Round-Robin Routing Policy

Value Functions and Mean Performance with Job- and Server-specific Costs

Esa Hyytiä
Department of Communications and Networking
Aalto University, Finland
esa.hyytia@aalto.fi

Samuli Aalto
Department of Communications and Networking
Aalto University, Finland
samuli.aalto@aalto.fi

ABSTRACT

We study the Round-Robin (RR) routing to a system of parallel queues. The cost structure comprises two components: a service fee and a queueing delay related component, where both can be job- and queue-specific random variables. With Poisson arrivals, the inter-arrival time to each queue obeys Erlang's distribution. This allows us to study the mean and transient behavior of the queues separately. The service fee is independent of the queueing, and we obtain the corresponding mean cost rate and value function in closed forms. With respect to queueing delay, we first derive integral expressions enabling efficient computation of the corresponding value function. By decomposition, these yield also the value function for the whole system of m parallel queues fed by RR. Given the value function, one can carry out the first policy iteration step with arbitrary holding cost rates (e.g., delay, slowdown etc.) yielding efficient size-, cost- and state-aware policies. Moreover, the mean waiting time in an M/G/m-RR system gets resolved at the same time. The results are demonstrated in the numerical examples, where we compute near optimal task assignment policies for a sample system with two servers.

Keywords

Round-Robin, M/G/m-RR, Erl/G/1 queue, Task assignment, Dispatching, Parallel queues, MDP

1. INTRODUCTION

In the task assignment (or routing) problems, one chooses a server for each new job immediately upon the arrival. The objective is to minimize the mean response time, slowdown, energy consumption, or some other performance quantity of interest. Within each queue, the First-Come-First-Served (FCFS) scheduling is usually assumed, but other scheduling disciplines can also be considered. Even though task assignment problems have been studied extensively in the literature, only a few optimality results are known, and these generally require homogeneous servers.

VALUETOOLS 2013, December 10-12, Torino, Italy
Copyright © 2013 ICST 978-1-936968-48-0
DOI 10.4108/icst.valuetools.2013.254379

The *Join-the-Shortest-Queue* (JSQ) policy assigns a new job to the server with the fewest tasks. Assuming exponentially distributed inter-arrival times and job sizes, and homogeneous servers, [33] showed that JSQ, followed by FCFS, minimizes the mean delay. Since then the optimality of JSQ has been shown in many other settings [32, 9, 18, 31, 29, 24, 1]. Similarly, *Round-Robin* (RR), followed by FCFS, has been shown to be the optimal policy when it is only known that the queues were initially in the same state, and the routing history is available [9, 27, 26]. For RR combined with the *Shortest-Remaining-Process-Time* (SRPT) scheduling, see [8]. With a Poisson arrival process and RR, the inter-arrival times to each queue obey Erl(m, λ) distribution, where m denotes the number of servers [15].

The *Least-Work-Left* (LWL) policy assigns a new job to the queue with the least amount of unfinished work. LWL is equivalent to M/G/m with a shared queue [16, 15], and thus it makes sure no server is idle when there are jobs in the queue. Interestingly, with constant service times, LWL, JSQ and RR make equivalent decisions as in M/D/m (with a shared queue), which can also be shown to be optimal with respect to the mean delay.

The M/G/m system is non-trivial to analyze. More results are available for M/D/m. The first analytical result for the distribution of waiting time in M/D/m is by Crommelin [6]. Numerically much more stable expressions are given by Franx [11], who has also analyzed its transient behavior in [12]. In particular, [11] gives the waiting time distribution as a function of the queue length distribution, allowing also the determination of the mean waiting and sojourn times. For extensive surveys on the topic we refer to [30, 23].

Value functions for single server queues with *Poisson arrivals* have been derived in [25, 2, 5, 21], which form the basis also for value functions for task assignment systems operating under a state-independent policy such as the Bernoulli-split. With state-dependent policies, such as RR and LWL, the queues are coupled and the analysis gets more complicated. In this paper, we derive a set of integral equations that enable efficient computation of the value function with respect to arbitrary holding cost based cost structure in a size-aware task assignment system subject to RR routing policy and FCFS scheduling discipline. Jobs arrive according to a Poisson process with rate λ and the servers are assumed to be identical.[1] As an interesting and useful side

[1] In fact, the analysis does not depend on this and the results hold with very minor modifications also for heterogeneous systems. However, the Round-Robin policy is not an ideal candidate if the service rates are highly asymmetric.

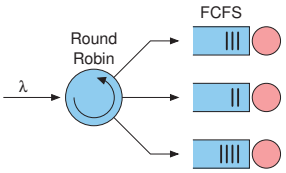

Figure 1: Round-Robin routing to m servers.

product, these expressions also provide a new approach to determine the mean waiting time in an M/G/m-RR system, whereas using the results given in [11] for M/D/m-RR requires the determination of the queue length distribution as an intermediate result.

The rest of the paper is organized as follows. First, in Section 2, we analyze a single Erl/G/1 queue and derive both differential and integral expressions for the value function. Due to the decomposition, the value function for M/G/m-RR is also obtained. Section 3 discusses the task assignment problem and policy iteration, and gives some numerical examples. Section 4 sheds light on possible more elaborate applications, and Section 5 concludes the paper.

2. ANALYSIS OF THE ROUND-ROBIN

In this section, we analyze the Round-Robin routing policy illustrated in Fig. 1. With Poisson arrivals, the arrival process to each queue is Erl(m,λ)/G/1, which facilitates the analysis. First we describe the model and the cost structure, and then derive expressions for calculating the value functions for *arbitrary job- and queue-specific service fees and holding cost rates*, which also give the mean delay.

2.1 Model and Cost Structure

We consider a stable M/G/m-RR system illustrated in Fig. 1. With a Poisson arrival process, each queue behaves according to an Erl(m,λ)/G/1 queue, where the inter-arrival time in each queue is a sum of m independent and exponentially distributed time-intervals, i.e., phases, with mean durations of $1/\lambda$. Jobs arrive at the end of phase m, after which a new phase 1 starts. The evolution of the queues is naturally coupled as they see the same Poisson process, while each of them is in a different phase $(1,\ldots,m)$. The cost structure comprises two components. First, Job j pays a *service fee* of S_j upon entering a queue. Second, it incurs costs at *holding cost rate* H_j while waiting in a queue.[2]

We further assume that the service time X, the service fee S and the holding cost rate H all may depend on the queue a job is assigned to. Thus, in general, Job j is defined by triples $(X_{j,k},H_{j,k},S_{j,k})$, where k corresponds to the queue, $k=1,\ldots,m$. In practice, the holding cost rate is often job-specific, $H_{j,k}=H_j$ and the service fee is either a server-specific constant or a function of service time, $S_{j,k}=g_k(X_{j,k})$ (e.g., per bit charging in mobile networks). That is, one can associate the holding cost with the queueing delay a job experiences and the service fee with a (server-specific) service cost/time (e.g., energy). In particular, we assume that jobs are i.i.d.,

$$(\mathbf{X}_j,\mathbf{H}_j,\mathbf{S}_j)\sim(\mathbf{X},\mathbf{H},\mathbf{S}),$$

while the different variables of a single job may depend on each other (e.g., the service fee can be equal to the service time). Note that with $H=1$ and $S=X$, the costs incurred are equal to the sojourn time.

With RR, the jobs are assigned sequentially, independently of their holding cost rates and service fees, and hence the above cost structure is unnecessarily complicated. However, later in Section 3, we consider also routing policies that take into account the job- and server-specific characteristics, and hence the notation.

