On Value Functions for FCFS Queues with Batch Arrivals and General Cost Structures

Esa Hyytiä[a,*], Rhonda Righter[b], Jorma Virtamo[c], Lauri Viitasaari[d]

[a] *University of Iceland, Department of Computer Science, Iceland*
[b] *Department of Industrial Engineering and Operations Research, University of California Berkeley*
[c] *Department of Communications and Networking, Aalto University, Finland*
[d] *Department of Information and Service Management, Aalto University, Finland*

Abstract

We develop a unified framework for analyzing and optimizing costs for systems of FCFS queues with batch arrivals, setup delays and a general nonlinear cost structure that includes costs associated with energy used, setup times and Quality of Service (QoS) measures. We focus on the $M^X/G/1$ and $Geo^X/G/1$ queues with i.i.d. service times, but our results hold also for arbitrary i.i.d. batch structures where service times within a batch may depend on the batch size and have different, possibly dependent, distributions. We use the notion of value functions from the theory of Markov decision processes (MDPs), along with exponential cost functions, to develop the notion of, and simple expressions for, value generating functions. These can be used to find efficient energy control and job dispatching policies.

Keywords: $M^X/G/1$, $Geo^X/G/1$, batch arrivals, generating function, value function

1. Introduction

Many practical and important problems can be formulated as Markov decision problems (MDPs), where optimal or near-optimal control of a system subject to stochastic phenomena is sought. Such control problems can be found in a wide-variety of fields, e.g., inventory management, stock portfolio management, computer networks, data centers, and transportation.

In this paper, we are interested in single and parallel server systems. In order to optimize such a system, one needs to first define a meaningful objective. A relatively common choice is to minimize the expected waiting time in queue (i.e., the time before a job receives service), or alternatively, the overall response time (i.e., the mean time spent in the system) [1, 2, 3, 4]. According to Little's result, these are equivalent to minimizing the mean number of jobs in the queue and in the system, respectively. Depending on the model and information, policies such as JSQ (join the shortest queue) are optimal for these objectives.

From the fairness perspective, it might be useful to also consider higher moments of the waiting or response time [5]. Another popular objective, reflecting the idea that longer jobs should have to wait longer, is minimization of slowdown, i.e., the ratio of the response time to the size of the jobs (e.g., [6, 7, 8]). In a more general context, one can introduce job- or job-class-specific holding costs to differentiate between more important and less important jobs. On the other hand, for certain applications, the most important thing is to get jobs processed before their deadlines. Consequently, deadline cost structures incur a (unit) cost whenever a deadline for waiting or response time is missed. Often these deadlines are assumed to be known [9, 10]. Moreover, jobs missing their deadlines may leave the system immediately. Alternatively, an admission control system may reject jobs upon arrival so as to minimize the number of deadline violations [11]. When the individual deadlines are not known, one obtains a system with (random) abandonments [12]. More recently, the minimization of energy used has become an important objective, e.g., [5, 13]. The costs can also be server specific if one server is, e.g., more expensive to use or should be avoided for some other reason. All of these objective functions are special cases of our general cost structure.

In particular, we define general server-specific cost functions $c^{(i)}(w, x, k)$, where i denotes the server, w is the waiting time in queue, x is the service time, and $k \in \mathcal{K}$ defines the class of the job. Some special cases of these cost structures (for identical customers) are given in Table 1. We will show how they can be incorporated into our framework.

*Corresponding author

Table 1: Common cost structures that all are special cases of our general framework.

Type	Cost function	References
waiting time	$c_1(w) = w$	[1, 2, 3, 4]
higher moments	$c_n(w) = w^n, \quad n = 1, 2, \ldots$	[5]
slowdown	$c_s(w, x) = 1 + w/x$	[6, 7, 8]
deadline(s)	$c_\tau(w) = \mathbf{1}(w > \tau)$	[9, 10, 11]
w is the waiting time, x the service time of a job, and τ the deadline.		

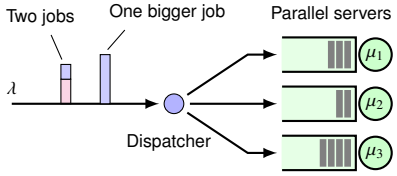


Figure 1: Model for a parallel server system with batch arrivals.

Our basic models for single-server systems are the continuous time $M^X/G/1$ queue and the discrete time $Geo^X/G/1$ queue, where the inter-arrival times of batches are, respectively, exponentially or geometrically distributed. In both cases, jobs are served in first-come-first-served (FCFS) order, and the service times are independent and identically distributed (i.i.d.) random variables. Our framework, however, is more general and allows for service times within a batch to depend on the batch size and have different, possibly dependent, distributions. General costs can be associated with waiting and response time at job and at batch level. 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 imposes an additional delay on jobs.

We also consider a multi-server setting with n heterogeneous parallel servers, where each job is dispatched immediately upon arrival to one of the servers (see Figure 1). In this case, the set of control actions includes both server switch-off and job routing decisions. The cost structures can be server specific.

We study the optimal control policies in the MDP framework by using the so-called value functions that characterize the expected future costs for each state. Often the system state is defined by the number of jobs (per server). We refer to such models as *number-aware* systems.

In contrast, we assume *size-aware* systems in which exact service times become available upon arrival. Moreover, we allow batch arrivals and consider general cost structures both in continuous and discrete time. Our main contributions are twofold: First, we give compact expressions for the value functions of the $M^X/G/1$ and $Geo^X/G/1$ queues subject to random setup delays and *arbitrary cost functions* defined in terms of the job-specific waiting and sojourn times. Second, we study "exponential" cost structures that yield or *generate all typically used cost structures* listed in Table 1 as special cases. These value functions have particularly convenient forms from which one can identify, e.g., the well-known Pollaczek-Khinchin transform formulæ and the penalty due to the setup delay.

The rest of the paper is organized as follows. In Section 2 we introduce the continuous- and discrete-time FCFS queues and necessary notation. In Section 3, we analyze the size-aware FCFS queues with arbitrary cost functions in the MDP framework. In Section 4 we consider exponential cost functions. In Section 5 we introduce a more general batch structure that allows dependencies in jobs' service times within each batch. In Section 6 we consider the special case where only the number of jobs is known, and in Section 7, we derive policies for switching off servers and routing jobs to parallel servers. Section 8 concludes the paper.

1.1. Related work

Prior related work has studied value functions in order to find efficient dispatching rules in different settings. In particular, value functions enable the first policy iteration (FPI) [14, 15], where the idea is to improve a given basic policy by choosing an alternative action yielding lower expected future costs in an infinite time horizon. The standard approach is to start from a static basic policy under which the system decomposes and it is sufficient to analyze a single server (pool). The necessary information to this end is provided by the value function of the given sub system (e.g. the M/G/1 queue, or Erlang's M/M/s system).

For Markovian systems, a sufficient state description is the number in the system (e.g., how many jobs each server has). Consequently, such systems are often referred to as number-aware systems, as the dispatching decision is based on that information. Several results already exist for minimizing the mean delay in number-aware systems. In the context of routing packets in a data network, Krishnan [16] obtained the value function for the M/M/s system, of which the M/M/1 queue is a special case. With identical jobs, this result holds for all work-preserving scheduling disciplines. When jobs have different holding costs, the value function must be adjusted accordingly. For example, Argon et al.

[9] consider the M/M/1-FCFS queue with a general cost $c(t)$ for sojourn time t, and Doroudi et al. [17] give the value function for the M/M/1-PS queue with job-specific holding costs. In particular, number-aware systems correspond to standard Markov processes with a discrete state space, and the corresponding value functions, in principle, can be obtained by means of dynamic programming [18, 19]. For example, Ansell et al. [20] consider multi-class FCFS (and priority) queues with class-specific holding costs and service rates. Similarly, Leino and Virtamo [21] consider the multi-class PS queue with class-specific arrival and service rates, for which the corresponding value function for sojourn time is derived.

However, other state descriptions can also be relevant, which often require the assumption that more information about a job becomes available upon its arrival. Bhulai [22] derived the value function for the M/Cox(r)/1-FCFS queue, with a state description of (n, k), where n denotes the number in the system, and k the service phase of the customer being processed (if any). A state description with service phases can sometimes be impractical. Sassen et al. [23] consider the M/G/1-FCFS queue. In order to avoid dealing with the remaining service times, they resort to an approximation and assume that the remaining processing time of the customer in service (if any) obeys the equilibrium excess distribution.

In a *size-aware* system, the service times become known upon arrival, and server-specific workloads can be utilized when dispatching jobs. The size-aware value function for the M/M/1 queue (with FCFS) is given by Aalto and Virtamo [24], for the M/G/1 queue with FCFS, LCFS, SPT and SRPT the value function is given by Hyytiä et al. [4], while M/D/1-PS and M/M/1-PS queues are treated by Hyytiä et al. in [25] and [26], respectively. Note that by conditioning, the size-aware results can be utilized in the number- and age-aware context, and Hyytiä et al. [27] gives the corresponding results for the age-aware M/G/1-FCFS and LCFS queues, where the number in the queue and the service received (the age) are available.

In the context of preserving energy, one may consider switching a server off, which then may induce a setup delay when the server needs to be reactivated. The value function for the M/G/1 queue with setup delays is derived by Hyytiä et al. [5] subject to switching and processing costs in addition to the traditional costs related to the waiting and response times. LCFS and PS scheduling disciplines in this context are considered in Hyytiä et al. [28]. Hyytiä and Righter [29] consider the M/G/1 queue with FCFS and LCFS scheduling disciplines, setup delay, and linearly increasing holding costs (cf. fairness).

In general, the FPI method yields a good heuristic policy that can be further improved by the so-called lookahead approach of Hyytiä [30], which considers also the next action explicitly. For example, Hyytiä et al. [31] use this method to control the set of running servers proactively; this also demonstrates the wide applicability of the MDP framework.

For QoS, important quantities are the tails of the waiting and response time distributions, i.e., whether a job's waiting or response time exceeds some threshold referred to as a deadline. The M/G/1-FCFS queue with deadlines is considered by the authors in [10] and [11], where in the latter the option to reject jobs (with some cost) is included. The explicit result for the M/D/1 and M/iD/1 queues was later given in [32].

In blocking systems rejecting a job is not optional but mandated by the number of system places. In this context, Krishnan and Ott [33] derive the value function for the Erlang M/M/s/s loss system with respect to the blocking probability. This result was generalized by Leeuwaarden et al. [34] to the M/M/s/k system, where the example application was the assignment of telephone calls to base stations in a mobile network.

All work summarized above assumes a Poisson arrival process, which is customary in this field. In contrast, Hyytiä and Aalto [35] consider the Erl/G/1 queue with FCFS and LCFS, and Hyytiä et al. [36] consider the D/M/1 queue (with general admission costs).

Some of the results given in this paper were initially presented in [37], where the M^X/G/1 queue with batch arrivals, setup delay and a general cost structure was studied. Many of the results described above are special cases of the results of [37]. In this paper, we include an analysis of the discrete time Geo^X/G/1 queues and study the number-aware cases both in continuous and discrete time (Section 6). Moreover, we generalize our results to a much larger class of queueing models by introducing and analyzing a class-based batch structure that allows dependencies among service times for jobs in the same batch (Section 5).

2. FCFS queueing models with batch arrivals

The first 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 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*

Table 2: Notation.

Quantity	Description
B	size of a batch (number of jobs)
A	number of jobs ahead in the same batch (depends on B)
X	service time of a job (size of a job)
H	service time of a batch, $H = X_1 + \ldots + X_B$
λ_b	arrival rate of batches
λ	arrival rate of jobs, $\lambda = \lambda_b \, \mathrm{E}[B]$
ρ	offered load, $\rho = \lambda \, \mathrm{E}[H]$
D	setup delay of an idle server
$c(w, x, k)$	cost function (waiting time, service time and job's class)
r	mean cost rate, $r = \lambda \, \mathrm{E}[c(W, X, K)]$

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,

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

In parallel, we consider also an equivalent discrete-time queue referred to as the $\mathrm{Geo}^{\mathrm{X}}$/G/1 queue that is defined as follows.

Definition 2 ($\mathrm{Geo}^{\mathrm{X}}$/G/1) *The discrete-time Geo^{X}/G/1 queue receives a new batch of B jobs with discrete sizes $X_1, \ldots, X_B$, at every time step with probability λ_b, where the X_i are i.i.d. strictly positive integer-valued (discrete) random variables, $X_i \sim X \geq 1$. Hence, the interarrival-time of batches obeys a geometric distribution and the total service time of a batch is again $H = X_1 + \ldots + X_B$. Between time steps, one unit of work, if present, is discharged.*

Alternatively, we can also consider the following characterization of the discrete-time queue.

Definition 3 (D^{X}/G/1) *The discrete-time D^{X}/G/1 queue receives N jobs, $X_1, \ldots, X_N$, at every time step, where $N \geq 0$ and the X_i are strictly positive integer-valued, i.i.d., random variables. Between time steps, one unit of work, if present, is discharged.*

These two models characterize the same system given $\lambda_b = \mathrm{P}\{N > 0\}$ and $\mathrm{P}\{N = k\} = \lambda_b \, \mathrm{P}\{B = k\}$ for $k > 0$. The latter model simply makes it explicit that the arrival process can be arbitrary. However, in what follows, we will consider the $\mathrm{Geo}^{\mathrm{X}}$/G/1 queue when discussing discrete-time systems.

In both continuous and discrete time, 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.

