Size- and State-Aware Dispatching Problem with Queue-Specific Job Sizes

Esa Hyytiä, Aleksi Penttinen and Samuli Aalto

Aalto University School of Electrical Engineering, Finland
email: firstname.lastname@tkk.fi

Abstract

We consider the dispatching problem in a size- and state-aware multi-queue system with Poisson arrivals and queue-specific job sizes. By size- and state-awareness, we mean that the dispatcher knows the size of an arriving job and the remaining service times of the jobs in each queue. By queue-specific job sizes, we mean that the time to process a job may depend on the chosen server. We focus on minimizing the mean sojourn time (i.e., response time) by an MDP approach. First we derive the so-called size-aware relative values of states with respect to the sojourn time in an M/G/1 queue operating under FIFO, LIFO, SPT or SRPT disciplines. For FIFO and LIFO, the size-aware relative values turn out to be insensitive to the form of the job size distribution. The relative values are then exploited in developing efficient dispatching rules in the spirit of the first policy iteration.

Keywords: Routing, Parellel queues, Mean response time, M/G/1, MDP, Relative values

1. Introduction

A dispatching system consists of a dispatcher and a set of servers. The task of the dispatcher is to route the arriving jobs to one of the available servers. The dispatching problem has been studied extensively since the early work by Winston (1977). Applications arise in many contexts such as manufacturing units, toll booths in motorways (Conolly (1984)), web server farms (Gupta et al. (2007)), supercomputing (Schroeder and Harchol-Balter (2004)) and other distributed server systems (Crovella et al. (1998); Harchol-Balter et al. (1999)), and CPU scheduling. The key objective is to minimize the mean sojourn time which is sometimes also referred to as the average response time reflecting, e.g., the CPU scheduling application. Also other performance metrics such as queue length, throughput and fairness have been considered (Wierman (2011)).

The dispatching problem is a joint-optimization problem of two interacting policies: (i) a dispatching policy assigning a queue for each job immediately upon arrival, and (ii) an internal scheduling policy of the queues, i.e., queueing discipline or service order. If the dispatching policy is fixed and the job sizes are known, then the optimal scheduling discipline is the (preemptive) *Shortest Remaining Processing Time* (SRPT), proposed by Schrage and Miller (1966) (see also Schrage (1968)). Among non-preemptive policies, the *Shortest Processing Time* (SPT) optimizes the mean response time. However, if the job sizes are not available, one may still be aware of the job size distribution. For example, with the exponentially distributed but unknown job sizes, the service order is irrelevant, and with constant job sizes *First In First Out* (FIFO) is clearly the optimal discipline. In contrast, Righter et al. (1990) show that FIFO is the worst possible when the job sizes have a non-increasing hazard function. In that case, e.g., a preemptive *Last In First Out* (LIFO) improves the situation.

In this paper, we assume that job sizes (service requirements) and the internal queueing discipline are known, and the task is to find the optimal dynamic dispatching strategy. We now discuss some results on dispatching policies that have been presented in the literature. Assuming exponentially distributed interarrival times and job sizes, Winston (1977) shows that the join the shortest queue (JSQ) with FIFO minimizes the mean waiting time when the number of jobs in each queue is known. Also Ephremides et al. (1980) argue for the optimality of JSQ/FIFO when the number of jobs in each queue is available, while the

round robin (RR), followed by FIFO, is shown to be the optimal policy when one only knows that initially the queues were in the same state. Liu and Towsley (1994) proved that RR/FIFO is optimal with the absence of queue length information if the job sizes have a non-decreasing hazard function. The RR results were later generalized by Liu and Righter (1998). Recently, Down and Wu (2006) also analyze JSQ and RR under different job size distributions (see also Wu and Down (2009)). Whitt (1986) provides several counter examples, where JSQ/FIFO policy fails to minimize the mean waiting time. Closely related, Hajek (1984) considers a two server system, a special case of which is a dispatching problem with exponentially distributed job sizes. In the context of the processor sharing (PS), Bonomi (1990) compares FIFO and PS disciplines and proposes dispatching policies that outperform JSQ. Analysis of JSQ/PS is difficult, and, e.g., Gupta et al. (2007) resort to an approximative approach to study the system with a general job size distribution.

Dispatching problem has also been tackled within the Markov decision process (MDP) framework by means of the policy iteration, for which the so-called relative values of state are a prerequisite (as explained in more detail in Section 2.2). In particular, Krishnan (1987) gives the relative values for an M/M/m queue, Sassen et al. (1997) provides an approximation for an age based M/G/1-FIFO queue, Bhulai (2006) for an M/Cox(r)/1-FIFO queue, while Hyytiä et al. (2011a) and Hyytiä et al. (2011b) consider size-aware M/D/1-PS and M/M/1-PS queues, respectively. Additionally, Krishnan (1990) applies the same approach to a routing problem so as to minimize the blocking probability.

Most of the work above assumes that the job sizes are not available, e.g., JSQ requires the knowledge about the number of jobs, and RR simply needs to know which queue the previous job was assigned to. In contrast, Crovella et al. (1998); Harchol-Balter et al. (1999) assume the FIFO discipline and that the dispatching policy is aware of the size of a given job, but not of the state of the queues, and propose policies based on job size intervals (e.g., short jobs to one queue, and the rest to another). Feng et al. (2005) showed that such a policy is the optimal size-aware state-independent policy for homogenenous servers.

We consider the state-dependent dispatching problem where the router is aware of the job sizes and also of the queue states. Moreover, the size of each job *may be different for each queue*, e.g., in a deterministic fashion similarly as in Sharifnia (1997). Consequently, the first objective is to ensure the stability of the system under given dispatching policy, and only then focus on minimizing the mean sojourn time.

The rest of the paper is organized as follows. First, in Section 2, we describe the model and notation, review several heuristic dispatching policies, and introduce a general Markov decision process (MDP) based approach to derive efficient state-dependent policies by means of the first policy iteration (FPI). To this end, we need the relative values of states, which characterize the value of a given state in terms of expected future costs or profits with respect to the given objective. The key observation we exploit is that if the basic policy preserves the Poisson nature of the arrival process to each queue, then the decomposition of the dispatching system to independent queueing systems means that it is sufficient to consider individual queues, find their relative values, and the relative value of the whole system is the sum of the queue specific relative values. Consequently, in Section 3, we derive the exact formulæ for the size-aware relative values with regard to the mean sojourn time (i.e., the average response time) for an *M/G/1 queue with FIFO, LIFO, SPT and SRPT queueing disciplines*. For FIFO and LIFO, the results are closed form expressions depending only on the current state, arrival rate and the mean job size, i.e., the size-aware relative values are *insensitive* to the form of the job size distribution (cf., Erlang's blocking formula). For SPT and SRPT, the results are integral expressions. These results are general and directly applicable in many other contexts such as fair pricing, admission control, in addition to the dispatching problems with or without queue-specific job sizes. In Section 4, we utilize the previous results and give FPI based dispatching policies for different scenarios. By means of numerical examples, we show how the FPI policies consistently manage to decrease the mean sojourn time with different types of job size distributions. Section 5 concludes the paper.

2. Model and Methodology

We assume that jobs arrive according to a Poisson process with rate λ to a dispatcher/router, which assigns each job to one of the m queues *immediately* upon arrival, as depicted in Fig. 1. We study a *size and state aware* system, where the dispatcher is aware of (i) the size of arriving new task (for each queue),

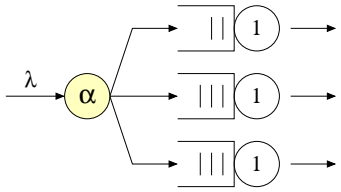

Figure 1: Dispatching problem in a multi-queue system.

(ii) the state of each queue (job sizes and their service order), and (iii) the queueing discipline (scheduling rule) that is applied within each queue. Each job is characterized by an m-tuple, $\mathbf{X} = (X_1; ..; X_m)$, where X_i denotes the amount of work (measured in time) if the given job is assigned to queue i. That is, the dispatcher chooses one of the conditional job sizes X_i, which is then routed to the corresponding queue. We assume that the jobs $\mathbf{X}^{(j)}$ are i.i.d. random vectors, $\mathbf{X}^{(j)} \sim \mathbf{X}$, while the internal structure of $\mathbf{X}^{(j)}$ has no restrictions, i.e., the components $X_i^{(j)}$ can be independent, directly related, or anything in between within the same job j.

For example, in the ordinary case of identical servers, the $X_i^{(j)}$ are equal, $X_i^{(j)} = X^{(j)}$, $\forall\, i$; in an i.i.d. case $X_i^{(j)} \sim X$ and independent for all i, j; and with heterogeneous servers $X_i^{(j)} = X^{(j)}/c_i$, where c_i denotes the service rate of queue i. One can also envision situations where some servers have *special traits* (skills), by which we mean that some servers can process more efficiently certain job types than others. All these problem types fit in our problem definition, and can be tackled with the FPI approach as described in Section 2.2. Thus, without lack of generality, we can assume a constant unit service rate for each queue i and include the possible heterogeneous service rates (and special traits) in the queue-specific job sizes $X_i^{(j)}$. Essentially, the random variables $X_i^{(j)}$ describe how well a job j suits to each server i.

The formulation above is rather generic. In contrast, Sharifnia (1997) considers a deterministic case where for each job class i, π_{ij} defines a processing time on server j, and shows how JSQ/FIFO and a plain multi-server FIFO with a common queue may have stability issues, which an appropriate dispatching and scheduling resolves. The same stability issues are naturally present also here.

2.1. Basic Dispatching Policies

Classic dispatching policies from the literature serve as a reference and as a starting point for this work:

1. *Random* queue (RND), i.e., Bernoulli splitting, chooses a queue in random (Ephremides et al. (1980); Becker et al. (2000)). In our numerical examples, uniform splitting probabilities are used so that all queues have the same arrival rate. However, non-uniform splitting probabilities are a viable option, e.g., when servers are not identical.
2. *Join the Shortest Queue* (JSQ) chooses the queue with the fewest number of jobs. Ties are broken by flipping a fair coin.
3. *Round-Robin* (RR) rotates between the queues in some predefined order.
4. *Least Work Left* (LWL) policy chooses the queue with the smallest backlog. If a job requires different effort from each server, we consider the *a posteriori* backlogs (Sharifnia (1997)).
5. *Myopic* policy chooses the queue which minimizes the total sojourn time of the known jobs in all queues together on condition that no further jobs arrive (Bonomi (1990); Gupta et al. (2007)).
6. *Size Interval Task Assignment* (SITA) policy assigns the jobs to queues based on non-overlapping job size intervals (Crovella et al. (1998); Harchol-Balter et al. (1999)).
7. *Minimum Effort* (ME) policy chooses the queue for which the conditional job size is the smallest. Thus, ME is only applicable when the effort to serve a job varies between the servers.

We say that a dispatching policy is *state-independent* when the decision does not depend on the states of the queues or past decisions. Thus, RND, ME and SITA are state-independent. With such policies, the arrival process to each queue remains Poissonian as illustrated in Fig. 2.

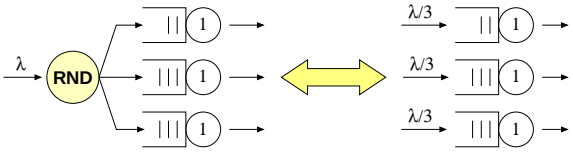

Figure 2: State-independent policies such as RND preserve the Poissonian arrival property and the server queues can be analyzed in isolation as standard M/G/1 queues.

