2005

Optimal Scheduling in a Queue with Differentiated Impatient Users

Amy Csizmar Dalal
Carleton College

Scott Jordan
University of California, Irvine

Recommended Citation

A. Csizmar Dalal and S. Jordan, "Optimal Scheduling in a Queue with Differentiated Impatient Users," *Performance Evaluation*, vol. 59, no. 1, pp. 73-84. Available at: https://doi.org/10.1016/j.peva.2004.08.001. , Elsevier, Jan 2005. Accessed via Faculty Work. Computer Science. *Carleton Digital Commons*. https://digitalcommons.carleton.edu/cs_faculty/1
The definitive version is available at https://doi.org/10.1016/j.peva.2004.08.001

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

ELSEVIER

Performance Evaluation 59 (2005) 73–84

PERFORMANCE EVALUATION
An International Journal

www.elsevier.com/locate/peva

Optimal scheduling in a queue with differentiated impatient users☆

Amy Csizmar Dalal[a,*], Scott Jordan[b]

[a] Department of Mathematics and Computer Science, Carleton College, Northfield, MN 55057, USA
[b] Department of Electrical Engineering and Computer Science, University of California, Irvine, CA 92697, USA

Received 22 November 2003; received in revised form 9 June 2004

Available online 23 September 2004

Abstract

We consider a M/M/1 queue in which the average reward for servicing a job is an exponentially decaying function of the job's sojourn time. The maximum reward and mean service times of a job are i.i.d. and chosen from arbitrary distributions. The scheduler is assumed to know the maximum reward, service rate, and age of each job. We prove that the scheduling policy that maximizes average reward serves the customer with the highest product of potential reward and service rate.

Keywords: Queuing theory; Reward; Scheduling

1. Introduction

We are concerned here with optimal scheduling in queues with customers who have limits on their queueing or sojourn times. Such time limits have been previously used to model transmission of real-time packets over a packet-switched network [1–3], overload control in call processing systems [4,5], and call handoff in cellular networks [6]. We consider in this paper a queue with Poisson arrivals and exponential service times, in which the average reward for servicing a job is an exponentially decaying function of the job's sojourn time. The maximum rewards (defined as the reward that would be earned if a job's sojourn time is zero) of the jobs are i.i.d. and chosen from an arbitrary distribution. Similarly, the mean service times, or equivalently the service rates, of jobs are i.i.d. and chosen from an arbitrary distribution. The scheduler is assumed to know the maximum reward, service rate, and age of each job.

[*] Corresponding author. Tel.: +1 507 646 5632.
E-mail addresses: adalal@carleton.edu (A.C. Dalal); sjordan@uci.edu (S. Jordan)

0166-5316/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.peva.2004.08.001

Much prior work has considered queues with impatient users with identical maximum rewards and service rates. Early papers modeled such queues under a first in first out (FIFO) scheduling policy, and derived various performance measures [7]. Later papers have focused on characterizing the optimal scheduling policy. The literature on such systems splits into two categories, depending on whether the deadlines of individual customers are known by the server.

If the deadlines of individual customers are known by the server, it has been shown that shortest time to extinction with idling (STEI) is often optimal. In [2], Panwar et. al. consider a M/G/1 queue in which customers have deadlines on queueing time and will drop out of the queue if service is not started before the deadline (called “expired jobs dropped”). They prove that STEI minimizes the average number of dropped jobs among the class of non-preemptive scheduling policies. In [3], Bhattacharya and Emphremides consider a G/M/1 queue in which customers have deadlines on queueing time or on sojourn time with expired jobs dropped, and prove that STEI minimizes dropped jobs (in the sense of stochastic ordering) among the class of non-preemptive policies. In [8], Bhattacharya and Emphremides consider two queues sharing a single server, with geometric arrivals and service, in which customers have deadlines on sojourn times but will not drop out of the queue when the deadline passes. The server accumulates a penalty for tardy customers, which is linear in the tardiness. They prove that the scheduling policy that minimizes average and discounted penalty is preemptive STEI, with an optimal switch-over policy among the two queues.

If the deadlines of individual customers are not known by the server, then the scheduling policy must base its decision on the distribution of deadlines and the age of the customer, rather than the customer’s actual deadline. The literature on such queues typically assumes that expired jobs are not dropped, i.e. the server must serve all jobs. In [9], Doshi and Lipper consider a M/G/1 queue in which customers have deadlines on queueing time and expired jobs are not dropped. A reward is earned for each served job, but the reward depends on the customer’s queueing time. They prove that if the reward function is convex decreasing, then last in first out (LIFO) maximizes average reward among work-conserving non-preemptive scheduling policies. Similarly, they also show that if the reward function is concave, then the optimal policy is first in first out (FIFO). In [10], Kallmes et al. consider a G/G/1 queue in which customers have deadlines on sojourn times and expired jobs are not dropped. They prove that if the distribution function of deadlines is concave, then LIFO maximizes the probability that the sojourn time is less than the deadline among work-conserving, non-preemptive scheduling policies. They also show that in a M/G/1 queue under similar conditions, last in first out preemptive resume (LIFO-PR) is optimal among work-conserving scheduling policies.

