

# Admission Control and Routing to Parallel Queues with Delayed Information via Marginal Productivity Indices

Peter Jacko  
peter.jacko@uc3m.es

José Niño-Mora  
jnimora@alum.mit.edu

Department of Statistics  
Universidad Carlos III de Madrid  
Avda. Universidad 30  
28911 Leganés (Madrid), Spain

## ABSTRACT

This paper addresses the problem of designing and computing a tractable index policy for dynamic job admission control and/or routing in a discrete time Markovian model of parallel loss queues with one-period delayed state observation, which comes close to optimizing an infinite-horizon discounted or average performance objective involving linear holding costs and rejection costs. Instead of devising some ad hoc indices, we deploy a unifying fundamental design principle for design of priority index policies in dynamic resource allocation problems of multiarmed restless bandit type, based on decoupling the problem into subproblems and defining an appropriate marginal productivity index (MPI) for each subproblem. In the model of concern, such subproblems represent admission control problems to a single queue with one-period feedback delay, for which the structure of optimal policies has been characterized in previous work as being of bi-threshold type, yet without giving an algorithm to compute the optimal thresholds. We deploy in such subproblems theoretical and algorithmic results on restless bandit indexation, which yields a fast algorithm that computes the MPI for a subproblem with a buffer size of  $n$  performing only  $O(n)$  arithmetic operations. Such MPI values can be used both to immediately obtain the optimal thresholds for the subproblem, and to design an index policy for the admission control and/or routing problem in the multi-queue system. The results readily extend to models with infinite buffer space.

## Categories and Subject Descriptors

G.3 [Probability and Statistics]: Markov processes, Queueing theory;

C.2.2 [Computer-Communication Networks]: Network Protocols—Routing protocols;

F.2.1 [Analysis of Algorithms and Problem Complexity]: Numerical Algorithms and Problems

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ValueTools 2008 October 21–23, 2008, Athens, GREECE  
Copyright 2008 ICST ISBN # 978-963-9799-31-8.

## General Terms

Algorithms, Performance, Design, Theory

## Keywords

admission control, routing, parallel queues, delayed information, index policies, restless bandits, marginal productivity index

## 1. INTRODUCTION

This paper addresses the problem of designing and computing a tractable heuristic policy for dynamic job admission control and routing in a discrete time Markovian model of parallel loss queues with one-period delayed state observation, which comes close to optimizing an infinite-horizon discounted or average performance objective involving linear holding costs and rejection costs. Two versions of the model are considered, depending on whether the admission control capability is enabled or not. The results will also apply to the corresponding models with infinite buffer space.

Jobs arrive to the system as a Bernoulli stream with probability  $0 < \lambda < 1$  of arrival per period. Upon a job's arrival, a central controller must decide: (i) in the case that admission control is enabled, whether to admit the job or to reject it; and, if admitted, (ii) to which of  $K$  queues in parallel to route the job for service. Queue  $k \in \mathbb{K} \triangleq \{1, \dots, K\}$  is endowed with a finite buffer with room for holding  $n_k \geq 1$  jobs waiting or in service, and has a single geometric server, which serves jobs in FCFS order and completes the service of a job in a period with probability  $0 < \mu_k < 1$ . Arrival and service processes are mutually independent.

Denote by  $X_k(t)$  the *state* of queue  $k$  at the start of period  $t$ , given by the number of jobs it holds waiting or in service, and by  $a_k(t) \in \{0, 1\}$  the *action* indicator that takes the value 1 when a job arriving at time  $t$  is *not to be routed to queue  $k$* . We assume that at the start of period  $t$  the controller does not know the current state, but has information on previous states and actions, knowing in particular  $X_k(t-1)$  and  $a_k(t-1)$  for each queue  $k$ .

Action choice is based on adoption of an admission control and routing policy (if admission control is enabled), or just a routing policy (if it is not), denoted by  $\pi$ . This is to be chosen from the corresponding class  $\Pi$  of (possibly randomized) policies that use previous state and action information.

The system incurs *holding costs*, at rate  $c_k(i_k) > 0$  per period that  $i_k$  jobs are held in queue  $k$ ; and *loss costs*, at rate  $\nu$  per rejected job, due either to active rejection or to

forced rejection (blocking) when an arrival finds all buffers full.