2.2 State description for Round-Robin

When all servers are identical, the service fees can be omitted and the state of the Round-Robin system can be described by an m-tuple,

$$\mathbf{z}=(u_1,\ldots,u_m),$$

where u_i denotes the backlog in the queue currently in phase i. Similarly, when considering the service fees, only the phase matters and a sufficient state description is

$$\mathbf{z}=(q_1,\ldots,q_m),$$

where q_i is the index of the queue currently in phase i. In general, the state of the system can be described by

$$\mathbf{z}=((q_1,u_1),\ldots,(q_m,u_m)),$$

where (q_i,u_i) denotes the server and its backlog that is currently in phase i. Therefore, $(q_1,\ldots,q_m)$ is some permutation of $(1,\ldots,m)$, whereas $u_i\geq 0$ for all i.

2.3 Service Fees

Let us first consider the service fees each Erl/G/1 queue incurs. Let $S_j\sim S$ denote the service fee of the jth job assigned to a given queue. Then let i denote the current phase in the arrival process, $i=1,\ldots,m$, such that at the end of phase m a job arrives and a cost of S_j is incurred.

A sufficient state description with respect to service fees is the current phase of the arrival process. Consequently, the corresponding *value function* depends also only on the current phase, and it is defined as the expected difference between a system initially in phase i and a system initially in equilibrium,

$$v_i\triangleq\lim_{t\to\infty}\mathrm{E}[V_i(t)-r_st], \tag{1}$$

where $V_i(t)$ denotes the service fees incurred during $(0,t)$ when initially in phase i, and r_s is the mean rate at which service fees are incurred,

$$r_s=\frac{\lambda\,\mathrm{E}[S]}{m}.$$

Proposition 1. *The value function with respect to service fees for an Erl(m,λ)/G/1 queue is*

$$v_i=\frac{2i-m-1}{2m}\mathrm{E}[S]. \tag{2}$$

where i denotes the current phase of the arrival process, $i=1,\ldots,m$, and $\mathrm{E}[S]$ is the mean service fee.

Proof. Value function, as defined in (1), measures the expected difference in the cumulative costs from the given initial phase i to the mean cost rate. For an arbitrary phase i, the so-called Howard's equation is

$$v_i=\frac{m-i+1}{\lambda}(0-r_s)+\mathrm{E}[S]+v_1.$$

[2]This is slightly different than, e.g., in [20, 22], where the holding costs are incurred also during the service time. In our case, an equivalent cost can be defined using service fees.

The first term corresponds to the time interval before the next arrival. During this time no service fees are collected and the difference to the mean cost rate is $0-r_s$. The factor $(m-i+1)/\lambda$ corresponds to the mean time duration.

The second term is the mean immediate cost due to the following arrival, after which the arrival process enters to phase 1. The mean difference in costs between phase 1 and the mean cost rate is v_1 by definition, i.e., v_1 takes care of the future costs from that point onwards (recall the Markov property). Consequently,

$$v_i - v_1 = \frac{i-1}{m}\mathrm{E}[S]. \tag{3}$$

As each phase is equally likely, we have $\sum_i v_i = 0$. Taking a sum of (3) over i gives v_1, which in turn yields (2). □

Consider next the whole M/G/m-RR system and let $S^{(k)}$ denote the service fee in Queue k. Due to the decomposition, we have the following result:

Corollary 1. *The value function w.r.t. service fees for the M/G/m-RR system in state* $\mathbf{z} = (q_1, \ldots, q_m)$ *is*

$$v(\mathbf{z}) = \frac{1}{2m}\sum_{i=1}^{m}(2i-m-1)\mathrm{E}[S^{(q_i)}]. \tag{4}$$

Note that the value function is insensitive to the arrival rate λ and depends only on the phases (i.e., the RR sequence) and the mean service fees. In fact, the constant offset in the value functions is irrelevant to us, and we can equally use

$$v(\mathbf{z}) = \frac{1}{m}\sum_{i=1}^{m} i\,\mathrm{E}[S^{(q_i)}], \tag{5}$$

from which it is obvious that the Round-Robin sequence assigning the jobs (initially) in the increasing order of the mean service fee so that $\mathrm{E}[S^{(q_i)}] \le \mathrm{E}[S^{(q_{i+1})}]$ incurs the least costs, as expected.

2.4 Virtual Waiting Time

We are often interested in the mean waiting and sojourn time in a system. To this end, we define the *virtual waiting time* in the system as the backlog of the queue receiving the next customer. More precisely, we define the holding cost rate of the system to be equal to the backlog of the queue in phase m. Considering an individual Erl(m,λ)/G/1 queue, it thus incurs costs at the rate equal to the backlog only during phase m, at the end of which a new job arrives. This is illustrated in Fig. 2. Due to the PASTA property, this corresponds to the waiting time the jobs arriving to the M/G/m-RR system see. Let $\tilde{r}$ denote the mean cost rate in a single queue and r in the whole system, $r = \mathrm{E}[W]$. With identical servers, we have the elementary relationship

$$\tilde{r} = r/m.$$

In general, the service times may be heterogeneous and

$$\left\{\begin{array}{l} r = \mathrm{E}[W] = (1/m)\sum_k \mathrm{E}[W^{(k)}] \\ r = \sum_k \tilde{r}^{(k)} \end{array}\right. \Rightarrow \mathrm{E}[W^{(k)}] = m\,\tilde{r}^{(k)}.$$

2.4.1 Single Erl/G/1 Queue

Consider next a single Erl(m,λ)/G/1 queue and let $F(x)$ denote the cdf of the service time X. Let $I_t(i)$ and $U_t(i,u)$ denote the phase of the arrival process and the backlog in the queue at time t, where (i,u) denote the initial phase,

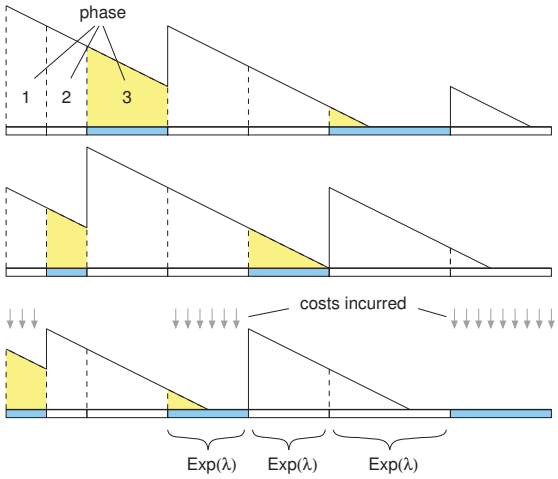

Figure 2: Sample path with m = 3 queues.