With i.i.d. queue-specific job sizes, $X_i \sim X$, the ME policy can support the highest load, making it both a robust and rational choice. With homogeneous servers the (random) service requirement is the same for each queue i, $X_i = X$, and the SITA policy is the prime candidate. SITA-V (with variable load) policy was proposed by Crovella et al. (1998), who argued that defining size intervals yielding unequal server loads can be advantageous with heavy-tailed job sizes. In this paper, we use such size intervals $\{(\gamma_0, \gamma_1), (\gamma_1, \gamma_2), \ldots, (\gamma_{m-1}, \gamma_m)\}$, $\gamma_0 < \gamma_1 < \ldots < \gamma_m$, that split the load evenly between the m queues, thus sustaining the highest possible load. Harchol-Balter et al. (1999) refer to this policy as the *SITA with Equal Load* (SITA-E).

2.2. Markov Decision Process Approach to Dispatching

Let us next describe the general policy iteration approach. The key idea in the policy iteration is to improve a given policy by deviating "once" from the standard action when the expected costs decrease and assume that the consecutive actions are according to the basic policy. To this end, one needs to know the relative values of states according to the basic policy. Formally, these are defined as the expected difference in the infinite time horizon cumulative costs when a system starts from state i instead of equilibrium,

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

where $V_i(t)$ denotes the costs accrued in time $(0, t)$, and r the mean cost rate. For example, given a choice to start either from state i or j, one should choose state i if $v_i < v_j$, and vice versa. For more details, we refer to standard text books such as Bellman (1957); Howard (1971); Ross (1970).

In the context of the dispatching problem, our approach is as follows. As a starting point, we consider size-aware but state-independent policies, where the dispatching decision is based on the information of the new job, not on the state of the queues or previous decisions. Consequently, the resulting arrival process to each queue remains Poissonian, and the decomposition allows one to study the queues in isolation. In particular, the size-aware relative value of the whole system is obtained as a sum of the queue specific relative values, thereby allowing the FPI on the dispatching decisions giving an improved state-dependent policy.

Single queue in isolation. As mentioned, the state-independent dispatching policies have a very convenient property of preserving the Poisson arrival process. For example, assuming i.i.d. queue-specific job sizes, $X_i \sim \mathrm{Exp}(\mu)$ for queue i, and the ME policy, jobs arriving to each queue are i.i.d., $Y \sim \min\{X_1, .., X_m\}$, i.e., $Y \sim \mathrm{Exp}(m\mu)$, and each queue behaves according to an M/M/1 queue.

Minimizing the average response time means that the costs are accrued by the jobs currently in the system. More concretely, focusing on a single queue, let $\mathbf{z} = (\Delta_1; ..; \Delta_n)$ denote the job sizes present in the given queue. The cost rate is simply the number of jobs in the system, $r_{\mathbf{z}} = n$, so that the mean cost rate is

$$r = \int_{\mathbf{z}} r_{\mathbf{z}} \, \mathrm{P}\{\mathbf{Z} \in d\mathbf{z}\} = \mathrm{E}[N],$$

where $\mathrm{E}[N]$ denotes the mean number of jobs in the system (single queue). However, the costs can be associated in different yet equivalent ways. For example, we can define that costs are accrued in state $\mathbf{z}$ at rate,

$$r_{\mathbf{z}} = \lambda \cdot \mathrm{E}[T \,|\mathbf{z}],$$

where $\mathrm{E}[T\,|\mathbf{z}]$ denotes the mean sojourn time of a job arriving at state $\mathbf{z}$ (the state prior the arrival). Then,

$$r = \lambda \int_{\mathbf{z}} \mathrm{E}[T\,|\mathbf{z}]\,\mathrm{P}\{\mathbf{Z} \in d\mathbf{z}\} = \lambda \cdot \mathrm{E}[T] = \mathrm{E}[N],$$

which is thus equivalent with the previous definition. One can also equivalently define "lump-sum" costs upon arrival that are based on the *expected sojourn times* of the arriving job at the given state.

We define cumulative costs as a sum of sojourn times during $(0, t)$ when the system is initially at state $\mathbf{z}$,

$$V_{\mathbf{z}}(t) \triangleq \text{"accrued sojourn time during } (0,t) \text{ starting from state } \mathbf{z}\text{"}.$$

We note that the cumulative cost is closely related to the total completion time $C(t)$ (Ephremides et al. (1980); Bonomi (1990)), which is defined as the sum of the sojourn times of the jobs arriving during $(0, t)$ in infinite time horizon. From (1), with the mean cost rate of $r = \mathrm{E}[N]$, we obtain the relative value of state $\mathbf{z}$,

$$v_{\mathbf{z}} = \lim_{t\to\infty} \mathrm{E}[V_{\mathbf{z}}(t) - \mathrm{E}[N] \cdot t].$$

An arriving job x triggering a state change $\mathbf{z} \to \mathbf{z}'$ means that the sojourn times accrued in the future increase on average by $v_{\mathbf{z}'} - v_{\mathbf{z}}$, which is thus the cost associated with accepting the new job. Our first task is to find the relative values for an M/G/1 queue under different disciplines.

Policy Iteration Step. Next we assume that jobs in each queue are served according to some queueing discipline, and that a state-independent basic policy would assign a given job to some queue. As we know the relative values, and in particular, the cost associated with accepting the cost to each queue, we can carry out the first policy iteration step (FPI). That is, we deviate from the default action if the expected cost is smaller with some other action, thereby decreasing the expected cumulative sojourn time globally in infinite time horizon. In practice, the policy iteration is known to converge quickly to the optimal policy, and the FPI step typically yields the highest improvement. Therefore, our assumption is that as long as one starts from a rational basic dispatching policy, the resulting state-dependent FPI policy is often close to optimal.

To summarize, the steps taken are as follows:

1. *Basic policy*: Start with a state-independent dispatching policy with a robust and predictable behaviour. The state-independence allows one to consider the queues in isolation.
2. *Relative values*: Derive an exact expression for the difference in the relative values between two states for a single M/G/1 queue with known job sizes and a given queueing discipline.
3. *FPI step*: Carry out the FPI step, i.e., improve the basic policy by choosing a different queue when an alternative action yields a lower expected increase in the cumulative sojourn time. Note that the resulting FPI based policy is no longer state-independent, making further policy iteration difficult.

3. Analysis of a Single M/G/1 Queue

In this section, we analyze M/G/1 queues and derive the size-aware relative values with respect to sojourn times for FIFO, LIFO, SPT and SRPT disciplines. For SPT and SRPT disciplines, we first give expressions for the mean remaining sojourn time of present customers in a given state, and then utilize it in the derivation of the relative values. Note that these results are generic, i.e., not restricted to the dispatching problem itself.

3.1. First In First Out (FIFO)

Let $\mathbf{z} = (\Delta_1; ..; \Delta_n)$ denote the amount of unfinished work in n tasks that receive service according to the FIFO discipline (i.e., FCFS) starting from Δ_n. The *backlog*, i.e., the amount of unfinished work, is

$$u_{\mathbf{z}} = \sum_{i=1}^{n} \Delta_i.$$

The complete understanding of the future costs is summarized in the relative values $v_{\mathbf{z}}$. For comparing states, it is sufficient to know the difference between, e.g., an arbitrary state $\mathbf{z}$ and a state 0 having no jobs:

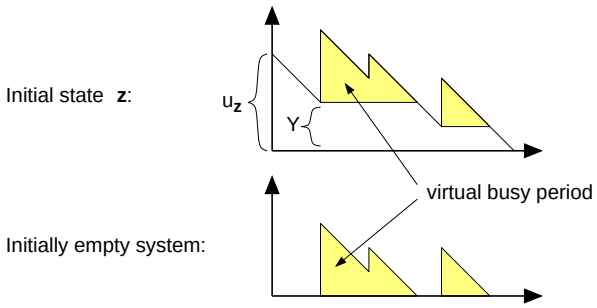

Figure 3: Difference in the cumulative sojourn times in M/G/1-FIFO between a system in state $\mathbf{z}$ and an empty system.

Proposition 3.1 (M/G/1-FIFO). *The size-aware relative value of state* $\mathbf{z} = (\Delta_1; ..; \Delta_n)$ *with respect to sojourn time in an M/G/1-FIFO queue is given by*

$$v_{(\Delta_1;..;\Delta_n)} - v_0 = \frac{\lambda\, u_{\mathbf{z}}^2}{2(1-\rho)} + \sum_{i=1}^{n} i\, \Delta_i, \tag{2}$$

where Δ_i *denotes the known (remaining) service time (amount of work) of job* i, $i = 1, .., n$, *the* n*th job is served first,* λ *is the Poisson arrival rate, and* $\rho = \lambda\, \mathrm{E}[X]$ *with* $\mathrm{E}[X]$ *denoting the mean job size.*

Proof Let us consider two systems under the same arrival realisations: System 1 initially in state $\mathbf{z} = (\Delta_1; ..; \Delta_n)$ and System 2 initially empty. Both systems behave identically once System 1 becomes empty. The total sojourn time of the n jobs in System 1 is already fixed in FIFO, and equals $\sum_{i=1}^{n} i\, \Delta_i$.

An arriving task starts a busy period in System 2, which corresponds to a virtual busy period in System 1, as illustrated in Fig. 3. During each (virtual) busy period, arriving jobs increase the cumulative sojourn time by an amount equal to the post arrival backlog. The key observation is that these jobs experience an additional delay of Y in System 1 when compared to System 2. Otherwise the contributions to the cumulative sojourn times are equal. The mean number of (virtual) busy periods before System 1 becomes idle is $\lambda\, u_{\mathbf{z}}$. The mean number of jobs served during a busy period of M/G/1 queue is $1/(1-\rho)$ (Kleinrock (1975); Ross (2000)). Moreover, the mean $\mathrm{E}[Y]$ is $u_{\mathbf{z}}/2$, which completes the proof. □

Corollary 3.2 (Insensitivity). *The size-aware relative values of an M/G/1-FIFO queue are insensitive to the form of the job size distribution.*

Interestingly, the first policy iteration *on the service order* already yields the shortest remaining processing time (SRPT) discipline that is optimal in minimizing the mean sojourn time (Schrage (1968)).

Dispatching problem. The cost in terms of cumulative sojourn time due to a new task with size x is

$$c_{\mathbf{z}}(x) = v_{(x;\Delta_1;..;\Delta_n)} - v_{(\Delta_1;..;\Delta_n)} = \frac{\lambda}{2(1-\rho)}(2\, u_{\mathbf{z}}\, x + x^2) + u_{\mathbf{z}} + x,$$

where $u_{\mathbf{z}}$ is the pre-arrival backlog, $u_{\mathbf{z}} = \sum_{i=1}^{n} \Delta_i$. According to FPI, in a multi-queue dispatching system one chooses the queue for which the $c_{\mathbf{z}}(x)$ is the smallest.

Corollary 3.3 (Ordinary dispatching system). *In a homogeneous dispatching system, without queue-specific job sizes, applying the FPI step to the RND basic policy with FIFO queues yields the LWL policy, FPI-RND/FIFO = LWL/FIFO, where the queue with the smallest amount of work is chosen.*

Note that with RND $\lambda/(1-\rho)$ is a common constant for all queues. Moreover, it is easy to see that *with the FIFO queueing discipline, the Myopic policy reduces to LWL* (with and without the queue-specific job sizes).