Several other optimization metrics have been considered in the literature. In [4,11], customer rejection mechanisms are considered in conjunction with scheduling in order to maximize the number of customers served before their deadlines in finite queues. In [12–14], optimal scheduling policies in queues with constraints on average or maximum delays are analyzed.

In this paper, we consider a queue with differentiated impatient users in which deadlines are not known and expired jobs are not dropped. In addition to knowing the age of each customer, in the differentiated customer model the scheduler is also presumed to know the maximum reward and service rate of each customer. The literature on queues with impatient users in which deadlines are not known, outlined above, provides a starting point for our work. In addition, there is extensive literature on scheduling in *multiclass* queues *without* impatient users. A common form for optimal policies in such systems is to service the job (or queue) that maximizes some index, often the product of a reward and service rate (the $c\mu$ rule) [15–19].

Specifically, we consider a queue with Poisson arrivals and exponential service times, in which the average reward for servicing a job is an exponentially decaying function of the job's sojourn time. The maximum reward and mean service times of a job are i.i.d. and chosen from arbitrary distributions. The scheduler is assumed to know the maximum reward, service rate, and age of each job. We prove that the scheduling policy that maximizes average reward serves the customer with the highest product of *potential reward* and service rate, where the potential reward is defined as the reward function evaluated at the customer's current age. For a queue in which all customers have identical maximum rewards and service rates, such a greedy policy reduces to LIFO-PR. If we interpret the reward as the probability that a job is served before its deadline, then this result is consistent with previous results in the literature.

The rest of this paper is structured as follows. In Section 2, we introduce the model. In Section 3, we derive the optimal scheduling policy. In Section 4, we illustrate the behavior of the optimal policy versus simpler policies. Finally, in Section 5, we briefly compare various scheduling policies under overload conditions.

2. Model

We consider a single server queue with Poisson arrivals. The service time of job i is assumed to have an exponential distribution with rate μ_i, where μ_i are i.i.d. random variables with an arbitrary distribution. The service times of jobs are assumed to be independent. Once a job has entered the system, it does not leave until it completes service. We assume that swap times are negligible.

Let x_i denote the arrival time of job i at the queue and $D_{i,p}$ denote the departure time of job i under policy p. Then job i has a *sojourn time* under a policy p equal to $D_{i,p} - x_i$. When job i departs from the queue, we assume that the server earns an expected reward equal to $g_i(D_{i,p}) = C_i \mathrm{e}^{-c(D_{i,p}-x_i)}$, where C_i are i.i.d. random variables with an arbitrary distribution.

The queue can be analyzed by considering a single cycle consisting of a busy period and an idle period. By the Renewal Reward Theorem [20], the average reward per unit time earned by the server under policy p is

$$V_p = \frac{E\left[\sum_{i=1}^{N} g_i(D_{i,p})\right]}{EZ} \tag{2.1}$$

where N is the number of jobs served in the first busy period and Z is the length of the first cycle. An optimal server policy satisfies $V = \max_p V_p$.

Denote by S all scheduling policies that choose which job to serve (if any) based solely on the set of ages, maximum rewards, and service rates of jobs in the queue, namely $\{t - x_i, C_i, \mu_i\}$. The set S therefore includes preemptive policies, processor-sharing policies, and non work-conserving policies.

3. Optimal scheduling policy

Our derivation of an optimal scheduling policy proceeds by showing that an optimal policy can be found in a subset of S containing only work-conserving, non processor-sharing, Markov policies that switch jobs in service only when jobs arrive or depart the system. A characterization of an optimal policy is then identified within this smaller class.

Our first lemma excludes from the set of optimal policies those in which the server may remain idle when jobs are in the queue.

Lemma 3.1. *Any optimal scheduling policy in S is work-conserving.*

Outline of proof. The proof is by contradiction. Suppose there exists a policy p that is not work-conserving and that achieves a higher average reward than any work-conserving policy. We construct a modified policy p' by swapping an interval when (under p) the server is idle and the queue is not empty with a later interval when the server is busy. The key idea is the identification of the time intervals. The two time intervals are chosen as consecutive infinitesimal periods from the same busy period, intersecting when the server switches from idle to busy. It can be shown that under the modified policy there is a nonzero probability that a job completes at an earlier time, and therefore achieves a higher reward. This portion of the proof is similar to Lemma 3.2 and is omitted here. (The full proof can be found in [21].) It follows that p cannot achieve the maximum average reward. □

Our next lemma further restricts the set in which optimal policies may be found.

Lemma 3.2. *Any optimal scheduling policy in S switches between jobs in service only upon an arrival to or departure from the queue.*

Proof. The proof is by contradiction. Suppose there exists an optimal policy p which switches between jobs in service at some time other than a departure time or arrival time over some interval of a sample path in the first busy period. At some time interval $[a_1 - \mathrm{d}l, a_1)$ during this busy period, the server works on job k. At time a_1, the server switches to job $k + 1$, and processes that job for at least the interval $[a_1, a_1 + \mathrm{d}l)$. We define a_1 to be a time at which no arrivals to or departures from the system occur. At some future time a_2, the server switches back to serving job k. We make no assumptions as to the jobs served in the interval $[a_1 + \mathrm{d}l, a_2)$. Under this policy, job k cannot depart the system earlier than time $a_2 + \mathrm{d}l$, and likewise job $k + 1$ cannot depart the system earlier than time $a_1 + \mathrm{d}l$. The server can only earn a reward once a job departs the system; thus, under policy p, the server can collect a reward for jobs k and $k + 1$ no earlier than at times $a_2 + \mathrm{d}l$ and $a_1 + \mathrm{d}l$, respectively.