$i = 1, \ldots, m$, and the initial backlog, $U_0(i,u) = u$. In this RR-specific cost structure, the queue incurs costs at rate

$$C_t(i,u) \triangleq 1(I_t(i) = m)\cdot U_t(i,u).$$

Let $v_i(u)$ denote the value function, where i is the initial phase and u the initial backlog. Formally,

$$v_i(u) \triangleq \lim_{t\to\infty}\mathrm{E}[V_i(u,t) - \tilde{r}\,t],$$

where $V_i(u,t)$ denotes the costs a queue initially in state (i,u) incurs during time t,

$$V_i(u,t) \triangleq \int_0^t C_s(i,u)\,ds.$$

Proposition 2. *The value function of an Erl(m,λ)/G/1 queue with respect to the virtual waiting time satisfies the following system of integro-differential equations,*

$$\begin{aligned} v_i'(u) &= -\tilde{r} + \lambda(v_{i+1}(u) - v_i(u)), \quad i = 1, .., m-1 \\ v_m'(u) &= u - \tilde{r} + \lambda\int_0^\infty (v_1(u+s) - v_m(u))\,dF(s). \end{aligned} \tag{6}$$

Proof. For $u > 0$, small $\delta > 0$, and for phases $i = 1, \ldots, m-1$

$$v_i(u) = (0-\tilde{r})\delta + (1-\lambda\delta)\,v_i(u-\delta) + \lambda\delta\,v_{i+1}(u-\delta).$$

The first term corresponds to the difference between the current cost rate (zero for phases $i \neq m$) and the mean cost rate $\tilde{r}$ multiplied by the time-interval δ. With the probability of $(1-\lambda\delta)$, the phase remains the same and $v_i(u-\delta)$ gives the future contribution, and with the probability of $\lambda\delta$, the arrival process moves to the next phase $i+1$, the value of which is given by $v_{i+1}(u-\delta)$. In contrast, at the end of last phase m a new job arrives, and thus for $v_m(u)$ we have

$$v_m(u) = (u-\tilde{r})\delta + (1-\lambda\delta)\,v_m(u-\delta) + \lambda\delta\int_0^\infty v_1(u+s)\,dF(s).$$

As $\delta \to 0$, the above yields (6). □

For the special case of a constant service time Δ, the latter equation in (6) reads

$$v_m'(u) = u - \tilde{r} + \lambda(v_1(u+\Delta) - v_m(u)),$$

and we have a first-order system of differential equations. Considering an empty system with $u = 0$ gives

$$\begin{aligned} \tilde{r} &= \lambda(v_{i+1}(0) - v_i(0)), \quad i = 1, \ldots, m-1, \\ \tilde{r} &= \lambda \int_0^\infty (v_1(s) - v_m(0))\, dF(s). \end{aligned} \tag{7}$$

With a constant service time Δ, the latter equation reads

$$\tilde{r} = \lambda(v_1(\Delta) - v_m(0)).$$

Combining (6) and (7) gives,

$$\begin{aligned} v_i'(0) &= 0, \qquad i = 1, \ldots, m, \\ v_i''(0) &= 0, \qquad i = 1, \ldots, m-1. \end{aligned} \tag{8}$$

Adding the equations (7) together gives

$$m\tilde{r} = r = \lambda \int_0^\infty (v_1(s) - v_1(0))\, dF(s), \tag{9}$$

which for a constant service time Δ reduces to

$$m\tilde{r} = \lambda(v_1(\Delta) - v_1(0)).$$

Similarly, we have for all $i = 1, \ldots, m-1$

$$v_{i+1}(0) - v_1(0) = \frac{i\tilde{r}}{\lambda}. \tag{10}$$

That is, initially the value functions $v_i(u)$ at $u = 0$ differ by a constant amount of $\tilde{r}/\lambda$.

Note that both (6) and (7) are insensitive to a constant term in the $v_i(u)$. The constant offset in the value functions indeed is generally superfluous and we can set, e.g., $v_1(0) = 0$. Unfortunately, (6) and (7) are difficult to solve even numerically. If the mean cost rate $\tilde{r}$ was available in a closed form, (7) would give the initial values $v_i(0) = 0$ also for $i = 2, \ldots, m$. Moreover, even if the initial values were available, $v_m'(u)$ still depends on the $v_1(u + s)$, $s > 0$, and therefore the standard Runge-Kutta method could not be applied to compute the $v_i(u)$ for $u > 0$. However, one could solve the $v_i(u)$ numerically backwards for $u < u^*$ given the $v_i(u^*)$ were available for some $u^* > 0$. We describe an elegant approach to solve the $v_i(u)$ and $\tilde{r}$ in Section 2.4.4.

Finally, with $m = 1$ the Erl/G/1 queue reduces to M/G/1, for which the exact value function is available [21]

$$v(u) - v(0) = \frac{u^2}{2(1-\rho)}. \tag{11}$$

It is easy to see that (11) satisfies (6), and applying to (7) gives, as expected, the Pollazcek-Khinchine formula for the mean waiting time.

2.4.2 M/G/m-RR System

Consider next the whole M/G/m-RR system. In general case, the servers may be heterogeneous and the value functions have to be determined separately for each of them. Let $v_i^{(k)}(u)$ denote the value function of Queue k currently in phase i. Due to the decomposition to m parallel Erl/G/1 queues, we again have:

Corollary 2. *The value function w.r.t. virtual waiting time for M/G/m-RR in state* $\mathbf{z} = ((q_1, u_1), \ldots, (q_m, u_m))$ *is*

$$v(\mathbf{z}) = v_1^{(q_1)}(u_1) + \ldots + v_m^{(q_m)}(u_m). \tag{12}$$

If the service times $X_{j,k}$ are identical for every queue k, then

$$v(\mathbf{z}) = v_1(u_1) + \ldots + v_m(u_m).$$

2.4.3 Asymptotic Behavior

Below we argue that, with relatively broad assumptions, the asymptotic behavior of the value function of the G/G/1-FCFS queue is quadratic. For large u, the backlog decreases with an average rate of $1 - \rho'$, where ρ' denotes the queue specific offered load. The virtual waiting time incurred during the remaining busy period corresponds to a triangle with initial height u and base (= duration) $u/(1-\rho')$. Therefore,

$$v(u) \approx \frac{u^2}{2(1-\rho')} - \frac{u}{1-\rho'} \cdot r + v(0),$$

where the first term corresponds to the costs incurred during the remaining busy period, the second term to the mean cost rate during the same time-interval, and the third term corresponds to what happens after that. For large u, the first quadratic term dominates.

With Erl(m, λ)/G/1, in the context of RR, the costs are accrued only in the final phase m as explained above (see also Fig. 2). Let ρ denote the offered load to the whole system, $\rho = \lambda \mathrm{E}[X]$, so that $\rho' = \rho/m$. The costs accrued during the remaining busy period are roughly $1/m$ of the "full triangle", i.e., for $u \gg 1$ we have,

$$v_i(u) \approx \frac{u^2}{2(m-\rho)}, \quad \text{and} \quad v_i'(u) \approx \frac{u}{m-\rho}. \tag{13}$$

2.4.4 Numerical Solution for $v_i(u)$

The equations (6) that the value functions $v_i(u)$ must satisfy can be written in an integral form that is suitable for numerical computations:

Proposition 3. *For an Erl(m, λ)/G/1 queue, the value function $v_i(u)$ with respect to the virtual waiting time satisfies the following system of integral equations,*

$$\begin{aligned} v_i(u) &= \left(e^{-\lambda u} - \frac{1}{i}\right) v_{i+1}(0) \\ &\quad + \lambda \int_0^u e^{-\lambda(u-s)} v_{i+1}(s)\, ds, \quad i = 1, \ldots, m-1, \\ v_m(u) &= \left(e^{-\lambda u} - \frac{1}{m}\right) \int_0^\infty v_1(s)\, dF(s) + \frac{e^{-\lambda u} + \lambda u - 1}{\lambda^2} \\ &\quad + \lambda \int_0^u e^{-\lambda(u-s)} \int_0^\infty v_1(s+\ell)\, dF(\ell)\, ds. \end{aligned} \tag{14}$$

