Value (generating) functions for the M^X/G/1 queue

Esa Hyytiä
Department of CS
University of Iceland

Rhonda Righter
IEOR
UC Berkeley

Jorma Virtamo
COMNET
Aalto University

Lauri Viitasaari
COMNET
Aalto University

***Abstract*—We analyze the M^X/G/1 queue in the framework of Markov decision processes (MDPs). The service times become known upon arrival, and each job incurs a cost according to a given cost function. The value function is a central concept in MDP theory as it characterizes the value of the system's state with respect to future developments. We derive compact expressions for the generating functions for general families of value functions corresponding to often used cost structures defined in terms of waiting and sojourn times. Moreover, we consider systems with and without setup delays.**

***Index Terms*—M^X/G/1; batch arrivals; generating function; value function; MDP; LST; Laplace transform**

I. Introduction

Many stochastic service systems can be modeled as networks of queues, and optimal control policies can be obtained for them by solving Markov decision problems (MDPs). Such control problems can be found in a wide-variety of fields, e.g., inventory management, computer networks, data centers, and transportation.

In this paper, we are especially interested in routing and energy control for parallel server systems, but our approach may be useful in other MDPs. Routing and energy control has become a critical issue in modern server farms, where quality of service must be balanced with energy costs, including greenhouse gas emissions.

In order to optimize such a system, one needs to first define a meaningful objective. A relatively common choice is to minimize the mean response time, i.e., the mean time from when a request arrives until it is processed and leaves the system. Our approach allows for much more general cost functions of waiting times or response times, including nonlinear functions to encourage fairness, having rewards (equivalently costs) for service before (after) deadlines, and costs and rewards that depend on both the job and the server. We can also include energy costs associated with running the server. Our approach can handle, for example, linear combinations of the cost functions shown in Table I.

For a general dispatching or routing control problem, solving the corresponding MDP becomes prohibitive, because the queues are inter-related through the sequence of dispatching decisions, and we have state-space explosion. A very effective heuristic to handle this is to do one step of policy iteration, starting with a basic policy for which the value function can be evaluated. By using a basic policy that decouples the system into independent queues, such as random routing, we can compute the system's value function for the base step of policy improvement, by finding the value functions for the individual queues. Our main objective in this paper is to derive the value functions that characterize the expected future costs for each state, for each queue. We also show how to use our general value generating functions in a one-step policy iteration heuristic, for both energy and dispatch control.

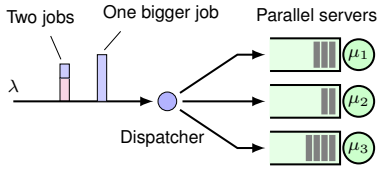


Fig. 1. Model for a parallel server system with batch arrivals.

Our model for each server is the M^X/G/1 queue, where batches of jobs arrive to the server according to a Poisson process, they are served in first-come-first-served (FCFS) order, and the service times are independent and identically distributed (i.i.d.) random variables. Optionally, we also assume that an idle server can be switched to a low power state, from which it is woken up when the next batch of jobs arrives. The wake-up time is non-negligible and referred to as the setup delay, which induces an additional delay to jobs.

In the multi-server setting we have n heterogeneous parallel servers, where each job is dispatched immediately upon arrival to one of the servers (see Figure 1). We assume jobs in the same batch are randomly ordered, and the dispatching decisions are made sequentially and instantaneously. At the time of the decision, the job's size (service time) is known, and the decision can be based on this information. That is, our policies are *size-aware*, and the state of a queue is the total unfinished work, or backlog. The space of control actions includes both server switch-off and job routing decisions.

Prior work on dynamic routing with policy iteration has generally assumed Poisson arrivals (without batches) and exponential service times under the basic routing policy, and that job sizes are unobserved, so that the system's state is the number of jobs (per server). We refer to these as *number-aware* systems. For example, Krishnan [1] minimized the mean sojourn time, and Argon et al. [2] considered a system where the admission cost is an arbitrary function of the number of jobs at the server. The size-aware setting has recently been considered, e.g., in [3] with job-specific holding costs (linear functions of sojourn time), and [4] with setup delays (cf. energy consumption) and admission costs that are arbitrary powers of the waiting or sojourn time.

978-0-9883-0453-6/17 $31.00 © 2017 ITC
DOI 10.1109/ITC.2017.54

TABLE I
COMMON COST STRUCTURES.

Type	Cost function	References
waiting time	$c_1(w) = w$	[5]–[8]
higher moments	$c_n(w) = w^n$, $n = 1, 2, \ldots$	[4]
slowdown	$c_s(w, x) = w/x$ (per job)	[3], [9]
deadline(s)	$c_\tau(w) = \mathbf{1}(w > \tau)$	[2], [10], [11]
energy	$c_e(w) = \mathbf{1}(w > 0)$	[4], [12]
w is the waiting time, x the service time of a job, and τ the deadline.		

We also assume a size-aware system with setup delays, but consider *more general cost structures* defined by arbitrary functions of the waiting or sojourn time. These include all *energy consumption and QoS metrics* that can be defined as functions of waiting, service or sojourn time. Moreover, in contrast to [3] and [4], we allow *batch arrivals*. The number-aware case can be obtained by unconditioning.

The main contribution of this paper is twofold: First, we give compact expressions for the value function of the M^X/G/1 queue subject to random setup delays and arbitrary cost functions defined in terms of job-specific waiting and sojourn times. Second, we study "exponential" cost structures that yield all cost structures listed in Table I as special cases. These value functions have particularly convenient forms from which one can identify the well-known Pollaczek-Khinchin transform formulæ and the penalty due to the setup delay. We call these value generating functions, in analogy to moment generating functions.

The rest of the paper is organized as follows. First, in Section II we introduce the basic model. In Section III, we analyze the M^X/G/1 queue with costs defined in terms of the waiting time. In Section IV, we consider an exponential cost function of waiting time. Sojourn time based costs are then discussed in Section V. In Section VI, we utilize the obtained results and derive policies for switching off servers and routing jobs to parallel servers. Section VII concludes the paper.

II. MODEL

The basic queueing model we consider is the M^x/G/1 queue defined as follows.