A sufficient state description for both continuous- and discrete-time FCFS queues is the backlog (unfinished work) denoted by u. With discrete-time systems, it is necessary to be more specific and also define whether the queue is observed just before or immediately after every time step. The former corresponds to the *late arrival system*, and the latter to the *early arrival system*. In this paper, we assume the early arrival system for convenience as then many expressions have the same form in discrete and continuous time. This means that if the discrete-time system is in state $u > 0$, then the state when the next batch arrives is at most $u - 1$.

The main advantage of studying queueing models with batch arrivals instead of requiring $B = 1$ is that by adjusting the batch size distribution we can model more bursty arrival processes while maintaining the memoryless property of the arrival process. Somewhat orthogonally, in Section 5, we relax the i.i.d. requirement for service times within a batch, and allow arbitrary dependencies within the batch. For example, the service time of the first and second job of a batch may obey a different distribution. The sequence of batches are still assumed to be i.i.d.

We also consider systems where the server is switched off when idle. The motivation for this is the presumed energy savings. The drawback is that restarting the server takes some time.

Definition 4 (Setup delay D) *A server is assumed to enter a low-power state immediately upon becoming idle to preserve energy. The low-power state lasts until the next batch of jobs arrives, at which time the server enters a setup phase. The time durations spent in the setup phase, referred to as* the setup delays*, are non-negative i.i.d. random variables, $D_i \sim D$. The service of the first job of the batch starts immediately after the corresponding setup delay.*

With the M^X/G/1 queue, D can be a continuous random variable (i.e., its support is the non-negative real numbers), whereas with the Geo^X/G/1 queue D is an integer-valued random variable. The waiting time distribution and its mean in the M^X/G/1 queue with setup delays D has been studied in [38] and [39], see also [40, 41].

2.1. Busy periods

Let us first recall some standard results regarding the length of the busy period denoted by T_b. Note that when considering busy periods, each batch of jobs can be seen as one larger job of equivalent size.

In the case of the M^X/G/1 queue, batches arrive according to a Poisson process with rate λ_b, i.e., the batch interarrival time obeys the exponential distribution with parameter λ_b. The service time X and setup delay D are generally distributed continuous random variables. Consequently, also the duration of the busy period T_b can have arbitrary values.

In contrast, with the discrete-time Geo^X/G/1 queue, the random quantities X and D are integer-valued. We let $\text{Geo}_0(p)$ denote a geometrically distributed random variable starting from zero, i.e., $p_i = (1-p)^i p$, $i = 0, 1, \ldots$, and $\text{Geo}_1(p)$ a geometrically distributed random variable starting from one, $p_i = (1-p)^{i-1} p$, $i = 1, 2, \ldots$. The inter-arrival time of batches, measured in time slots, obeys the $\text{Geo}_1(\lambda_b)$ distribution. The busy period we define as follows.

Definition 5 *The busy period in Geo^X/G/1 lasts until the system becomes empty and there is no immediate arrival, i.e., until a genuine idle period starts (see Figure 3(a)).*

Consequently, in the discrete time system also the length of the busy period T_b is integer-valued.

Given the early arrival system in discrete time, the mean results related to the busy periods have same forms:

Lemma 1 *The mean remaining busy period both in the M^X/G/1 and Geo^X/G/1 queues with setup delay is*

$$\mathrm{E}[T_u] = \frac{u}{1-\rho}, \tag{2}$$

where u denotes the backlog. Consequently, the mean busy period is given by

$$\mathrm{E}[T_b] = \frac{\mathrm{E}[H+D]}{1-\rho}.$$

Both systems form a renewal process where the renewal point is the time instant when a system becomes idle. The mean renewal interval is given by

$$\mathrm{E}[T_r] = \frac{1}{\lambda_b} + \mathrm{E}[T_b] = \frac{1+\lambda_b \mathrm{E}[D]}{\lambda_b(1-\rho)}. \tag{3}$$

Finally, the mean number of jobs served during a busy period is

$$\mathrm{E}[N_r] = \frac{\mathrm{E}[B]}{1-\rho}\left(1+\lambda_b \mathrm{E}[D]\right).$$

Proof: The results are well-known and can be found in many text books. □

2.2. Generating Functions and Transform Domain

Both the M^X/G/1 and Geo^X/G/1 queue can be analyzed in the transform domain.

Definition 6 *For a non-negative discrete random variable B, the generating function (z-transform) is*

$$\widetilde{B}(z) \triangleq \mathrm{E}[z^B].$$

For continuous random variables, we use the Laplace-Stieltjes transform:

Definition 7 (LST) *The Laplace-Stieltjes transform of the random variable $X \geq 0$ is (see, e.g., [42] and [43])*

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

Throughout we will use an asterisk to denote the transform of a continuous random variable. As is well known, see, e.g., [42, 43], these transforms, when defined, encompass all the necessary information about a random variable in a single function and, e.g., the LST of the sum of two independent continuous random variables, $Z = X + Y$, is $Z^*(s) = X^*(s) \cdot Y^*(s)$. Similarly, for random sums of independent and identically distributed continuous and discrete random variables, $Z = \sum_{i=1}^{N} X_i$, where N is an integer-valued random variable, we have

$$Z^*(s) = \widetilde{N}(X^*(s)), \quad \text{and} \quad \widetilde{Z}(z) = \widetilde{N}(\widetilde{X}(z)). \tag{4}$$

2.3. Batch size distribution B

In all our models, the fundamental assumption is that batches are i.i.d. This is clearly the case with the $M^X/G/1$ and $Geo^X/G/1$ queues, where also the service times of the jobs are i.i.d., but it also holds with the more general batch structures considered later in Section 5. Consequently, batch sizes B_i, defining the number of jobs in the i^{th} batch, are i.i.d. random variables, $B_i \sim B$.

The probability that a randomly chosen job came from a size i batch is

$$\frac{i\,\mathrm{P}\{B=i\}}{\mathrm{E}[B]},$$

and the probability that a job is the i^{th} 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,

$$\mathrm{P}\{A=i\} = b_{i+1} = \frac{\mathrm{P}\{B\geq i+1\}}{\mathrm{E}[B]}. \tag{5}$$

That is, (5) gives the distribution for job's position within the same batch. For example, the probability of being the first job is $\mathrm{P}\{A=0\} = 1/\mathrm{E}[B]$. Similarly, the mean number of jobs ahead in the same batch is given by

$$\mathrm{E}[A] = \frac{1}{\mathrm{E}[B]}\sum_{i=1}^{\infty}\mathrm{P}\{B\geq i+1\}\cdot i = \frac{1}{2}\left(\frac{\mathrm{E}[B^2]}{\mathrm{E}[B]}-1\right), \tag{6}$$

and the corresponding generating function is

$$\widetilde{A}(z) = \mathrm{E}[z^A] = \frac{1}{z\,\mathrm{E}[B]}\sum_{i=1}^{\infty}\mathrm{P}\{B\geq i\}\,z^i. \tag{7}$$

2.4. M/G/1 in the transform domain

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

$$W_0^*(s) = \frac{(1-\rho)s}{s-\lambda(1-X^*(s))}. \tag{8}$$

From [44], 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 is 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

$$p_0 = \frac{1-\rho}{1+\lambda\,\mathrm{E}[D]}. \tag{9}$$

Additionally, in our case also the first job experiences the setup delay D, yielding

$$W^*(s) = W_x^*(s) + p_0(D^*(s)-1) = \frac{1-\rho}{1+\lambda\,\mathrm{E}[D]}\cdot\frac{\lambda+D^*(s)(s-\lambda)}{s-\lambda(1-X^*(s))}, \tag{10}$$

where the term $p_0(D^*(s)-1)$ accounts for the waiting time of the first job. Thus, we obtain

$$W^*(s) = W_0^*(s)\cdot W_e^*(s), \tag{11}$$

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 and Cooper's decomposition property, illustrated by (11), holds for a large class of M/G/1 queues with different vacation models, see [45] and [46]. (The setup delay D can also be interpreted as a vacation.) 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.

2.5. $M^X/G/1$ in the transform domain

Consider next the batch arrival process of B jobs with i.i.d. service times $X_i \sim X$. The service time of the whole batch is a random sum, $H = X_1 + \ldots + X_B$, for which we have (see (4))

$$\begin{aligned}
\mathrm{E}[H] &= \mathrm{E}[B] \cdot \mathrm{E}[X], \\
\mathrm{E}[H^2] &= \mathrm{E}[B] \cdot \mathrm{V}[X] + \mathrm{E}[B^2] \cdot \mathrm{E}[X]^2, \\
H^*(s) &= \widetilde{B}(X^*(s)),
\end{aligned} \tag{12}$$

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

2.5.1. 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 W_b and the service times of those jobs from the same batch that happen to be ahead of the given job, W_a. For the former, we can utilize (8),

$$W_b^*(s) = \frac{s(1-\rho)}{s - \lambda_b(1 - \widetilde{B}(X^*(s)))},$$

and the latter is given by

$$W_a^*(s) = \widetilde{A}(X^*(s)). \tag{13}$$

Thus the LST for the waiting time of a job is given by $W_0^*(s) = W_b^*(s) \cdot W_a^*(s)$, yielding

$$W_0^*(s) = \frac{(1-\rho)s\,\widetilde{A}(X^*(s))}{s - \lambda_b(1 - \widetilde{B}(X^*(s)))}. \tag{14}$$

2.5.2. 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 later batches increase the backlog by $H_i \sim H$. Consequently, from (11) 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

$$W_e^*(s) = \frac{\lambda_b + (s - \lambda_b)\,D^*(s)}{(1 + \lambda_b\,\mathrm{E}[D])s}. \tag{15}$$

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

$$W_j^*(s) = W_b^*(s) \cdot W_e^*(s) \cdot W_a^*(s). \tag{16}$$

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

$$\mathrm{E}[W_e] = \lim_{s \to 0^+} W_e^{*\prime}(s) = \frac{2\,\mathrm{E}[D] + \lambda_b\,\mathrm{E}[D^2]}{2(1 + \lambda_b\,\mathrm{E}[D])}. \tag{17}$$

2.5.3. Examples

Example 1 *According to* (16)*, the waiting time in the M^X/G/1 queue is a sum of three terms with mean [38, 39, 41, 37],*

$$\mathrm{E}[W] = \mathrm{E}[W_b] + \mathrm{E}[W_e] + \mathrm{E}[W_a],$$

where

$$\mathrm{E}[W_b] = \frac{\lambda_b \mathrm{E}[H^2]}{2(1-\rho)} = \frac{\lambda_b(\mathrm{E}[B]\cdot \mathrm{V}[X] + \mathrm{E}[B^2]\cdot \mathrm{E}[X]^2)}{2(1-\rho)}, \qquad \textit{(batch-level)}$$

$$\mathrm{E}[W_e] = \frac{2\,\mathrm{E}[D] + \lambda_b\,\mathrm{E}[D^2]}{2(1+\lambda_b\,\mathrm{E}[D])}, \qquad \textit{(setup delay)}$$

$$\mathrm{E}[W_a] = \mathrm{E}[A]\cdot \mathrm{E}[X] = \frac{\mathrm{E}[X]}{2}\left(\frac{\mathrm{E}[B^2]}{\mathrm{E}[B]} - 1\right), \qquad \textit{(intra-batch)},$$

and $\mathrm{E}[A]$ *is given in* (6)*. Similarly, according to* (22)*, the waiting time in the Geo^X/G/1 queue is also a sum of three equivalent terms with*

$$\mathrm{E}[W_b] = \frac{\lambda_b \mathrm{E}[H^2] - \rho}{2(1-\rho)} = \frac{\lambda_b(\mathrm{E}[B]\cdot \mathrm{V}[X] + \mathrm{E}[B^2]\cdot \mathrm{E}[X]^2) - \rho}{2(1-\rho)}, \qquad \textit{(batch-level)}$$

$$\mathrm{E}[W_e] = \frac{(2-\lambda_b)\,\mathrm{E}[D] + \lambda_b\,\mathrm{E}[D^2]}{2(1+\lambda_b\,\mathrm{E}[D])}, \qquad \textit{(setup delay)}$$

$$\mathrm{E}[W_a] = \frac{\mathrm{E}[X]}{2}\left(\frac{\mathrm{E}[B^2]}{\mathrm{E}[B]} - 1\right). \qquad \textit{(intra-batch)}$$

Example 2 *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 from (5)

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

i.e., $A \sim \mathrm{Geo}_0(q)$. Hence, the mean number of jobs ahead of a random job in a batch is (follows also from (6)*)*

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

The mean sojourn time of a job in this M^X/G/1 queue, $\mathrm{E}[T]$*, is then*

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

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

2.6. Geo^X/G/1 in the transform domain

Let us next state some necessary results concerning the waiting time distribution in the Geo^X/G/1 queue, i.e., the discrete time counterparts of the Pollaczek-Khinchin transform formulæ. The generating function of the waiting time distribution W in the Geo/G/1 queue without a setup delay is given by [47, Sec. 4.6.1]

$$\widetilde{W}_0(z) = \frac{(1-z)(1-\rho)}{(1-z) - \lambda(1-\widetilde{X}(z))}. \tag{18}$$

The setup delay shows up again as an independent random variable (cf. decomposition).

Proposition 1 *The generating function of the waiting time distribution in the Geo/G/1 queue with a random integer-valued setup delay D is*

$$\widetilde{W}(z) = \widetilde{W}_e(z) \cdot \widetilde{W}_0(z), \tag{19}$$

where the generating function of the additional delay W_e is

$$\widetilde{W}_e(z) = \frac{(1 - z - \lambda)\widetilde{D}(z) + \lambda}{(1 - z)(1 + \lambda \,\mathrm{E}[D])}. \tag{20}$$

The proof is given in the Appendix.