We will find it convenient to formulate the total cost incurred in a period in which the joint system state is  $\mathbf{j} = (j_k)$  and action  $\mathbf{a} = (a_k)$  prevails as a constant plus a term that is separably additive across queues, using the identity

$$\sum_{k \in \mathbb{K}} c_k j_k + \nu \lambda \left\{ 1 - \sum_{k \in \mathbb{K}} (1 - a_k) \right\} = -(K-1)\lambda\nu + \sum_{k \in \mathbb{K}} (c_k j_k + \nu \lambda a_k). \quad (1)$$

Note that the term  $1 - \sum_{k \in \mathbb{K}} (1 - a_k)$  in (1) takes the value 1 if an arrival is to be rejected ( $a_k = 1$  for every  $k$ ), and takes the value 0 otherwise ( $a_k = 0$  for exactly one  $k$ ).

The operation of such a system raises the following performance optimization problems: (i) find a policy minimizing the expected total discounted value of costs accrued,

$$\min_{\pi \in \Pi} \mathbb{E}_{(\mathbf{i}^-, \mathbf{a}^-)}^{\pi} \left[ \sum_{t=0}^{\infty} \sum_{k \in \mathbb{K}} \left\{ c_k (X_k(t)) + \nu \lambda a_k(t) \right\} \beta^t \right], \quad (2)$$

where  $0 < \beta < 1$  is the discount factor and  $\mathbb{E}_{(\mathbf{i}, \mathbf{a})}^{\pi}[\cdot]$  denotes expectation under policy  $\pi$  conditioned on the initial previous joint state and action vectors being equal to  $\mathbf{X}(-1) = \mathbf{i}^- = (i_k^-)$  and  $\mathbf{a}(-1) = \mathbf{a}^- = (a_k^-)$ ; and (ii) find a policy minimizing the long-run average cost rate per period,

$$\min_{\pi \in \Pi} \limsup_{T \rightarrow \infty} \frac{1}{T} \mathbb{E}_{(\mathbf{i}^-, \mathbf{a}^-)}^{\pi} \left[ \sum_{t=0}^T \sum_{k \in \mathbb{K}} \left\{ c_k (X_k(t)) + \nu \lambda a_k(t) \right\} \right]. \quad (3)$$

Note that in the objectives in (2)–(3) we have disregarded the additive constant  $-(K-1)\lambda\nu$  in (1).

Such problems are relevant in a variety of application domains, most notably in the operation of packet-switched communication networks and distributed computer systems, where there are nonnegligible propagation delays on system state information, which force the controller to take decisions based on stale information.

As for our considering joint admission control and routing problems, instead of restricting attention to the conventional pure-routing case, the motivation is that it allows the system designer to take into account the tradeoff between rejection and holding costs. The key insight is that, when the system is heavily congested, denying access to further arrivals until the congestion is sufficiently reduced can substantially decrease holding costs at a relatively small expense in terms of increased rejection costs.

The above problems are naturally formulated as *partially observed Markov decision processes* (POMDPs), which in turn are readily reformulated as conventional *Markov decision processes* (MDPs) by redefining the state of each queue  $k$  as the *augmented state*  $Y_k(t) \triangleq (a_k(t-1), X_k(t-1))$ , and using the joint state and action process  $\mathbf{Y}(t) = (Y_k(t))$  and  $\mathbf{a}(t) = (a_k(t))$ . Computation of optimal policies for the resultant multidimensional MDPs by solving the associated *dynamic programming* (DP) equations is, however, hindered by the curse of dimensionality in large-scale models. We will thus focus attention on the more realistic and practical goals of designing and computing well-grounded heuristic policies that are readily implementable.