Definition 1 (M^x/G/1) *The M^x/G/1 queue queue is a single server queue where jobs arrive in batches, each batch consisting of a random number of jobs, $B_i \sim B$ (i.i.d.), where B is referred to as the batch size. The batch arrival process is a Poisson process at rate λ_b. The service times of jobs are i.i.d., $X_i \sim X$, and thus the service time of the batch is a random sum,*

$$H = X_1 + \ldots + X_B.$$

The offered load, denoted by ρ is

$$\rho \triangleq \lambda_b \, \mathrm{E}[H] = \lambda \, \mathrm{E}[X], \tag{1}$$

where λ denotes the *job arrival rate*, $\lambda = \lambda_b \, \mathrm{E}[B]$. In general, we assume that $\rho < 1$ so that the system is stable and ergodic.

The main advantage of studying the M^x/G/1 queue instead of the M/G/1 queue (with $B = 1$) is that by adjusting the batch size distribution we can model more bursty arrival processes without giving up the memoryless property of Poisson process.

We also consider systems where the server is switched off when idle. The motivation for this is the presumed energy savings. However, the drawback is that restarting the server takes some time, referred to as *the setup delay*. We assume that setup delays, $D_i \sim D$, are i.i.d., and that the server is switched back on as soon as the first batch of jobs arrives.

III. VALUE GENERATING FUNCTION FOR WAITING TIME

In this section, we consider the M^x/G/1 queue with a general cost structure that is some function of the waiting time. The main result is Proposition 1, which characterizes the corresponding value function, defined later, in a compact and useful expression.

Definition 2 (Cost structure) *Each job incurs a cost immediately upon arrival defined by a cost function $c(w)$, where w denotes the job's waiting time.*

Typically, $c(w)$ is non-negative and increasing, even though the results also hold more generally. Some examples are given in Table I.

Remark 1 *Several generalizations are possible with very little effort. First, the cost function can be a job- or batch-specific random function (i.i.d.), allowing, e.g., job classes with different weights, and taking into account the job's service time (cf. slowdown, see Table I). Second, the cost function can also depend on the position of the job in the batch. Third, the cost structures can be server-specific.*

The systems we consider in this paper are *size-aware*, which means that service times become known upon arrival and, consequently, the state of the queue under FCFS can be described by the backlog (unfinished work) u. Let W be the waiting time of a random job in steady state. Note that W includes both the batch waiting time, i.e., the time until the first job in the batch starts service, and the with-in batch waiting time. We defer the detailed analysis of this breakdown to Section IV.

We let r denote the mean cost rate (per unit time) and $\bar{c} = \mathrm{E}[c(W)]$ the mean cost incurred per arriving job,

$$r \triangleq \lambda \, \mathrm{E}[c(W)] = \lambda \, \bar{c}. \tag{2}$$

We carry out our analysis in the context of Markov decision processes. The central concept is the value function.

Definition 3 (Value function) *The value function is the expected cost difference in the infinite time-horizon between a system initially in state u and a system in equilibrium,*

$$v(u) \triangleq \lim_{t \to \infty} \mathrm{E}[V(u, t) - rt], \tag{3}$$

where the random variable $V(u, t)$ denotes the costs incurred during time $(0, t)$ when the system is initially in state u.

We assume that $\rho < 1$, so that the M^x/G/1 queue is stable and the system ergodic, and that the above limit exists and is finite. In general, a constant term in $v(u)$ is irrelevant as the important quantity for making decisions is the difference $v(u+x) - v(u)$. Most of our results are given for $v(u) - v(0)$, which characterizes the expected deviation (in terms of costs) from the system that is initially empty, whereas $v(u)$ characterizes the expected deviation from the mean cost (rate).

Proposition 1 *The value function $v(u)$ for the M^x/G/1 queue with i.i.d. setup delays D and admission costs defined by an arbitrary function $c(w)$ satisfies*

$$v(u) - v(0) = \frac{\lambda u}{1-\rho}\,\mathrm{E}[c(W_0 + Y) - c(W)], \qquad (4)$$

where W and W_0 denote waiting time of a job in steady state with and without the setup delay, and $Y = Y(u) \sim \mathrm{U}(0,u)$ and independent of W_0.

To keep the proof of Proposition 1 short, we first prove the following Lemma regarding the total cost S accumulated during a busy period when the cost of an individual job is $c(W)$, where W is the waiting time of the job.

Lemma 1 *Consider an M^x/G/1 queue with i.i.d. setup delays $D_i \sim D$. Jobs incur costs according to $c(W)$, where W denotes job's waiting time and $c(w)$ is an arbitrary function $\mathbb{R}^+ \to \mathbb{R}$. Let random variable S denote the cost incurred during a busy period. Then*

$$\mathrm{E}[S] = \frac{(1+\lambda_b\,\mathrm{E}[D])\mathrm{E}[B]}{1-\rho}\,\mathrm{E}[c(W)]. \qquad (5)$$

Proof: Recall that batches arrive at rate λ_b and jobs at rate $\lambda = \lambda_b\,\mathrm{E}[B]$. The mean cost rate is by definition

$$r = \lambda\,\mathrm{E}[c(W)] = \lambda_b\,\mathrm{E}[B]\,\mathrm{E}[c(W)]. \qquad (6)$$

Consider a renewal process where the renewal point is the time instant when the system becomes idle. The renewal interval is the sum of an idle and busy period, so that

$$\mathrm{E}[T_r] = \frac{1}{\lambda_b} + \frac{\mathrm{E}[H+D]}{1-\rho} = \frac{1+\lambda_b\,\mathrm{E}[D]}{\lambda_b(1-\rho)},$$

where H is the service time of a batch. Hence, an alternative expression for the mean cost rate is

$$r = \frac{\mathrm{E}[S]}{\mathrm{E}[T_r]} = \frac{\lambda_b(1-\rho)}{1+\lambda_b\,\mathrm{E}[D]} \cdot \mathrm{E}[S]. \qquad (7)$$

Combining (6) and (7) yields (5). □

The following corollary allows us to include an offset in the cost function as needed in our proof of Proposition 1.

Corollary 1 *When jobs incur costs according to $c(W+y)$, where y is a given constant, then noting that $\tilde{c}(w) \triangleq c(w+y)$ is just another cost function, it follows that the mean cost incurred during a busy period is*

$$\mathrm{E}[S(y)] = \frac{(1+\lambda_b\,\mathrm{E}[D])\mathrm{E}[B]}{1-\rho}\,\mathrm{E}[c(W+y)],$$

and if the server has no setup delay ($D = 0$), then the mean cost incurred during a busy period is

$$\mathrm{E}[S_0(y)] = \frac{\mathrm{E}[B]}{1-\rho}\,\mathrm{E}[c(W_0+y)], \qquad (8)$$

where W_0 denotes job's waiting time in the M^x/G/1 queue without a setup delay.