Proof. For $i = 1, \ldots, m-1$, multiplying both sides of (6) with $e^{\lambda u}$ gives

$$e^{\lambda u} v_i'(u) + \lambda e^{\lambda u} v_i(u) = e^{\lambda u}(-\tilde{r} + \lambda v_{i+1}(u)).$$

The left-hand side is equal to $(d/du)\, e^{\lambda u} v_i(u)$, yielding

$$e^{\lambda u} v_i(u) = \int_0^u e^{\lambda s}(-\tilde{r} + \lambda v_{i+1}(s))\, ds + v_i(0),$$

$$v_i(u) = \frac{\tilde{r}}{\lambda}(e^{-\lambda u} - 1) + e^{-\lambda u} v_i(0) + \lambda \int_0^u e^{-\lambda(u-s)} v_{i+1}(s)\, ds.$$

Similarly, for phase m one obtains

$$\begin{aligned} v_m(u) = \frac{\tilde{r}}{\lambda}(e^{-\lambda u} - 1) + e^{-\lambda u} v_m(0) + \frac{e^{-\lambda u} + \lambda u - 1}{\lambda^2} \\ + \lambda \int_0^u e^{-\lambda(u-s)} \int_0^\infty v_1(s+\ell)\, dF(\ell)\, ds. \end{aligned}$$

As we are generally interested in the differences between the relative values, we can set $v_1(0) = 0$ so that

$$v_i(0) = \frac{(i-1)\tilde{r}}{\lambda}, \qquad \text{for } i = 1, \ldots, m,$$

and

$$\frac{\tilde{r}}{\lambda} = \begin{cases} \dfrac{v_{i+1}(0)}{i}, & \text{for } i = 1, \ldots, m-1, \\ \dfrac{1}{m}\displaystyle\int_0^\infty v_1(s)\, dF(s). \end{cases}$$

Substituting these into the above gives (14). □

COROLLARY 3. *For a constant service time Δ, the latter equation in* (14) *reads*

$$v_m(u) = \left(e^{-\lambda u} - \frac{1}{m}\right) v_1(\Delta) + \frac{e^{-\lambda u} + \lambda u - 1}{\lambda^2} + \lambda \int_0^u e^{-\lambda(u-s)} v_1(s+\Delta)\, ds. \tag{15}$$

Note that (14) expresses $v_i(u)$ as a function $v_{i+1}(u)$ for $i = 1, \ldots, m-1$, and $v_m(u)$ as a function of $v_1(u)$. Given an initial guess for any $v_i(u)$, provided, e.g., by (13), the equations (14) can be iterated until they converge. In practice, the convergence turns out to be fast.

As a convenient side product of being able to determine the value functions efficiently, also the *mean waiting time*, $r = \mathrm{E}[W]$, is obtained (with $v_1(0) = 0$):

COROLLARY 4. *Solving the value functions using* (14) *gives also the mean waiting time* $\mathrm{E}[W]$ *in the system,*

$$\mathrm{E}[W] = \lambda\, m\, v_2(0), \tag{16}$$

In order to compute $\mathrm{E}[W]$, we do not need to find the waiting time distribution first (which itself is non-trivial, [11]).

For $m = 1$, the insensitive solution (11) can be shown to satisfy (14). That is, for the M/G/1 queue we have,

$$v(u) = \left(e^{-\lambda u} - 1\right) \int_0^\infty v(s)\, dF(s) + \frac{e^{-\lambda u} + \lambda u - 1}{\lambda^2} + \lambda \int_0^u e^{-\lambda(u-s)} \int_0^\infty v(s+\ell)\, dF(\ell)\, ds,$$

which "trial"

$$v(u) = \frac{u^2}{2(1-\rho)},$$

satisfies independently of the service time distribution.

2.4.5 Generalized Round-Robin (GRR)

RR is a special case of the *Generalized Round-Robin* policies defined by predefined typically periodic sequences [13, 14, 3]. RR is optimal under certain assumptions when the servers are identical. However, if some servers have different service rates, then the even split of tasks that RR carries out may no longer make sense. In [13], Hajek proves the intuitive result that among a very large class of arrival process, the one with constant inter-arrival times is optimal for a single server queue with exponentially distributed service times. Then, in [14], he shows that the so-called most-regular-sequence is optimal for two, not necessarily identical, servers when jobs again obey exponential distribution.

Suppose that the (external) sequence a_i defining the task assignments is periodic with m denoting the length of the period. Without lack of generality, we can consider Queue 1. With respect to service fees, both the mean rate and value function are straightforward to deduce. We omit these for brevity. For the virtual waiting time, let $v_i(u)$ again denote its value function with respect to the backlog (in phases with actual arrivals, $a_i = 1$). For notational convenience, we define $v_{i+m}(u) \triangleq v_i(u)$. Similarly as earlier, we have a system of differential equations,

$$v_i'(u) = \begin{cases} -\tilde{r} + \lambda(v_{i+1}(u) - v_i(u)), & \text{if } a_i \neq 1, \\ u - \tilde{r} + \lambda \displaystyle\int (v_{i+1}(u+t) - v_i(u))\, dF(t), & \text{if } a_i = 1, \end{cases}$$

where the initial values are coupled,

$$v_i'(0) = \begin{cases} -\dfrac{\tilde{r}}{\lambda} + v_{i+1}(0), & \text{if } a_i \neq 1, \\ -\dfrac{\tilde{r}}{\lambda} + \displaystyle\int v_{i+1}(t)\, dF(t), & \text{if } a_i = 1. \end{cases}$$

That is, the generalized (periodic) RR can be analyzed essentially the same way as RR. Also probabilistic variants, where subsequences are chosen with certain probabilities (for load-balancing reasons), are amenable to the same approach.

2.5 Waiting Time and General Holding Costs

Consider next M/G/m-RR with identical servers. The backlog based holding cost rate $c(\mathbf{z}) = u_m$ can be seen as a penalty for a long queue length. However, often one is interested in the actual *waiting time*, possibly weighted with arbitrary job-specific holding costs. Let $U(t) = U_m(t)$ denote the virtual waiting time at time t. With FCFS, this is the waiting time an arriving customer sees, $W \sim U$. The mean cost rate w.r.t. waiting time is

$$r_W = \lambda \cdot r = \lambda \cdot \mathrm{E}[W],$$

i.e., the rate at which the system incurs waiting time.

PROPOSITION 4. *The value function for an M/G/m-RR system with respect to waiting time is*

$$\tilde{v}(\mathbf{z}) = \lambda\, v(\mathbf{z}) \tag{17}$$

where $v(\mathbf{z})$ is the value function w.r.t. virtual waiting time.