3.2. Last In First Out (LIFO)

Let us next consider the LIFO discipline (i.e., LCFS). The following result holds for the preemptive LIFO:

Proposition 3.4 (Preemptive M/G/1-LIFO). *The size-aware relative value of state* $(\Delta_1;..;\Delta_n)$ *with respect to sojourn time in an M/G/1 queue with a preemptive LIFO discipline is given by*

$$v_{(\Delta_1;..;\Delta_n)} - v_0 = \frac{1}{1-\rho}\sum_{i=1}^{n} i \cdot \Delta_i, \tag{3}$$

where task Δ_n *is served first and task* Δ_1 *last, and* $\rho = \lambda \mathrm{E}[X]$ *where* λ *is the Poissonian arrival rate and* $\mathrm{E}[X]$ *the mean job size.*

Proof Let D_i denote the (remaining) sojourn time of job $i, = 1, .., n$. With the preemptive LIFO the current state has no effect on the future arrivals' sojourn times and the difference between the relative value of state $(\Delta_1;..;\Delta_n)$ and an empty state is given by the mean total sojourn time of the n present jobs,

$$v_{(\Delta_1;..;\Delta_n)} - v_0 = \sum_{i=1}^{n} \mathrm{E}[D_i].$$

The sojourn time of the latest customer n currently receiving service is given by a random sum,

$$D_n = \Delta_n + \left(B_1 + \ldots + B_{A(\Delta_n)}\right) \tag{4}$$

where $A(\Delta_n)$ denotes the number of arrivals during time Δ_n, and B_i is the duration of the ith busy period in an M/G/1 queue, for which $\mathrm{E}[B_i] = \mathrm{E}[X]/(1-\rho)$ (see Kleinrock (1975); Ross (2000)). Taking the expectation on both sides of (4) gives (cf. Wald's equation, $A(\Delta_n)$ and the B_i are independent)

$$\mathrm{E}[D_n] = \Delta_n + \mathrm{E}[A(\Delta_n)] \cdot \mathrm{E}[B] = \frac{\Delta_n}{1-\rho}.$$

Similarly, the mean sojourn time of the ith customer, $i = 1, .., n$, is $\mathrm{E}[D_i] = (1-\rho)^{-1}\sum_{j=i}^{n}\Delta_j$, and therefore

$$\sum_{i=1}^{n}\mathrm{E}[D_i] = \frac{1}{1-\rho}\sum_{i=1}^{n} i \cdot \Delta_i.$$

□

Proposition 3.5 (Non-preemptive M/G/1-LIFO). *The size-aware relative value of state* $(\Delta_1;..;\Delta_n)$ *with respect to sojourn time in an M/G/1 queue with a non-preemptive LIFO discipline is given by*

$$v_{(\Delta_1;..;\Delta_n)} - v_0 = \frac{1}{1-\rho}\sum_{i=1}^{n}(i-\rho)\Delta_i + \frac{\lambda}{2(1-\rho)}\sum_{i=1}^{n}\Delta_i^2, \tag{5}$$

where task Δ_n *is served first and task* Δ_1 *last, and* $\rho = \lambda \mathrm{E}[X]$ *with* λ *denoting the Poisson arrival rate and* $\mathrm{E}[X]$ *the mean job size.*

Proof For the existing jobs $i = 1, ..., n$, the mean waiting time is $\mathrm{E}[W_i] = (1-\rho)^{-1}\sum_{j=i+1}^{n}\Delta_j$. Therefore, the cumulative *sojourn time* of the n known jobs is

$$V_1 = \sum_{i=1}^{n}(\mathrm{E}[W_i] + \Delta_i) = \frac{1}{1-\rho}\sum_{i=1}^{n}(i-\rho)\Delta_i.$$

Consider next the difference in future arrivals' sojourn times between System 1 initially at state $(\Delta_1;..;\Delta_n)$ and System 2 initially empty. For each arrival realization, both systems behave identically after time t^*

when System 1 becomes empty for the first time. Also the service times of the new arrivals are equal in both systems and thus can be neglected. This leaves us to consider the waiting times of the jobs arriving during $(0, t^*)$. Instead of considering their actual waiting times, one can assume that waiting costs are accrued at the expected rate the cumulative waiting time increases, $r_{\mathbf{z}} = \lambda \cdot \Delta/(1-\rho)$ where Δ denotes the remaining service time of a job currently receiving service. The mini busy periods due to new arrivals in both systems, albeit not necessarily concurrent, cancel each other. Thus, the cost difference is accrued while the n original jobs receive service, which reduces to an integral over n triangles with heights Δ_i,

$$V_2 = \sum_{i=1}^{n} \frac{\lambda}{1-\rho} \int_0^{\Delta_i} x\, dx = \frac{\lambda}{2(1-\rho)} \sum_{i=1}^{n} \Delta_i^2.$$

Consequently, the difference in the expected cumulative sojourn times, $v_{\mathbf{z}} - v_0 = V_1 + V_2$, yielding (5). □

We note that (5) can also be written as

$$v_{\mathbf{z}} - v_0 = \sum_{i=1}^{n} \Delta_i + \frac{1}{1-\rho} \sum_{i=1}^{n} (i-1)\Delta_i + \frac{\lambda}{2(1-\rho)} \sum_{i=1}^{n} \Delta_i^2,$$

where the first sum corresponds to the service times of the known jobs, the second sum to the expected waiting times of the known jobs, and the third sum to the expected difference in the waiting times the future arrivals experience between state $(\Delta_1; ..; \Delta_n)$ and an empty state.

Corollary 3.6 (Insensitivity). *The relative values of size-aware preemptive and non-preemptive M/G/1-LIFO queues are also insensitive to the form of the job size distribution.*

Similarly as with the FIFO, also with the LIFO queues, applying the first policy iteration *on the service order* yields the optimal SRPT policy (Schrage (1968)).

Dispatching problem. The cost of accepting a job with size x to a queue is equal to the difference in the relative values. In the case of preemptive LIFO, this gives,

$$c_{\mathbf{z}}(x) = v_{(\Delta_1;..;\Delta_n;x)} - v_{(\Delta_1;..;\Delta_n)} = \frac{1}{1-\rho}(n+1)x.$$

In the case of non-preemptive discipline the interesting quantity is

$$c_{\mathbf{z}}(x) = v_{(\Delta_1;..;x;\Delta_n)} - v_{(\Delta_1;..;\Delta_n)} = \begin{cases} \dfrac{\lambda}{2(1-\rho)}\, x^2 + \dfrac{n-\rho}{1-\rho}\, x + \dfrac{\Delta_n}{1-\rho}, & \text{when } n > 0, \\ \dfrac{\lambda}{2(1-\rho)}\, x^2 + x, & \text{when } n = 0. \end{cases}$$

Therefore, with the preemptive LIFO, the cost $c_{\mathbf{z}}(x)$ depends only on the job size x, the number of tasks in queue n, and the offered load ρ. In particular, for a homogeneous state-independent basic policy, for which the $\rho^{(i)}$ are equal for all queues i, $\rho^{(i)} = \rho^{(j)}\ \forall\, i, j$, the FPI criteria yields a policy that chooses queue i iff

$$\frac{x^{(i)}}{x^{(j)}} \leq \frac{n^{(j)}+1}{n^{(i)}+1}, \qquad \forall\, j,$$

where $x^{(i)}$ denotes the conditional job size for queue i, and $n^{(i)}$ denotes the number of jobs in queue i.

Corollary 3.7. *With preemptive LIFO queues, the FPI-α policy reduces to the Myopic policy for all state-independent basic policies α that balance the load, $\rho^{(i)} = \rho^{(j)}$, $\forall\, i, j$.*

Corollary 3.8 (Ordinary dispatching system). *In a homogeneous dispatching system with preemptive LIFO queues, and without queue-specific job sizes, applying the FPI step to the RND policy yields JSQ policy, FPI-RND/LIFO = JSQ/LIFO, where the queue with the smallest number of jobs is chosen.*

3.3. Shortest Processing Time (SPT)

Next we focus on the shortest processing time (SPT) discipline where each job gets a fixed priority that is defined by the *initial* job size. We consider an M/G/1 queue with arbitrary i.i.d. job sizes. For simplicity of notation, we assume a continuous job size distribution and let $p(x)$ denote the probability density function of the job size X. Consequently, $X_i \neq X_j$ almost surely for $i \neq j$ and there is no need to define a service order for jobs with equal sizes. Furthermore, $\lambda(x)$ denotes the arrival rate of tasks smaller than x, $\lambda(x) = \lambda \,\mathrm{P}\{X < x\}$, $S(x)$ the mean job size of jobs smaller than x, $S(x) = \mathrm{E}[X \,|X < x]$, and $\rho(x)$ the offered load consisting of jobs smaller than x, $\rho(x) = \lambda(x) \cdot S(x) = \lambda \int_0^x t\, p(t)\, dt$. Thus, $\lim_{x\to\infty} S(x) = \mathrm{E}[X]$ and $\lim_{x\to\infty} \rho(x) = \rho$.

3.3.1. Preemptive M/G/1-SPT

The classical mean delay formula in a preemptive M/G/1 queue with k priority classes for class j, $j = 1, .., k$, gives (see, e.g., Gross and Harris (1998))

$$\mathrm{E}[T_j] = \frac{\mathrm{E}[X_j]\,(1 - \sum_{i=1}^{j} \rho_i) + R_j}{(1 - \sum_{i=1}^{j-1} \rho_i)(1 - \sum_{i=1}^{j} \rho_i)},$$

where $\mathrm{E}[X_j]$ and $\rho_j = \lambda_j \,\mathrm{E}[X_j]$ are the class specific mean job size and offered load, and $R_j = \frac{1}{2} \sum_{i=1}^{j} \lambda_i \,\mathrm{E}[X_i^2]$ is the mean residual service time a class j job observes upon arrival. With SPT, each job size Δ constitutes its own class, and, at the continuum limit when $k \to \infty$, the above becomes

$$\mathrm{E}[T(\Delta)] = \frac{\Delta}{1 - \rho(\Delta)} + \frac{R(\Delta)}{(1 - \rho(\Delta))^2}, \quad \text{where } R(\Delta) = \frac{\lambda}{2} \int_0^{\Delta} x^2\, p(x)\, dx. \tag{6}$$

The mean sojourn time is obtained by taking an integral over the job sizes Δ,

$$\mathrm{E}[T] = \int_0^{\infty} p(x) \cdot \mathrm{E}[T(x)]\, dx.$$

Let $\mathbf{z} = ((\Delta_1, \Delta_1^*); ..; (\Delta_n, \Delta_n^*))$ denote the state of a queue where the Δ_i are the remaining service times and the Δ_i^* the initial job sizes. The amount of work having a higher priority than x in state $\mathbf{z}$ is,

$$u_{\mathbf{z}}(x) = \sum_{i\,:\,\Delta_i^* < x} \Delta_i.$$

We are interested in knowing the expected sojourn time of a job. Due to the preemptive service discipline, it is irrelevant if a job arrives immediately or if it has already arrived. For example, a (Δ, Δ^*)-job sees a virtual state corresponding to the amount of higher priority work $u_{\mathbf{z}}(\Delta^*)$.