We establish a contradiction by demonstrating that there exists at least one modified policy with different potential completion times for requests k and $k + 1$ that earns the server a higher expected reward than that earned under policy p. This modified policy, p', is identical to policy p in the intervals $[0, a_1 - \mathrm{d}l)$ and $[a_1 + \mathrm{d}l, \infty)$, and swaps the order of jobs k and $k + 1$ in $[a_1 - \mathrm{d}l, a_1 + \mathrm{d}l)$.

Note that the earliest departure time for job k under policy p' is the same as it is under policy p. The earliest departure time for job $k + 1$ under p' is a_1, while under under p it is $a_1 + \mathrm{d}l$. The difference in expected reward earned under the two policies is thus simply the expected reward earned by policy p' if job $k + 1$ departs the system at time a_1, which occurs with probability $\mu_{k+1}\mathrm{d}l$, minus the expected reward earned by policy p if job $k + 1$ departs the system at time $a_1 + \mathrm{d}l$, also with probability $\mu_{k+1}\mathrm{d}l$:

$$\begin{aligned} E\left[\sum_{i=0}^{N} g_i(D_{i,p'})\right] - E\left[\sum_{i=0}^{N} g_i(D_{i,p})\right] &= E[g_{k+1}(a_1) + g_k(a_1 + \mathrm{d}l)] - E[g_k(a_1) + g_{k+1}(a_1 + \mathrm{d}l)] \\ &= \mu_{k+1}\mathrm{d}l[g_{k+1}(a_1)] - \mu_{k+1}\mathrm{d}l[g_{k+1}(a_1 + \mathrm{d}l)] \\ &= \mu_{k+1}\mathrm{d}l\, C_{k+1}\mathrm{e}^{-c(a_1 - x_{k+1})}(1 - \mathrm{e}^{-c\,\mathrm{d}l}) \end{aligned}$$

which is a positive quantity.

We have shown that with some nonzero probability, $V_{p'} > V_p$, and therefore p cannot be an optimal policy. □

Our next lemma states that an optimal policy can always be found among the class of policies that work on at most one job at a time, which we denote as *non processor-sharing policies*.

Lemma 3.3. *An optimal policy in S can be found in the class of non processor-sharing policies.*

Proof. Suppose there exists a generalized processor-sharing policy PS that generates a higher reward than the best non processor-sharing policy NPS. Then under policy PS, over some interval [*a*,*b*], the server splits its resources between at least two jobs. We consider here the case in which PS splits its service rate between two jobs, with a constant proportion to each job, until the first job departs, and then devotes all of its service capacity to the second job until it departs. The general case follows in a similar fashion.

Suppose the server devotes a proportion q of its service rate μ to job 1 and the remainder to job 2. Denote by τ_i the time job i has spent in the queue prior to time a, and denote by $T_{i,\mathrm{PS}}$ the remaining time job i will spent in the system after time a until the job's departure. The expected total reward that will be gained from jobs 1 and 2 under policy PS is thus:

$$\mathrm{ER}_{\mathrm{PS}} = E[C_1 \mathrm{e}^{-c(\tau_1+T_{1,\mathrm{PS}})} + C_2 \mathrm{e}^{-c(\tau_2+T_{2,\mathrm{PS}})}].$$

Let J1 denote the event that job 1 completes service before job 2, under policy PS, and J2 denote the event that job 2 completes service before job 1. Conditioning on J1 gives

$$\mathrm{ER}_{\mathrm{PS}} = P(\mathrm{J1})E[R_{\mathrm{PS}}|\mathrm{J1}] + P(\mathrm{J2})E[R_{\mathrm{PS}}|\mathrm{J2}]. \tag{3.1}$$

Using classic results from the multiplexing of Poisson processes, $P(\mathrm{J1}) = q\mu_1/\mu$ and $P(\mathrm{J2}) = 1 - P(\mathrm{J1})$, where $\mu \equiv q\mu_1 + (1-q)\mu_2$. Furthermore, $(T_{1,\mathrm{PS}}|\mathrm{J1}) \sim \exp(\mu)$, and $(T_{2,\mathrm{PS}}|\mathrm{J1}) = T_{1,\mathrm{PS}} + Z$, where $Z \sim \exp(\mu_2)$.

Now consider the best non processor-sharing policy NPS. This policy must either serve job 1 to completion and then serve job 2 (denoted NPS1), or vice-versa (denoted NPS2). Under NPS1, the remaining sojourn times are $T_{1,\mathrm{NPS1}} \sim \exp(\mu_1)$ and $T_{2,\mathrm{NPS1}} = T_{1,\mathrm{NPS1}} + Z$, where $Z \sim \exp(\mu_2)$.

