. Solving the system with $r = 0.08$ yields the lower curve. However, as $v'(0)$ is clearly positive, the boundary condition is not satisfied. Similarly, the upper curve is obtained with $r = 0.16$, which turns out to be too high. The correct mean cost rate with the optimal admission control is $r \approx 0.118$, corresponding to the curve in the middle. In this case, the boundary condition is satisfied, $v'(0) = 0$.

As discussed, the optimal admission policy is obtained at the same time. This is depicted in Figure 6. We note that Class 1 packets are rejected when $u > \tau$, and otherwise they are more likely to be admitted. In contrast, Class 2 packets are admitted to the system also when $u > \tau$ given their size is less than $h_2^* \approx 0.52$.

V. PARALLEL SERVERS AND POLICY IMPROVEMENT

The standard application of the value functions of single server queues has been policy improvement in the dispatching systems with parallel servers [10], [11], [12], [13], [14], [15]. For simplicity of explanation, let us assume a single job class.

First we assume the static *basic policy*, e.g., the random Bernoulli split (RND), so that the arrival process to server i is a Poisson process with some rate λ_i , $\sum_i \lambda_i = \lambda$, and the system decomposes into n independent servers. In general, the chosen basic policy defines the server-specific parameters (λ_i, X_i) , which define a (fictitious) arrival process to each server, and the corresponding (server-specific) value functions $v_i(\cdot)$, $i = 1, \dots, n$.

Let $\mathbf{z} = (u_1, \dots, u_n)$ denote the state of the system. The first policy improvement step gives us a policy, referred to as FPI, that either rejects the job or assigns it to a queue, whichever yields the smallest expected cost in the given state. The cost of rejecting the job is

$$a_0 = b,$$

whereas the admission cost to server i is

$$a_i = a_i(u_i, x) = c_i(u_i) + v_i(u_i + x) - v_i(u_i).$$

where $c_i(u_i)$ denotes the immediate cost of choosing server i , and $v_i(\cdot)$ is the value function of server i . The policy improvement then gives the FPI policy,

$$\alpha(\mathbf{z}, x) = \arg \min_i a_i,$$

where action $i = 0$ corresponds to rejecting the given job. Note that the basic policy may (tentatively) reject some jobs. For example, under a heavy load a sensible default action could be to reject large jobs. The dynamic FPI policy then deviates from the default action and assigns a job to a server i if $a_i \leq a_j$ for all j .

VI. CONCLUSIONS

We have considered the problem of admission control to the single server M/G/1 queue in the multi-class setting with general cost structure. In particular, jobs have class-specific admission costs that depend also on the current waiting time. The cost of rejecting a job depends only on the class, but not on the current state of the system.

We analyzed the system in the framework of Markov decision processes. In particular, we showed that both the optimal class-specific admission control, defined in terms of the critical class-specific size thresholds $\xi_j(u)$, and the corresponding value function $v(u)$, can be determined at the same time when solving a particular differential equation backwards from high workloads to an empty system with $u = 0$. The system involves only one unknown parameter, the mean cost rate r , that can easily be determined by iteration.

These expressions can be used to determine the optimal admission control to a wide range of single server systems.

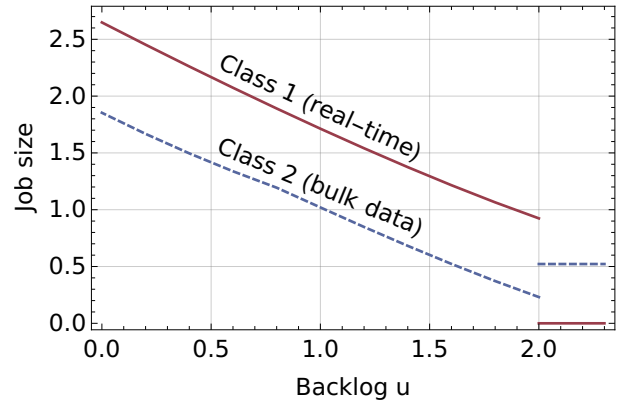


Fig. 6. Optimal admission thresholds $\xi_j(u)$ for Example 2 scenario.

Moreover, the knowledge of the value function can be used in the context of parallel server systems to develop efficient dispatching policies using first policy iteration.

REFERENCES

- [1] E. Hyttiä, R. Righter, and J. Virtamo, "Meeting soft deadlines in single- and multi-server systems," in *28th International Teletraffic Congress (ITC'28)*, Würzburg, Germany, Sep. 2016.
- [2] R. McNaughton, "Scheduling with deadlines and loss functions," *Management Science*, vol. 6, no. 1, Oct. 1959.
- [3] J. P. Lehoczky, "Real-time queueing theory," in *17th IEEE Real-Time Systems Symposium*, Dec. 1996, pp. 186–195.
- [4] D. Li and K. D. Glazebrook, "An approximate dynamic programming 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.
- [5] 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.
- [6] J. Niño-Mora, "Resource allocation and routing in parallel multi-server queues with abandonments for cloud profit maximization," *Computers & Operations Research*, vol. 103, pp. 221–236, 2019.
- [7] E. Hyttiä and R. Righter, "Routing jobs with deadlines to heterogeneous parallel servers," *Operations Research Letters*, vol. 44, no. 4, pp. 507–513, 2016.
- [8] R. A. Howard, *Dynamic Probabilistic Systems, Volume II: Semi-Markov and Decision Processes*. Wiley Interscience, 1971.
- [9] S. M. Ross, *Applied Probability Models with Optimization Applications*. Holden-Day Inc., 1970.
- [10] K. R. Krishnan and T. J. Ott, "State-dependent routing for telephone traffic: Theory and results," in *IEEE Conference on Decision and Control*, vol. 25, Dec. 1986, pp. 2124–2128.
- [11] K. R. Krishnan, "Joining the right queue: a Markov decision rule," in *Proc. of the 28th Conference on Decision and Control*, Dec. 1987, pp. 1863–1868.
- [12] —, "Markov decision algorithms for dynamic routing," *IEEE Communications Magazine*, pp. 66–69, Oct. 1990.
- [13] S. Bhulai, "On the value function of the M/Cox(r)/1 queue," *Journal of Applied Probability*, vol. 43, no. 2, pp. 363–376, Jun. 2006.
- [14] P. Whittle, *Optimal Control: Basics and Beyond*. Wiley, 1996.
- [15] E. Hyttiä, A. Penttinen, and S. Aalto, "Size- and state-aware dispatching problem with queue-specific job sizes," *European Journal of Operational Research*, vol. 217, no. 2, pp. 357–370, Mar. 2012.