Lemma 3.9. *The mean remaining sojourn time of a* (Δ, Δ^*)*-job in a* preemptive *M/G/1-SPT queue in state* $\mathbf{z} = ((\Delta_1, \Delta_1^*); ..; (\Delta_n, \Delta_n^*))$*, where the* (Δ_i, Δ_i^*) *denote the remaining and initial job sizes, is given by*

$$\mathrm{E}[T_{\mathbf{z}}(\Delta, \Delta^*)] = \frac{\Delta + u_{\mathbf{z}}(\Delta^*)}{1 - \rho(\Delta^*)}. \tag{7}$$

Proof The sojourn time is initially $\Delta + u_{\mathbf{z}}(\Delta^*)$ as illustrated in Fig. 4. From (Δ, Δ^*)-job point of view, we can focus on jobs smaller than Δ^*, which arrive according to a Poisson process with rate $\lambda(\Delta^*)$. Moreover, the service order of higher priority jobs is irrelevant for the (Δ, Δ^*)-job. Thus, we can assume that each new job smaller than Δ^* starts a mini busy period in an M/G/1-queue with a mean duration of $S(\Delta^*)/(1-\rho(\Delta^*))$. The mean prolongation is $\lambda(\Delta^*)\,(\Delta + u_{\mathbf{z}}(\Delta^*))\, S(\Delta^*)/(1 - \rho(\Delta^*)) = (\Delta + u_{\mathbf{z}}(\Delta^*))\, \rho(\Delta^*)/(1 - \rho(\Delta^*))$, giving

$$\mathrm{E}[T_{\mathbf{z}}(\Delta, \Delta^*)] = \Delta + u_{\mathbf{z}}(\Delta^*) + (\Delta + u_{\mathbf{z}}(\Delta^*)) \frac{\rho(\Delta^*)}{1 - \rho(\Delta^*)} = \frac{\Delta + u_{\mathbf{z}}(\Delta^*)}{1 - \rho(\Delta^*)}.$$

□

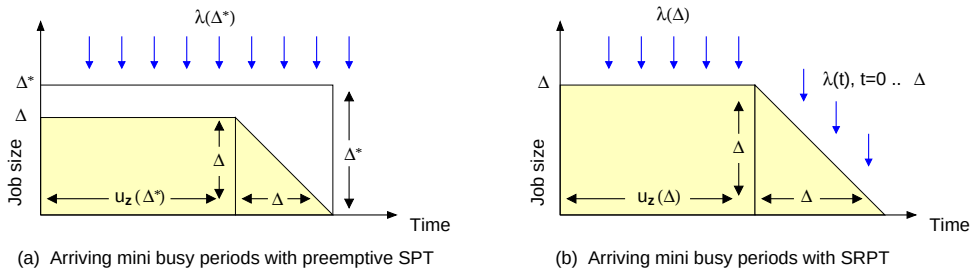

Figure 4: Derivation of the state dependent mean sojourn times for preemptive SPT and SRPT.

Thus, (6) allows us to compute the mean delay for an arriving (Δ, Δ)-job, while (7) gives the state dependent mean delay time for an arbitrary (Δ, Δ^*)-job. In the MDP framework, the costs can be associated as lump sums of the expected arrivals. To this end, let us define the cost (accumulating) rate at state $\mathbf{z}$ as

$$r_{\mathbf{z}} = \lambda \int_0^\infty p(x)\, \mathrm{E}[T_{\mathbf{z}}(x)]\, dx,$$

where $T_{\mathbf{z}}(x) = T_{\mathbf{z}}(x, x)$. Next we will utilize this definition to derive the relative value of a state:

Proposition 3.10 (preemptive M/G/1-SPT). *For the size-aware relative value of state* $\mathbf{z} = ((\Delta_1, \Delta_1^*); \ldots ; (\Delta_n, \Delta_n^*))$ *with respect to sojourn time in a preemptive M/G/1-SPT queue it holds that*

$$v_{\mathbf{z}} - v_0 = \sum_{i=1}^{n} \frac{\Delta_i + u_{\mathbf{z}}(\Delta_i^*)}{1 - \rho(\Delta_i^*)} + \int_0^\infty \frac{\lambda\, p(x) \cdot u_{\mathbf{z}}(x)^2}{2(1-\rho(x))^2}\, dx. \tag{8}$$

where λ *is the Poissonian arrival rate,* $p(x)$ *the pdf of job size distribution,* $u_{\mathbf{z}}(x)$ *the amount of unfinished work due to jobs initially shorter than* x*, and* $\rho(x) = \lambda \int_0^x t\, p(t)\, dt$*.*

Proof The expected sojourn times of the present n jobs are given by (7),

$$V_1 = \sum_{i=1}^{n} \frac{\Delta_i + u_{\mathbf{z}}(\Delta_i^*)}{1 - \rho(\Delta_i^*)}.$$

What is left is to evaluate the difference in the sojourn times that later arriving jobs experience between System 1 initially at state $\mathbf{z}$, and System 2 initially empty. Let V_2 denote this quantity, so that

$$v_{\mathbf{z}} - v_0 = V_1 + V_2.$$

Next consider job sizes $(x, x + dx)$ and let $s(x)\, dx$ denote their contribution to V_2. These jobs observe only the shorter jobs having a higher preemptive priority. Such jobs arrive at rate $\lambda(x)$. Due to the PASTA property, we can as well consider the backlog that an arriving x-job would observe and utilize (7). That is,

$$s(x) = \mathrm{E}\left[\int_0^\infty \lambda\, p(x) \left[\frac{x + U_{\mathbf{z}}(x,t)}{1-\rho(x)} - \frac{x + U_0(x,t)}{1-\rho(x)}\right] dt\right], \tag{9}$$

where the expectation is over all arrival realizations, and $U_{\mathbf{z}}(x,t)$ denotes the higher priority backlog at time t with initial state $\mathbf{z}$, i.e., $U_{\mathbf{z}}(x,0) = u_{\mathbf{z}}(x)$. For each arrival realization, both systems behave identically after System 1 becomes empty. Let T denote this time, $T = u_{\mathbf{z}}(x) + T_1 + \ldots + T_N$, where the T_j are durations of the N mini busy periods due to jobs shorter than x, and $N \sim \mathrm{Poisson}(\lambda(x)\, u_{\mathbf{z}}(x))$. Thus, $s(x)$ can be expressed as

$$s(x) = \frac{\lambda\, p(x)}{1-\rho(x)}\, \mathrm{E}\left[\int_0^T U_{\mathbf{z}}(x,t) - U_0(x,t)\, dt\right]. \tag{10}$$

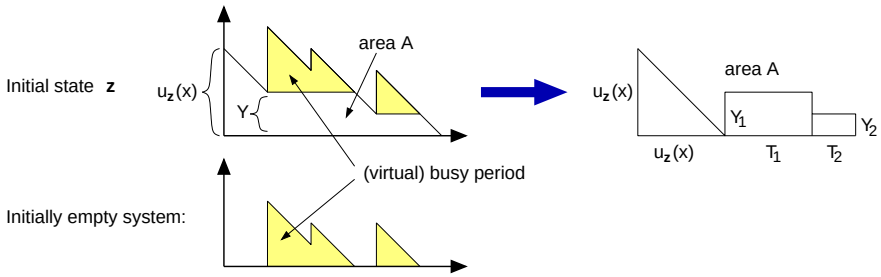

Figure 5: Difference in the cumulative sojourn times in an M/G/1-SPT queue between state $\mathbf{z}$ and an empty system for $(x, x+dx)$-jobs with aid of PASTA property.

Next we refer to Fig. 5 and observe that the integral inside the expectation corresponds to the white area in the upmost graphs. This area consists of one triangle and N $T_j \times Y_j$ rectangles. As the arrival process is Poisson, the heights Y_j are i.i.d., $Y_j \sim \mathrm{U}(0, u_{\mathbf{z}}(x))$. Similarly, the durations of the busy periods are i.i.d. with mean $S(x)/(1-\rho(x))$. Therefore, the expected area, i.e., the expectation in (10), is

$$\frac{u_{\mathbf{z}}(x)^2}{2} + \lambda(x)\, u_{\mathbf{z}}(x) \cdot \frac{u_{\mathbf{z}}(x)}{2} \cdot \frac{S(x)}{1-\rho(x)} = \frac{u_{\mathbf{z}}(x)^2}{2(1-\rho(x))}.$$

Substituting this to (10) then gives

$$s(x) = \frac{\lambda\, p(x) \cdot u_{\mathbf{z}}(x)^2}{2(1-\rho(x))^2}, \tag{11}$$

and finally,

$$V_2 = \int_0^\infty s(x)\, dx = \int_0^\infty \frac{\lambda\, p(x) \cdot u_{\mathbf{z}}(x)^2}{2(1-\rho(x))^2}\, dx.$$

Combining V_1 and V_2 yields (8). □

Note that the approximation of the relative value based on the expected sojourn time of the present customers corresponds to the first summation term. The latter integral term corresponds to the relative sojourn time of the later arriving jobs. Expanding the $u_{\mathbf{z}}(x)$ in (8) gives a more explicit form:

$$v_{\mathbf{z}} - v_0 = \sum_{i=1}^{n} \frac{\Delta_i + \sum_{j=i+1}^{n} \Delta_j}{1-\rho(\Delta_i^*)} + \frac{\lambda}{2} \sum_{i=1}^{n} \left(\sum_{j=i}^{n} \Delta_j \right)^2 \int_{\Delta_i^*}^{\Delta_{i-1}^*} \frac{p(x)}{(1-\rho(x))^2}\, dx, \tag{12}$$

where $\Delta_1^* > \Delta_2^* > \ldots > \Delta_n^*$ and $\Delta_0^* = \infty$. We note that it is also straightforward to write a similar expression for the case of a discrete job size distribution. Finally, we have the following corollary which generalizes the result on SPT to an arbitrary fixed priority scheme (e.g., the longest processing time (LPT) discipline):

Corollary 3.11 (Fixed priorities). *The same approach holds also for an arbitrary fixed priority system, where each job is assigned a unique priority number x_i upon arrival. A job i is characterized by a (Δ_i, x_i) pair, where Δ_i is the (remaining) job size. In this case, $\rho(x)$ denotes the offered load due to jobs with a higher priority than x, $u_{\mathbf{z}}(x)$ the amount of work with a higher priority than x in state $\mathbf{z}$, and $p(x)$ is the pdf of the job priority distribution. With this notation,* (8) *becomes*

$$v_{\mathbf{z}} - v_0 = \sum_{i=1}^{n} \frac{\Delta_i + u_{\mathbf{z}}(x_i)}{1-\rho(x_i)} + \int_0^\infty \frac{\lambda\, p(x) \cdot u_{\mathbf{z}}(x)^2}{2(1-\rho(x))^2}\, dx. \tag{13}$$

The proof is essentially the same as with the SPT. Also (6) and (7) are straightforward to express in a more general context if necessary. Similar expressions can be written also for the classical priority queues with a finite set of priorities assuming the FIFO or LIFO discipline within each priority class. As an example, one can introduce *random priorities* by assigning each job an i.i.d. random priority number P_i upon arrival.