With these, we are ready to prove Proposition 1:

Proof: [Proposition 1] Let T_u denote the remaining time of the busy period when currently in state u. By definition,

$$\begin{aligned} v(u) &= \lim_{t\to\infty} \mathrm{E}[V(u,t) - rt] \\ &= \mathrm{E}[V(u) - T_u\, r] + \lim_{t\to\infty} \mathrm{E}[V(0,t) - rt] \\ &= \mathrm{E}[V(u)] - \mathrm{E}[T_u]\, r + v(0), \end{aligned}$$

where $\mathrm{E}[V(u)]$ denotes the mean cost incurred during the remaining busy period, $r = \lambda\,\mathrm{E}[c(W)]$ is the mean cost rate (with the setup delay), and $\mathrm{E}[T_u] = u/(1-\rho)$. Hence,

$$\mathrm{E}[T_u]\, r = \frac{\lambda u}{1-\rho}\,\mathrm{E}[c(W)], \qquad (9)$$

where W corresponds to the waiting time in the actual system subject to a possible setup delay D.

Consider next the first term $\mathrm{E}[V(u)]$. When a batch of jobs arrives before the system is empty, a new mini busy period is started that will end when the backlog returns to the same level as it was before (see Figure 2 (left)). As the server was already processing jobs, the mini busy periods do not involve any setup delays. Let N_b denote the number of mini busy periods, and Y_i the original unfinished work u in system when the ith mini busy period starts. With these,

$$\mathrm{E}[V(u)] = \mathrm{E}[\sum_{i=1}^{N_b} S_0(Y_i)]. \qquad (10)$$

Referring to Figure 2 (right), we see that $Y_i = u - T_i$, where the T_i represent the instants where the service of the original unfinished work u is interrupted in a system where the durations of these interruptions are squeezed to zero. The intervals between successive instants T_i are exponentially (and independently) distributed (taking $0 = T_0$) and consequently constitute a Poisson process with intensity λ_b. Given the number N_b of arrivals from this Poisson process in the interval $(0,u)$, the set of arrival instants may be obtained drawing instants $\tilde{T}_i$, $i = 1,\ldots,N_b$, independently of the others, from the uniform distribution $\mathrm{U}(0,u)$. Consequently, the set of values $\tilde{Y}_i$ is obtained drawing each of them independently from the distribution $U(0,u)$. We denote by Y a generic variable of this kind, $Y(u) \sim \mathrm{U}(0,u)$. The actual values Y_i represent an ordered set of the $\tilde{Y}_i$. Ordering does not affect the sum in

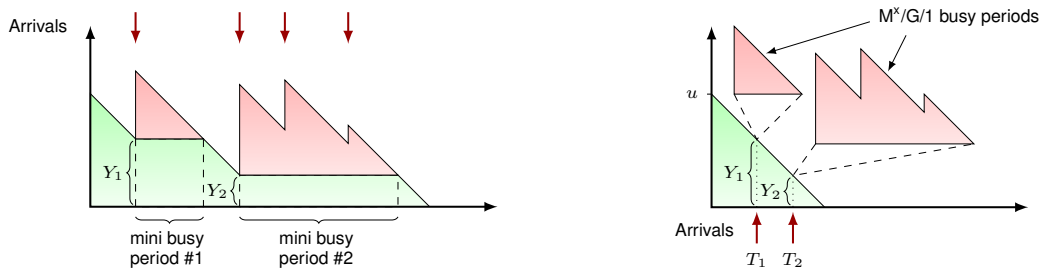


Fig. 2. A sample realization with two mini busy periods until the server becomes idle (left). Mini busy periods interrupting the service of the initial backlog of u originate from the Poisson process with rate λ_b (right).

(10), and noting that $\mathrm{E}[N_b] = \lambda_b u$ and using the result (8), we can write (10) in the form

$$\begin{aligned}\mathrm{E}[V(u)] &= \mathrm{E}[\sum_{i=1}^{N_b} S_0(Y)] \\ &= \mathrm{E}[\mathrm{E}[\sum_{i=1}^{N_b} S_0(Y_i) \,|\, N_b]] \\ &= \mathrm{E}[N_b] \cdot \mathrm{E}[S_0(Y)] = \mathrm{E}[N_b] \cdot \mathrm{E}[\mathrm{E}[S_0(Y)] \,|\, Y] \\ &= \lambda_b u \cdot \frac{\mathrm{E}[B]}{1-\rho}\,\mathrm{E}[c(W_0+Y)] = \frac{\lambda u}{1-\rho}\,\mathrm{E}[c(W_0+Y)],\end{aligned}$$

where W_0 denotes the waiting time in the system with no setup delay (mini busy periods do not involve setup delay). Combining the above with (9) completes the proof. □

The above yields immediately a corollary (see [4]):

Corollary 2 *Setup delay D gives rise to an additive term in the value function,*

$$\begin{aligned}&v(u) - v(0) = \\ &\quad v^{(0)}(u) - v^{(0)}(0) + \frac{\lambda u}{1-\rho}\,\mathrm{E}[c(W_0) - c(W)],\end{aligned} \tag{11}$$

where $v(u)$ and $v^{(0)}(u)$ denote the value functions with and without the setup delay of D, and W and W_0 denote the waiting time with and without the setup delay.

As $Y \sim \mathrm{U}(0,u)$ in (4), we can write explicitly that

$$v(u) - v(0) = \frac{\lambda}{1-\rho}\int_0^u \mathrm{E}[c(W_0+y) - c(W)]\,dy. \tag{12}$$

Thus,

$$v'(u) = \frac{\lambda}{1-\rho}\,\mathrm{E}[c(W_0+u) - c(W)]. \tag{13}$$

In general, if a random variable Z and $Y \sim \mathrm{U}(0,u)$ are independent, then for any differentiable[1] (cost) function $c(\cdot)$ with integrable derivative the following holds identically:

$$u\,\mathrm{E}[c'(Z+Y)] = \mathrm{E}[c(Z+u)] - \mathrm{E}[c(Z)].$$

Without setup delay, $W \sim W_0$, and (13) thus reduces to

$$v'(u) = \frac{\lambda u}{1-\rho}\,\mathrm{E}[c'(W_0+Y)]. \tag{14}$$

[1]The cost function $c(w)$ can also be non-differentiable at countably many points. Consider, e.g., the deadline cost $c_\tau(w)$ given in Table I.