Example 3 *The mean waiting time in the Geo/G/1 queue with a random setup delay D is*

$$\mathrm{E}[W] = \frac{\lambda \,\mathrm{E}[X^2] - \rho}{2(1 - \rho)} + \frac{(2 - \lambda)\mathrm{E}[D] + \lambda \,\mathrm{E}[D^2]}{2(1 + \lambda \,\mathrm{E}[D])}. \tag{21}$$

The corresponding results with batch arrivals follow immediately from (18), which in this case gives the batch-level waiting time:

Corollary 1 *For the discrete-time Geo^X/G/1 queue with a random setup delay D, the generating function of the waiting time distribution W is*

$$\widetilde{W}(z) = \widetilde{W}_b(z) \cdot \widetilde{W}_e(z) \cdot \widetilde{A}(\widetilde{X}(z)), \tag{22}$$

where $\widetilde{W}_b$ is the batch-level waiting time without setup delay (cf. (18)*),*

$$\widetilde{W}_b(z) = \frac{(1 - z)(1 - \rho)}{1 - z - \lambda_b(1 - \widetilde{B}(\widetilde{X}(z)))},$$

W_e corresponds to the additional waiting time due to setup delay,

$$\widetilde{W}_e(z) = \frac{(1 - z - \lambda_b)\widetilde{D}(z) + \lambda_b}{(1 - z)(1 + \lambda_b \,\mathrm{E}[D])}, \tag{23}$$

and $\widetilde{A}(\cdot)$, given in (7)*, is the delay due to jobs ahead in the same batch. Alternatively to* (22)*, we can also write*

$$\widetilde{W}(z) = \widetilde{W}_0(z) \cdot \widetilde{W}_e(z),$$

where $\widetilde{W}_0(z)$ corresponds to the waiting time in the Geo^X/G/1 queue with no setup delay,

$$\widetilde{W}_0(z) = \widetilde{W}_b(z) \cdot \widetilde{A}(\widetilde{X}(z)). \tag{24}$$

In explicit form, we thus have

$$\widetilde{W}(z) = \frac{(1 - \rho)\left[(1 - z - \lambda_b)\widetilde{D}(z) + \lambda_b\right]\widetilde{A}(\widetilde{X}(z))}{(1 + \lambda_b \,\mathrm{E}[D])\left[1 - z - \lambda_b(1 - \widetilde{B}(\widetilde{X}(z)))\right]}. \tag{25}$$

For the sojourn time distribution we have $\widetilde{T}(z) = \widetilde{W}(z) \cdot \widetilde{X}(z)$.

3. Value functions with general costs

In this section, we consider the two FCFS queues with general cost structures that are functions of waiting time, sojourn time and job class. The main results, Propositions 2 and 3, characterize the corresponding value functions, defined later, in compact and useful expressions.

Definition 8 (Cost structure) *We let $\mathcal{K}$ denote the set of job classes. Each job incurs a cost immediately upon arrival defined by a cost function $c(w, x, k)$, where w denotes the waiting time in queue, x the service time, and $k \in \mathcal{K}$ the class of the job.*

The above cost structure is very general. We note that class k can be, e.g., equal to the batch size B. The cost can be or can include a job- or batch-specific random variable (i.i.d.), allowing, e.g., job classes with different weights. Similarly, the cost can take into account the job's service time x (cf. slowdown in Table 1). Moreover, the cost can also depend on the position of the job within the batch. Common special cases include costs that depend solely on the waiting or sojourn time, in which case we write simply $c_W(w)$ and $c_T(t)$, respectively. Some examples are given in Table 1.

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, where u may also include any unfinished setup time. In discrete time systems, we observe the state of the queue, u, **right after the new job, if any, has arrived**, and has been added to the queue. Hence, the first thing that will happen is that u decreases by 1 if $u > 0$, after which comes the next opportunity for a batch to arrive.

We let U denote the backlog in steady state. The backlog is updated instantly as each job of the batch gets added to the queue one at a time. When a job arrives to an empty queue, the setup delay is also added to the backlog. The backlog U corresponds to the waiting time the batch sees (cf. PASTA). Additionally, individual jobs must wait until the jobs ahead of them within the same batch have been processed.

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

$$r \triangleq \lambda \,\mathrm{E}[c(W, X, K)] = \lambda\, \bar{c}. \tag{26}$$

where W is the waiting time in queue, X is the service time, and K denotes the job's class.

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

Definition 9 (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,*

$$v(u) \triangleq \lim_{t\to\infty} \mathrm{E}[V(u, t) - rt], \tag{27}$$

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

Because we assume $\rho < 1$, the queues are stable and ergodic. We further assume that the cost structure is such that the above limit exists and is finite. This depends both on the service time distribution and the cost structure (cf. Pollaczek-Khinchin mean waiting time formula; a stable queue may have infinite mean waiting time).

In general, the constant term in $v(u)$ is irrelevant as the important quantity for making decisions is the difference $v(u + x) - v(u)$. Consequently, most of our results are given in terms of $v(u) - v(0)$. That is, for initial backlog u, $v(u)$ characterizes the expected deviation (in terms of costs) from the mean cost (rate), whereas $v(u) - v(0)$ characterizes the expected deviation from the cost when the system is initially empty.

Moreover, as the time instants when a queue becomes idle are renewal points, it is sufficient to consider the remaining busy period, and by dynamic programming, we have

$$v(u) = \mathrm{E}[V(u, T_u) - r\,T_u] + v(0), \tag{28}$$

where T_u denotes the length of the remaining busy period.

3.1. Continuous time M^X/G/1 queue

Let W and W_0 denote the waiting time of a job in steady state for systems with and without the setup delay D, and let Y denote a continuous uniformly distributed random variable, $Y = Y(u) \sim \mathrm{U}(0, u)$, that is independent of W_0, X and K. Then we have the following result for the value function.

Proposition 2 *The value function $v(u)$ for the M^X/G/1 queue with setup delays satisfies*

$$v(u) - v(0) = \frac{\lambda u}{1-\rho}\,\mathrm{E}[c(W_0 + Y, X, K) - c(W, X, K)]. \tag{29}$$

We first prove the following Lemma regarding the total cost S accumulated during a busy period when the cost of an individual job is $c(W, X, K)$.

Lemma 2 *The mean total cost an M^X/G/1 queue with setup delays incurs during a busy period is*

$$\mathrm{E}[S\,] = \frac{(1 + \lambda_b\,\mathrm{E}[D])\mathrm{E}[B]}{1-\rho}\,\mathrm{E}[c(W, X, K)]. \tag{30}$$

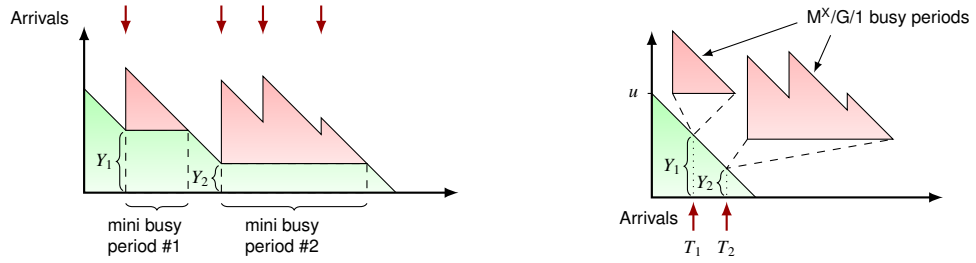


Figure 2: A sample realization with two mini busy periods until the server becomes idle (left). Mini busy periods interrupting the service of the initial backlog of u originate from the Poisson process with rate λ_b (right).

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

$$r = \lambda \, \mathrm{E}[c(W, X, K)] = \lambda_b \, \mathrm{E}[B] \, \mathrm{E}[c(W, X, K)]. \tag{31}$$

Consider a renewal process where the renewal point is the time instant when the system becomes idle. The mean renewal interval is given by (3)

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

Hence, an alternative expression for the mean cost rate is

$$r = \frac{\mathrm{E}[S]}{\mathrm{E}[T_r]} = \frac{\lambda_b (1 - \rho)}{1 + \lambda_b \, \mathrm{E}[D]} \cdot \mathrm{E}[S]. \tag{32}$$

Combining (31) and (32) yields (30). □

The following corollary allows us to include an offset in the cost function as needed in our proof of Proposition 2.

Corollary 2 *When jobs incur costs according to $c(W + y, X, K)$, where y is a given constant, then noting that $c_2(w, x, k) \triangleq c(w + y, x, k)$ is just another cost function, it follows that the mean cost incurred during a busy period is*

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

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

$$\mathrm{E}[S_0(y)] = \frac{\mathrm{E}[B]}{1 - \rho} \, \mathrm{E}[c(W_0 + y, X, K)], \tag{33}$$

where $S_0(y)$ denotes the cost incurred during a busy period with initial work y and no setup delay.

With these, we are ready to prove Proposition 2:

Proof: (Proposition 2.) By dynamic programming, (28), the value function $v(u)$ satisfies,

$$v(u) = \mathrm{E}[V(u, T_u)] - \mathrm{E}[T_u] \, r + v(0),$$

where $\mathrm{E}[V(u, T_u)]$ denotes the mean cost incurred during the remaining time of the busy period T_u when currently in state u, and $r = \lambda \, \mathrm{E}[c(W, X, K)]$ is the mean cost rate (with the setup delay). We recall that (2) gives $\mathrm{E}[T_u] = u/(1 - \rho)$, and hence,

$$\mathrm{E}[T_u] \, r = \frac{\lambda u}{1 - \rho} \, \mathrm{E}[c(W, X, K)]. \tag{34}$$

Consider now the first term $\mathrm{E}[V(u, T_r)]$. Recall that costs are incurred only upon batch arrivals. When a batch of jobs arrives before the system is empty, i.e., with backlog $u > 0$, a new mini busy period is started that will end when the backlog returns to the same level (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 unfinished work in system at the time the i^{th} mini busy period starts. Then,

$$\mathrm{E}[V(u, T_u)] = \mathrm{E}\left[\sum_{i=1}^{N_b} S_0(Y_i) \right], \tag{35}$$

where $S_0(Y_i)$ is the cost during the mini busy period with initial work Y_i and no setup delay. Referring to Figure 2 (right), we see that $Y_i = u - T_i$, where the T_i's 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, 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 $\mathrm{U}(0, u)$. 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 (35), and noting that $\mathrm{E}[N_b] = \lambda_b u$ and using the result (33), we can write (35) in the form

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

where W_0 denotes the waiting time in the system with no setup delay (because mini busy periods do not involve setup delay). Combining the above with (34) completes the proof. □

The above immediately yields a corollary (see [5]) characterizing the role of the setup delay in the value functions.

Corollary 3 *Setup delay D gives rise to a negative linear term in the value function,*

$$v(u) - v(0) = v_0(u) - v_0(0) - \frac{\lambda u}{1-\rho}\,\mathrm{E}[c(W, X, K) - c(W_0, X, K)], \tag{36}$$

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.

Note that the expectation $\mathrm{E}[c(W, X, K) - c(W_0, X, K)]$ is a constant factor equal to the increase in the mean cost a job incurs due to the setup delay.

As $Y \sim \mathrm{U}(0, u)$ in (29), we can write explicitly that

$$v(u) - v(0) = \frac{\lambda}{1-\rho}\int_0^u \mathrm{E}[c(W_0 + y, X, K) - c(W, X, K)]\,dy. \tag{37}$$

Thus,

$$v'(u) = \frac{\lambda}{1-\rho}\,\mathrm{E}[c(W_0 + u, X, K) - c(W, X, K)]. \tag{38}$$

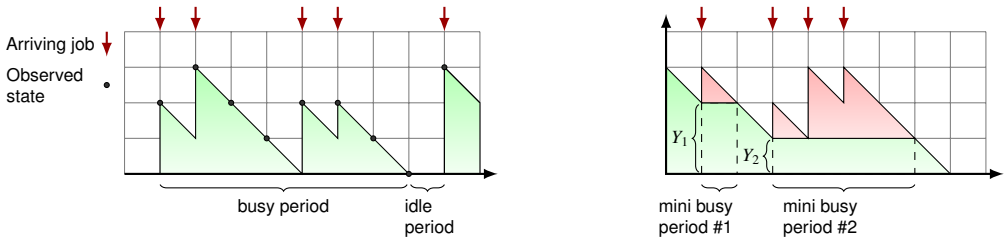
In general, if a random variable $Y \sim \mathrm{U}(0, u)$ is independent of everything else, then for any differentiable (cost) function $c(\cdot)$ with integrable derivative the following holds identically:

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

where, for clarity, we have omitted the other two parameters X and K. We note that 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 1. Without setup delay, $W \sim W_0$, and (38) thus reduces to

$$v_0'(u) = \frac{\lambda u}{1-\rho}\,\mathrm{E}[c'(W_0 + Y)]. \tag{39}$$

Note that (38) and (39) 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 $v(u) - v(0)$.



(a) Renewal process comprising a busy and idle period.

(b) Derivation of the expected costs.

Figure 3: Evolution of the Geo^{X}/G/1 queue is a renewal process (left). The expected costs during the remaining busy period can be deduced the same with as with the M^{X}/G/1 queue by noting that costs are incurred during the mini busy periods.

3.2. Discrete time Geo^{X}/G/1 queue

Lemma 3 *The mean costs incurred during a busy period for the Geo^{X}/G/1 queue is given by the same formula as that for the M^{X}/G/1 queue, Eq.* (30)*,*

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

Thus, the mean costs incurred during a busy period with an additional waiting time y when server has no setup delay is given by Eq. (33)*,*

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

where W_0 now denotes job's waiting time in the Geo^{X}/G/1 queue without a setup delay.