PROOF. Let $W_1, W_2, \ldots$ denote the waiting times related to the future arrivals. One can associate the costs in two equivalent ways for these arrivals until the end of the current busy period (renewal point),

$$\begin{aligned} \tilde{c}_1 &\triangleq \lambda\, \mathrm{E}[\int_0^{B_{\mathbf{z}}} U_m(t)\, dt], \\ \tilde{c}_2 &\triangleq \mathrm{E}[W_1 + \ldots + W_{N_{\mathbf{z}}}], \end{aligned} \tag{18}$$

where $B_{\mathbf{z}}$ denotes the duration of the (remaining) busy period (having a finite mean), and $N_{\mathbf{z}}$ the number of jobs arriving during it. Due to the PASTA property, $\tilde{c}_1 = \tilde{c}_2$. The first equation corresponds to the virtual waiting time based holding cost $c(\mathbf{z}) = u_m$ multiplied by the arrival rate λ, and the latter to the actually incurred waiting time. Then,

$$\begin{aligned} \tilde{v}(\mathbf{z}) - \tilde{v}(0) &= \mathrm{E}[W_1 + \ldots + W_{N_{\mathbf{z}}}] - r_W\, \mathrm{E}[B_{\mathbf{z}}] \\ &= \lambda\, \mathrm{E}[\int_0^{B_{\mathbf{z}}} (U_m(t) - r)\, dt] = \lambda(v(\mathbf{z}) - v(0)), \end{aligned}$$

and thus (17) holds. □

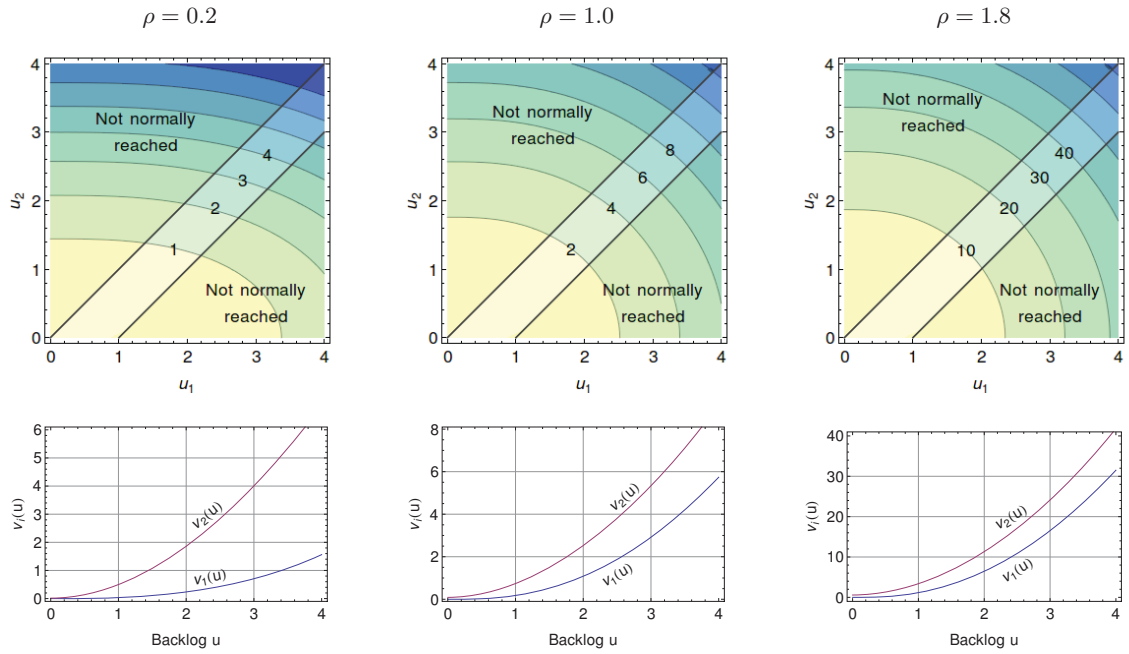

Figure 4: Value function for M/D/2-RR with respect to the virtual waiting time.

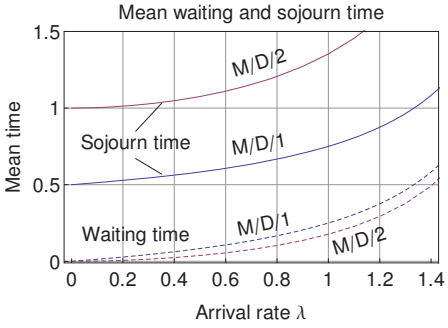

Figure 3: Mean waiting time (lower curves) and sojourn time (upper curves) for M/D/2 and M/D/1 (with service rate c = 2).

Consider next the case with arbitrary *job-specific holding costs*. Let (X_j, H_j) denote the size and the holding cost of Job j that are assumed to be i.i.d., $(X_j, H_j) \sim (X, H)$, while each X_j and H_j may still depend on each other. For example, the slowdown metric, defined as the ratio of the delay to the service time, is obtained with $H_j = 1/X_j$ and $S_j = 1$ [20]. The difference in the cumulative costs is incurred by later arriving jobs during their waiting time. Therefore:

COROLLARY 5. *The value function w.r.t. general holding costs (associated with the waiting time) for M/G/m-RR in state* $\mathbf{z} = ((q_1, u_1), \ldots, (q_m, u_m))$ *is*

$$\tilde{v}(\mathbf{z}) = \lambda \sum_i v^{(q_i)}(u_i)\, \mathrm{E}[H^{(q_i)}].$$

where $\mathrm{E}[H^{(k)}]$ *denotes the mean holding cost rate of a job in Queue k. In case of identical holding cost rates,*

$$\tilde{v}(\mathbf{z}) = \lambda\, v(\mathbf{z})\, \mathrm{E}[H], \qquad (19)$$

2.6 Examples with Value Function

Next we give some numerical examples with the virtual waiting time. The service fee is included to the cost structure later in Section 3.

2.6.1 Comparison of Equivalent M/D/1 and M/D/2

With identical servers and constant service times, both the Round-Robin and LWL are equivalent to M/D/m with a shared queue. According to (16), the mean waiting time can be obtained from the value functions, and we can compare the performance of an M/D/2 queue with an equivalent M/D/1 queue that is twice as fast ($c = 2$). Fig. 3 illustrates the mean waiting time and sojourn time for both systems. The mean waiting time with two servers is smaller than with one fast server. However, the mean sojourn time is obviously always shorter with one fast server.

2.6.2 M/D/2 w.r.t. Virtual Waiting Time

Consider next the virtual waiting time in M/D/2-RR. Fig. 4 depicts the resulting value function $v(\mathbf{z}) = v_1(u_1) + v_2(u_2)$ for the whole system (upper row) and its components (lower row) for $\rho = 0.2$, 1.0, 1.8. For the upper row, we have included also states with $u_2 > u_1$ and $u_1 > u_2 + 1$, even though the normal state space[3] of an initially empty M/D/2-RR queue is the narrow strip constrained by $0 \le u_2 \le u_1 \le u_2 + 1$. That is, we allow an arbitrary initial state. The value function becomes more symmetric as ρ increases. From the lower row, we observe that initially, at $u = 0$, the slope of each $v_i(u)$ is zero in accordance with (8).

2.6.3 Other Distributions

Let $D(x_1, x_2)$ denote the discrete probability distribution with two possible outcomes x_1 and x_2. We further as-

[3]Later, in Section 3, the so-called FPI policy may deviate from RR and any state is in principle possible.