Note that (13) and (14) are first order ordinary differential equations and, provided that the expectations on the right hand side can be evaluated, they can easily be integrated to yield a unique solution for the difference $v(u) - v(0)$.

IV. Exponential cost function

Let us next consider a cost function $c_W(w,s)$ defined as

$$c_W(w,s) = 1 - e^{-sw}, \qquad s > 0, \tag{15}$$

where w is the waiting time of the job (the backlog u observed by each job), and $s > 0$ is a free parameter. This cost could be interpreted as the probability the job completes before its deadline, where its deadline has an exponential distribution. Note that $c_W(0,s) = 0$.

A. Generating functions

It turns out that the cost function (15) is closely related to the Laplace-Stieltjes transform, and before continuing, let us first introduce some additional notation and results.

Definition 4 *For non-negative discrete random variable B, the generating function (z-transform) is*

$$\mathcal{G}_B(z) \triangleq \mathrm{E}[z^B].$$

Definition 5 (LST) *The Laplace-Stieltjes transform of the random variable $X \geq 0$ is [13], [14]*

$$X^*(s) \triangleq \mathrm{E}[e^{-sX}], \qquad s > 0.$$

Throughout we will use the asterisk to denote the transform of a random variable. As is well known, see, e.g., [13], [14], these transforms, when defined, encompass all the necessary information about a random variable in a single function and the LST of the sum of two independent continuous random variables, $Z = X + Y$, is $Z^*(s) = X^*(s) \cdot Y^*(s)$.

B. M/G/1 in transform domain

Next we will state some well-known basic results for the M/G/1 queue. Pollaczek-Khinchin transform formula for the waiting time in the M/G/1 queue (i.e., without setup delay) is

$$W_0^*(s) = \frac{(1-\rho)s}{s - \lambda(1 - X^*(s))}. \tag{16}$$

From [15], we find the LST of the waiting time for the M/G/1 queue, where the first job has an *exceptional service time*,

$$W_x^*(s) = p_0 \cdot \frac{s + \lambda(X^*(s) - X_0^*(s))}{s - \lambda(1 - X^*(s))},$$

where X_0 is the service time of the job starting the busy period, X the service time of all other jobs, and p_0 is the probability that the system is idle,

$$p_0 = \frac{1 - \lambda \,\mathrm{E}[X]}{1 - \lambda(\mathrm{E}[X] - \mathrm{E}[X_0])}.$$

Note that the waiting time of the first job of the busy period is still zero in this model. For the M/G/1 with setup delay, $X_0 = X + D$, and

$$p_0 = \frac{1 - \rho}{1 + \lambda \,\mathrm{E}[D]}. \tag{17}$$

and also the first job experiences the setup delay D, yielding

$$\begin{aligned} W^*(s) &= W_x^*(s) + p_0(D^*(s) - 1) \\ &= \frac{1 - \rho}{1 + \lambda \,\mathrm{E}[D]} \cdot \frac{\lambda + D^*(s)(s - \lambda)}{s - \lambda(1 - X^*(s))}, \end{aligned} \tag{18}$$

which means that

$$W^*(s) = W_0^*(s) \cdot W_e^*(s), \tag{19}$$

where

$$W_e^*(s) = \frac{\lambda + D^*(s)(s - \lambda)}{s(1 + \lambda \,\mathrm{E}[D])},$$

corresponds to the extra waiting time due to the setup delay. Fuhrmann's and Cooper's decomposition property, shown in (19), holds for a large class of M/G/1 queues with different vacation models [16], [17]. (The setup delay D can also be interpreted as one.) In the following section, we present such results for the M^x/G/1 queue that are the needed for our later developments with respect to different value functions.

C. M^x/G/1 in transform domain

Consider next the batch arrival process. The probability that a randomly chosen job belongs to a size i batch is $i \,\mathrm{P}\{B = i\}/\mathrm{E}[B]$. The probability that a job is the ith job of a batch is

$$b_i \triangleq \sum_{j=i}^{\infty} \frac{j \,\mathrm{P}\{B = j\}}{\mathrm{E}[B]} \frac{1}{j} = \frac{\mathrm{P}\{B \geq i\}}{\mathrm{E}[B]}.$$

For a randomly chosen job, let A denote the number of jobs there are ahead of it in the same batch,

$$\mathrm{P}\{A = i\} = b_{i+1} = \frac{\mathrm{P}\{B \geq i + 1\}}{\mathrm{E}[B]}.$$

and the corresponding generating function is

$$\mathcal{G}_A(z) = \mathrm{E}[z^A] = \frac{1}{z \,\mathrm{E}[B]} \sum_{i=1}^{\infty} \mathrm{P}\{B \geq i\} \, z^i.$$

The service time of the whole batch is a random sum, $H = X_1 + \ldots + X_B$, for which we have

$$\begin{aligned} \mathrm{E}[H] &= \mathrm{E}[B] \cdot \mathrm{E}[X], \\ \mathrm{E}[H^2] &= \mathrm{E}[B] \cdot \mathrm{V}[X] + \mathrm{E}[B^2] \cdot \mathrm{E}[X]^2, \\ H^*(s) &= \mathcal{G}_B(X^*(s)), \end{aligned}$$

where $\mathrm{V}[X]$ is the variance of X.