If $\mu_1 = \mu_2 = \mu$, then it follows that $T_{1,\mathrm{NPS1}} \sim (T_{1,\mathrm{PS}}|\mathrm{J1})$ and $T_{2,\mathrm{NPS1}} \sim (T_{2,\mathrm{PS}}|\mathrm{J1})$. Therefore, the expected total reward that will be gained from jobs 1 and 2 under policy NPS1, denoted $\mathrm{ER}_{\mathrm{NPS1}}$ is equal to $E[R_{\mathrm{PS}}|\mathrm{J1}]$. Similarly, $\mathrm{ER}_{\mathrm{NPS2}} = E[R_{\mathrm{PS}}|\mathrm{J2}]$. Note that both $\mathrm{ER}_{\mathrm{NPS1}}$ and $\mathrm{ER}_{\mathrm{NPS2}}$ are independent of q. Eq. (3.1) can thus be written as $\mathrm{ER}_{\mathrm{PS}} = q\mathrm{ER}_{\mathrm{NPS1}} + (1-q)\mathrm{ER}_{\mathrm{NPS2}}$. Since the expected revenue under policy PS is a linear weighted sum of the expected revenues under policies NPS1 and NPS2, it follows that $\max(\mathrm{ER}_{\mathrm{NPS1}}, \mathrm{ER}_{\mathrm{NPS2}}) \geq \mathrm{ER}_{\mathrm{PS}}$.

If, however, $\mu_1 \neq \mu_2$, then the distribution of $T_{1,\mathrm{NPS1}}$ is different than that of $(T_{1,\mathrm{PS}}|\mathrm{J1})$. We can explicitly evaluate the expected reward from serving jobs 1 and 2 under each policy:

$$\mathrm{ER}_{\mathrm{NPS1}} = g_1(a)\frac{\mu_1}{c+\mu_1} + g_2(a)\frac{\mu_1}{c+\mu_1}\frac{\mu_2}{c+\mu_2}, \quad \mathrm{ER}_{\mathrm{NPS1}} = g_1(a)\frac{\mu_1}{c+\mu_1}\frac{\mu_2}{c+\mu_2} + g_2(a)\frac{\mu_2}{c+\mu_2},$$

$$\mathrm{ER}_{\mathrm{PS}} = P(\mathrm{J1})\left[g_1(a)\frac{\mu}{c+\mu} + g_2(a)\frac{\mu}{c+\mu}\frac{\mu_2}{c+\mu_2}\right] + P(J2)\left[g_1(a)\frac{\mu}{c+\mu}\frac{\mu_1}{c+\mu_1} + g_2(a)\frac{\mu_2}{c+\mu_2}\right]$$

It can be shown after simplification of the above expression that $\mathrm{ER}_{\mathrm{PS}} > \mathrm{ER}_{\mathrm{NPS1}}$ iff $g_1(a)\mu_1 < g_2(a)\mu_2$, and that $\mathrm{ER}_{\mathrm{PS}} > \mathrm{ER}_{\mathrm{NPS2}}$ iff $g_1(a)\mu_1 > g_2(a)\mu_2$. It follows that $\max(\mathrm{ER}_{\mathrm{NPS1}}, \mathrm{ER}_{\mathrm{NPS2}}) \geq \mathrm{ER}_{\mathrm{PS}}$. □

The scheduler is assumed to know the set of ages, maximum rewards, and service rates of jobs in the queue, namely $\{t - x_i, C_i, \mu_i\}$. Since interarrival times and service times are memoryless, it suffices to keep track solely of the set of potential rewards (under policy p) and service rates, $\{g_i(t), \mu_i\}$.

Lemma 3.4. *An optimal policy in S can be found in the class of Markov policies.*

Outline of proof. Since interarrival and service times are independent and exponentially distributed, knowledge of past arrival times, past service times, or the current service time so far does not help to predict future arrivals or future service times. In addition, rewards are solely a function of sojourn times, and therefore given knowledge of the current potential reward for job i, $g_i(t)$, it is not useful to know past history of job arrivals to estimate future rewards. Finally, knowledge of past rewards or service times (or even the distribution of C_i or of μ_i) is not useful, since preemption is allowed. □

We can now state the form of the optimal policy:

Definition 3.5. Denote the job with the highest product of current potential reward and service rate as $i^*(t)$, namely $i^*(t) = \arg\max_i g_i(t)\mu_i$. The greedy scheduling policy p^* serves at time t job $i^*(t)$.

Theorem 3.6. *The greedy policy p^* maximizes the average reward among all scheduling policies in the class S.*

Proof. By Lemmas 3.1–3.4, it suffices to consider scheduling policies in a smaller class S', defined as policies in S that are work-conserving, non processor-sharing, Markov and only switch between jobs at jobs' arrivals or departures. Assume that there exists a policy $p \in S'$ that achieves a higher average reward than does p^*. Then, infinitely often, p consecutively serves two jobs for which product of potential reward and service rate of the first job is less than that of the second. Consider one such time, t_0, and one such pair of jobs, i and j. Denote the remaining sojourn times, after time t_0, for jobs i and j as $T_{i,p}$ and $T_{j,p}$. The revenue from these two jobs under policy p is

$$R_p \equiv g_i(t_0)\mathrm{e}^{-cT_{i,p}} + g_j(t_0)\mathrm{e}^{-cT_{j,p}}.$$

Now consider an alternate scheduling policy p' that interchanges the order of jobs i and j. The revenue from these two jobs under policy p' is similarly:

$$R_{p'} \equiv g_j(t_0)\mathrm{e}^{-cT_{j,p'}} + g_i(t_0)\mathrm{e}^{-cT_{i,p'}}.$$