Proof: (3.) The same proof as that for Lemma 2 holds in discrete time (see Figure 3(b)). □

Given that also the mean results regarding the busy period (see Lemma 1) are identical, it is no surprise that the result for the value function is similarly almost identical to that of the M^{X}/G/1 queue:

Proposition 3 *The value function for the Geo^{X}/G/1 queue with a setup delay D satisfies*

$$v(u) - v(0) = \frac{\lambda u}{1-\rho}\,\mathrm{E}[c(W_0+Y,X,K) - c(W,X,K)], \tag{40}$$

where Y is an independent integer-valued uniformly distributed random variable, $Y \sim \mathrm{U}\{0,1,\ldots,u-1\}$.

Proof: We are again interested in the mini-busy periods, during which new costs are incurred. As the current state u is observed right after a possible arrival instant, the mini-busy periods can start at levels $u-1,\ldots,0$ (cf. Def. 5), and the random offset in the observed backlog is $Y \sim \mathrm{U}\{0,1,\ldots,u-1\}$. Dynamic programming (see (28)) gives

$$v(u) = \mathrm{E}[V(u,T_u)] - \mathrm{E}[T_u]\,r + v(0),$$

where $V(u,T_u)$ denotes the costs incurred during the remaining busy period when initially in state u, r is the mean cost rate, and T_u the length of the remaining busy period. Utilizing (33) (cf. Lemma 3) gives

$$v(u) - v(0) = \sum_{i=0}^{u-1} \lambda_b\,\mathrm{E}[S_0(i)] - r\frac{u}{1-\rho}.$$

As $r = \lambda\,\mathrm{E}[c(W,X,K)]$, the above reduces to

$$\frac{\lambda u}{1-\rho}\left[\left(\sum_{i=0}^{u-1}\frac{1}{u}\mathrm{E}[c(W_0+i,X,K)]\right) - \mathrm{E}[c(W,X,K)]\right],$$

yielding (40). □

Moreover, let Δ denote the difference operator, $\Delta v(u) = v(u+1) - v(u)$ and $\Delta c(w,x,k) = c(w+1,x,k) - c(w,x,k)$. Note that with $u = 1$, (40) gives the boundary condition,

$$\Delta v(0) = v(1) - v(0) = \frac{\lambda}{1-\rho}\,\mathrm{E}[c(W_0,X,K) - c(W,X,K)].$$

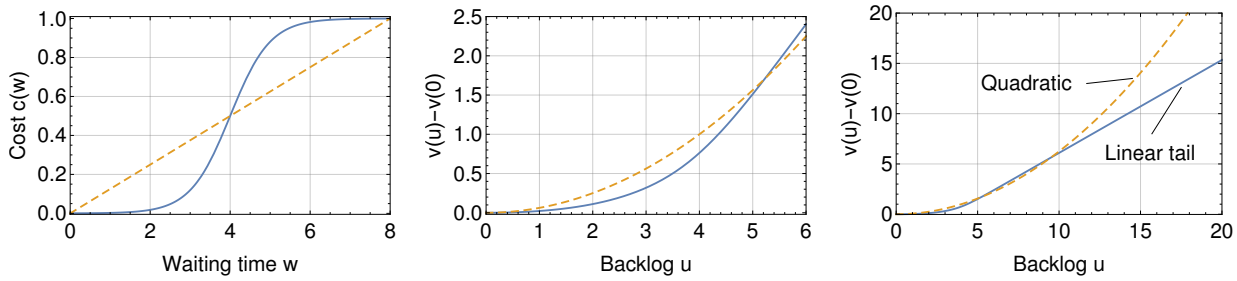


Figure 4: A logistic cost function (solid line) compared with the linear cost function (dashed line) corresponding to the mean waiting time. The left figure illustrates the cost functions and the middle and right figure the corresponding value functions (in different scales).

Thus without a setup delay $\Delta v_0(0) = 0$, i.e., $v_0(1) = v_0(0)$. The counterpart for (38) is

$$\Delta v(u) = \frac{\lambda}{1-\rho}\,\mathrm{E}[c(W_0 + u, X, K) - c(W, X, K)]. \tag{41}$$

In the discrete case, it holds that $u\,\mathrm{E}[\Delta c(Z + Y, X, K)] = \mathrm{E}[c(Z + u, X, K) - c(Z, X, K)]$, and without setup delays, we have a discrete-time counterpart also for (39),

$$\Delta v_0(u) = \frac{\lambda u}{1-\rho}\,\mathrm{E}[\Delta c(W_0 + Y, X, K)]. \tag{42}$$

3.3. Examples

Example 4 (QoE with M/M/1) *Suppose that when a customer's waiting time exceeds some threshold her perception of the quality quickly deteriorates to an unsatisfactory level. This can be modeled with a logistic cost function, say*

$$c(w) = \frac{1}{1 + e^{-k(w-\tau)}}.$$

The solid line in Figure 4 (left) illustrates $c(w)$ with $k = 2$ and $\tau = 4$. The dashed line corresponds to mean waiting time, $c(w) = (1/8)w$ (scaled to an equivalent range). Consider then the classical M/M/1 queue with $\lambda = 1/2$ and $\mu = 1$, for which $\mathrm{P}\{W > t\} = \rho\, e^{-(\mu-\lambda)t}$, and (39) yields $v'(u)$, the integral of which then gives the value function depicted in Figure 4 (middle and right). Note the asymptotically linear behavior (cf. [10, 11]) visible in the right figure.

Example 5 (Deadlines with M/M/1) *Let us next consider the M/M/1 queue with the deadline cost structure [10],*

$$c_\tau(w) = \mathbf{1}(w > \tau),$$

i.e., a unit cost is incurred if the waiting time exceeds threshold τ. Substituting this cost function into (37) gives

$$v_\tau(u) - v_\tau(0) = \frac{\lambda}{1-\rho}\int_0^u (\mathrm{P}\{W > \tau - y\} - \mathrm{P}\{W > \tau\})\,dy.$$

Given the waiting time distribution of the M/M/1 queue is known, $\mathrm{P}\{W > t\} = \rho e^{-(\mu-\lambda)t}$, the above expression can be evaluated, yielding a closed-form expression for the value function,

$$v_\tau(u) - v_\tau(\tau) = \begin{cases} \dfrac{\rho^2\, e^{-(\mu-\lambda)\tau}}{(1-\rho)^2}\left(e^{(\mu-\lambda)u} - e^{(\mu-\lambda)\tau} - (\mu-\lambda)(u-\tau)\right), & u \le \tau, \\ \dfrac{\lambda\left(1 - e^{-(\mu-\lambda)\tau}\right)}{1-\rho}(u-\tau), & u > \tau, \end{cases} \tag{43}$$

thus completing the results given in [10], where an exact expression was given only for the tail $u > \tau$. Note that in the heavy traffic limit, where $\rho \to 1$, we obtain, in accordance with [10, Prop. 3],

$$\lim_{\rho\to 1} v_\tau(u) - v_\tau(\tau) = \begin{cases} \dfrac{\lambda^2}{2}(u^2 - \tau^2), & u \le \tau, \\ \lambda(1 + \lambda\tau)(u-\tau), & u > \tau. \end{cases}$$

Example 6 (Waiting time cost for $\mathbf{M^X/G/1}$) *Let us consider the $M^X/G/1$ queue without setup delay and $c_1(w) = w$. Using* (29)*, it turns out that the value function is the same as with the M/G/1 queue [4],*

$$v_1(u) - v_1(0) = \frac{\lambda u}{1-\rho}\,\mathrm{E}[Y(u)] = \frac{\lambda u^2}{2(1-\rho)}, \tag{44}$$

as $\mathrm{E}[Y(u)] = u/2$. Alternatively, substituting $c_1'(w) = 1$ into (39) *gives*

$$v_1'(u) = \frac{\lambda u}{1-\rho},$$

which yields the same result (44)*. The dashed curves in Figure 4 illustrate $c_1(w)$ scaled by $1/8$ and the corresponding $v(u)$ when $\lambda = \rho = 1/2$.*

Example 7 *According to* (36)*, the setup delay shows up as a linear term. Utilizing* (17)*, we obtain the value function for the $M^X/G/1$ queue with setup delay D subject to a cost that is linear in the waiting time, $c_1(w) = w$,*

$$v_1(u) - v_1(0) = \frac{\lambda u^2}{2(1-\rho)} - \frac{\lambda u}{1-\rho}\,\frac{2\,\mathrm{E}[D] + \lambda_b\,\mathrm{E}[D^2]}{2(1+\lambda_b\,\mathrm{E}[D])}. \tag{45}$$

Recall that λ corresponds to the job arrival rate, and λ_b to the batch arrival rate, i.e., $\lambda = \lambda_b\,\mathrm{E}[B]$.

Example 8 (Polynomial waiting time cost for M/G/1) *Now consider the special case of the ordinary M/G/1 queue without setup delay when the cost function is $c_k(w) = w^k$. Then* (29) *reduces to*

$$v_k(u) - v_k(0) = \frac{\lambda u}{1-\rho}\,\sum_{i=1}^{k}\binom{k}{i}\mathrm{E}[Y(u)^i]\,\mathrm{E}[W^{k-i}].$$

The k^{th} moment of $Y(u) \sim \mathrm{U}(0,u)$ is

$$\mathrm{E}[Y^k(u)] = \frac{u^k}{k+1},$$

and thus

$$v_k(u) - v_k(0) = \frac{\lambda}{1-\rho}\,\sum_{i=1}^{k}\binom{k}{i}\frac{u^{i+1}}{i+1}\cdot\mathrm{E}[W^{k-i}].$$

The first $k-1$ moments of the waiting time are available from the Pollaczek-Khinchin transform formula.

Consequently, e.g., for $k = 2$, we have

$$v_2(u) - v_2(0) = \frac{\lambda u^2}{1-\rho}\left(\frac{u}{3} + \frac{\lambda\,\mathrm{E}[X^2]}{2(1-\rho)}\right). \tag{46}$$

Alternatively, we can utilize (39) *with $c_2'(w) = 2w$, giving*

$$v_2'(u) = \frac{\lambda u}{1-\rho}\,\mathrm{E}[2(W+Y)] = \frac{\lambda u}{1-\rho}\,(2\,\mathrm{E}[W] + u)\,,$$

which also yields (46)*.*

Example 9 (Waiting time cost for $\mathbf{Geo^X/G/1}$) *Consider the $Geo^X/G/1$ queue without a setup delay when the cost is either the waiting time, $c(w) = w$, or the sojourn time, $c(w,x) = w + x$. In both cases, substituting the given cost function into* (40) *gives*

$$v_1(u) - v_1(0) = \frac{\lambda u}{1-\rho}\,\mathrm{E}[Y] = \frac{\lambda(u-1)u}{2(1-\rho)}.$$

Note that $v_1(1) - v_1(0) = 0$, as desired. The same result can also be obtained by solving the difference equation (41)*, which in this case reduces to $v_1(u+1) - v_1(u) = \lambda u/(1-\rho)$.*

Example 10 (Polynomial waiting time cost for Geo/G/1) *Consider the Geo/G/1 queue without a setup delay when the cost is $c_k(w) = w^k$. Similarly as with the M/G/1 queue,*

$$v_k(u) - v_k(0) = \frac{\lambda u}{1-\rho} \sum_{i=1}^{k} \binom{k}{i} \mathrm{E}[Y(u)^i] \, \mathrm{E}[W^{k-i}],$$

where $Y \sim \mathrm{U}\{0, 1, \ldots, u-1\}$. Again, the first k and $k-1$ moments of $Y(u)$ and W are available, yielding an explicit expression for the value function. For example, with $k = 2$ we have

$$v_2(u) - v_2(0) = \frac{\lambda u}{1-\rho} \mathrm{E}[2WY + Y^2] = \frac{\lambda(u-1)u}{1-\rho} \left(\frac{2u-1}{6} + \mathrm{E}[W] \right),$$

where $\mathrm{E}[W] = \frac{\lambda \mathrm{E}[X^2] - \rho}{2(1-\rho)}$ (see later, Eq. (21)). Again, $v_2(1) - v_2(0) = 0$, as desired.

Example 11 (Position-specific costs) *Suppose that the class of the job, K, is its position within the batch, and moreover, that the job-specific holding cost is equal to K, $c(w, x, k) = wk$. First, the probability that a random job is in position i in the batch is*

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

and consequently a straightforward manipulation gives

$$\mathrm{E}[K] = \frac{1}{2} \left(\frac{\mathrm{E}[B^2]}{\mathrm{E}[B]} + 1 \right).$$

Consider then the expectation $\mathrm{E}[c(W_0 + Y, X, K) - c(W, X, K)]$ that appears both in the continuous and discrete time results, (29) and (40), respectively. With this cost function,

$$\mathrm{E}[c(W_0 + Y, X, K) - c(W, X, K)] = \mathrm{E}[YK] - \mathrm{E}[(W - W_0)K].$$

As Y is independent of everything else, the first expectation on the right-hand side simplifies to $\mathrm{E}[Y] \cdot \mathrm{E}[K]$. The second expectation on the right-hand side is zero in the absence of setup delays (when $D = 0$). With setup delays, we have

$$\begin{aligned} \mathrm{E}[(W - W_0)K] &= \sum_{i=1}^{\infty} \mathrm{P}\{K = i\} \cdot \mathrm{E}[(W - W_0)K \mid K = i] \\ &= \sum_{i=1}^{\infty} \mathrm{P}\{K = i\} \cdot \mathrm{E}[W_e] \cdot i \\ &= \mathrm{E}[K] \cdot \mathrm{E}[W_e], \end{aligned}$$

where $\mathrm{E}[W_e]$ denotes the extra delay due to the setup time, which is known for M^X/G/1 and Geo^X/G/1 queues (see Example 1). Combining the above with (29) and (40) gives that the corresponding value functions in both continuous and discrete time satisfy,

$$v_{pos}(u) - v_{pos}(0) = \frac{\lambda u}{2(1-\rho)} \cdot (\mathrm{E}[Y] - \mathrm{E}[W_e]) \cdot \left(\frac{\mathrm{E}[B^2]}{\mathrm{E}[B]} + 1 \right),$$

where $\mathrm{E}[Y]$ and $\mathrm{E}[W_e]$ have slightly different forms for M^X/G/1 and Geo^X/G/1 queues. Interestingly, the position-specific holding cost shows up as a constant factor,

$$v_{pos}(u) - v_{pos}(0) = (v_1(u) - v_1(0)) \cdot \left(\frac{\mathrm{E}[B^2]}{\mathrm{E}[B]} + 1 \right),$$

where $v_1(u)$ denotes the value functions with respect to waiting time, $c_1(w) = w$, given in Eq. (45).