1) Without setup delay: For M^x/G/1, the waiting time of a randomly chosen job is the sum of the batch-level waiting time and the service times of those jobs from the same batch that happen to be ahead of the given job. For the former, we can utilize (16),

$$W_b^*(s) = \frac{s(1 - \rho)}{s - \lambda_b(1 - \mathcal{G}_B(X^*(s)))},$$

and the latter is given by $\mathcal{G}_A(X^*(s))$. Thus the LST for the waiting time of a job is given by

$$W_0^*(s) = W_b^*(s) \cdot \mathcal{G}_A(X^*(s)),$$

yielding

$$W_0^*(s) = \frac{(1 - \rho)s \,\mathcal{G}_A(X^*(s))}{s - \lambda_b(1 - \mathcal{G}_B(X^*(s)))}. \tag{20}$$

2) With setup delay: Again, the M/G/1 result gives the waiting time for the batch in the context of M^x/G/1. The first batch of the busy period increases the backlog to $H + D$, whereas the size of the later batches is H. Consequently, from (19) we obtain the LST for the batch-level waiting time with a setup delay, $W_b^*(s) \cdot W_e^*(s)$, where the LST for the extra delay (penalty) due to the random setup time D is now given by

$$W_e^*(s) = \frac{\lambda_b + (s - \lambda_b) \, D^*(s)}{(1 + \lambda_b \,\mathrm{E}[D])s}. \tag{21}$$

Furthermore, the individual jobs may have to wait additionally until the jobs ahead of them within the same batch have been served. Thus,

$$W_j^*(s) = W_b^*(s) \cdot W_e^*(s) \cdot \mathcal{G}_A(X^*(s)), \tag{22}$$

where $\mathcal{G}_A(X^*(s))$ corresponds to the waiting time within the batch. We can also write

$$W_j^*(s) = W_0^*(s) \cdot W_e^*(s).$$

Note that W_e depends on the batch arrival rate λ_b and setup delay D, but not on the batch size, offered load ρ, or the service time distribution X. For example, the penalty in terms of the mean sojourn time is

$$\lim_{s \to 0^+} W_e^{*\prime}(s) = \frac{2 \,\mathrm{E}[D] + \lambda_b \,\mathrm{E}[D^2]}{2(1 + \lambda_b \,\mathrm{E}[D])}. \tag{23}$$

Example 1 *Suppose we have the M^x/G/1 queue with geometrically distributed batch size, $B \sim \mathrm{Geo}_1(q)$, so that*

$$\mathrm{E}[B] = \frac{1}{q}, \quad \textit{and} \quad \mathrm{E}[B^2] = \frac{2 - q}{q^2}.$$

Then, for A we have

$$\mathrm{P}\{A=i\} = \frac{\mathrm{P}\{B \geq i+1\}}{\mathrm{E}[B]} = \frac{(1-q)^i}{1/q} = (1-q)^i q,$$

i.e., $A \sim \mathrm{Geo}_0(q)$. This can be deduced also directly by taking a sequence of Bernoulli trials, each succeeding with probability of q. The number of failed trials preceding a successful trial obeys $\mathrm{Geo}_0(q)$ distribution. Hence, the mean number of jobs ahead of a random job in a batch is

$$\mathrm{E}[A] = \frac{1-q}{q}.$$

The mean sojourn time of a job in this M^x/G/1 queue is

$$\mathrm{E}[T] = \overbrace{\frac{2\,\mathrm{E}[D] + \lambda_b\,\mathrm{E}[D^2]}{2(1+\lambda_b\,\mathrm{E}[D])}}^{\textit{setup penalty}} + \overbrace{\frac{\lambda_b\,\mathrm{E}[H^2]}{2(1-\rho)}}^{\textit{batch waits}} + \overbrace{\frac{1-q}{q}\cdot\mathrm{E}[X]}^{\textit{job waits}},$$

where $\mathrm{E}[H^2] = \mathrm{E}[X^2]/q + \mathrm{E}[X]^2 \cdot 2(1-q)/q^2$, yielding

$$\mathrm{E}[T] = \overbrace{\frac{2\,\mathrm{E}[D] + \lambda_b\,\mathrm{E}[D^2]}{2(1+\lambda_b\,\mathrm{E}[D])}}^{\textit{setup penalty}} + \overbrace{\frac{\lambda\,\mathrm{E}[X^2]}{2(1-\rho)}}^{\textit{M/G/1}} + \overbrace{\frac{1-q}{q(1-\rho)}\,\mathrm{E}[X]}^{\textit{batch penalty}}.$$

Note that as $q \to 1$, batch sizes reduce to 1 and the batch penalty disappears. Similarly, when $D = 0$, the setup penalty vanishes.

D. Value functions for exponential cost function

Next we will give the main result of this section, the value function for the M^x/G/1 queue subject to the exponential cost function:

Proposition 2 (waiting time) *For the M^x/G/1 queue with batch arrival rate λ_b, batch size B, i.i.d. setup time D, and a generally distributed service time X, subject to the (job-specific) admission cost function* (15)*, the mean cost is*

$$\bar{c}_W(s) = 1 - W_j^*(s), \tag{24}$$

with $W_j^*(s)$ given in (22)*, and the corresponding value function is*

$$v_W(u,s) - v_W(0,s) = \frac{\mathrm{E}[B]\,\mathcal{G}_A(X^*(s))}{s/\lambda_b + \mathcal{G}_B(X^*(s)) - 1} \cdot \left[e^{-su} - 1 + \frac{s + (1-D(s))(\lambda_b - s)}{1+\lambda_b \mathrm{E}[D]}u\right]. \tag{25}$$

Proof: The mean cost with (15) is

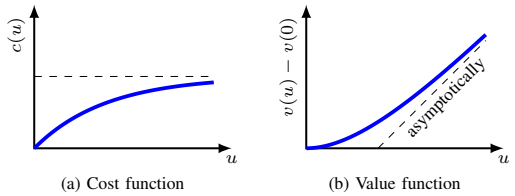
$$\mathrm{E}[1 - e^{-sW}] = 1 - W_j^*(s),$$

where $W_j^*(s)$ is given in (22). Similarly, the value function is obtained by substituting (15) into (4),

$$\begin{aligned} v_W(u,s) - v_W(0,s) &= \frac{\lambda u}{1-\rho}\,\mathrm{E}[c(W_0+Y) - c(W)], \\ &= \frac{\lambda u}{1-\rho}\,\mathrm{E}[e^{-sW} - e^{-sW_0}e^{-sY}], \end{aligned}$$

and thus,

$$v_W(u,s) - v_W(0,s) = \frac{\lambda u}{1-\rho}\,(W_j^*(s) - W_0^*(s)\,Y^*(s)). \tag{26}$$



(a) Cost function (b) Value function

Fig. 3. Bounded and strictly increasing cost function and the corresponding value function.