Since $T_{i,p} + T_{j,p} \sim T_{i,p'} + T_{j,p'}$, the average reward during the remainder of the busy cycle and the length of the busy cycle are identical under p and p'. The difference in average revenue in this busy cycle can thus be explicitly expressed as:

$$\begin{aligned}\mathrm{ER}_{p'} - \mathrm{ER}_p &= \frac{g_j(t_0)\mu_j}{c+\mu_j}\left(1 - \frac{\mu_i}{c+\mu_i}\right) - \frac{g_i(t_0)\mu_i}{c+\mu_i}\left(1 - \frac{\mu_j}{c+\mu_j}\right)\\ &= \frac{c}{(c+\mu_i)(c+\mu_j)}\left(g_j(t_0)\mu_j - g_i(t_0)\mu_i\right). \qquad (3.2)\end{aligned}$$

Each term is positive, so $\mathrm{ER}_{p'} > \mathrm{ER}_p$. Since such interchanges can occur infinitely often, $V_{p'} > V_p$ and hence p cannot be optimal. □

We note that in the case in which $C_i = C$, $\forall i$ and $\mu_i = \mu$, $\forall i$, the optimal policy reduces to LIFO-PR.

4. Simulation results

To illustrate the differences in average reward under various circumstances, we present a set of simulation results. Using the BONeS simulation software, we implemented the scheduler under four policies: the greedy policy, LIFO-PR, FIFO, and PS. For each policy, we measure the average reward per unit time and the variance of the sojourn time. Jobs arrive at the queue as a Poisson process with rate $\lambda = 50$. We vary the service rate μ so that the load λ/μ varies from zero to near one. (Loads greater than one are considered in Section 5.) The simulation times were chosen to include many busy cycles in each case, so that the confidence intervals for each plotted quantity are relatively small.

We first consider a system in which $C_i = 1$, $\forall i$ and $\mu_i = \mu$, $\forall i$. The greedy policy therefore reduces to LIFO-PR. In Fig. 1a we plot the average reward divided by λ versus the load. At low loads, the sojourn time is only slightly greater than the service time, and there is not much difference among the service policies in terms of the average reward. As the load increases, the differences in average reward become more pronounced, and for loads close to one the average reward for FIFO and PS drop much more quickly than the average reward for LIFO-PR.

The increase in the difference in average reward between various policies can be explained in part by the high variances of the LIFO-PR sojourn times as compared to the sojourn time variances under FIFO and PS. The measured sojourn time variances for each policy are plotted in Fig. 1b. It is known that among scheduling policies in S, FIFO attains the smallest sojourn time variance and LIFO-PR attains the largest [22]. Since the reward function is decreasing convex, an increase in sojourn time variance results in an increase in average reward. In particular, at high loads, the average reward under LIFO-PR will be dominated by jobs with small service times which earn high rewards, while the average reward under FIFO will drop quickly due to long average sojourn times.

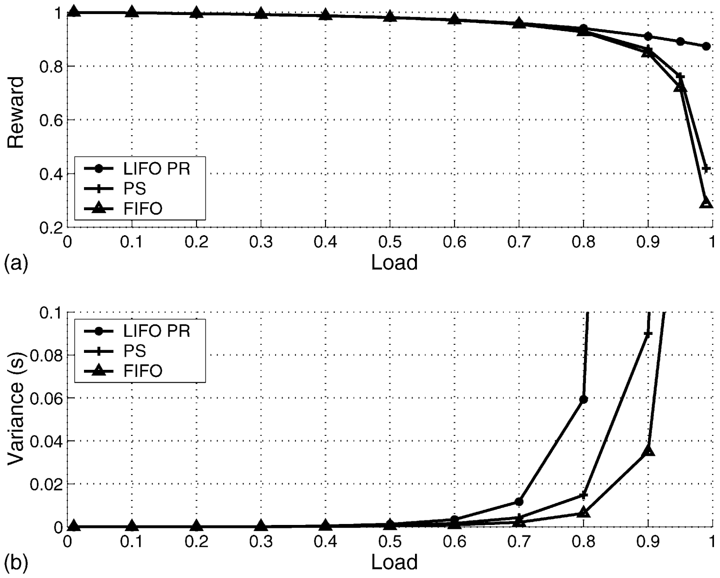

Fig. 1. Equal maximum reward model: (a) normalized average reward as a function of load and (b) sojourn time variance as a function of load.