3.3.2. Non-preemptive M/G/1-SPT

The classical mean delay formula for class j, $j = 1, \ldots, k$, in a non-preemptive M/G/1-SPT queue reads

$$\mathrm{E}[T_j] = \frac{R}{(1 - \sum_{i=1}^{j-1} \rho_i)(1 - \sum_{i=1}^{j} \rho_i)} + \mathrm{E}[X_j],$$

where $R = \frac{1}{2} \sum_{i=1}^{k} \lambda_i \mathrm{E}[X_i^2]$ is the mean residual service time upon arrival. The first term corresponds to the mean waiting time and the second term to the mean service time. At continuum limit one again obtains the conditional mean delay formula for the non-preemptive SPT discipline (see Phipps (1956)),

$$\mathrm{E}[T(\Delta)] = \frac{(\lambda/2)\,\mathrm{E}[X^2]}{(1 - \rho(\Delta))^2} + \Delta, \quad \text{and} \quad \mathrm{E}[T] = (\lambda/2)\,\mathrm{E}[X^2] \int_0^\infty \frac{p(x)}{(1 - \rho(x))^2}\, dx + \mathrm{E}[X].$$

Proposition 3.12 (Non-preemptive M/G/1-SPT). *Let $\mathbf{z} = (\Delta_1; ..; \Delta_n)$ denote an arbitrary state in a non-preemptive M/G/1-SPT queue with job n receiving service and jobs $1, \ldots, (n-1)$ waiting in the queue, $\Delta_1 > \Delta_2 > \ldots > \Delta_{n-1}$. For the size-aware relative value with respect to sojourn time it holds that,*

$$v_{(\Delta_1;..;\Delta_n)} - v_0 = \sum_{i=1}^{n} \left(\Delta_i + \frac{\sum_{j=i+1}^{n} \Delta_j}{1 - \rho(\Delta_i)} \right) + \frac{\lambda}{2} \sum_{i=1}^{n} \left(\left(\sum_{j=1}^{i-1} \Delta_j^2 + \big(\sum_{j=i}^{n} \Delta_j\big)^2 \right) \int_{\tilde{\Delta}_i}^{\tilde{\Delta}_{i-1}} \frac{p(x)}{(1 - \rho(x))^2}\, dx \right), \qquad (14)$$

where $p(x)$ denotes the pdf of the job size distribution, $\rho(x)$ is the offered load due to jobs shorter than x, $\rho(x) = \lambda \int_0^x t\, p(t)\, dt$, $\tilde{\Delta}_0 = \infty$, $\tilde{\Delta}_n = 0$ and for $i = 1, \ldots, (n-1)$, $\tilde{\Delta}_i = \Delta_i$.

Proof is similar to the proof given for the preemptive M/G/1-SPT and thus we only sketch it here. First, the mean remaining sojourn time of a job i is $\Delta_i + \left(\sum_{j=i+1}^{n} \Delta_j\right) / (1 - \rho(\Delta_i))$, which gives the first sum. The latter part corresponds to the mean difference in total sojourn time of the future arrivals between a system initially in state $(\Delta_1; ..; \Delta_n)$ and a system initially empty. Due to the non-preemptive discipline, it is sufficient to consider the waiting times, which, through a very similar reasoning as in the case of preemptive SPT, then finally yields (14), which is similar to (12). Moreover, an equivalent expression as in Corollary 3.11 can also be written in the non-preemptive case.

3.4. Shortest Remaining Processing Time (SRPT)

The shortest remaining processing time is the optimal queueing discipline for a single-server queue (Schrage and Miller (1966); Schrage (1968)), and thus has great importance both in theory and in practice. Also for the SRPT discipline with Poisson arrivals, an integral expression for the size specific mean sojourn time is known, which then gives the mean cost rate per job size similarly as (6) for the SPT. For more general arrival processes, we refer to Down et al. (2009). Again we assume i.i.d. job sizes X_i with a continuous job size distribution so that $X_i \neq X_j$ for $i \neq j$ almost surely, and let $\mathbf{z} = (\Delta_1; ..; \Delta_n)$ denote the state of the queue where Δ_i is the remaining amount of work in job i. With these we have:

Lemma 3.13. *In a preemptive M/G/1-SRPT queue at state $\mathbf{z} = (\Delta_1; ..; \Delta_n)$, the mean remaining sojourn time of a job with a (remaining) size Δ and an unfinished work $u_{\mathbf{z}}(\Delta)$ ahead in the queue is given by*

$$\mathrm{E}[T_{\mathbf{z}}(\Delta)] = \Delta + \frac{u_{\mathbf{z}}(\Delta)}{1 - \rho(\Delta)} + \int_0^\Delta \frac{\rho(t)}{1 - \rho(t)}\, dt. \qquad (15)$$

Proof Similarly as with the SPT, excluding new arrivals and the corresponding virtual busy periods for a moment, during time $u_{\mathbf{z}}(\Delta)$ the jobs shorter than Δ have a higher priority. After that the threshold decreases linearly to zero, as illustrated in Fig. 4 (right). Therefore, the mean prolongation due to new arrivals is

$$\lambda(\Delta)\, u_{\mathbf{z}}(\Delta) \cdot \frac{S(\Delta)}{1 - \rho(\Delta)} + \int_0^\Delta \lambda(t) \frac{S(t)}{1 - \rho(t)}\, dt.$$

Then adding the initial sojourn time, $\Delta + u_{\mathbf{z}}(\Delta)$, to the above yields (15). □

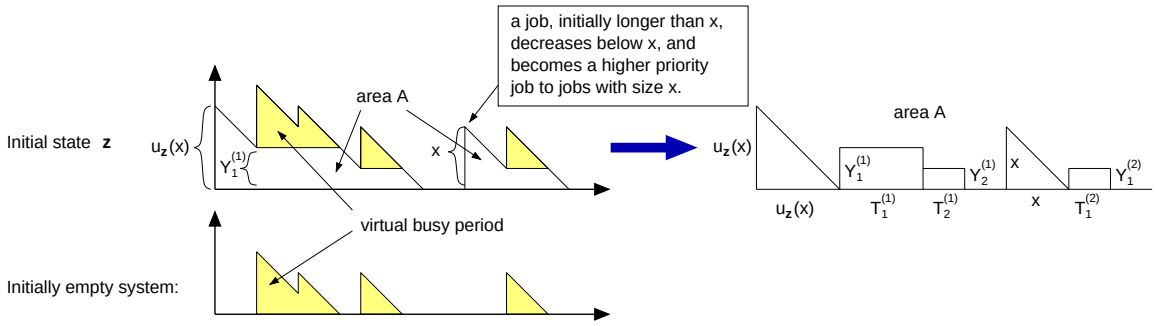

Figure 6: Difference in the cumulative sojourn times in an M/G/1-SRPT queue between state $\mathbf{z}$ and an empty system for $(x, x+dx)$-jobs corresponds to the white area.

The integral at the right-hand side complicates applying this equation directly. For example, $X \sim \mathrm{Exp}(\mu)$ yields no closed form expression and one has to resort to numerical integration. However, a closed form expression exists, e.g., for uniformly distributed job sizes. Moreover, the integral can be combined with the Δ term, yielding a more compact expression $\mathrm{E}[T_{\mathbf{z}}(\Delta)] = u_{\mathbf{z}}(\Delta)/(1-\rho(\Delta)) + \int_0^{\Delta}(1-\rho(t))^{-1}\,dt$. Below we use (15) as it is an explicit decomposition to the own service time and to the delay caused by the other jobs.

Proposition 3.14 (M/G/1-SRPT). *The size-aware relative value of state* $\mathbf{z} = (\Delta_1; ..; \Delta_n)$ *with respect to sojourn time in an M/G/1 queue with the SRPT discipline is given by*

$$v_{\mathbf{z}} - v_0 = \sum_{i=1}^{n}\left(\Delta_i + \frac{u_{\mathbf{z}}(\Delta_i)}{1-\rho(\Delta_i)} + \int_0^{\Delta_i}\frac{\rho(t)}{1-\rho(t)}\,dt\right) + \int_0^{\infty}\frac{\lambda\,p(x)\,\left(u_{\mathbf{z}}(x)^2 + n_{\mathbf{z}}(x)\,x^2\right)}{2(1-\rho(x))^2}\,dx, \tag{16}$$

where λ *is the arrival rate,* $p(x)$ *the pdf of job size distribution,* $u_{\mathbf{z}}(x)$ *the amount of unfinished work due to jobs shorter than* x *in state* $\mathbf{z}$*,* $n_{\mathbf{z}}(x)$ *the number of jobs longer than* x *in state* $\mathbf{z}$*, and* $\rho(x) = \lambda(x)\,S(x)$ *with* $\lambda(x)$ *denoting the arrival rate and* $S(x)$ *the mean size of jobs shorter than* x*.*

Proof Similarly as in Proposition 3.10, (15) gives the expected sojourn time of the n current jobs,

$$V_1 = \sum_{i=1}^{n}\left(\Delta_i + \frac{u_{\mathbf{z}}(\Delta_i)}{1-\rho(\Delta_i)} + \int_0^{\Delta_i}\frac{\rho(t)}{1-\rho(t)}\,dt\right).$$

For the later arriving jobs, similarly as in (9), we can write

$$\begin{aligned} s(x) &= \mathrm{E}\left[\int_0^{\infty}\lambda\,p(x)\left(\left(x + \frac{U_{\mathbf{z}}(x,t)}{1-\rho(x)} + \int_0^{x}\frac{\rho(h)}{1-\rho(h)}\,dh\right) - \left(x + \frac{U_0(x,t)}{1-\rho(x)} + \int_0^{x}\frac{\rho(h)}{1-\rho(h)}\,dh\right)\right)dt\right], \\ &= \frac{\lambda\,p(x)}{1-\rho(x)}\,\mathrm{E}\left[\int_0^{\infty}\left(U_{\mathbf{z}}(x,t) - U_0(x,t)\right)\,dt\right], \end{aligned}$$

which describes the expected difference in the cumulative sojourn times between System 1, initially at state $\mathbf{z}$, and System 2 initially empty. With SRPT also the jobs in System 1 initially longer than x affect the result as, at some point in time, the remaining length of such jobs, one at a time, decreases below the threshold x and triggers a new higher priority workload that is present only in System 1 (see Fig. 6). Consequently, instead of a single sequence of a $u_{\mathbf{z}}(x)$-triangle followed by rectangles, we have one $u_{\mathbf{z}}(x)$-sequence and $n_{\mathbf{z}}(x)$ times an x-sequence, where $n_{\mathbf{z}}(x)$ denotes the number of jobs longer than x in state $\mathbf{z}$. Each of these reduce to (11), which gives

$$s(x) = \frac{\lambda\,p(x)\,\left[u_{\mathbf{z}}(x)^2 + n_{\mathbf{z}}(x)\,x^2\right]}{2(1-\rho(x))^2},$$

and $V_2 = \int s(x)\,dx$. Then $V_1 + V_2$ yields (16). □

	At Dispatcher	At Queue after ME Dispatcher	
Arrivals	Poisson with rate λ	Poisson with rate λ/m	
Job size X	i) Exp(μ), $\mathrm{P}\{X > t\} = e^{-\mu t}$	Exp($m\mu$)	$\rho(x) = \frac{\lambda}{m^2\mu}\left(1 - e^{-m\mu x}(1 + m\mu x)\right) \to \frac{\lambda}{m^2\mu}$
	ii) U$(0, L)$, $\mathrm{P}\{X > t\} = 1 - t/L$	-	$\rho(x) = \frac{\lambda}{m} \cdot \frac{L^{m+1} - (L-x)^m(L+mx)}{(m+1)L^m} \to \frac{\lambda L}{m^2+m}$
	iii) Pareto(β), $\mathrm{P}\{X > t\} = (1+t)^{-\beta}$	Pareto($m\beta$)	$\rho(x) = \frac{\lambda}{m} \cdot \frac{(1+x)^{m\beta} - m\beta x - 1}{(m\beta-1)(1+x)^{m\beta}} \to \frac{\lambda}{m^2\beta - m}$

Table 1: Job size distributions, and how they appear to each queue with i.i.d. queue-specific job sizes after the ME policy.