The LST of $Y = Y(u)$ is

$$Y^*(s) = \frac{1 - e^{-su}}{su},$$

and $W_j^*(s)$ and $W_0^*(s)$ are given in (22) and (20), respectively. Substituting these into (26) yields (25). □

Without setup delay, $\mathrm{E}[D] = 0$ and $D^*(s) = 1$, and the expressions simplify considerably:

Corollary 3 *For the M^x/G/1 queue with batch arrival rate λ_b, batch size B, no setup delay, and a generally distributed service time X, subject to the (job-specific) admission cost function* (15)*, the mean cost is*

$$\bar{c}_W(s) = 1 - \frac{(1-\rho)s\,\mathcal{G}_A(X^*(s))}{s - \lambda_b(1 - \mathcal{G}_B(X^*(s)))}, \tag{27}$$

and the corresponding value function is

$$v_W(u,s) - v_W(0,s) = \frac{(su + e^{-su} - 1)\mathrm{E}[B]\,\mathcal{G}_A(X^*(s))}{s/\lambda_b + \mathcal{G}_B(X^*(s)) - 1}. \tag{28}$$

Corollary 4 *For the M/G/1 queue with arrival rate λ, generally distributed service time X and no setup delay, subject to admission cost function* (15)*, the mean cost is*

$$\bar{c}_W(s) = \frac{s\,\mathrm{E}[X] + X^*(s) - 1}{s/\lambda + X^*(s) - 1}, \tag{29}$$

and the corresponding value function is

$$v_W(u,s) - v_W(0,s) = \frac{su + e^{-su} - 1}{s/\lambda + X^*(s) - 1}. \tag{30}$$

Note that at the heavy-traffic limit when $\rho \to 1$, $1/\lambda \to \mathrm{E}[X]$ and (29) converges to 1, as expected. Figure 3 illustrates an example cost function of type (15) and the corresponding value function given by (30).

Example 2 *Consider next the M^x/G/1 queue without setup delay and Corollary 3. Dividing $c(w)$ by s, and letting $s \to 0$, one obtains the waiting time metric $c_1(w) = w$ [8],*

$$v_1(u) - v_1(0) = \frac{\lambda\,u^2}{2(1-\rho)}, \quad \bar{c}_1 = \frac{\lambda_b\,\mathrm{E}[H^2]}{2(1-\rho)} + \mathrm{E}[A]\mathrm{E}[X].$$

Example 3 *Consider then the ordinary M/G/1 queue without setup delay and Corollary 4. Dividing $c(u)$ by λ, and then*

taking the limit $s \to \infty$ yields the energy-consumption model from [12], $c_e(u) = \mathbf{1}(u > 0)$, and

$$v(u) - v(0) = u, \quad \bar{c}_e = \rho,$$

where $c_e(u) = \lambda\, c(u)$ is the cost rate in state u (zero if idle, otherwise one), and $\bar{c}_e$ is the mean cost rate (probability that the server is busy).

What we have just discovered is similar to LST and moment generating functions of random variables; elementary and compact expressions for the mean cost and value function for the M^x/G/1 queue subject to a rather general cost structure. From these results, many interesting special cases can be immediately obtained. First note that $c_W(w,s) = 1 - e^{-sw} = sw - (sw)^2/2! + \ldots$, so

$$c_W(w,s) = \sum_{i=1}^{\infty} (-1)^{i+1} \frac{s^i}{i!} c_i(w),$$

where the $c_i(w)$ are the polynomial cost functions for the *waiting time*,

$$c_i(w) = w^i, \qquad i = 1, 2, \ldots \tag{31}$$

Consequently,

$$v_W(u,s) = \sum_{i=1}^{\infty} (-1)^{i+1} \frac{s^i}{i!} v_i(u),$$

where the $v_i(u)$ are the value functions corresponding to costs $c_i(w)$. Therefore, in analogy with LST of a random variable, we can compute any $v_i(u)$ directly from $v_W(u,s)$,

$$v_i(u) = (-1)^{i+1} \lim_{s \to 0} \frac{d^i}{ds^i} v_W(u,s).$$

That is, the value function $v_W(u,s)$, given in (25), is the *generating function for the family of value functions* with respect to the polynomial costs (31). In the remainder of this paper, we refer to $v(u,s) = v_w(u,s)$ simply as the **value generating function** for waiting time in order to keep the discussion short.

The same steps can be taken also with the mean costs, i.e., it is straightforward to compute an arbitrary moment $\mathrm{E}[W^k]$ of the waiting time in the M^x/G/1 queue by using the identity

$$\bar{c}(s) = \mathrm{E}[c(W)] = \sum_{i=1}^{\infty} (-1)^{i+1} \frac{s^i}{i!} \mathrm{E}[W^i].$$

together with (24).

V. Value generating function for sojourn time

More generally, the cost paid upon arrival may depend also on the service time x of the arriving job, $c = c(w,x)$.

Proposition 3 *The value function $v(u)$ for the M^x/G/1 queue with i.i.d. setup delays D and admission costs defined by an arbitrary function $c(w,x)$ satisfies*

$$v(u) - v(0) = \frac{\lambda u}{1-\rho} \mathrm{E}[c(W_0 + Y, X) - c(W, X)], \tag{32}$$

where W and W_0 denote the waiting time of a job with and without the setup delay, X is the service time, and $Y = Y(u) \sim \mathrm{U}(0,u)$ is independent of W_0 and X.

Proof: Similar to that of Proposition 1, and omitted for brevity. □

Next we study a cost structure, where, instead of waiting time, the costs are defined as an exponential function of the sojourn time (response time),

$$c_T(w,x,s) = 1 - e^{-s(w+x)}, \qquad s > 0, \tag{33}$$

where w is the waiting time of the job, and x the service time of the job, and s is a free parameter.

Proposition 4 (sojourn time) *For the M^x/G/1 queue with batch arrival rate λ_b, batch size B, i.i.d. setup time D, and a generally distributed service time X, subject to the (job-specific) admission cost function* (33)*, the mean cost is*

$$\bar{c}_T(s) = 1 - W_j^*(s) \cdot X^*(s), \tag{34}$$

with $W_j^*(s)$ given in (22)*, and the corresponding value function is*

$$v_T(u,s) - v_T(0,s) = (v_W(u,s) - v_W(0,s))\, X^*(s). \tag{35}$$

Proof: The mean cost is

$$\bar{c}_T(s) = \mathrm{E}[c_T(U,X,s)] = 1 - W_j^*(s) X^*(s).$$

Substituting $c_T(u,x)$ into (32), one obtains

$$v_T(u,s) - v_T(0,s) = \frac{\lambda u}{1-\rho} \mathrm{E}[e^{-s(W+X)} - e^{-s(W_0+Y+X)}]$$

which yields (35). □

Corollary 5 (sojourn time for M/G/1) *For the M/G/1 queue with arrival rate λ and general service time distribution X, subject to costs* (33)*, the mean cost is*

$$\bar{c}_T(s) = \frac{s\,\mathrm{E}[X] + X^*(s) - 1}{s/\lambda + X^*(s) - 1}\, X^*(s), \tag{36}$$

and the corresponding value function is

$$\begin{aligned} v_T(u,s) - v_T(0,s) &= (v_W(u,s) - v_W(0,s))\, X^*(s) \\ &= \frac{su + e^{-su} - 1}{s/\lambda + X^*(s) - 1}\, X^*(s). \end{aligned} \tag{37}$$