4. Exponential costs and value generating functions

In this section, we consider exponential cost functions. These cost functions have a free parameter, similar to the LST and z-transform, that allows us to derive mean costs and value functions with respect to any polynomial cost of the waiting or sojourn time.

4.1. Exponential cost function with M^X/G/1

With the M^X/G/1 queue, we consider the following two cost functions:

$$c_W(w, s) = 1 - e^{-sw}, \qquad s > 0, \tag{47}$$

$$c_T(w, x, s) = 1 - e^{-s(w+x)}, \qquad s > 0, \tag{48}$$

where w and x denote the waiting and service times of the job, and s is a free parameter. These costs can be interpreted as the probability that the job completes either waiting or service before its deadline, where its deadline has an exponential distribution with parameter s.

Not surprisingly, the cost functions (47) and (48) are closely related to the Laplace-Stieltjes transform (see Section 2.2).

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 4 (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 admission cost function* (47)*, the mean cost is*

$$\bar{c}_W(s) = 1 - W_j^*(s), \tag{49}$$

with $W_j^*(s)$ given in (16)*, and the corresponding value function is*

$$v_W(u, s) - v_W(0, s) = \frac{\lambda u}{1-\rho}\left(W_e^*(s) - \frac{1 - e^{-su}}{su}\right) W_0^*(s). \tag{50}$$

where $W_j^*(s)$ and $W_0^*(s)$ are given in (16) *and* (14)*, respectively.*

Proof: The mean cost with (47) is

$$\bar{c}_W(s) = \mathrm{E}[1 - e^{-sW}] = 1 - W_j^*(s).$$

Similarly, the value function is obtained by substituting (47) into (29),

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

and thus,

$$v_W(u, s) - v_W(0, s) = \frac{\lambda u}{1-\rho}\,(W_j^*(s) - W_0^*(s)\,Y^*(s)). \tag{51}$$

Substituting the LST of $Y = Y(u)$,

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

into (51) yields (50). □

As $W_j^*(s)$ and $W_0^*(s)$ are given in (16) and (14), they can be substituted into (50) to get an explicit expression for the value function:

$$v_W(u, s) - v_W(0, s) = \frac{\mathrm{E}[B]\,\widetilde{A}(X^*(s))}{s/\lambda_b + \widetilde{B}(X^*(s)) - 1}\left[e^{-su} - 1 + \frac{s + (1 - D^*(s))(\lambda_b - s)}{1 + \lambda_b \mathrm{E}[D]}u\right]. \tag{52}$$

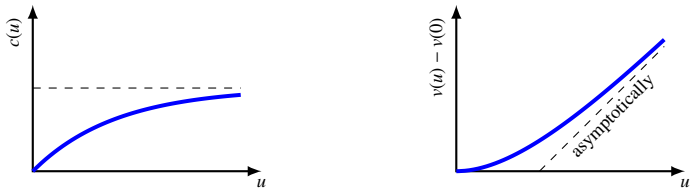
Without setup delay, $\mathrm{E}[D] = 0$ and $D^*(s) = 1$, and the expressions simplify further:

Corollary 4 *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* (47)*, the mean cost is*

$$\bar{c}_W(s) = 1 - \frac{(1-\rho)s\,\widetilde{A}(X^*(s))}{s - \lambda_b(1 - \widetilde{B}(X^*(s)))}, \tag{53}$$

and the corresponding value function is

$$v_W(u, s) - v_W(0, s) = \frac{(su + e^{-su} - 1)\mathrm{E}[B]\,\widetilde{A}(X^*(s))}{s/\lambda_b + \widetilde{B}(X^*(s)) - 1}. \tag{54}$$



(a) Cost function (b) Value function

Figure 5: Bounded and strictly increasing cost function and the corresponding value function.

Corollary 5 *For the M/G/1 queue with arrival rate λ, generally distributed service time X and no setup delay, subject to admission cost function* (47)*, the mean cost is*

$$\bar{c}_W(s) = \frac{s\,\mathrm{E}[X] + X^*(s) - 1}{s/\lambda + X^*(s) - 1}, \tag{55}$$

and the corresponding value function is

$$v_W(u,s) - v_W(0,s) = \frac{su + e^{-su} - 1}{s/\lambda + X^*(s) - 1}. \tag{56}$$

Note that in the heavy-traffic limit when $\rho \to 1$, $1/\lambda \to \mathrm{E}[X]$ and $\bar{c}_W(s)$, given in (55), converges to 1, as expected. Figure 5 illustrates an example cost function of type (47) and the corresponding value function given by (56).

Let us next consider the sojourn time, which is the sum of the waiting time and service time.

Proposition 5 (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* (48)*, the mean cost is*

$$\bar{c}_T(s) = 1 - W_j^*(s) \cdot X^*(s), \tag{57}$$

with $W_j^*(s)$ given in (16)*, and the corresponding value function is*

$$v_T(u,s) - v_T(0,s) = (v_W(u,s) - v_W(0,s))\, X^*(s). \tag{58}$$

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 (29), 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 (58). □

Corollary 6 (sojourn time for M/G/1) *For the M/G/1 queue with arrival rate λ, generally distributed service time X and no setup delay, subject to admission cost function* (48)*, the mean cost is*

$$\bar{c}_T(s) = 1 - \frac{s\,(1-\rho)\,X^*(s)}{s - \lambda(1 - X^*(s))}, \tag{59}$$

and the corresponding value function is

$$\begin{aligned} v_T(u,s) - v_T(0,s) &= (v_W(u,s) - v_W(0,s))\, X^*(s) \\ &= \frac{su + e^{-su} - 1}{s/\lambda + X^*(s) - 1}\, X^*(s). \end{aligned} \tag{60}$$

From these results, many interesting special cases can be immediately obtained. First note that

$$c_W(w,s) = 1 - e^{-sw} = sw - (sw)^2/2! + \ldots = \sum_{i=1}^{\infty} (-1)^{i+1} \frac{s^i}{i!} c_W^{(i)}(w),$$

where the $c_W^{(i)}(w)$ are the polynomial cost functions for the *waiting time*,

$$c_W^{(i)}(w) = w^i, \qquad i = 1, 2, \ldots \tag{61}$$

Consequently,

$$\begin{aligned} v_W(u,s) &= \frac{\lambda u}{1-\rho}\mathrm{E}\left[c_W(W+Y,s) - c_W(W,s)\right] \\ &= \frac{\lambda u}{1-\rho}\mathrm{E}\left[\sum_{i=1}^{\infty}(-1)^{i+1}\frac{s^i}{i!}\left(c_W^{(i)}(W+Y) - c_W^{(i)}(W)\right)\right] \\ &= \sum_{i=1}^{\infty}(-1)^{i+1}\frac{s^i}{i!}\left(\frac{\lambda u}{1-\rho}\mathrm{E}\left[c_W^{(i)}(W+Y) - c_W^{(i)}(W)\right]\right) \\ &= \sum_{i=1}^{\infty}(-1)^{i+1}\frac{s^i}{i!}v_W^{(i)}(u), \end{aligned}$$

where the $v_W^{(i)}(u)$ are the value functions corresponding to costs $c_W^{(i)}(w)$. The argument for v_T is similar. Moreover, possible setup delays change nothing. Therefore, in analogy with the LST of a random variable, we can compute any $v_W^{(i)}(u)$ and $v_T^{(i)}(u)$ directly from $v_W(u,s)$ and $v_T(u,s)$, respectively, by first taking the i^{th} derivative with respect to s, and then setting $s = 0$:

$$\begin{aligned} v_W^{(i)}(u) &= (-1)^{i+1}\lim_{s\to 0}\frac{d^i}{ds^i}v_W(u,s), \\ v_T^{(i)}(u) &= (-1)^{i+1}\lim_{s\to 0}\frac{d^i}{ds^i}v_T(u,s). \end{aligned}$$

That is, the value functions $v_W(u,s)$ and $v_T(u,s)$, given in (50), are *generating functions for the family of value functions* with respect to the polynomial costs (61). In the remainder of this paper, we refer to $v_w(u,s)$ and $v_T(u,s)$ simply as the **value generating functions** for waiting and sojourn time in order to keep the discussion short.

The same steps can be taken also with the mean costs. For example, it is straightforward to compute an arbitrary moment $\mathrm{E}[W^k]$ of the waiting time W in the $\mathrm{M^X/G/1}$ 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 (49).

4.2. Exponential cost function with Geo^X/G/1

With the discrete-time $\mathrm{Geo^X/G/1}$ queue, we consider the following exponential cost functions:

$$c_W(w,z) = 1 - z^w, \qquad 0 < z < 1, \tag{62}$$

$$c_T(w,x,z) = 1 - z^{(w+x)}, \qquad 0 < z < 1, \tag{63}$$

where w and x are again the waiting and service times of the job, and z is a free parameter, which for our purposes, is some small real number, $0 < z < 1$. The main results of this section are the value functions for the discrete time $\mathrm{Geo^X/G/1}$ queue subject to the exponential cost functions:

Proposition 6 *For the Geo^X/G/1 queue with a random setup delay D, subject to admission cost functions $c_W(w,z) = 1 - z^w$ and $c_T(w,x,z) = 1 - z^{w+x}$, the mean costs are*

$$\bar{c}_W(z) = 1 - \widetilde{W}(z), \tag{64}$$

$$\bar{c}_T(z) = 1 - \widetilde{W}(z)\,\widetilde{X}(z), \tag{65}$$

and the corresponding value functions are

$$v_W(u,z) - v_W(0,z) = \frac{\lambda_b u}{1-\rho}\left(\widetilde{W}_e(z) - \frac{1-z^u}{(1-z)u}\right)\widetilde{W}_0(z), \tag{66}$$

$$v_T(u,z) - v_T(0,z) = \frac{\lambda_b u}{1-\rho}\left(\widetilde{W}_e(z) - \frac{1-z^u}{(1-z)u}\right)\widetilde{T}_0(z). \tag{67}$$

where $\widetilde{T}_0(z)$ denotes the generating function of the sojourn time distribution in a system without a setup delay.

Proof: The mean cost (64) follows directly from the definition,

$$\bar{c}_W(z) = \mathrm{E}[c_W(W, z)] = 1 - \mathrm{E}[z^W],$$

and similarly (65) for the sojourn time. The value function (66) for waiting time follows from (A.1) by noting that $\widetilde{W}(z) = \widetilde{W}_e(z)\,\widetilde{W}_0(z)$. The value function also decomposes similarly as with the M^{X}/G/1 queue. Substituting $c(w, x, z) = 1 - z^w \cdot z^x$ into (40) gives immediately that

$$v_T(u, z) - v_T(0, z) = (v_W(u, z) - v_W(0, z)) \cdot \widetilde{X}(z),$$

yielding (67). □

Without setup delay, the expressions simplify and we have more explicit results. As the results for the sojourn time follow directly from the corresponding results for the waiting time, we give the special cases only for the waiting time metric.

Corollary 7 *For the Geo^X/G/1 queue without a setup delay and subject to admission cost function* (62)*, the mean cost is*

$$\bar{c}_W(z) = 1 - \frac{(1-\rho)(1-z)\,\widetilde{A}(\widetilde{X}(z))}{1 - z - \lambda_b(1 - \widetilde{H}(z))}, \tag{68}$$

where $\widetilde{H}(z) = \widetilde{B}(\widetilde{X}(z))$, and the corresponding value function is

$$v_W(u, z) - v_W(0, z) = \frac{(1-z)\,u + z^u - 1}{(1-z)(1-\rho)/\lambda_b}\,\widetilde{W}_0(z). \tag{69}$$

or more explicitly,

$$v_W(u, z) - v_W(0, z) = \frac{((1-z)\,u + z^u - 1)\,\widetilde{A}(\widetilde{X}(z))}{(1-z)/\lambda_b + \widetilde{H}(z) - 1}. \tag{70}$$

Without batch arrivals (when $B = 1$), one obtains the Geo/G/1 queue and the expressions simplify further.