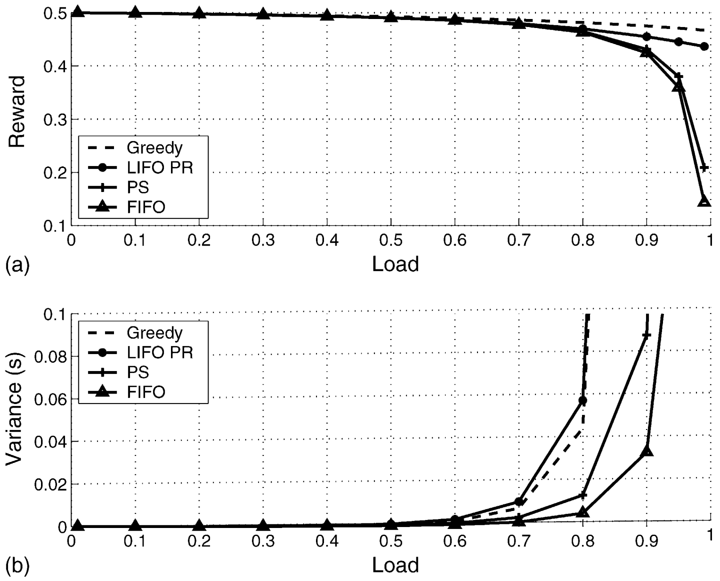

Fig. 2. Differentiated maximum reward model: (a) normalized average reward as a function of load and (b) sojourn time variance as a function of load.

We now consider a system in which the maximum reward C_i are i.i.d. and drawn from a uniform distribution on [0, 1), and $\mu_i = \mu,\ \forall i$. In Fig. 2a we again plot the average reward divided by λ versus the load. (The maximum average reward is now $\text{EC}_i = 0.5$.) Since FIFO, PS, and LIFO-PR serve jobs independent of C_i, the average rewards earned by these scheduling policies are identical to those in Fig. 1a, scaled by 0.5. The greedy policy, however, earns a slightly larger average revenue than LIFO-PR. The corresponding sojourn time variances for each policy are plotted in Fig. 2b. The greedy policy results in small increases in sojourn time variance above that achieved by LIFO-PR when the offered load is very high (above 0.95, not shown on the plot). In addition, the differences in average revenue will increase with the variance of C_i.

Finally, we consider a system in which the maximum rewards are set to $C_i = 1,\ \forall i$, and the service rates are independently drawn from a set $\{K, K/7, K/50\}$ with probabilities $\{0.2, 0.7, 0.1\}$, with K chosen so that the load λ/μ varies from zero to near one. The service time is thus drawn from a hyperexponential distribution, and the queue is M/H/1. In Fig. 3a we plot the average reward divided by λ versus the load, and in Fig. 3b we plot the corresponding sojourn time variances.

The difference in average reward between PS and FIFO is greater in the differentiated service time simulation than it was in the differentiated reward simulation, due to a significant decrease in FIFO's average reward. The difference in average reward between the greedy policy and LIFO-PR is also smaller. Even though the greedy policy uses additional information (service rate) in choosing which request to service, the fact that the majority of the rates are chosen from the same distribution means that this policy will behave similarly to the LIFO-PR policy, which solely considers reward and not service rate.

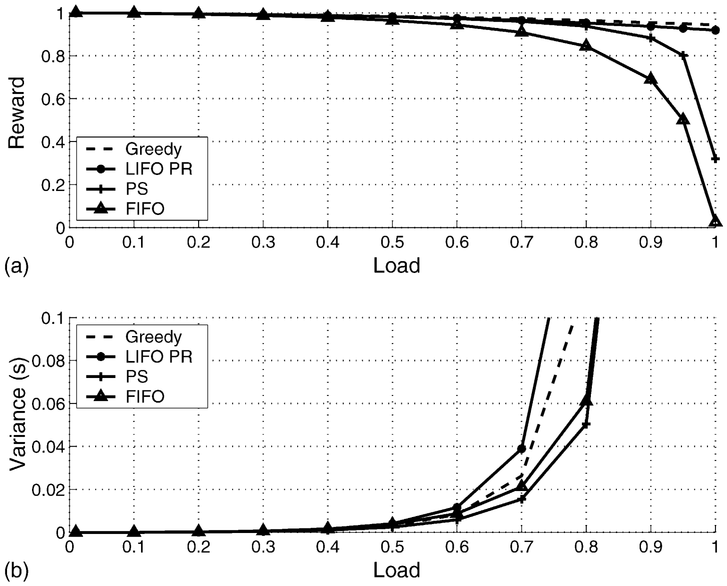

Fig. 3. Differentiated service rate model: (a) normalized average reward as a function of load and (b) sojourn time variance as a function of load.

5. Optimal policy under overload conditions

The simulation results presented in Section 4 demonstrate that at low system loads, FIFO, LIFO-PR, and PS generate almost the same average reward as the optimal policy. The differences in average reward among the policies increase as the system load increases towards one. In this section, we consider the performance of these policies under overload conditions, when the load exceeds one.

At loads above one, the queue becomes unstable and the average queue length and the average sojourn time will grow without bound. As a result, the average reward earned under FIFO will be zero, since the oldest customer's sojourn times will tend to infinity. Similarly, the average reward earned under PS will be zero. In contrast, the average rewards under LIFO-PR and the greedy policy will remain positive, since they serve recently arriving customers.

We can analyze this degradation of FIFO and PS by considering the average revenue per unit time over a finite period of time, starting with an empty queue. We consider the queue with homogeneous users, i.e. $C_i = 1$, $\forall i$ and $\mu_i = \mu$, $\forall i$. In the simulation results presented here, $\mu = 100$. In Fig. 4, we plot the normalized average reward per unit time earned by FIFO, PS, and LIFO-PR (the optimal policy) at loads of 1.0, 1.01, and 1.1, for time windows ranging from 10 s to 30 min.