A. Summary of the relationships

We have obtained compact expressions for the **value generating functions** with respect to the waiting and sojourn time, which we summarize here. For the M^x/G/1 queue with setup delay D, the value generating functions are

$$v_W(u,s) - v_W(0,s) = \frac{\lambda u}{1-\rho} \left(W_e^*(s) - \frac{1 - e^{-su}}{su} \right) W_0^*(s),$$

$$v_T(u,s) - v_T(0,s) = \frac{\lambda u}{1-\rho} \left(W_e^*(s) - \frac{1 - e^{-su}}{su} \right) T_0^*(s),$$

where $W_0^*(s)$ and $T_0^*(s) = W_0^*(s) X^*(s)$ denote the LST of the waiting and sojourn time in the M^x/G/1 queue without a setup delay, and $W_e^*(s)$ is the LST of the additional waiting time due to setup delay W_e and given in (21).

VI. Applications

In this section, we show how the new results can be utilized.

A. To switch off, or not to switch off

Let us first discuss briefly switching-off policies. We determine if a server becoming idle should be switched off or kept running. Inherently, this is a question about the energy-performance trade-off. By switching off, we hope to save some energy, but at the same time, response times increase. We let e denote the cost of energy per unit time incurred when the server is running. The mean energy cost if the server is switched-off follows immediately from (17),

$$\bar{c}_e = (1-p_0)e = \frac{\rho + \lambda_b \operatorname{E}[D]}{1+\lambda_b \operatorname{E}[D]}\, e. \tag{38}$$

The value function with respect to energy consumption is [8],

$$v_e(u) - v_e(0) = \frac{u}{1+\lambda_b \operatorname{E}[D]}\, e. \tag{39}$$

The QoS is characterized by an arbitrary cost function, e.g., a polynomial

$$c(w) = \sum_i a_i w^i.$$

Then let the random variable $C(w)$ denote the total cost incurred by a batch,

$$C(w) \triangleq c(w) + c(w+X_1) + \ldots + c(w+X_1+\ldots+X_{B-1}).$$

The optimal switch off decision can be deduced in (at least) three different ways:

1) The straightforward method is to compute the mean cost $\bar{c}$ for system (i) that keeps the server running and thus avoids setup delays, and system (ii) that switches the server off when idle and thereby exposes some jobs to the ensuing setup delay. Given the mean costs, one can simply choose the strategy that has lower mean cost.
2) Suppose that we decide to keep the server always running, and have computed the corresponding value function for the whole system (without setup delay), denoted by $v^{(0)}(u)$. The energy consumption is fixed at a constant level e, and it thus does not show up in the value function. Then consider deviating from the standard action by switching the server off for one idle period. Switching off decreases the expected costs if
$$\operatorname{E}[C(D) - C(0) + v^{(0)}(D+H) - v^{(0)}(H)] < \frac{e}{\lambda_b}.$$
3) Alternatively, let $v(u)$ denote the value function of the system when the policy is to switch the server off whenever it becomes idle. In this case, the energy consumption, given by (39), is included explicitly in the value function $v(u)$. Then consider keeping the server running when the last customer exits, which is equivalent to keeping the backlog artificially in state $u=\epsilon$ by feeding very small (infinitesimal) virtual jobs until the next real job arrives. Thus, keeping the server running is advantageous if $v(\epsilon) < v(0)$, which yields
$$\frac{v(\epsilon) - v(0)}{\epsilon} < 0,$$
and as $\epsilon \to 0$, the condition reduces to checking the sign of the derivative at $u=0$,
$$\left[\frac{d}{du} v(u)\right]_{u=0} < 0.$$
Considering the weighted cost model for jobs, with the aid of the value generating function (25), we can write an explicit expression for keeping the server running,
$$\left[\frac{d}{du}\sum_i a_i \left[\frac{d^i}{ds^i} v(u,s)\right]_{s=0}\right]_{u=0} > \frac{e}{1+\lambda_b \operatorname{E}[D]}.$$
The summation on the left-hand side is a generic cost function related to the QoE as perceived by jobs, and the term on the right-hand side with the constant factor e corresponds to the energy consumption.

From each of the above, we can deduce the (same) critical cost for energy above which the server should be switched off when idle. Note that in the latter two cases, we only need to analyze one system, whereas with the first option one needs to compute the mean cost for two systems. Because we have Poisson arrivals we need not consider policies that switch a server off and switch it back on again before there is an arrival. Similarly, we ignored the policies that delay the switch on after a job has arrived.

B. Dispatching policies

The value function for the M^x/G/1 queue is also useful in the context of parallel servers (see Figure 1), where jobs arriving according to a Poisson batch process are routed immediately upon arrival to servers. Jobs of a batch can be split to multiple servers (optionally) in different order than they were in the batch. Moreover, servers can be heterogeneous with individual cost structures and service times.

The value function of the M^x/G/1 queue can be used to develop efficient dispatching (routing) policies by carrying out one policy improvement step. This method, first proposed in [18], has been used in numerous papers, see, e.g., [19]–[22], and [23, Sec.10.3. and 11.5]. The basic steps are as follows:

1) Choose an appropriate *basic dispatching policy*, denoted with α_0, such that the arrival process to each server is a (server-specific) Poisson batch arrival process.
2) With α_0, the Poisson batch arrival process is split into n independent streams, with server-specific batch size distributions, and the system *decomposes* into n parallel M^x/G/1 queues that can be analyzed independently.
3) The value function for each M^X/G/1 can be computed, and the system's value function is the sum of those,
$$v(\mathbf{z}) = v_1(u_1) + \ldots + v_n(u_n),$$
where $\mathbf{z} = (u_1, \ldots, u_n)$ denotes the state of the system (here the backlogs u_i is a sufficient state description).