Computationally, it is convenient to evaluate the latter integral for V_2 in (16) in $n+1$ parts. That is, in each interval $(0, \Delta_1)$, (Δ_1, Δ_2), .. (Δ_n, ∞), both $u_{\mathbf{z}}(x)$ and $n_{\mathbf{z}}(x)$ are constants and one can compute (and tabulate) in advance the state-independent cumulative functions,

$$F(t) = \int_0^t \frac{p(x)}{2(1-\rho(x))^2}\, dx \quad \text{and} \quad G(t) = \int_0^t \frac{p(x)\, x^2}{2(1-\rho(x))^2}\, dx.$$

4. Application to the Size-aware Dispatching Problem

Next, we utilize the derived relative values of M/G/1 queues in the context of dispatching problems. As described in Section 2, any state-independent dispatching policy can be improved by the policy iteration once the relative values of states are known. In the following, we refer to these improved policies by the FPI prefix, and compare their performance to heuristic dispatching policies given in Section 2. The numerical results presented in this section are obtained by simulations.

Jobs arrive according to a Poisson process with rate λ to a dispatcher, which assigns them to one of the m queues immediately. The job size distributions considered are given in the left column of Table 1: uniform $U(0, 2)$, exponential Exp(1) and Pareto(β). The uniform distribution has a finite range and strictly increasing hazard function, the exponential distribution has no memory and a constant hazard function, and the Pareto distribution is heavy-tailed with a strictly decreasing hazard function.

First we consider the ordinary case with homogeneous servers, where the (random) job sizes are equal to all queues, $X_i = X$. The second example is with the queue-specific i.i.d. job sizes (service requirements), $X_i \sim X$, and the right column in Table 1 indicates the resulting arrival process to individual queues with the ME dispatching policy, and the corresponding conditional offered load $\rho(x)$ due to the jobs, the sizes of which are at most x. With the exponentially distributed job sizes, the job size distribution remains exponential with a m times smaller mean, and with the Pareto(β) distribution the chosen jobs obey Pareto distribution with parameter $m\beta$. In passing, we mention that the minimum operation the ME policy carries out is very efficient in reducing the heavy-tailedness. For example, with $\beta = 2$ the variance of a Pareto distribution is infinite, but choosing a smaller job among two increases the parameter β to 4, which is not such a tricky distribution to deal with. Finally, we note that even though our example settings are elementary, the approach lends itself also to situations where a server may have special traits (skills) enabling it to process certain job types faster than the other servers. Moreover, the basic policy can also be based on the job types as long as the dispatching decisions are independent of the queue states.

4.1. Ordinary Dispatching System

Let us start with ordinary dispatching/routing problems, where the job sizes are the same for each queue and servers have homogeneous service rates, i.e., $\mathbf{X} = (X_1, .., X_m)$ with $X_i = X$. In this case, deviating from the default action does not increase the total amount of work conducted.

4.1.1. Two FIFO queues without queue-specific job sizes

First we assume identical FIFO queues without queues-specific job sizes, i.e., an effort to serve a given job is the same for each queue. In addition to RND, JSQ, SITA-E, and their FPI variants, we consider also *SITA with Switch* policy defined as follows:

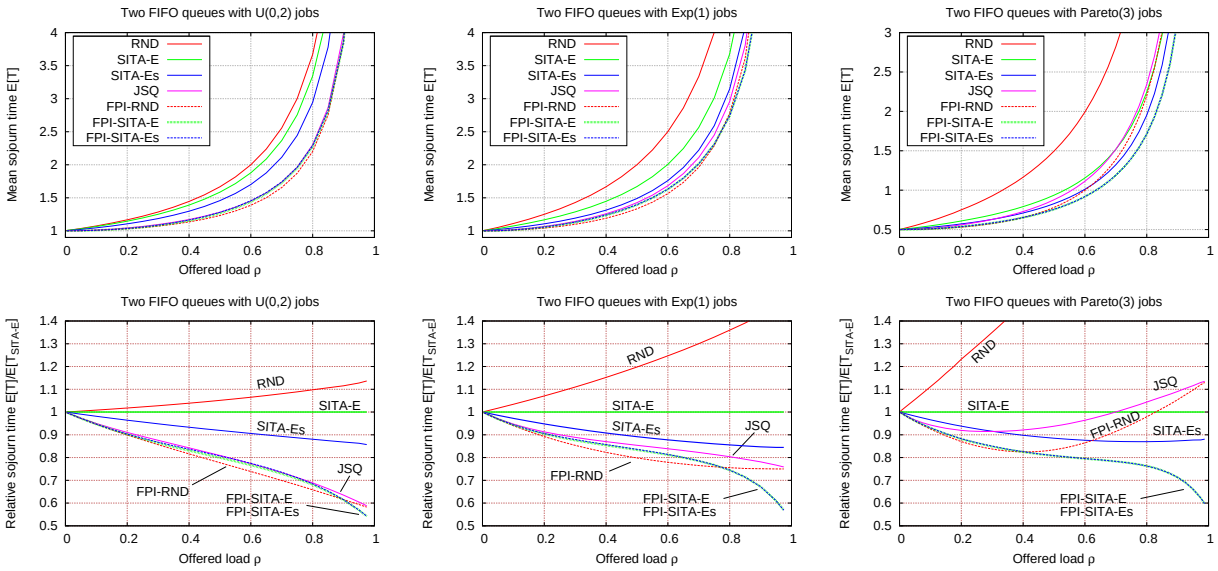

Figure 7: Mean sojourn time in an ordinary two queue dispatching system *without* queue-specific job sizes under FIFO queueing disciplines with uniformly, exponentially and Pareto distributed job sizes (from left to right).

Definition 4.1. *In SITA with Switch policy, upon arrival the roles of the queues (size intervals associated with each queue) are switched according to Proposition 3.1 so as to minimize the (initial) relative value.*

Consequently, SITA with Switch is always at least as good as the corresponding SITA policy. Stability concerns may arise with heterogeneous servers if the offered load in some size interval is too high for some server. With identical FIFO servers, this bears no meaning and the role switching reduces to the following:

Corollary 4.1. *Suppose m identical FIFO queues are numbered in decreasing order of the current backlog, $u_{\mathbf{z}}^{(1)} \geq u_{\mathbf{z}}^{(2)} \geq \ldots \geq u_{\mathbf{z}}^{(m)}$, and such size intervals of SITA that $\xi_i \triangleq \lambda_i/(1-\rho_i)$ for size interval i are in increasing order, $\xi_1 \leq \xi_2 \leq \ldots \xi_m$. Then, SITA with Switch assigns size interval i job to queue i.*

With SITA-E, the queue specific loads are equal and, in particular, the arrival rate λ_i of tasks belonging to size interval i is inversely proportional to the mean job size in the corresponding size interval. Therefore,

Corollary 4.2. *SITA-E with Switch assigns the shortest jobs to the shortest queue etc.*

That is, e.g., for two queues, *the short jobs* up to a certain threshold are assigned to the queue with a shorter backlog, and *the long jobs* to the other queue. Hence, knowledge of the relative value facilitates an elegant way to improve the SITA policy by dynamically switching the roles of the queues.

The numerical results with two FIFO queues are given in Fig. 7. As expected, the gain from SITA-E over RND increases as the job sizes vary more. With uniformly distributed job sizes, RND, SITA-E and SITA-Es (with switch) have somewhat similar performance, and most of the time the FPI-RND policy yields the lowest mean sojourn time. In contrast, with heavy-tailed job sizes, the SITA policies are significantly better than RND. Especially at higher loads, they also offer a better starting point for FPI. The FPI-RND policy cannot compete with the state-independent SITA policies, even though it does show a significant improvement over the basic policy RND. In contrast, the FPI-SITA policies manage to improve the mean sojourn time by about 40% in each case about $\rho = 1$ when compared to the optimal state-independent policy SITA-E (at that point). Thus, the gain from the queue state information can be significant.

Fig. 8 illustrates SITA-E and FPI-SITA-E policies for a two server system with FIFO queues. Both policies are of *switch-over* type, i.e., there is a certain threshold which defines to which queue a job is assigned to. The difference is that FPI-SITA-E implements a state-dependent threshold.

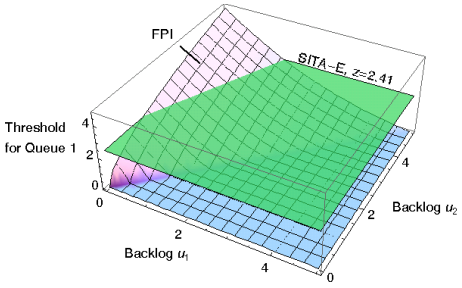

Figure 8: Both SITA-E and FPI-SITA-E are of switch-over type.

Figure 9: Dispatching system vs. a shared queue with SRPT. The results with two servers and exponentially distributed (*not queue-specific*) job sizes suggest that the disadvantage due to the dispatching decision can be insignificant (e.g., order of 5%).

4.1.2. Shared queue vs. dispatching system with SRPT

Next, we compare the situation between a multi-server and a dispatching system which both operate under the SRPT discipline. The multi-server system has a shared queue (Fig. 9 left), which means that at any given moment, the m servers will work on the m shortest jobs present (or all jobs if less than m jobs). In contrast, in the dispatching system it is possible that one queue has job(s) while the other queues are empty. Thus, the multi-server system is clearly in a better position to minimize the mean sojourn time. Two graphs in Fig. 9 illustrate the performance with $m = 2$ servers. The tiny bars in the middle graph correspond to $\pm 2\sigma$ confidence intervals, where σ denotes the sample standard deviation. For the dispatching system, we have chosen to include also the round-robin (RR) in addition to RND, JSQ, LWL and FPI-RND. We can observe that JSQ, LWL and FPI-RND work relatively well, FPI-RND being the best among them, i.e., the best policy for the dispatching system even though the starting point is the worst policy (RND). The RND and RR policies assume the least amount of information, and to no surprise, their performance is also limited when compared to the others. In this case, the penalty due to a dispatching decision in contrast to the shared queue appears to be about 5% or less.

4.2. Queue-specific Job Sizes

Next we consider dispatching systems with queue-specific job sizes, $\mathbf{X} = (X_1, .., X_m)$ where X_i denotes the service requirement if the given job is processed at queue i. For simplicity, we assume that the X_i are i.i.d., $X_i \sim X$. By definition, the default action of ME, $j = \underset{i}{\text{argmin}}\{X_i\}$, ensures that the system conducts as little work as possible. Conversely, deviating from the default action means that the system unavoidably conducts some amount of extra work. Therefore, e.g., RND and JSQ policies, ignoring this fact, have a limited capacity region.