Corollary 8 *For the Geo/G/1 queue without a setup delay, where a job of size X arrives with probability λ at the end of each time step, subject to admission cost function* (62)*, the mean cost is*

$$\bar{c}_W(z) = \frac{(1-z)\,\mathrm{E}[X] + \widetilde{X}(z) - 1}{(1-z)/\lambda + \widetilde{X}(z) - 1}, \tag{71}$$

and the corresponding value function is

$$v_W(u, z) - v_W(0, z) = \frac{(1-z)\,u + z^u - 1}{(1-z)/\lambda + \widetilde{X}(z) - 1}. \tag{72}$$

Example 12 *For the Geo/Geo/1 queue with $X \sim \mathrm{Geo}_1(q)$, the generating function of X is $\widetilde{X}(z) = qz/(1 - z + qz)$ and*

$$v_W(u, z) - v_W(0, z) = \frac{\lambda(1 - z + qz)\,((1-z)\,u + z^u - 1)}{(1-z)(1 - z + qz - \lambda)}, \tag{73}$$

$$v_T(u, z) - v_T(0, z) = \frac{\lambda\,qz\,((1-z)\,u + z^u - 1)}{(1-z)(1 - z + qz - \lambda)}, \tag{74}$$

from which, e.g., for the sojourn time costs $c(t) = t$, one obtains the familiar result (see Example 9),

$$v_T(u) - v_T(0) = \frac{\lambda\,(u-1)u}{2(1-\rho)}.$$

Table 3: Summary of the value generating functions in continuous and discrete time.

	$\mathbf{M^X/G/1}$: $v_W(u,s) - v_W(0,s)$	$\mathbf{Geo^X/G/1}$: $v_W(u,z) - v_W(0,z)$
With setup delay	$\frac{\lambda u}{1-\rho}\left(W_e^*(s) - \frac{1-e^{-su}}{su}\right) W_0^*(s)$	$\frac{\lambda_b u}{1-\rho}\left(\widetilde{W}_e(z) - \frac{1-z^u}{(1-z)u}\right) \widetilde{W}_0(z)$
Without setup delay	$\frac{\lambda(su + e^{-su} - 1)}{(1-\rho)s} W_0^*(s)$	$\frac{\lambda_b\left((1-z)\,u + z^u - 1\right)}{(1-z)(1-\rho)} \widetilde{W}_0(z)$
Without setup delay or batches	$\frac{su + e^{-su} - 1}{s/\lambda + X^*(s) - 1}$	$\frac{(1-z)\,u + z^u - 1}{(1-z)/\lambda + \widetilde{X}(z) - 1}$

4.3. Summary of the relationships

In the previous sections, we derived compact expressions for the **value generating functions** for the $M^X/G/1$ and $Geo^X/G/1$ queues with setup delay D with respect to the waiting and sojourn time. Table 3 summarizes the results for the waiting time. Without a setup delay, $W_e^*(s) = 1$ and $\widetilde{W}_e(z) = 1$. Without batch arrivals, $B = 1$.

For the $M^X/G/1$ queue, $W_0^*(s)$ denotes the LST of the waiting 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. These are given in (14) and (15), respectively. Moreover, in every case it holds for the corresponding value generating function with respect to sojourn time that

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

For the discrete time $Geo^X/G/1$ queue, $\widetilde{W}_0(z)$ denotes the generating function of the waiting time distribution in a system without a setup delay, and $\widetilde{W}_e(z)$ corresponds to the additional waiting time due to the setup delay D. These are given in (24) and (23). Similarly, for the sojourn time we always have

$$v_T(u,z) - v_T(0,z) = (v_W(u,z) - v_W(0,z))\, X^*(z).$$

5. Batches with dependencies in service times

So far we have assumed that batch sizes are i.i.d. random variables and jobs have i.i.d. service times. However, e.g., the proof of Proposition 2 does not require such assumptions; it is sufficient that the batches are i.i.d. random tuples. Consequently, it is straightforward to generalize our results both in continuous and discrete time to allow the service time of a job to depend, e.g., on its position in the batch and the batch size. For brevity, in what follows, we discuss the continuous time case, but equivalent results hold also in the discrete time case. Our general batch structure is defined using batch classes:

Definition 10 (Class-based batch structure) *Class $k \in \mathcal{K}$ batches consist of $b(k)$ jobs with service times*

$$(X_1^{(k)}, \ldots, X_{b(k)}^{(k)}),$$

where the $X_i^{(k)}$ may depend on each other (and on k). Then we let K denote the class and B the size of a random batch, $B = b(K)$, so that the batch consists of jobs

$$(X_1^{(K)}, \ldots, X_{b(K)}^{(K)}).$$

Note that without loss of generality we can assume a constant batch size within each class. The set of batch classes $\mathcal{K}$ can be a finite set, a countable set, or even uncountable, depending on the modeled scenario. Otherwise the batch arrival process is as before, i.e., i.i.d. batches arrive according to a Poisson process with rate λ_b.

Let $H^{(k)}$ denote the total service time of class k batch,

$$H^{(k)} \sim X_1^{(k)} + \ldots + X_{b(k)}^{(k)}.$$

The class-based batch structure induces a distribution for the total service time at the batch-level,

$$H = H^{(K)} \sim X_1^{(K)} + \ldots + X_{b(K)}^{(K)}.$$

The total service times of consecutive batches, $H_1, H_2, \ldots$, thus constitute a sequence of i.i.d. random variables, $H_i \sim H$, as before. As mentioned, the proofs given for Propositions 2 and 3 are general and hold also for the class-based batch structure:

Corollary 9 *Propositions 2 and 3 hold also with the class-based batch structure, where the cost function may depend both on the batch class k and the position of the job i in the batch, $i \in 1, \ldots, b(k)$.*

The following two examples illustrate the possibilities of the class-based batch structure:

Example 13 (Short queries and long computing jobs) *Suppose we have two types of batches. Type 1 batches consist of 1, 2 or 3 short queries (i.i.d.), and Type 2 batches consist of a single long computing job, so $\mathcal{K}$ consists of 4 batch classes, (X_1), (X_1, X_2), (X_1, X_2, X_3), (Y), where $X_i \sim X$ correspond to short queries and Y to the computing job.*

Note that we did not combine (X_1) and (Y) for two reasons, even though $B = 1$ in both cases. First, the short queries and long computing jobs may have different costs, and this information is carried in the batch class parameter. Second, in case of parallel servers, the dispatching policies may be aware of the batch class.

Example 14 (DFT) *Suppose that a server analyzes the submitted data files by computing the discrete Fourier transform (DFT) at B points. This application can be modelled with batch arrivals consisting of B jobs with (approximately) equal service times, $(X, X, \ldots, X)$, where the common random variable X is related to the length of the file. In the basic case, we assume that X is independent of the batch size B and set $K = B$; the class of a batch is its size. However, we permit the service times within each batch to be highly correlated, in contrast to the $M^X/G/1$ queue with i.i.d. service times.*

Example 15 *Consider a single server queue, where jobs arrive in batches according to a Poisson process with rate λ_b and batches are defined by a class-based batch structure (the M/G/1 and $M^X/G/1$ queues are thus special cases). Without a setup delay, the value function with respect to waiting time, $c_1(w) = w$, is the same as with the M/G/1 queue,*

$$v_1(u) - v_1(0) = \frac{\lambda u^2}{2(1-\rho)},$$

which is interesting as the class-based batch structure allows dependencies between consecutive jobs. For example, with $B = 2$ and $(X_1, X_2) \sim (X_{odd}, X_{even})$, the service time of every second job is from X_{odd}, and every other from X_{even}.

Remark 1 *Our general cost functions and general within-batch dependencies also allow us to include within-batch cost-minimizing preprocessing. For example, we could have jobs with different costs and sizes, and assume that, within a batch, they are preordered to minimize within-batch costs, i.e., by the cμ-rule.*

Also with the class-based batches, we can, in principle, easily determine distributions for such quantities as $W_b = W_b(\lambda_b, H)$ and $W_e = W_e(\lambda_b, D)$ that apply to every job in each batch. Referring to Eqs. (16) and (22), what remains is the last factor corresponding to the intra-batch delay due to the jobs ahead in the same batch.

The intra-batch delays depend solely on the batch itself, not on the state of the system, and thus they are independent of W, W_0 and D. The intra-batch delay of the m^{th} job of a class-k batch, referred to as the type (k, m) job, is given by the partial sum

$$A_{k,m} \sim X_1^{(k)} + \ldots + X_m^{(k)}, \qquad m = 1, \ldots, b(k),$$

and $A^*_{k,m}(s) = \mathrm{E}[e^{-s(X_1^{(k)} + \ldots + X_m^{(k)})}]$. Note that $A_{k,b(k)} \sim H^{(k)}$. Letting $A_{k,0} = 0$, the LST of the intra-batch waiting and sojourn times of a random class-k job are

$$W^*_{a,k}(s) = \frac{1}{b(k)} \sum_{m=0}^{b(k)-1} \mathrm{E}[e^{-sA_{k,m}}] = \frac{1}{b(k)} \sum_{m=0}^{b(k)-1} A^*_{k,m}(s),$$

$$T^*_{a,k}(s) = \frac{1}{b(k)} \sum_{m=1}^{b(k)} \mathrm{E}[e^{-sA_{k,m}}] = \frac{1}{b(k)} \sum_{m=1}^{b(k)} A^*_{k,m}(s).$$

We are interested in the waiting and sojourn times of a random job, so need to weight each batch class by its size,

$$W^*_a(s) = \frac{\mathrm{E}[b(K) \cdot W^*_{a,K}(s)]}{\mathrm{E}[B]} = \frac{\mathrm{E}[A^*_{K,0}(s) + \ldots + A^*_{K,b(K)-1}(s)]}{\mathrm{E}[B]}. \tag{75}$$

$$T^*_a(s) = \frac{\mathrm{E}[b(K) \cdot T^*_{a,K}(s)]}{\mathrm{E}[B]} = \frac{\mathrm{E}[A^*_{K,1}(s) + \ldots + A^*_{K,b(K)}(s)]}{\mathrm{E}[B]}. \tag{76}$$

The total waiting time of a random job in steady state thus decomposes into three independent terms, $W_j = W_b + W_e + W_a$, and the corresponding LST is $W^*_j(s) = W^*_b(s) \cdot W^*_e(s) \cdot W^*_a(s)$. Similarly, for the total sojourn time experienced by a random job, $T_j = W_b + W_e + T_a$, and the corresponding LST is $T^*_j(s) = W^*_b(s) \cdot W^*_e(s) \cdot T^*_a(s)$.

Remark 2 *Let X denote the service time of a random job and $X^*(s)$ its LST. Introducing dependencies into the service times within a batch implies that W_a and X are no longer necessarily independent, and therefore $T_a^*(s)$ may not be equal to $W_a^*(s) \cdot X^*(s)$.*

A trivial example is a batch structure where the service time of the i^{th} job of a batch is i. Then X and W_a are directly related, $W_a = 1 + 2 + \ldots + (X-1) = X(X-1)/2$ (a bijection).

Example 16 *In the M^X/G/1 queue (with $K = B$), service times $(X_1^{(k)}, \ldots, X_k^{(k)})$ are i.i.d. for all batch sizes k. Thus $A_{k,m}^*(s) = (X^*(s))^m$ and* (75) *reduces to*

$$W_a^*(s) = \frac{\mathrm{E}[B \cdot A_B^*(s)]}{\mathrm{E}[B]} = \frac{\sum_{k=1}^{\infty} \mathrm{P}\{B=k\}\left((X^*(s))^0 + \ldots + (X^*(s))^{k-1}\right)}{\mathrm{E}[B]} = \frac{\sum_{k=1}^{\infty} \mathrm{P}\{B \geq k\} \cdot X^*(s)^k}{X^*(s) \cdot \mathrm{E}[B]} = \widetilde{A}(X^*(s)),$$

where $\widetilde{A}(z)$ is from (7)*, as expected. The same holds also for the discrete time Geo^X/G/1 queue and sojourn times.*

As mentioned earlier, our results generalize to the class-based batch structure. For example, Propositions 4 and 5 give the mean cost rates and the value generating functions with respect to the exponential waiting time and sojourn time costs $c_w(w) = 1 - e^{-sw}$ and $c_t(w,x) = 1 - e^{-s(w+x)}$.

Corollary 10 *Propositions 4 and 5 hold also for the class-based batch structure with*

$$W_0^*(s) = W_b^*(s) \cdot W_a^*(s), \quad \textit{and} \quad T_0^*(s) = W_b^*(s) \cdot T_a^*(s),$$

where $W_a^*(s)$ and $T_a^*(s)$ are now given by (75) *and* (76)*.*

We can also define job-specific holding costs separately for each job of batch class k.

We now return to the two example batch structures introduced at the start of the section:

Example 17 (short queries and long computing jobs) *Let $\widetilde{B}(z)$ denote the generating function for the number of short queries in a type 1 batch, and suppose that the fraction of type 1 batches is p. Then the LST for the total service time of a batch is*

$$H^*(s) = p \cdot \widetilde{B}(X^*(s)) + (1-p) \cdot Y^*(s).$$

For the intra-batch waiting time we have

$$W_a^*(s) = \frac{1}{1 + p(\mathrm{E}[B]-1)}\left[1 + p\left(\mathrm{E}[B]\widetilde{A}(X^*(s)) - 1\right)\right],$$

where $\widetilde{A}(X^*(s)$ corresponds to the number of queries ahead of a random (short) query in the same batch, see (7)*.*

Example 18 (DFT) *Define*

$$A_{k,m}^*(s) = \mathrm{E}[e^{-smX}] = X^*(sm),$$

so

$$H^*(s) = \mathrm{E}[X^*(sK)] = \sum_k \mathrm{P}\{B=k\} \cdot X^*(ks),$$

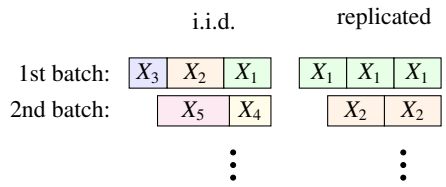
$$W_a^*(s) = \frac{\mathrm{E}[B \cdot A_B^*(s)]}{\mathrm{E}[B]} = \frac{1}{\mathrm{E}[B]} \sum_k \mathrm{P}\{B=k\}\left(X^*(0) + \ldots + X^*((k-1)s)\right) = \frac{1}{\mathrm{E}[B]} \sum_k \mathrm{P}\{B \geq k\} \cdot X^*((k-1)s).$$

Note that with the i.i.d. service times of the M^X/G/1 queue, according to (7) *and* (12)*, we have*

$$H^*(s) = \sum_k \mathrm{P}\{B=k\} \cdot X^*(s)^k, \quad \textit{and} \quad W_a^*(s) = \frac{1}{\mathrm{E}[B]} \sum_k \mathrm{P}\{B \geq k\} \cdot X^*(s)^{k-1}.$$

Table 4 shows the mean and variance of H and W_a when $B \sim \mathrm{U}(1,2,3)$ for the three cases where X is a constant, uniformly distributed and exponentially distributed, for the two cases of i.i.d. X_i's and perfectly correlated (replicated) X_i's ($X_1 = X_2 \ldots$). With i.i.d. service times, the system benefits from statistical multiplexing; it is unlikely that all three jobs are "long".