4) The one step policy improvement yields a new policy. Let $\xi = \xi(\mathbf{x})$ denote a routing decision that assigns all jobs of a batch $\mathbf{x}$ in some order to the servers, and $c(\mathbf{z}, \xi)$ the corresponding immediate cost, $\mathbf{z} \oplus \xi$ the resulting state. Then

$$\alpha_{\text{FPI}}(\mathbf{z}, \mathbf{x}) \in \arg\min_{\xi} \left(c(\mathbf{z}, \xi) + v(\mathbf{z} \oplus \xi) - v(\mathbf{z}) \right).$$

Ties can be resolved, e.g., randomly.

The basic dispatching decision can depend on any batch-specific parameters such as size, class, weight or priority. The basic policy can also *split* batches to several servers as long as the decision is static and each server receives a Poisson batch arrival process. One example is the size-interval-task-assignment (SITA) policy that dispatches, e.g., all short jobs to server 1 and long jobs to server 2 [24]. Similarly, a random Bernoulli split to two or more servers decreases the burstiness in the arrival process, which in turn tends to improve the performance with FCFS. However, the static policy may not depend on the state of the system (i.e., the backlogs in the servers), or on any parameters of a batch in such a way that the arrival process to some server is not a Poisson batch process.

Unfortunately, it is often difficult to carry out more than the first policy improvement step, because the first step already yields a dynamic policy for which it is significantly harder to compute a value function. In this case, one can consider the lookahead approach, where the tentative action deviating from the basic policy includes also the next arriving job(s) [25].

VII. Conclusions

We have studied the M^x/G/1 queue subject to random setup delay and general cost structures, and obtained compact expressions for the value function defined in terms of job-specific waiting and sojourn times. Then a particular attention was paid to exponential cost structures with a free parameter s. In this case, we obtained closed-form expressions, defined solely in terms of the arrival rate, the batch-size specific generating functions $\mathcal{G}_A(z)$ and $\mathcal{G}_B(z)$, and LST of the service time distribution, $X^*(s)$, for the mean costs and the value functions. These cost functions include many often used cost structures such as the waiting time, the slowdown, and energy consumption. We refer to the value functions as *value generating functions*, because, as with, e.g., moment generating functions, they summarize whole families of cost structures and their corresponding value functions. Moreover, as the exponential cost functions have convenient simple expressions for the mean costs and value functions, they also serve as a building block for more complex scenarios.

Acknowledgements

This work was supported by the Academy of Finland in the FQ4BD and TOP-Energy projects (grant nos. 296206 and 268992). We thank the anonymous referees and our shepherd, Mark Squillante, for helping to improve the paper.

References

[1] 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.
[2] N. Argon, L. Ding, K. Glazebrook, and S. Ziya, "Dynamic routing of customers with general delay costs in a multiserver queuing system," *Probability in the Engineering and Informational Sciences*, vol. 23, pp. 175–203, 2009.
[3] E. Hyytiä, S. Aalto, and A. Penttinen, "Minimizing slowdown in heterogeneous size-aware dispatching systems," *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, pp. 29–40, Jun. 2012, (ACM SIGMETRICS/Performance conference).
[4] E. Hyytiä, R. Righter, and S. Aalto, "Task assignment in a heterogeneous server farm with switching delays and general energy-aware cost structure," *Performance Evaluation*, vol. 75–76, no. 0, pp. 17–35, May-June 2014.
[5] R. R. Weber, "On the optimal assignment of customers to parallel servers," *Journal of Applied Probability*, vol. 15, no. 2, pp. 406–413, Jun. 1978.
[6] A. Hordijk and G. Koole, "On the optimality of the generalised shortest queue policy," *Prob. Eng. Inf. Sci.*, vol. 4, pp. 477–487, 1990.
[7] O. Akgun, R. Righter, and R. Wolff, "Multiple server system with flexible arrivals," *Advances in Applied Probability*, vol. 43, pp. 985–1004, 2011.
[8] E. Hyytiä, 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.
[9] M. Harchol-Balter, K. Sigman, and A. Wierman, "Asymptotic convergence of scheduling policies with respect to slowdown," *Perform. Eval.*, vol. 49, no. 1-4, pp. 241–256, Sep. 2002.
[10] E. Hyytiä and R. Righter, "Routing jobs with deadlines to heterogeneous parallel servers," *Operations Research Letters*, vol. 44, no. 4, pp. 507–513, 2016.
[11] E. Hyytiä, 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.
[12] A. Penttinen, E. Hyytiä, and S. Aalto, "Energy-aware dispatching in parallel queues with on-off energy consumption," in *30th IEEE International Performance Computing and Communications Conference (IPCCC)*, Orlando, FL, USA, Nov. 2011.
[13] L. Kleinrock, *Queueing Systems, Volume I: Theory*. Wiley Interscience, 1975.
[14] H. C. Tijms, *Stochastic Models, An Algorithmic Approach*. John Wiley & Sons, 1994.
[15] P. D. Welch, "On a generalized M/G/1 queuing process in which the first customer of each busy period receives exceptional service," *Operations Research*, vol. 12, no. 5, pp. 736–752, 1964.
[16] S. Fuhrmann and R. Cooper, "Stochastic decomposition in the M/G/1 queue with generalized vacations," *Operations Research*, vol. 33, no. 5, pp. 1117–1129, 1985.
[17] J. G. Shanthikumar, "On stochastic decomposition in m/g/1 type queues with generalized server vacations," *Operations Research*, vol. 36, no. 4, pp. 566–569, 1988.
[18] J. M. Norman, *Heuristic procedures in dynamic programming*. Manchester University Press, 1972.
[19] 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.
[20] 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.
[21] S. A. E. Sassen, H. C. Tijms, and R. D. Nobel, "A heuristic rule for routing customers to parallel servers," *Statistica Neerlandica*, vol. 51, no. 1, pp. 107–121, 1997.
[22] 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.
[23] P. Whittle, *Optimal Control: Basics and Beyond*. Wiley, 1996.
[24] M. Harchol-Balter, A. Scheller-Wolf, and A. R. Young, "Surprising results on task assignment in server farms with high-variability workloads," in *Proc. of SIGMETRICS*, 2009, pp. 287–298.
[25] E. Hyytiä, "Lookahead actions in dispatching to parallel queues," *Performance Evaluation*, vol. 70, no. 10, pp. 859–872, 2013, (IFIP Performance'13).