4.2.1. Two queue system with the FIFO and preemptive LIFO disciplines

First we assume two queues both operating either under FIFO or a preemptive LIFO queueing discipline. The job sizes obey one of the three distributions given in Table 1: uniform, exponential and Pareto. We

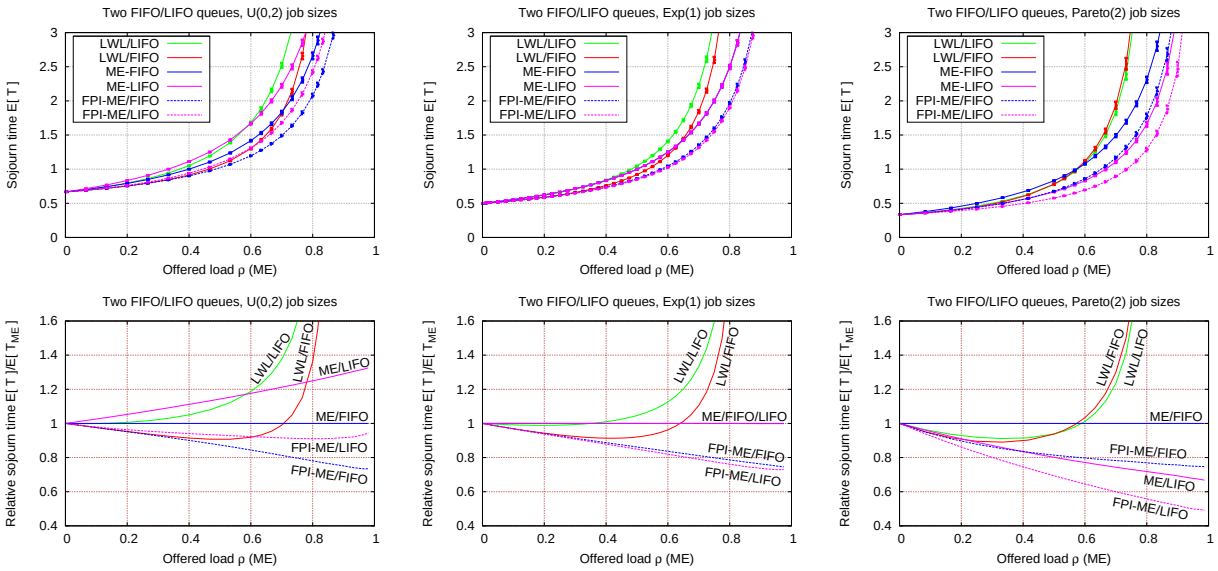

Figure 10: Mean sojourn time in a two queue system *with queue-specific* job sizes under FIFO and preemptive LIFO queueing disciplines. Job size distribution is uniform, exponential and Pareto (from left to right).

consider the ME, LWL, and FPI-ME dispatching policies together with the FIFO and LIFO queueing disciplines, i.e., in total 6 different systems. For the FPI policies, the exact expressions for the relative values were derived earlier in Sections 3.1 and 3.2. As mentioned earlier, the Myopic policy with FIFO and LIFO queueing disciplines reduces to the LWL/FIFO and FPI-ME/LIFO, respectively.

The numerical results obtained by means of simulation are illustrated in Fig. 10 for the three job size distributions: uniform, exponential and Pareto (from left to right). In the upper figures, the y-axis represents the mean sojourn time, and the x-axis the offered load where $\rho_{\mathrm{ME}} = 1$ corresponds to the maximum arrival rate the ME policy can support. The tiny bars correspond to $\pm 2\,\sigma$ confidence intervals. The lower figures represent the performance against the ME/FIFO policy, i.e., y-axis is the relative sojourn time (the lower the better). We can observe that the LWL policies "lose the ball" somewhere around $\rho \approx 0.2 \ldots 0.4$ in each case, and eventually become instable (cf. Sharifnia (1997)). The FPI policies are able to accomodate the full feasible load range $0 \le \rho_{\mathrm{ME}} < 1$, i.e., the one which the ME policies support. Moreover, the FPI based approaches yield consistently better performance than the corresponding ME policy, as the theory suggests. The relative performance appears to improve linearly until about $\rho \approx 0.95$. As λ increases further, the ME policy becomes the only feasible one and the others converge to that. Note that around $\rho = 0.95$, the improvement is about 25%..50%, which is rather significant. Among the two FPI based policies, the LIFO variant appears to be considerably better when $X \sim$ Pareto(2), and marginally better when $X \sim$ Exp(1). This is due to the fact, that with LIFO the new short jobs can be placed at the front of a queue, similarly as what SRPT does. In contrast, when $X \sim \mathrm{U}(0, 2)$, the FPI-ME/FIFO yields the lowest mean sojourn time (cf. a strictly increasing hazard function). The same phenomenon is well-known for single queue systems.

4.2.2. Two-queue system with the preemptive SPT discipline

Next we assume a two-queue system with the preemptive SPT discipline and the same three job size distributions. Substituting $\rho(x)$ from Table 1 into (7) gives the relative values, and one can again proceed with the FPI approach. Numerical results with the 4 dispatching policies, ME, LWL, Myopic and FPI-ME, are given in Fig. 11. For the ME policy, an analytical expression is readily available (apply (6) for individual queues) while the other 3 policies need to be simulated. We have simulated all 4 cases, and the tiny bars in the upper graph correspond to $\pm 2\,\sigma$ confidence intervals. Again, the LWL policy, which simply looks at the backlogs, faces the same capacity barrier as with the FIFO and LIFO (or any other) queueing discipline.

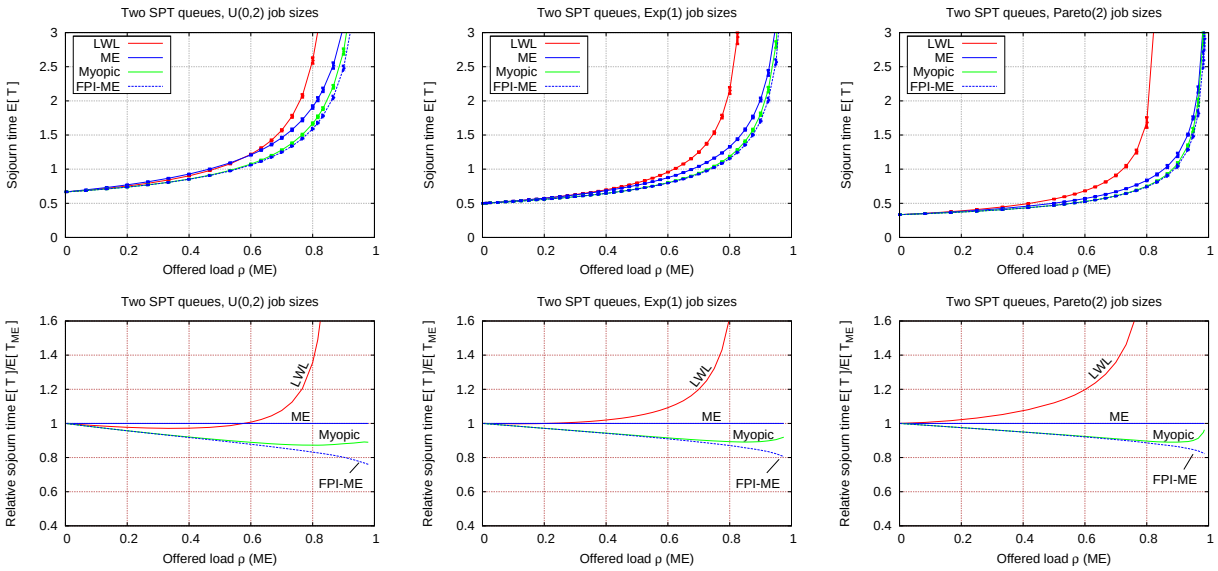

Figure 11: Mean sojourn time under preemptive SPT discipline *with queue-specific job sizes* from different distributions.

Also with lower loads, the LWL fails in improving the mean sojourn time.

In contrast, the FPI-ME and the Myopic policy are aware of the underlying discipline, and manage to decrease the mean sojourn time clearly below that of the basic ME policy. At higher loads, the FPI turns out to be superior with all job size distributions, and the gain is highest with the uniform distribution. Comparing Fig. 11 to Fig. 10, suggests that the improvement over the ME policy is somewhat smaller in this case. One logical reason contributing to this is the fact that the preemptive SPT discipline can locally correct poor dispatching choices by re-arranging the processing order in the queue.

4.2.3. Two-queue system with the SRPT discipline

Finally, let us consider the shortest remaining processing time (SRPT), which is the most efficient queueing discipline. In this case, the integral terms in (16) either have to be omitted, or evaluated numerically. We have chosen the latter, and use the composite Simpson's method and interpolation, which is computationally fast. Substituting the conditional offered load $\rho(x)$ from Table 1 to (16) gives an expression for the difference in the two relative values that can be evaluated numerically. In other words, it allows carrying out the FPI starting from the ME policy, yielding an improved policy which we refer to as the FPI-ME/SRPT policy.

Fig. 12 depicts the performance with uniform, exponential and Pareto distributed job sizes (from left to right). The bars in the upper graph again correspond to $\pm 2\,\sigma$ confidence intervals. In all cases, the FPI based approach utilizes the available servers better and push down the mean sojourn time. Similarly as with the SPT, the gain appears to decrease a bit with the heavy-tailed job size distribution. At higher loads, the performance gain is again significant, an order of $20\% \ldots 25\%$.

4.2.4. Comparison of queueing disciplines

With exponentially distributed job sizes, ME/FIFO and ME/LIFO have the same mean sojourn time. Let us next compare the ME policies to the FPI-ME policies with the four queueing disciplines: FIFO, preemptive LIFO, preemptive SPT and SRPT. The simulation results are depicted in Fig. 13. The $\pm 2\sigma$ confidence intervals are very small and have been omitted. One can identify two groups: FPI-ME/FIFO and FPI-ME/LIFO systems offer a reasonable improvement over the basic ME policies. However, the FPI-ME with SPT and SRPT attain a significantly lower mean sojourn time than the other policies. Interestingly, the difference between the static SPT and the dynamic SRPT disciplines is rather marginal.

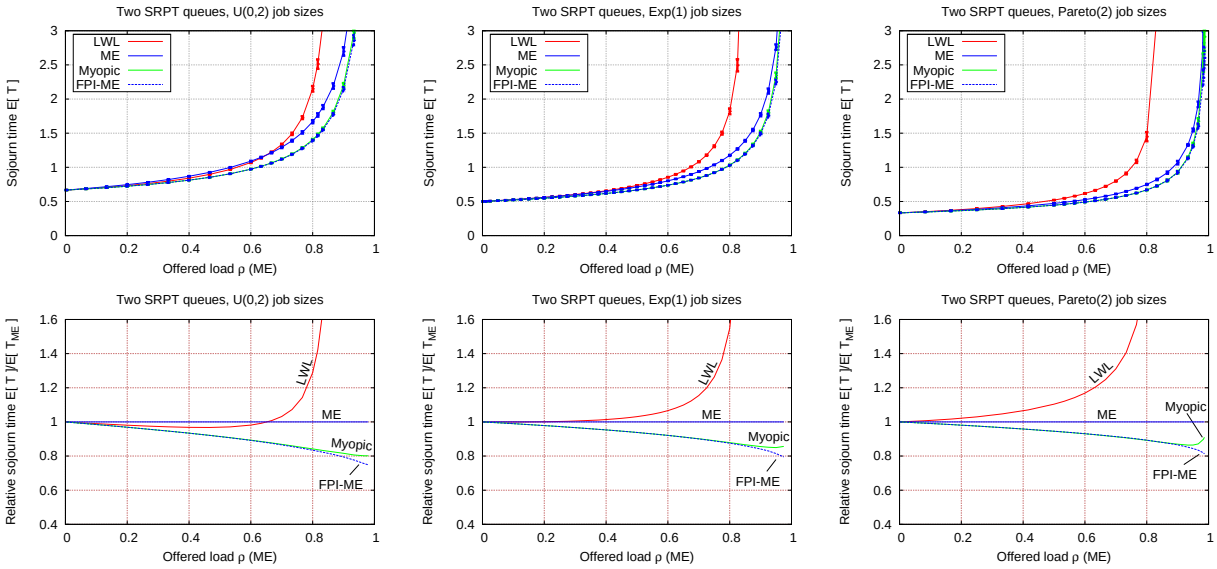

Figure 12: Mean sojourn time under SRPT discipline *with queue-specific job sizes* from different distributions.