Since in such problems the controller must dynamically assess the relative values of alternative rejection and routing actions, it is intuitively appealing to do so based on an *index policy*. In the present model, such policies are based on

attaching to each queue  $k$  a numeric *index*  $\nu_k(a_k^-, i_k^-)$ , which can be thought of as a *measure of undesirability for routing a job to queue  $k$* , given as a function of the queue's augmented state, which we denote by  $(a_k^-, i_k^-)$  to emphasize that it refers to the observed action-state pair at the previous period. The resultant index policy prescribes the following actions, when at time  $t$  the augmented state of each queue  $k$  is known to be  $Y_k(t) = (a_k^-, i_k^-)$ . In the problem version with admission control capability, the policy prescribes to admit an arriving job if  $\nu > \nu_k(a_k^-, i_k^-)$  for at least one queue  $k$ , i.e., if the cost of rejecting the job exceeds the undesirability of routing it to some queue; otherwise, the job is rejected. If accepted, the job is routed to a queue of *lowest index* value, breaking ties arbitrarily, among those queues  $k$  for which  $\nu > \nu_k(a_k^-, i_k^-)$ . Note that such policies may well prescribe to accept and route a job to a queue that is actually full, unbeknownst to the controller, in which case the job will be blocked and hence rejected.

For the special case of the pure-routing problem in which there are two *symmetric* infinite-buffer queues ( $\mu_k \equiv \mu$  and  $c_k \equiv 1$ ), it is shown in [6] that an index policy is optimal: the *Join the Shortest Expected Queue* (JSEQ) rule, which routes an incoming job to a queue  $k$  of lowest index

$$\nu_k^{\text{JSEQ}}(a_k^-, i_k^-) \triangleq \begin{cases} i_k^- - \mu & \text{if } a_k^- = 1, i_k^- \geq 1 \\ 0 & \text{if } a_k^- = 1, i_k^- = 0 \\ i_k^- + \lambda - \mu & \text{if } a_k^- = 0, i_k^- \geq 1 \\ \lambda(1 - \mu) & \text{if } a_k^- = 0, i_k^- = 0, \end{cases}$$

where the index represents the expected value of  $X_k(t)$  conditioned on  $(a_k(t-1), X_k(t-1)) = (a_k^-, i_k^-)$ . Such a result partially extends to queues with delays classical results in [16, 4] for symmetric queues without delays on optimality of the *Join the Shortest (Nonfull) Queue* (JSQ) rule.

For the case of routing to two nonsymmetric queues with infinite buffers, in which index policies need no longer be optimal, [2] shows (in a variation on the above model) that the optimal routing policy is characterized by a monotone switching curve, extending a classical result in [3] for a model without delayed information. Still, one can easily devise a variety of heuristic routing index rules by defining indices based on ad hoc arguments, analogously to the Shortest Expected Delay routing rule in [5]. Yet, a drawback of such conventional indices, which typically measure a queue's expected weighted load, is that they only give a routing rule, being of no use to obtain a reasonable combined admission control and routing rule as outlined above, since consideration of rejection costs does not play a role in their definition.

We are thus led to address the issue of how to define appropriate indices  $\nu_k(a_k^-, i_k^-)$  for the above admission control and routing problems. Instead of proposing some ad hoc index via heuristic arguments, we will deploy a unifying fundamental design principle for priority allocation policies in *multiarmed restless bandit problems* (MARBPs), of which (2) and (3) are special cases, based on the economically intuitive concept of *marginal productivity index* (MPI). Such an approach was introduced in [15], and has been developed and applied in a variety of models by the second author in work including [7, 8, 10, 9], which is reviewed in [12]. In particular, [8, 13] introduced such an approach to the design of index policies for admission control and routing to parallel exponential queues without delayed information. As for use of MPI policies for problems with delayed state in-

formation, they were introduced in [14] in the setting of a dynamic scheduling model.

In the present setting, and focusing for concreteness on discounted problem (2) under combined admission control and routing, such a restless bandit indexation approach is based on decoupling the problem into individual *single-queue admission control subproblems*, one for each queue  $k \in \mathbb{K}$ :

$$\min_{\pi_k \in \Pi_k} \mathbb{E}_{(i_k^-, a_k^-)}^{\pi_k} \left[ \sum_{t=0}^{\infty} \left\{ c_k(X_k(t)) + \nu \lambda a_k(t) \right\} \beta^t \right], \quad (4)$$

where  $\Pi_k$  denotes the class of admission control policies based on one-period delayed state observation for operating queue  $k$  in isolation, and  $\mathbb{E}_{(i_k^-, a_k^-)}^{\pi_k}[\cdot]$  denotes expectation conditioned on the initial observed state and action pair being equal to  $Y_k(0) = (X_k(-1), a_k(-1)) = (i_k^-, a_k^-)$ . Note that, in such a setting, taking action  $a_k(t) = 1$  at period  $t$  means denying access to potential arrivals, which can be conveniently visualized as the action of *shutting the queue's entry gate* which is taken by a *gatekeeper*.