Table 4: The variance of H and W_a with different service time distributions X with $\mathrm{E}[X] = m$ when $B \sim \mathrm{U}(1, 2, 3)$ and the jobs are i.i.d. or replicated as in Example 18. The means remain the same, $\mathrm{E}[H] = \mathrm{E}[B] \cdot \mathrm{E}[X] = 2m$ and $\mathrm{E}[W_a] = (2/3)m$, but replication increases the variances as expected.



Job's service time X		$\mathrm{V}[H]$		$\mathrm{V}[W_a]$	
Distribution	$X^*(s)$	i.i.d.	replicate	i.i.d.	replicate
Const(m)	e^{-sm}	$\frac{2m^2}{3}$	$\frac{2m^2}{3}$	$\frac{5m^2}{9}$	$\frac{5m^2}{9}$
U$(0, 2m)$	$\frac{1-e^{-sm}}{sm}$	$\frac{4m^2}{3}$	$\frac{20m^2}{9}$	$\frac{7m^2}{9}$	$\frac{8m^2}{9}$
Exp(μ), $m = \frac{1}{\mu}$	$\frac{1}{1+sm}$	$\frac{8m^2}{3}$	$\frac{16m^2}{3}$	$\frac{11m^2}{9}$	$\frac{14m^2}{9}$

6. Number-aware value functions

So far, we have assumed that exact job size information is available. This is a justified assumption in many modern applications, where, e.g., file sizes are known in advance. However, traditionally it has been more common to assume that only the number of jobs present is known, a scenario referred to as the number-aware setting. In this case, it is common to assume exponentially or geometrically distributed service times to take advantage of the lack of memory property.

By unconditioning, the size-aware results can be transformed to cases where only inaccurate information about the service times and setup delay are available (e.g., we may have class-specific service time distributions). In particular, one can easily obtain the number-aware value functions from the size-aware value functions. For example, in [27], the size-aware value functions for the M/G/1-FCFS and LCFS queues (without setup delay) are transformed to the corresponding number-aware value functions when the cost structure is simply the sojourn time, $c(w, x) = w + x$. In this section, we derive the number-aware value functions for the M^{X}/G/1 and $\mathrm{Geo}^{\mathrm{X}}$/G/1 queues with setup delay and exponential cost functions, from which, e.g., arbitrary polynomial cost functions follow as special cases.

In the number-aware case, the backlog is a random variable,

$$U = X_1 + \ldots + X_{n-1} + R_t,$$

where n denotes the number of jobs in the queue, and R_t is the residual service time of a job given that it has already been in service for time t, $R_t \sim (X - t \mid X > t)$. With arbitrary service time distributions, this leads to an age-based system where the conditional distribution R_t needs to be evaluated. For the M^{X}/G/1 queue, the value generating functions, given in (50) and (58), depend on su and e^{-su}, for which we have

$$\mathrm{E}[sU] = s\left((n-1)\mathrm{E}[X] + \mathrm{E}[R_t]\right),$$

and

$$\mathrm{E}[e^{-sU}] = [X^*(s)]^{n-1} \cdot R_t^*(s),$$

with similar expressions for the $\mathrm{Geo}^{\mathrm{X}}$/G/1 queue.

For simplicity, we assume no setup delay and memoryless (exponential or geometric) service time distributions, so we have the M^{X}/M/1 and $\mathrm{Geo}^{\mathrm{X}}$/Geo/1 queues, and in particular, $R_t \sim X$ in both cases.

6.1. $M^X/M/1$ Queue

For the M^{X}/M/1 queue, we now have

$$\begin{aligned} X^*(s) &= \mu/(s+\mu), \\ \mathrm{E}[sU] &= sn/\mu, \\ \mathrm{E}[e^{-sU}] &= (X^*(s))^n = \left[\frac{\mu}{s+\mu}\right]^n. \end{aligned}$$

Let us consider the exponential cost function of the waiting time. For the M^{X}/M/1 queue, (54) becomes

$$v_W(n,s) - v_W(0,s) = \frac{\left(\left[\frac{\mu}{s+\mu}\right]^n + \frac{sn}{\mu} - 1\right)\mathrm{E}[B]\,\widetilde{A}(\mu/(s+\mu))}{s/\lambda_b + \widetilde{B}(\mu/(s+\mu)) - 1}. \tag{77}$$

Without batch arrivals, i.e., for the basic M/M/1 queue, we have

$$v_W(n,s) - v_W(0,s) = \frac{\lambda(s+\mu)}{s(s+\mu-\lambda)} \left(\left[\frac{\mu}{s+\mu} \right]^n + \frac{sn}{\mu} - 1 \right), \tag{78}$$

and

$$v_T(n,s) - v_T(0,s) = \frac{\lambda\mu}{s(s+\mu-\lambda)} \left(\left[\frac{\mu}{s+\mu} \right]^n + \frac{sn}{\mu} - 1 \right). \tag{79}$$

The admission cost, e.g., with respect to the exponential cost of the sojourn time is

$$a_T(n,s) = \mathrm{E}[c_T(n,X,s)] + v_T(n+1,s) - v_T(n,s),$$

which reduces to

$$a_T(n,s) = \frac{s+\mu-\mu\left(\frac{\mu}{s+\mu}\right)^n}{s+\mu-\lambda} = 1 + \frac{\lambda-\mu\left(\frac{\mu}{s+\mu}\right)^n}{s+\mu-\lambda}. \tag{80}$$

The above is the admission cost to the (number-aware) M/M/1 queue with the exponential cost of the sojourn time.

Example 19 *The number-aware M/M/1 queue with an arbitrary cost function is studied also in [9]. Instead of deriving the value function, the admission cost, $a(i) \triangleq c(i) + v(i+1) - v(i)$, involving an infinite sum, is derived directly [9, Eq. (8)]. An explicit closed-form admission cost is then derived for some example cases including the squared sojourn time, $c(t) = t^2$. From* (80)*, we obtain for the linear cost of the sojourn time $c(t) = t$,*

$$a_T(n) = -\left[\frac{\partial}{\partial s} a_T(n,s) \right]_{s=0} = \frac{n+1}{\mu-\lambda},$$

whereas for $c(t) = t^2$, we have

$$a_{T^2}(n) = -\left[\frac{\partial^2}{\partial^2 s} a_T(n,s) \right]_{s=0} = \frac{(n+1)(n(1-\rho)+2)}{(\mu-\lambda)^2},$$

which agrees with the admission cost given in [9, Example 2].

6.2. Geo^X/Geo/1 Queue

For the equivalent discrete-time Geo^X/Geo$_1$/1 system we have

$$\begin{aligned} \widetilde{X}(z) &= \frac{qz}{1-z+qz}, \\ \mathrm{E}[U] &= \frac{n}{q}, \\ \mathrm{E}[z^U] &= [\widetilde{X}(z)]^n = \left[\frac{qz}{1-z+qz} \right]^n. \end{aligned}$$

Substituting these into (70) yields the number-aware value function with respect to the number-aware exponential cost function of the waiting time (62),

$$v_W(n,z) - v_W(0,z) = \frac{((1-z)\frac{n}{q} + \left[\frac{qz}{1-z+qz}\right]^n - 1)\,\widetilde{A}\left(\frac{qz}{1-z+qz}\right)}{(1-z)/\lambda_b + \widetilde{B}\left(\frac{qz}{1-z+qz}\right) - 1}.$$

Without batch arrivals ($B = 1$) one obtains the Geo/Geo/1 queue and the expression simplifies further yielding,

$$\begin{aligned} v_W(n,z) - v_W(0,z) &= \frac{\lambda(1-z+qz)\left[(1-z)\,\frac{n}{q} + \left[\frac{qz}{1-z+qz}\right]^n - 1\right]}{(1-z)(1-z+qz-\lambda)}, \\ v_T(n,z) - v_T(0,z) &= \frac{\lambda qz\left[(1-z)\,\frac{n}{q} + \left[\frac{qz}{1-z+qz}\right]^n - 1\right]}{(1-z)(1-z+qz-\lambda)}. \end{aligned}$$

The (expected) immediate cost with respect to the exponential cost of the sojourn time is

$$c_T(n,z) \triangleq \mathrm{E}[c_T(U,X,z)\,|\,n] = \mathrm{E}[1-z^{(U+X)}\,|\,n] = 1-\left[\frac{qz}{1-z+qz}\right]^{n+1},$$

and the corresponding admission cost, $a_T(n,z) = c_T(n,z) + v_T(n+1,z) - v_T(n,z)$, is then

$$a_T(n,z) = \frac{1-z+qz-\lambda(1-z)-qz\left[\frac{qz}{1-q+qz}\right]^n}{1-z+qz-\lambda} = 1+\frac{\lambda z - qz\left[\frac{qz}{1-q+qz}\right]^n}{1-z+qz-\lambda}.$$

Example 20 *For the (linear) cost of the sojourn time, $c(t) = t$, the admission cost in the number-aware Geo/Geo/1 queue (without setup delay) is*

$$a_T(n) = -\left(\frac{\partial}{\partial z}a_T(n,z)\right)_{z=1} = \frac{1+n-\lambda}{q-\lambda}.$$

and for the quadratic cost of the sojourn time, $c(t) = t^2$, we have

$$a_{T^2}(n) = -\left(\frac{\partial}{\partial z}z\frac{\partial}{\partial z}a_T(n,z)\right)_{z=1} = \frac{(1-\rho)n(n+1)+(2-q-\lambda)(n+1-\lambda)}{(q-\lambda)^2}.$$

As mentioned, the continuous time system can obtained by an appropriate scaling at the limit where the time step tends to zero. More specifically, for example by scaling the arrival and service rates by δ,

$$\lambda = \delta\lambda, \quad q = \delta\mu,$$

and the admission costs by δ and δ^2, respectively, the above results yield the corresponding admission costs for the M/M/1 queue (see Example 19) when $\delta \to 0$.

7. Applications

In this section, we show how the new results can be utilized. As it is obvious by now, all our results are similar in continuous and discrete time. Thus, for simplicity, in this section we consider only the continuous time $\mathrm{M^X}$/G/1 queue and leave the discrete-time system for the reader.

7.1. To switch off, or not to switch off

Let us first discuss briefly switching-off policies [31]. We determine when an idle server should be switched off, and when it should be 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 because of the setup delay. 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 whenever idle follows immediately from (9),

$$\bar{c}_e = (1-p_0)e = \frac{\rho+\lambda_b\,\mathrm{E}[D]}{1+\lambda_b\,\mathrm{E}[D]}\,e. \tag{81}$$

The value function with respect to energy consumption is [4],

$$v_e(u) - v_e(0) = \frac{u}{1+\lambda_b\,\mathrm{E}[D]}\,e. \tag{82}$$

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

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

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

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

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

1. The straightforward method is to compute the mean cost $\bar{c} = \bar{c}_e + \bar{c}_W$ 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
$$\mathrm{E}[C(D) - C(0) + v_0(D + H) - v_0(H)] < \frac{e}{\lambda_b}.$$
3. Alternatively, let $v_0(u)$ 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 (82), is included explicitly in the value function $v_0(u)$. 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 $v_0(\epsilon) < v_0(0)$, which yields
$$\frac{v_0(\epsilon) - v_0(0)}{\epsilon} < 0,$$
and as $\epsilon \to 0$, the condition reduces to checking the sign of the derivative at $u = 0$,
$$\left[\frac{\partial}{\partial u} v_0(u)\right]_{u=0} < 0.$$
Considering the weighted cost model for jobs, with the aid of the value generating function (50), we can write an explicit expression for keeping the server running,
$$\left[\frac{\partial}{\partial u} \sum_i a_i \left[\frac{\partial^i}{\partial s^i} v(u, s)\right]_{s=0}\right]_{u=0} > \frac{e}{1 + \lambda_b \mathrm{E}[D]}.$$
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.

In each case 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 can ignore policies that delay the switch on after a job has arrived. Our analysis also extends easily to the case where the energy cost depends on whether the server is working or is setting up.