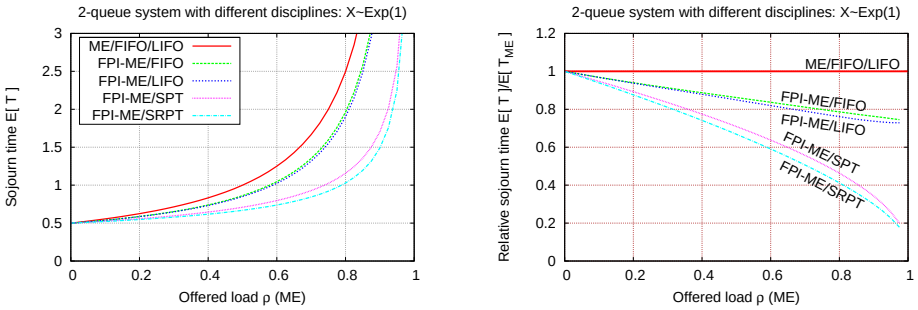

Figure 13: Different FPI based policies for the different queueing policies with exponentially distributed *queue-specific job sizes*, $X_i \sim \text{Exp}(1)$. Left figure depicts the absolute performance and the right the gain over ME/FIFO policy.

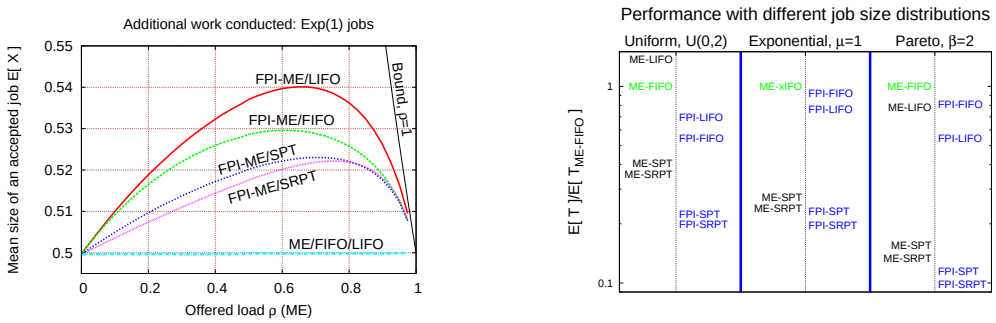

Figure 14: Left: Mean size of an accepted job with *queue-specific job sizes*, $X_i \sim \text{Exp}(1)$. Policies seeking to minize sojourn times have to accept jobs with sizes longer than the minimum. Right: Relative performance of the ME and the corresponding FPI-ME policy with different job size distributions at $\rho_{\text{ME}} = 0.9$. The y-axis is in logarithmic scale.

The mean sizes of the accepted jobs are illustrated in Fig. 14 (left). Recall that with ME, the mean job size is 0.5 (minimum of two Exp(1) random variables). However, if the dispatcher decides to deviate from this, unavoidably the mean job size also increases. From the figure, one can make three observations: (i) Initially, FPI-ME/FIFO/LIFO policies deviate from the ME policy more, as the strict queueing discipline (FIFO or LIFO) is non-optimal. (ii) As the load increases, also the preemptive FPI-ME/SPT, and in particular FPI-ME/SRPT, also find more options to improve the performance by switching the queue. (iii) Near the capacity limit at $\rho_{\text{ME}} = 1$, all policies have to adhere to ME and the mean job size converges back to 0.5.

Fig. 14 (right) depicts the achieved performance in different scenarios at $\rho_{\text{ME}} = 0.9$ relative to the ME/FIFO policy. For each job size distribution, the left column consists of the ME policies with different queueing disciplines (FIFO, preemptive LIFO, preemptive SPT, SRPT), and the corresponding FPI policy is on the right column. The y-axis is in logarithmic scale. We can observe that in each case the FPI improves the performance considerably. The order between different queueing disciplines remains otherwise the same, except that FIFO and LIFO change their relative order as one moves from a finite job sizes (uniform distribution) to a heavy-tailed one. The similar phenomenon is known to exist for ordinary queues.

5. Conclusions

We have considered dispatching or task assignment problems in the Markov decision process framework. In addition to the traditional cases, the approach supports queue-specific job sizes where the same job requires different efforts from different servers. This allows one to model a very large class of real life dispatching problems with heterogeneous servers. The dispatcher is assumed to have full knowledge about the new job and the state of each queue. The optimal dispatching decision, seeking to minimize the mean sojourn time, depends also on the queues' internal scheduling discipline (e.g., FIFO or SRPT).

Our approach stems from the MDP framework. First we have analyzed M/G/1 queues in isolation. In particular, we have derived exact expressions for the size-aware relative values in an M/G/1 queue under FIFO, LIFO, SPT and SRPT queueing disciplines. Moreover, for FIFO and LIFO, the corresponding relative values are shown to be insensitive to the form of job size distribution. These analytical results have also some general importance as they are readily applicable in many other contexts (e.g., admission control, pricing), while in this paper we have exploited them in the context of dispatching problems.

In particular, starting from a state-independent basic policy, the size-aware relative values allow us to carry out the first policy iteration (FPI) yielding a new improved state-dependent policy. The FPI policies take into account the arrival process, job size distribution and the internal queueing discipline, resulting in a sound and robust approach to a wide range of size-aware dispatching problems. This includes also settings where some servers are specialized to certain job types and can process such jobs faster than other servers. The methodology is demonstrated by means of numerical examples, where the FPI based approaches consistently yield considerably better performance in terms of mean sojourn time than any of the heuristic

policies. For example, policies known to be optimal in other settings such as JSQ, where the routing decision is based on the number of jobs per queue, and LWL, which considers the backlogs, are not generally optimal.

References

W. Winston, Optimality of the shortest line discipline, Journal of Applied Probability 14 (1977) 181–189.

B. W. Conolly, The Autostrada Queueing Problem, Journal of Applied Probability 21 (2) (1984) 394–403.

V. Gupta, M. Harchol-Balter, K. Sigman, W. Whitt, Analysis of Join-the-Shortest-Queue Routing for Web server Farms, Performance Evaluation 64 (9-12) (2007) 1062–1081.

B. Schroeder, M. Harchol-Balter, Evaluation of Task Assignment Policies for Supercomputing Servers: The Case for Load Unbalancing and Fairness, Cluster Computing 7 (2) (2004) 151–161.

M. E. Crovella, M. Harchol-Balter, C. D. Murta, Task Assignment in a Distributed System: Improving Performance by Unbalancing Load, in: Proceedings of SIGMETRICS '98, Madison, Wisconsin, USA, 268–269, 1998.

M. Harchol-Balter, M. E. Crovella, C. D. Murta, On Choosing a Task Assignment Policy for a Distributed Server System, Journal of Parallel and Distributed Computing 59 (1999) 204–228.

A. Wierman, Fairness and scheduling in single server queues, Surveys in Operations Research and Management Science 16 (1) (2011) 39–48.

L. E. Schrage, L. W. Miller, The queue M/G/1 with the shortest remaining processing time discipline, Operations Research 14 (4) (1966) 670–684.

L. Schrage, A proof of the optimality of the shortest remaining processing time discipline, Operations Research 16 (3).

R. Righter, J. G. Shanthikumar, G. Yamazaki, On Extremal Service Disciplines in Single-Stage Queueing Systems, Journal of Applied Probability 27 (2) (1990) 409–416.

A. Ephremides, P. Varaiya, J. Walrand, A simple dynamic routing problem, IEEE Transactions on Automatic Control 25 (4) (1980) 690–693.

Z. Liu, D. Towsley, Optimality of the Round-Robin Routing Policy, Journal of Applied Probability 31 (2) (1994) 466–475.

Z. Liu, R. Righter, Optimal Load Balancing on Distributed Homogeneous Unreliable Processors, Operations Research 46 (4) (1998) 563–573.

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

R. Wu, D. G. Down, Round robin scheduling of heterogeneous parallel servers in heavy traffic, European Journal of Operational Research 195 (2) (2009) 372–380.

W. Whitt, Deciding Which Queue to Join: Some Counterexamples, Operations Research 34 (1) (1986) 55–62.

B. Hajek, Optimal control of two interacting service stations, IEEE Transactions on Automatic Control 29 (6) (1984) 491–499.

F. Bonomi, On Job Assignment for a Parallel System of Processor Sharing Queues, IEEE Trans. Comput. 39 (7) (1990) 858–869.

K. R. Krishnan, Joining the right queue: a Markov decision rule, in: Proc. of the 28th Conference on Decision and Control, 1863–1868, 1987.

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.

S. Bhulai, On the Value Function of the M/Cox(r)/1 Queue, Journal of Applied Probability 43 (2) (2006) 363–376.

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, 190–197, 2011a.

E. Hyytiä, J. Virtamo, S. Aalto, A. Penttinen, M/M/1-PS Queue and Size-Aware Task Assignment, Performance Evaluation doi:10.1016/j.peva.2011.07.011, available online.

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

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

A. Sharifnia, Instability of the Join-The-Shortest-Queue and FCFS Policies in Queuing Systems and Their Stabilization, Operations Research 45 (2) (1997) 309–314.

K. J. Becker, D. P. Gaver, K. D. Glazebrook, P. A. Jacobs, S. Lawphongpanich, Allocation of tasks to specialized processors: A planning approach, European Journal of Operational Research 126 (1) (2000) 80–88.

R. Bellman, Dynamic programming, Princeton University Press, 1957.

R. A. Howard, Dynamic Probabilistic Systems, Volume II: Semi-Markov and Decision Processes, Wiley Interscience, 1971.

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

L. Kleinrock, Queueing Systems, Volume I: Theory, Wiley Interscience, 1975.

S. M. Ross, Introduction to Probability Models, Academic Press, 7th edn., 2000.

D. Gross, C. M. Harris, Fundamentals of Queueing Theory, John Wiley & Sons, third edn., 1998.

T. E. Phipps, Jr., Machine Repair as a Priority Waiting-Line Problem, Operations Research 4 (1) (1956) 76–86.

D. G. Down, H. C. Gromoll, A. L. Puha, Fluid Limits for Shortest Remaining Processing Time Queues, Mathematics of Operations Research 34 (4) (2009) 880–911.