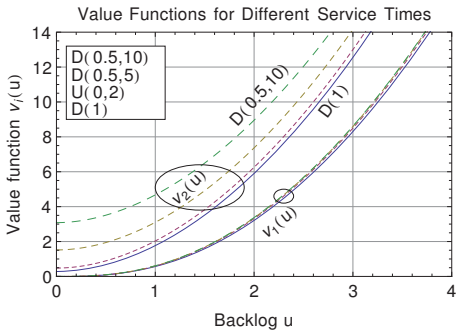

Figure 5: Value function $\mathbf{v_i(u)}$ for $\mathbf{m = 2}$ phases $\mathbf{(i = 1, 2)}$ when service times obey $\mathbf{D(0.5, 10)}$, $\mathbf{D(0.5, 5)}$, $\mathbf{U(0, 2)}$ and $\mathbf{D(1)}$ with unit mean and $\boldsymbol{\lambda = 1.6}$. For both $\mathbf{v_1(u)}$ and $\mathbf{v_2(u)}$, service time distribution $\mathbf{D(0.5, 10)}$ corresponds to the highest curve and $\mathbf{D(1)}$ to the lowest (in the order of variability).

sume that $x_1 < 1 < x_2$ so that with a suitable choice of point probabilities the mean is equal to one. Similarly, $D(x)$ denotes the deterministic distribution with one outcome x and $U(x_1, x_2)$ the uniform distribution on interval (x_1, x_2). Fig. 5 illustrates the resulting value functions with $D(0.5, 10)$, $D(0.5, 5)$, $U(0, 2)$ and $D(1)$. In each case, the mean service time is 1 and arrival rate $\lambda = 1.6$. We can observe that the higher the variance in the service times is, the higher the initial difference $v_2(0) - v_1(0)$ is. The $v_1(u)$ behave rather similarly (note the initial choice $v_1(0) = 0$). Consequently, we have the following result:

COROLLARY 6. *The value function for an Erl(m, λ)/G/1-FCFS queue w.r.t. waiting time (or delay) is not insensitive to the job size distribution (for $m > 1$).*

PROOF. See Fig. 5. □

We note that this is in contrast to the M/G/1-FCFS queue, which value function is insensitive to the job size distribution [21]. This implies that the value function of the corresponding Round-Robin system, due to the decomposition, is also sensitive to the job size distribution, which again is not the case if the routing is by any state-independent (static) policy such as the random Bernoulli-split.

3. TASK ASSIGNMENT PROBLEM

In this section, we consider the system of m parallel servers with job- and server-specific service fees and holding costs. As reference routing policies, we consider the following:

RR: Round-Robin assigns the jobs using a predefined sequence $s_1, s_2, \ldots, s_m, s_1, s_2, \ldots$ where $s_i \neq s_j$ for $i \neq j$.

RND: Bernoulli-split assigns jobs randomly and independently using probabilities $p_1, \ldots, p_m$.

JSQ: Join-the-shortest-queue assigns a new job to queue with the least number of jobs.

LWL: Least-work-left assigns a new job to the queue with the shortest backlog.

Myopic chooses the queue that minimizes the costs assuming no other jobs arrive in future.

Ties are broken in favor of the queue with a smaller index.

3.1 Policy Iteration

The policy iteration is a standard technique of the MDP framework to improve a given policy based on a value function [4, 19, 28]. In layman's terms, at every state, it chooses the action a for which the sum of the immediate cost and the change in the future cumulative costs is the smallest. In our case, the immediate cost of Job j is the sum of the service fee s_j^a and the waiting time w_j^a times the holding cost h_j^a,

$$s_j^a + w_j^a \cdot h_j^a,$$

where we have made it explicit that also the service fee and holding cost may depend on the chosen queue, not just on the job. Note also that with FCFS, the waiting time w_j^a gets fixed at the task assignment by action a (the later arriving jobs do not affect the sojourn time of the present jobs). If the basic policy is RR, then an action may define two things:

1. The queue for the new job.
2. The future RR sequence (the phases for queues).

Similarly, utilizing (19), the expected increase in the holding costs incurred in the future is

$$\tilde{v}(\mathbf{z} \oplus a) - \tilde{v}(\mathbf{z}) = \lambda \left(v(\mathbf{z} \oplus a) - v(\mathbf{z}) \right) \mathrm{E}[H],$$

where $\mathbf{z}$ is the current state of the system, $\mathbf{z} \oplus a$ the state after action a, and $v(\mathbf{z})$ is the value function with respect to the virtual waiting time. The improved policy α' is then

$$\alpha'(j, \mathbf{z}) = \underset{a \in \mathcal{A}}{\operatorname{argmin}} \left(s_j^a + w_j^a \, h_j^a + (\tilde{v}(\mathbf{z} \oplus a) - \tilde{v}(\mathbf{z})) \right), \quad (20)$$

where $\mathcal{A}$ denotes the set of possible actions (the m servers). We will utilize (20) in the next section.

3.2 Numerical Examples

Let us consider $m = 2$ equally fast servers. First we assume a constant service time and then we experiment with some other elementary service time distributions.

3.2.1 Two Policy Iteration Steps

As RR/LWL is optimal with respect to the mean delay for M/D/m, let us consider a system with arbitrary job-specific holding cost rates. Let h denote the holding cost rate of the new job. If $h > \mathrm{E}[H]$, the intuition suggests that the new job should be assigned to the shorter queue according to RR/LWL. However, if $h < \mathrm{E}[H]$, it may be beneficial to assign the new job to the longer queue, thus keeping the other queue shorter for later arriving, possibly more important, jobs. The potential pitfall is that no such "important" job arrives and one of the servers is unnecessarily idle (which never happens with RR/LWL). For the policy improvement, it is more convenient to consider the waiting time based cost structure and (20).

Let us start with the state-independent *Bernoulli-split policy*. With two identical servers, this policy assigns the new job to Server 1 with probability of 0.5, and otherwise to Server 2. The value function of the whole system is [21]

$$\tilde{v}_{\mathrm{RND}}(u_1, u_2) = \frac{\lambda' \, \mathrm{E}[H]}{2(1 - \rho')} (u_1^2 + u_2^2),$$

where λ' is the queue-specific arrival rate, $\lambda' = \lambda/2$, and ρ' the queue-specific load, $\rho' = \lambda/2 \cdot \Delta$. The mean difference

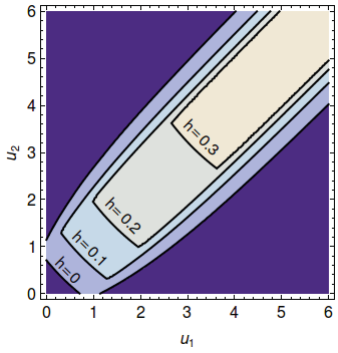

Figure 6: States in the diagonal where SPI suggests assigning the new Job j with holding cost h_j to the longer queue (assuming RR afterwards).

in the expected costs between assigning the new job with holding cost h to Server 1 and Server 2 is

$$\Delta c = h(u_1 - u_2) + \tilde{v}_{\mathrm{RND}}(u_1{+}\Delta, u_2) - \tilde{v}_{\mathrm{RND}}(u_1, u_2{+}\Delta)$$
$$= \left(h + \frac{\lambda\Delta\,\mathrm{E}[H]}{2-\lambda\Delta}\right)\cdot(u_1 - u_2),$$

i.e., Δc is negative when $u_1 < u_2$, and vice versa. This means that the *first policy iteration* (FPI) step (20), choosing the action with the lowest expected overall cost if the consecutive decisions are according the basic policy (Bernoulli-split), yields LWL. Moreover, we recall that LWL was equivalent to RR with a constant service time Δ.

As the first policy iteration step yielded RR (that was equivalent to LWL in this case), the value function of which we can now compute, we can proceed further and carry out the *second policy iteration step* (SPI),

$$\mathrm{RND} \overset{PI}{\Longrightarrow} \mathrm{LWL} \overset{PI}{\Longrightarrow} \mathrm{SPI}.$$

Fig. 6 illustrates the regions in the state space where SPI chooses the alternative action, i.e., assigns the new job to the longer queue. The arrival rate λ was chosen to be 0.5. We note that SPI changes the FPI policy (RR) only near the diagonal where both queues have roughly equal amount of unfinished work. Higher the holding cost h of the new job is, higher the backlogs must be before the change, on average, pays off. Jobs with $h \geq \mathrm{E}[H]$ are categorically assigned to the shorter queue.