The average reward under FIFO and PS must fall to zero as the length of the time window increases. Indeed, they also drop off more quickly as the load increases. In contrast, the average reward under LIFO-PR will converge to a positive value as the time window increases.

In Fig. 5, we plot the normalized average reward per unit time under LIFO-PR for loads ranging from one to two and for time windows ranging from 30 s to 30 min. We observe that the normalized average reward per unit time increases as the load increases.

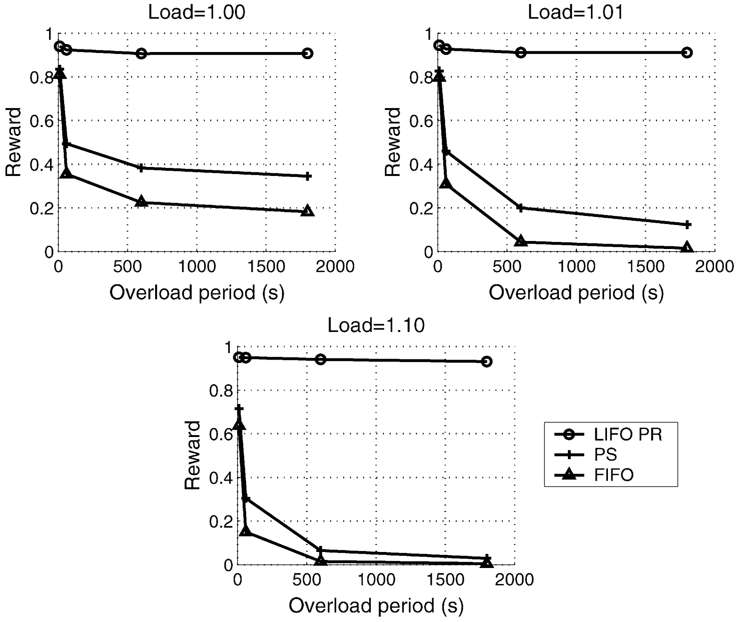

Fig. 4. Average reward earned by LIFO-PR and other service orders under overload conditions.

Indeed, we can analyze the limiting average revenue per unit time earned under LIFO-PR as the load tends to infinity. At arbitrarily high loads, LIFO-PR will almost always be serving the newest job. It will complete service for this job with probability $\mu/\lambda + \mu$. Conditioned on service completion, the sojourn time has an exponentially distributed distribution with rate $\lambda + \mu$, so the expected reward

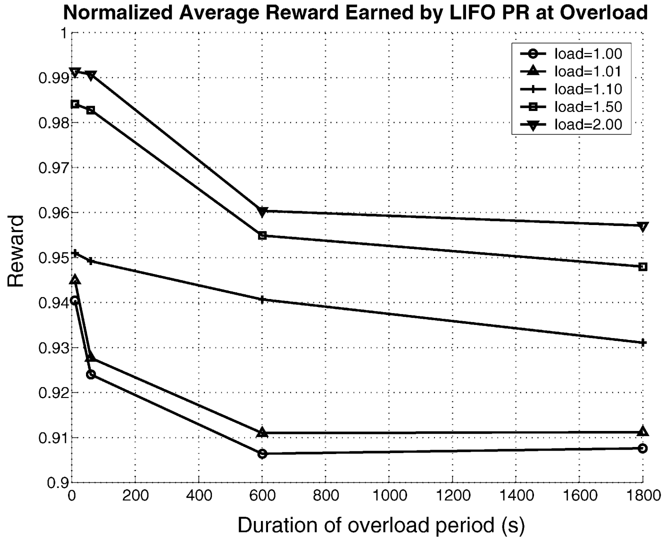

Fig. 5. Average reward at various stages of overload, LIFO-PR only.

per service completion will be $\mathrm{EC}_i(\lambda+\mu)/(c+\lambda+\mu)$. The expected reward per arrival will thus be $(\mu/(\lambda+\mu))\mathrm{EC}_i((\lambda+\mu)/(c+\lambda+\mu)) = \mathrm{EC}_i(\mu/(c+\lambda+\mu))$, and the average reward per unit time will thus be $\mathrm{EC}_i(\lambda\mu/(c+\lambda+\mu))$. As the load tends to infinity, the average reward per unit time tends to $\mathrm{EC}_i\mu$.

6. Conclusions

This paper discusses optimal scheduling policies for single-server queues with Poisson arrivals and exponential service times, where the average reward for servicing a job is an exponentially decaying function of the job's sojourn time. The scheduling policy that maximizes average reward serves the customer with the highest product of *potential reward* and service rate, where potential reward is defined as the reward function evaluated at the customer's current age. When maximum rewards are equal and service times are drawn from a single exponential distribution, this policy defaults to LIFO-PR. We have demonstrated via simulation the difference in average reward generated by the scheduler per unit time under the optimal scheduling policy and other policies such as FIFO and processor-sharing. As the offered load approaches one, the differences in average reward become more pronounced, and the optimal policy significantly outperforms both FIFO and processor-sharing policies. Finally, we have shown via simulation that the optimal policy continues to generate a high average reward even when the system is overloaded for a significant time period.

Acknowledgment

This work was supported by NSF under grant 9523580 and by DARPA under grant N66001-00-8935.