7.2. Dispatching policies

The value functions for single-server queues are also useful in the context of (heterogeneous) parallel servers (see Figure 1). In particular, they can be used to develop efficient dispatching (routing) policies by carrying out one policy improvement step. This method, first proposed by Norman [48], has been used in numerous papers, see, e.g., [33, 49, 23, 22], and [14, Sec.10.3. and 11.5]. The basic steps are as follows (similarly in discrete time):

1. Choose a static *basic dispatching policy* α_0 that *splits* the Poisson arrival process to to different servers so that each of them receives jobs according to a (server-specific) Poisson batch arrival process.
2. With α_0 the system then *decomposes* into n independent parallel queues that can be analyzed separately, and the value function of the whole system is the sum of queue-specific value functions,
$$v(\mathbf{z}) = v^{(1)}(u_1) + \ldots + v^{(n)}(u_n),$$
where the backlogs, $\mathbf{z} = (u_1, \ldots, u_n)$, are a sufficient state description in our setting.
3. One step of 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, $c(\mathbf{z}, \xi)$ the corresponding immediate cost, and $\mathbf{z} \oplus \xi$ the resulting state. Then
$$\alpha_{\mathrm{FPI}}(\mathbf{z}, \mathbf{x}) \in \arg\min_{\xi} \left(c(\mathbf{z}, \xi) + v(\mathbf{z} \oplus \xi) - v(\mathbf{z})\right),$$
where α_{FPI} is the set of servers chosen by the FPI policy. Ties can be resolved, e.g., at random. Note that $v(\mathbf{z})$ is a common constant for all actions and can be omitted.

The basic dispatching policy may depend on batch size B, or any batch-specific parameters such as class, weight or priority. The basic policy can also *split* batches among several servers (cf. [50]) as long as the decision is static and each server receives a Poisson batch arrival process. One such 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. SITA is often (but not always) a good policy with FCFS (cf. [51]). Similarly, a random Bernoulli split of jobs in a batch, instead of the whole batch, decreases the burstiness in the arrival process to each server, 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 at the servers), or the parameters of the batch in such a way that the resulting arrival process to some server would not be a Poisson batch arrival process.

Value functions for a single queue thus allow us to carry out the one policy improvement step to determine a policy that uses server-specific state information. It is often difficult to go further, i.e., to carry out a second policy improvement step, because the first step already yields a dynamic policy for which it is significantly harder to compute the value function than for a single queue. 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) [30]. For numerical examples, we refer to the cited publications.

8. Conclusions

We have studied continuous and discrete time queues with general batch arrivals subject to general cost structures. This includes the $M^X/G/1$ and $Geo^X/G/1$ queues as special cases. We obtained compact closed-form expressions for the mean costs and the value functions defined solely in terms of the arrival intensity, the batch-specific generating functions and either the LST (for $M^X/G/1$) or z-transform (for $Geo^X/G/1$) of the service time distribution. These cost functions include many often used cost structures such as the waiting time, the slowdown, and energy consumption. We refer to these 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. Our results, with convenient simple expressions for the mean costs and value functions, serve as building blocks for more complex scenarios that attempt to optimize dynamic actions, e.g., for dispatching jobs to parallel servers. Finally, it is worth noting that it is possible to introduce dependencies and different service time distributions for jobs in a batch with very little additional analysis.

Acknowledgements

This work was partially supported by the University of Iceland Research Fund in the RL-STAR project.

References

[1] R. R. Weber, On the optimal assignment of customers to parallel servers, Journal of Applied Probability 15 (2) (1978) 406–413.
[2] A. Hordijk, G. Koole, On the optimality of the generalised shortest queue policy, Prob. Eng. Inf. Sci. 4 (1990) 477–487.
[3] O. Akgun, R. Righter, R. Wolff, Multiple server system with flexible arrivals, Advances in Applied Probability 43 (2011) 985–1004.
[4] E. Hyytiä, A. Penttinen, S. Aalto, Size- and state-aware dispatching problem with queue-specific job sizes, European Journal of Operational Research 217 (2) (2012) 357–370.
[5] E. Hyytiä, R. Righter, S. Aalto, Task assignment in a heterogeneous server farm with switching delays and general energy-aware cost structure, Performance Evaluation 75–76 (0) (2014) 17–35.
[6] S. Yang, G. de Veciana, Size-based adaptive bandwidth allocation: optimizing the average QoS for elastic flows, in: IEEE INFOCOM, Vol. 2, 2002, pp. 657–666.
[7] M. Harchol-Balter, K. Sigman, A. Wierman, Asymptotic convergence of scheduling policies with respect to slowdown, Perform. Eval. 49 (1-4) (2002) 241–256.
[8] E. Hyytiä, S. Aalto, A. Penttinen, Minimizing slowdown in heterogeneous size-aware dispatching systems, ACM SIGMETRICS Performance Evaluation Review 40 (2012) 29–40, (ACM SIGMETRICS/Performance conference).
[9] N. Argon, L. Ding, K. Glazebrook, S. Ziya, Dynamic routing of customers with general delay costs in a multiserver queuing system, Probability in the Engineering and Informational Sciences 23 (2009) 175–203.
[10] E. Hyytiä, R. Righter, Routing jobs with deadlines to heterogeneous parallel servers, Operations Research Letters 44 (4) (2016) 507–513.
[11] E. Hyytiä, R. Righter, J. Virtamo, Meeting soft deadlines in single- and multi-server systems, in: 28th International Teletraffic Congress (ITC'28), Würzburg, Germany, 2016.
[12] J. Niño-Mora, Resource allocation and routing in parallel multi-server queues with abandonments for cloud profit maximization, Computers & Operations Research 103 (2019) 221–236.
[13] A. Penttinen, E. Hyytiä, 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, 2011.
[14] P. Whittle, Optimal Control: Basics and Beyond, Wiley, 1996.
[15] M. L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, Wiley, 2005.

[16] K. R. Krishnan, Joining the right queue: a Markov decision rule, in: Proc. of the 28th Conference on Decision and Control, 1987, pp. 1863–1868.
[17] S. Doroudi, E. Hyytiä, M. Harchol-Balter, Value driven load balancing, Performance Evaluation 79 (2014) 306–327, (IFIP Performance'14). doi:10.1016/j.peva.2014.07.019.
[18] R. Bellman, Dynamic programming, Princeton University Press, 1957.
[19] R. A. Howard, Dynamic Probabilistic Systems, Volume II: Semi-Markov and Decision Processes, Wiley Interscience, 1971.
[20] P. S. Ansell, K. D. Glazebrook, C. Kirkbride, Generalised 'join the shortest queue' policies for the dynamic routing of jobs to multi-class queues, The Journal of the Operational Research Society 54 (4) (2003) 379–389.
[21] J. Leino, J. Virtamo, Determining the moments of queue-length distribution of discriminatory processor-sharing systems with phase-type service requirements, in: Proceedings of NGI 2007, Trondheim, Norway, 2007, pp. 205–208.
URL http://dx.doi.org/10.1109/NGI.2007.371217
[22] S. Bhulai, On the value function of the M/Cox(r)/1 queue, Journal of Applied Probability 43 (2) (2006) 363–376.
[23] S. A. E. Sassen, H. C. Tijms, R. D. Nobel, A heuristic rule for routing customers to parallel servers, Statistica Neerlandica 51 (1) (1997) 107–121.
[24] S. Aalto, J. Virtamo, Basic packet routing problem, in: The thirteenth Nordic teletraffic seminar NTS-13, Trondheim, Norway, 1996, pp. 85–97.
[25] E. Hyytiä, A. Penttinen, S. Aalto, J. Virtamo, Dispatching problem with fixed size jobs and processor sharing discipline, in: 23rd International Teletraffic Congress (ITC'23), San Fransisco, USA, 2011, pp. 190–197.
[26] E. Hyytiä, J. Virtamo, S. Aalto, A. Penttinen, M/M/1-PS queue and size-aware task assignment, Performance Evaluation 68 (11) (2011) 1136–1148, (IFIP Performance'11).
[27] E. Hyytiä, S. Aalto, A. Penttinen, J. Virtamo, On the value function of the M/G/1 FCFS and LCFS queues, Journal of Applied Probability 49 (4) (2012) 1052–1071.
[28] E. Hyytiä, R. Righter, S. Aalto, Energy-aware job assignment in server farms with setup delays under LCFS and PS, in: 26th International Teletraffic Congress (ITC'26), Karlskrona, Sweden, 2014.
[29] E. Hyytiä, R. Righter, Fairness through linearly increasing holding costs in systems of parallel servers with setup delays, in: 27th International Teletraffic Congress (ITC'27), Ghent, Belgium, 2015.
[30] E. Hyytiä, Lookahead actions in dispatching to parallel queues, Performance Evaluation 70 (10) (2013) 859–872, (IFIP Performance'13).
[31] E. Hyytiä, D. Down, P. Lassila, S. Aalto, Dynamic control of running servers, in: 19th International GI/ITG Conference on "Measurement, Modelling and Evaluation of Computing Systems" (MMB), Vol. LNCS 10740, Springer Verlag, 2018, pp. 127–141.
[32] E. Hyytiä, R. Righter, O. Bilenne, X. Wu, Dispatching discrete-size jobs with multiple deadlines to parallel heterogeneous servers, in: A. Puliafito, K. Trivedi (Eds.), Systems modeling: methodologies and tools, EAI/Springer Innovations in Communications and Computing, Springer, 2018, pp. 29–46.
[33] K. R. Krishnan, T. J. Ott, State-dependent routing for telephone traffic: Theory and results, in: IEEE Conference on Decision and Control, Vol. 25, 1986, pp. 2124–2128.
[34] J. van Leeuwaarden, S. Aalto, J. Virtamo, Load balancing in cellular networks using first policy iteration, Technical report, Networking Laboratory, Helsinki University of Technology (2001).
[35] E. Hyytiä, S. Aalto, On round-robin routing policy with FCFS and LCFS scheduling, Performance Evaluation 97 (2016) 83–103.
[36] E. Hyytiä, G. Magnússon, R. Righter, Controlling queues with constant interarrival times, in: 31st International Teletraffic Congress (ITC'31), Budapest, Hungary, 2019.
[37] E. Hyytiä, R. Righter, J. Virtamo, L. Viitasaari, Value (generating) functions for the M^X/G/1 queue, in: 29th International Teletraffic Congress (ITC'29), Genoa, Italy, 2017.
[38] G. Choudhury, An M^X/G/1 queueing system with a setup period and a vacation period, Queueing Systems 36 (1) (2000) 23–38.
[39] J.-C. Ke, Batch arrival queues under vacation policies with server breakdowns and startup/closedown times, Applied Mathematical Modelling 31 (7) (2007) 1282–1292.
[40] R. Bolla, R. Bruschi, A. Carrega, F. Davoli, Green networking with packet processing engines: Modeling and optimization, IEEE/ACM Transactions on Networking 22 (1) (2014) 110–123.
[41] R. Bolla, R. Bruschi, A. Carrega, F. Davoli, J. F. Pajo, Corrections to: "Green Networking With Packet Processing Engines: Modeling and Optimization", IEEE/ACM Transactions on Networking PP (99).
[42] L. Kleinrock, Queueing Systems, Volume I: Theory, Wiley Interscience, 1975.
[43] H. C. Tijms, Stochastic Models, An Algorithmic Approach, John Wiley & Sons, 1994.
[44] 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 12 (5) (1964) 736–752.
[45] S. Fuhrmann, R. Cooper, Stochastic decomposition in the M/G/1 queue with generalized vacations, Operations Research 33 (5) (1985) 1117–1129.
[46] J. G. Shanthikumar, On stochastic decomposition in M/G/1 type queues with generalized server vacations, Operations Research 36 (4) (1988) 566–569.
[47] S. K. Bose, An Introduction to Queueing Systems, Springer, 2002.
[48] J. M. Norman, Heuristic procedures in dynamic programming, Manchester University Press, 1972.
[49] K. R. Krishnan, Joining the right queue: a state-dependent decision rule, IEEE Transactions on Automatic Control 35 (1) (1990) 104–108.
[50] E. Hyytiä, S. Aalto, To split or not to split: Selecting the right server with batch arrivals, Operations Research Letters 41 (4) (2013) 325–330.
[51] M. Harchol-Balter, A. Scheller-Wolf, A. R. Young, Surprising results on task assignment in server farms with high-variability workloads, in: Proc. of SIGMETRICS, 2009, pp. 287–298.

Appendix A. Proofs

Proof: (Proposition 1) We can prove (19), giving the generating function of the waiting time W in the Geo/G/1 queue with a random setup delay D, by using the exponential cost function of the waiting time as they share the same term, z^w. Thus, suppose the costs are incurred according to (62), $c_W(w, z) = 1 - z^w$. From Proposition 3, it follows that the

corresponding value function satisfies the following:

$$v_W(u,z) - v_W(0,z) = \frac{(1-z)\,u\,\widetilde{W}(z) - (1-z^u)\,\widetilde{W}_0(z)}{(1-z)(1-\rho)/\lambda}. \tag{A.1}$$

Consider next the state where the system is empty, which is the only state where the setup delay takes place. The average time the system is empty is $1/\lambda$, and by dynamic programming,

$$v_W(0,z) = \frac{1}{\lambda}\cdot(0-r) + \mathrm{E}[c(D)] + \mathrm{E}[v_W(D+X,z)],$$

where $r = \lambda\,\mathrm{E}[c(W)]$ denotes the mean cost rate, yielding

$$\mathrm{E}[v_W(D+X,z) - v_W(0,z)] = \mathrm{E}[c(W) - c(D)]. \tag{A.2}$$

Combining (A.1) and (A.2), and substituting $\widetilde{W}(z) = \widetilde{W}_e(z)\cdot\widetilde{W}_0(z)$, where $\widetilde{W}_0(z)$ is given by (18), gives, after some algebraic manipulation,

$$\widetilde{W}_e(z) = \frac{(1-z-\lambda)\widetilde{D}(z) + \lambda}{(1-z)(1+\lambda\,\mathrm{E}[D])}.$$

□