3.2.2 Two Servers with Varying Service Fees

Let us next consider a server system with primary and secondary server with fixed size jobs illustrated in Fig. 7. The servers are equally fast, i.e., the service time of a job is the same in both queues. The cost structure is

$$H = 1, \quad \text{and} \quad S_1 = 1, S_2 = 4,$$

i.e., the secondary server has a four times higher service fee. Alternatively, the secondary server costs, say 3 dollars per job, the primary server is free, and the jobs incur costs at the rate of 1 dollar per unit time during their sojourn time. Note that without service fees, LWL/RR are optimal minimizing the mean waiting and sojourn times.

With two servers, both LWL and Myopic are clearly the so-called switch-over policies that can be defined by a curve

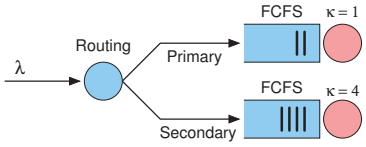

Figure 7: Primary and secondary server with unit holding cost $H = 1$ and service fees $S_1 = 1$ and $S_2 = 4$ processing jobs with constant service time.

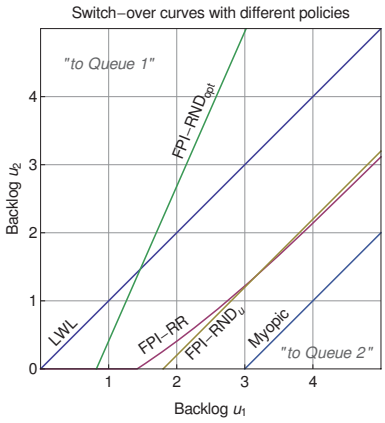

Figure 8: Dynamic switch-over policies illustrated for a system with a constant service time, and two equally fast servers with service fees $S_1 = 1$ and $S_2 = 4$. Each policy assigns a new job to Queue 1 when the current state is above the corresponding curve, and otherwise to Queue 2.

$f(u_1)$ such that a new job is routed to Queue 2 if $u_2 < f(u_1)$, and otherwise to Queue 1. It turns out that also the FPI policies based on RND and RR yield a switch-over policy. Fig. 8 illustrates the switch-over curves for $\lambda = 0.8$. RND_{opt} uses the optimal splitting probabilities, and RND_u splits the jobs equally, $p_1 = p_2 = 0.5$. We note that the curves with FPI-RND policies are straight lines, while with FPI-RR the switch-over curve is a slowly turning curve. Simulating the system gives the mean costs per job:

LWL:	2.20	FPI-RND$_{opt}$:	1.92
Myopic:	2.12	FPI-RND$_u$:	1.91
		FPI-RR:	1.88

We observe that FPI-RR achieves the lowest mean cost rate, closely followed by the other two FPI policies. Numerically experimenting one can see that when λ approaches 2 (the stability bound for this system), all FPI policies converge to LWL. Similarly, when $\lambda \to 0$, Myopic is optimal and all three FPI policies reduce to it.

3.2.3 Varying Holding Cost and Service Times

Let us consider an elementary system comprising two identical servers. Jobs arrive according to the Poisson process with rate λ. Job sizes and holding costs are i.i.d. random variables. The job size is 1 with probability of 0.9, and otherwise 91, so that $\mathrm{E}[X] = 10$ and the variance much higher.[4]

[4]We experimented also with uniform distribution, but the results were boring as the differences between RR, LWL, and FPI-RR were small. In this respect, uniform distribu-

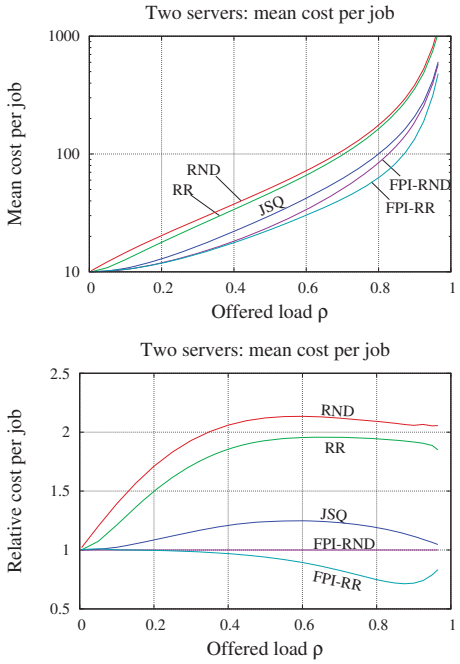

Figure 9: Mean holding costs in the elementary example setting with two identical servers. FPI-RR achieves clearly the lowest cost rate.

The holding costs are assumed to obey exponential distribution with unit mean, $H \sim \text{Exp}(1)$.

Simulation results are depicted in Fig. 9 for Bernoulli-split (RND), JSQ, RR, FPI-RND (i.e., LWL) and FPI-RR. The offered load ρ is on the x-axis, and the y-axis corresponds to the mean costs incurred per job. The first figure shows the absolute performance in logarithmic scale, and the second the performance relative to LWL. Note that LWL minimizes the holding cost of the current job, i.e., it makes the same greedy decision as selfish users do.

When $\rho \approx 0$, LWL and JSQ both, obviously, work well. The task there is merely to avoid situations when two or more jobs are in one server while the other server is idle. As ρ increases beyond about 0.1, the performance of JSQ starts to degrade. At $\rho \approx 0.4$ or higher, also the performance of LWL, due to its greedy behavior degrades. Around $\rho = 0.9$, FPI-RR is about 25% better than LWL. As $\rho \to 1$, JSQ, LWL and FPI-RR appear to converge to the same point. Indeed, at that limit the mean queue lengths explode and it is sufficient to (dynamically) balance the load.

4. MORE ADVANCED APPLICATIONS

The ability to determine a value function for Erl/G/1 queues enables the analysis of far more complex server systems than the plain Round-Robin system. One example is illustrated in Fig. 10(a), where a multi-layer RR is constructed: Queue 1 behaves according to Erl$(2, \lambda)$/G/1 and Queues 2 and 3 according to Erl$(4, \lambda)$/G/1. This type of arrangements can be advantageous when the service rates

tion is sufficiently close to a constant value. The chosen job size distribution has sufficiently high variance so that it is important to take into account jobs arriving in the future.

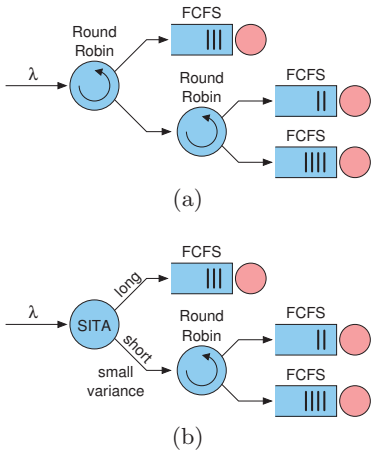

Figure 10: (a) Multi-layer round-robin system feeds tasks to each queue with Erlang-distributed inter-arrival times. (b) In the hybrid system, Queue 1 behaves according to M/G/1 and Queues 2 and 3 according to Erl$(2,\lambda')$/G/1 with reduced variance in the service times.

and/or operating costs are asymmetric.