References

[1] J. Kurose, M. Schwartz, Y. Yemini, Controlling window protocols for time-constrained communication in multiple access networks, IEEE Trans. Commun. 36 (1) (1988) 41–49.

[2] S.S. Panwar, D. Towsley, J.K. Wolf, Optimal scheduling policies for a class of queues with customer deadlines to the beginning of service, J. ACM 35 (4) (1988) 832–844.

[3] P. Bhattacharya, A. Ephremides, Optimal scheduling with strict deadlines, IEEE Trans. Autom. Control 34 (7) (1989) 721–728.

[4] B. Doshi, H. Heffes, Overload performance of several processor queueing disciplines for the M/M/1 queue, IEEE Trans. Commun. 34 (1986) 538–546.

[5] R.R. Pillai, A distributed overload control algorithm for delay-bounded call setup, IEEE/ACM Trans. Networking 9 (6) (2001) 780–789.

[6] D. McMillan, Delay analysis of a cellular mobile priority queueing system, IEEE/ACM Trans. Networking 3 (3) (1995) 310–319.

[7] F. Baccelli, P. Boyer, G. Hebuterne, Single-server queues with impatient customers, Adv. Appl. Probability 16 (1984) 887–905.

[8] P. Bhattacharya, A. Ephremides, Optimal allocation of a server between two queues with due times, IEEE Trans. Autom. Control 36 (1991) 1417–1423.

[9] B. Doshi, E. Lipper, Comparisons of service disciplines in a queueing system with delay dependent customer behaviour, in: R. Disney, T. Ott (Eds.), Applied Probability—Computer Science: the Interface, vol. II, Birkhauser, Cambridge, MA, 1982, pp. 269–301.

[10] M. Kallmes, D. Towsley, C. Cassandras, Optimality of the last-in-first-out (LIFO) service discipline in queueing systems with real-time constraints, in: Proceedings of the 28th Conference on Decision and Control, IEEE, Tampa, FL, 1989, pp. 1073–1074.

[11] Z.-X. Zhao, S. Panwar, D. Towsley, Queueing performance with impatient customers, in: Proceedings of the INFOCOM, IEEE, 1991, pp. 400–409.

[12] K.W. Ross, B. Chen, Optimal scheduling of interactive and noninteractive traffic in telecommunication systems, IEEE Trans. Autom. Control 33 (3) (1988) 261–267.

[13] C. Cassandras, V. Julka, Optimal scheduling in systems with delay-sensitive traffic, in: Proceedings of the 32nd IEEE Conference on Decision and Control, San Antonio, TX, 1993, pp. 1680–1685.

[14] M. Hamdaoui, P. Ramanathan, A dynamic priority assignment technique for streams with (m,k)-firm deadlines, IEEE Trans. Comput. 44 (1995) 1443–1451.

[15] P. Varaiya, J. Walrand, C. Buyukkoc, Extensions of the multiarmed bandit problem: the discounted case, IEEE Trans. Autom. Control 30 (5) (1985) 426–439.

[16] V. Anantharam, P. Varaiya, J. Walrand, Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays, IEEE Trans. Autom. Control 32 (11) (1987) 968–982.

[17] R. Agrawal, M. Hegde, D. Teneketzis, Asymptotically efficient adaptive allocation rules for the multiarmed bandit problem with switching cost, IEEE Trans. Autom. Control 33 (10) (1988) 899–906.

[18] E. Altman, A. Shwartz, Optimal priority assignment: a time sharing approach, IEEE Trans. Autom. Control 34 (1989) 1098–1102.

[19] D. Pandelis, D. Teneketzis, On the optimality of the gittins index rule in multi-armed bandits with multiple plays, in: Proceedings of the 34th IEEE Conference on Decision and Control, New Orleans, LA, 1995, pp. 1408–1414.

[20] S.M. Ross, Introduction to Stochastic Dynamic Programming, Academic Press, New York, 1983.

[21] A.C. Dalal, Characterization of user and server behavior in web-based networks, Ph.D. thesis, Northwestern University, December 1999.

[22] J.G. Shanthikumar, U. Sumita, Convex ordering of sojourn times in single-server queues: extremal properties of FIFO and LIFO service disciplines, J. Adv. Probability 24 (1987) 737–748.

Amy Csizmar Dalal is an Assistant Professor of Computer Science at Carleton College in Northfield, Minnesota. From 2000 to 2003, she was a postdoctoral researcher at Hewlett-Packard Laboratories in Palo Alto, California. She received her BS degree in Electrical Engineering from the University of Notre Dame in 1994 and her MS and PhD degrees in Electrical Engineering from Northwestern University in 1997 and 1999, respectively. Her research interests include streaming media and performance measurement and analysis of computer networks. She is a member of IEEE.

Scott Jordan received the AB degree in Applied Mathematics and the BS, MS, and PhD degrees in Electrical Engineering and Computer Science from the University of California, Berkeley, in 1985, 1985, 1987, and 1990, respectively. From 1990 until 1999, he served as a faculty member at Northwestern University. Since 1999, he has served as an Associate Professor in the Department of Electrical Engineering and Computer Science at the University of California, Irvine. Professor Jordan's teaching and research interests currently include pricing and differentiated services in the Internet, and resource allocation in wireless multimedia networks.