The main strength in RR comes from the fact that it reduces the variability in the inter-arrival times (see Section 2.4.5). On the other hand, a high variability in the job sizes can be equally harmful (due to the second moment in the Pollazcek-Khinchine formula for the mean waiting time). The so-called *Size-Interval-Task-Assignment* (SITA) policy [7, 16, 10, 17] seeks to reduces the variability in the job sizes by assigning jobs with a similar size to the same queue. To this end, the support of the job sizes is divided into m non-overlapping intervals $[\xi_i, \xi_{i+1})$, $i = 1, \ldots, m$, and a job with size x is assigned to Server i iff $x \in [\xi_i, \xi_{i+1})$.

Fig. 10(b) illustrates a server system which combines the worthwhile features of SITA (variance reduction in service times) and RR (which was the optimal policy w.r.t. delay for tasks with a constant service time). The arrival process to Queue 1 is a Poisson process as SITA is a state-independent policy. Queues 2 and 3 behave according to Erl$(2, \lambda')$/G/1, where the service time of tasks can have a significantly smaller variance thanks to SITA. Moreover, dedicated jobs arriving according to some other Poisson process can be directed to any point already receiving a Poisson process, such as the Queue 1 and the second level RR dispatcher in Fig. 10(b). Thus, the analysis of systems that hierarchically combine dispatchers remains tractable, their value function can be determined, and the policy improvement step can be carried out.

5. CONCLUSIONS

We have analyzed the Round-Robin (RR) routing to a system of parallel queues. RR is a commonly used robust technique to balance the load by assigning tasks to different servers sequentially. It decreases the burstiness in the arrival process to each queue, which is important especially when the queues process the jobs in FCFS order.

The availability of the value function for RR-systems, via the corresponding value function of Erl/G/1 queues, provides new insight to this mechanism itself (and to G/G/1

queues). The value functions that we considered characterize the system state with respect to the service fees and virtual waiting time, and also enable the policy iteration step with respect to a very versatile cost structure defined by job- and server-specific service fees and holding costs, yielding robust cost- and state-aware routing policies. Moreover, as an useful side-product, we obtain the mean waiting time in the Round-Robin system, which itself is a non-trivial result even for an M/D/m-RR queue.

Acknowledgments

This work was supported by the Academy of Finland in the Top-Energy project (grant no. 268992) and the PDP project (grant no. 260014).

6. REFERENCES

[1] O. Akgun, R. Righter, and R. Wolff. Multiple server system with flexible arrivals. *Advances in Applied Probability*, 43:985–1004, 2011.

[2] P. S. Ansell, K. D. Glazebrook, and C. Kirkbride. Generalised 'join the shortest queue' policies for the dynamic routing of jobs to multi-class queues. *The J. of the Oper. Res. Society*, 54(4):379–389, Apr. 2003.

[3] Y. Arian and Y. Levy. Algorithms for generalized round robin routing. *Oper. Res. Lett.*, 12(5):313–319, 1992.

[4] R. Bellman. *Dynamic programming*. Princeton University Press, 1957.

[5] S. Bhulai. On the value function of the M/Cox(r)/1 queue. *Journal of Applied Probability*, 43(2):363–376, June 2006.

[6] C. D. Crommelin. Delay probability formulas when the holding times are constant. *Post Office Electrical Engineers Journal*, 25:41–50, 1932.

[7] M. E. Crovella, M. Harchol-Balter, and C. D. Murta. Task assignment in a distributed system: Improving performance by unbalancing load. In *Proceedings of SIGMETRICS '98*, pages 268–269, Madison, Wisconsin, USA, June 1998.

[8] D. Down and R. Wu. Multi-layered round robin routing for parallel servers. *Queueing Systems*, 53(4):177–188, 2006.

[9] A. Ephremides, P. Varaiya, and J. Walrand. A simple dynamic routing problem. *IEEE Transactions on Automatic Control*, 25(4):690–693, Aug. 1980.

[10] H. Feng, V. Misra, and D. Rubenstein. Optimal state-free, size-aware dispatching for heterogeneous M/G/-type systems. *Performance Evaluation*, 62(1-4):475–492, 2005.

[11] G. J. Franx. A simple solution for the M/D/c waiting time distribution. *Oper. Res. Lett.*, 29(5):221–229, Dec. 2001.

[12] G. J. Franx. The transient M/D/c queueing system, 2002.

[13] B. Hajek. The proof of a folk theorem on queuing delay with applications to routing in networks. *J. ACM*, 30(4):834–851, Oct. 1983.

[14] B. Hajek. Extremal splittings of point processes. *Mathematics of Oper. Res.*, 10(4):543–556, Nov. 1985.

[15] M. Harchol-Balter. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge University Press, 2013.

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

[17] M. Harchol-Balter, A. Scheller-Wolf, and A. R. Young. Surprising results on task assignment in server farms with high-variability workloads. *ACM SIGMETRICS Performance Evaluation Review*, 37:287–298, 2009.

[18] A. Hordijk and G. Koole. On the optimality of the generalised shortest queue policy. *Prob. Eng. Inf. Sci.*, 4:477–487, 1990.

[19] R. A. Howard. *Dynamic Probabilistic Systems, Volume II: Semi-Markov and Decision Processes*. Wiley, 1971.

[20] E. Hyytiä, S. Aalto, and A. Penttinen. Minimizing slowdown in heterogeneous size-aware dispatching systems. *ACM SIGMETRICS Performance Evaluation Review*, 40:29–40, June 2012.

[21] E. Hyytiä, A. Penttinen, and S. Aalto. Size- and state-aware dispatching problem with queue-specific job sizes. *European J. of Oper. Res.*, 217(2), 2012.

[22] E. Hyytiä, S. Aalto, A. Penttinen and J. Virtamo. On the value function of the M/G/1 FCFS and LCFS queues. *J. of Applied Probability*, 49(4), 2012.

[23] A. J. E. M. Janssen and J. S. H. Van Leeuwaarden. Back to the roots of the M/D/s queue and the works of Erlang, Crommelin and Pollaczek. *Statistica Neerlandica*, 62(3):299–313, 2008.

[24] G. Koole, P. D. Sparaggis, and D. Towsley. Minimizing response times and queue lengths in systems of parallel queues. *Journal of Applied Probability*, 36(4):1185–1193, Dec. 1999.

[25] K. R. Krishnan. Joining the right queue: a state-dependent decision rule. *IEEE Transactions on Automatic Control*, 35(1):104–108, Jan. 1990.

[26] Z. Liu and R. Righter. Optimal load balancing on distributed homogeneous unreliable processors. *Operations Research*, 46(4):563–573, 1998.

[27] Z. Liu and D. Towsley. Optimality of the round-robin routing policy. *Journal of Applied Probability*, 31(2):466–475, June 1994.

[28] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 2005.

[29] P. D. Sparaggis and D. Towsley. Optimal routing and scheduling of customers with deadlines. *Prob. Eng. Inf. Sci.*, 8(1):33–49, 1994.

[30] H. Tijms. New and old results for the M/D/c queue. *AEÜ - International Journal of Electronics and Communications*, 60(2):125–130, 2006.

[31] D. Towsley, P. Sparaggis, and C. Cassandras. Stochastic ordering properties and optimal routing control for a class of finite capacity queueing systems. In *Proc. of the 29th IEEE Conference on Decision and Control*, pages 658–663, Dec. 1990.

[32] R. R. Weber. On the optimal assignment of customers to parallel servers. *Journal of Applied Probability*, 15(2):406–413, June 1978.

[33] W. Winston. Optimality of the shortest line discipline. *Journal of Applied Probability*, 14:181–189, 1977.