Problem (4) is a single *restless bandit problem* (RBP), i.e., a binary-action ( $a_k(t) = 1$ : active;  $a_k(t) = 0$ : passive) MDP, on which we can deploy the powerful theoretical and algorithmic results available for *restless bandit indexation* (cf. [12]). Let us say that problem (4) is *indexable* if there exists an index  $\nu_k^*(i_k^-, a_k^-)$  that characterizes its optimal policies for every value of rejection cost parameter  $\nu \in \mathbb{R}$ , as follows: it is optimal to take the active action (shut the entry gate) in augmented state  $Y_k(t) = (i_k^-, a_k^-)$  iff  $\nu_k^*(i_k^-, a_k^-) \geq \nu$ .

In such a case, we term  $\nu_k^*(i_k^-, a_k^-)$  the queue's MPI, due to its economic interpretation as a measure of the rate of marginal reduction in expected holding cost relative to the marginal increase in expected rejections that results from shutting the gate in state  $(i_k^-, a_k^-)$  instead of opening it, which characterizes the expected holding cost versus rejections tradeoff curve. Such is the index we propose to use as the basis for designing an index rule for admission control and/or routing for the multi-queue problems of concern.

Two issues need thus be addressed: (i) show that problem (4) is indeed indexable; and (ii) design an efficient index-computing algorithm. As for the first issue, we will deploy the sufficient indexability conditions based on *partial conservation laws* (PCLs) introduced in [7, 8]. Such conditions require one to identify a family of stationary deterministic policies among which an optimal policy for problem (4) exists for every value of the parameter  $\nu$ . For such a purpose, we draw on results in [1, 6] that characterize the structure of optimal policies for such an admission control problem (in an infinite-buffer model) with one-period delayed state information. Such work shows that it suffices to consider policies that are characterized by two thresholds  $k_1 \geq k_0 \geq 0$ , as follows: if the previous observed number of jobs in the system was  $i^-$  and the previous action was to open, i.e.,  $a^- = 0$  (resp. shut, i.e.,  $a^- = 1$ ) the queue's entry gate, the  $(k_0, k_1)$ -policy prescribes to shut the gate iff  $i^- > k_0$  (resp. iff  $i^- > k_1$ ). The intuition behind such a result is that, if it is optimal to shut the entry gate given that it was previously shut, then, other things being equal, it should also be optimal to shut it when it was previously open, as in the latter case the actual number of jobs in the system cannot be smaller than in the former. It is further shown in [1] that one need only consider threshold pairs that differ in at most one unit:  $0 \leq k_1 - k_0 \leq 1$ . Note that, in order to be con-

sistent with such *bi-threshold policies*, the MPI  $\nu^*(a^-, i^-)$  must be monotone nondecreasing in  $i^-$ , and satisfy

$$\nu^*(0, i^-) \geq \nu^*(1, i^-).$$

As for the second issue, that of index computation, provided PCL-indexability is established relative to such a family of policies, one can use the adaptive-greedy index algorithm introduced in [7, 8] to compute the MPI. Using the general fast-pivoting implementation given in [11] such an algorithm has a cubic arithmetic operation complexity in the number of restless bandit states, which in the present setting corresponds to an  $O(n^3)$  operation count. While tractable, such a complexity can be overly burdensome for online computation in high-speed communication switches.

Relative to the above two issues, this paper presents the following contributions: (i) it shows that problem (4) is PCL-indexable relative to bi-threshold policies, which ensures both existence of the MPI and the validity of the adaptive-greedy index algorithm for its computation; and (ii) by exploiting special structure, a substantially faster index algorithm is presented that computes the MPI in  $O(n)$  operations.

The remainder of the paper is organized as follows. Section 2 discusses the formulation of the admission control problems with and without feedback delay as *Markov decision problems* (MDPs). Section 3 discusses the restless bandit indexation approach to be deployed in a single-queue admission control problem with delayed information. Finally, Section 4 discusses the theoretical and algorithmic results.

Proofs of all results along with an extensive computational study testing the performance of the proposed index policies will be included in the full version of this paper, currently under preparation.

## 2. MDP FORMULATIONS

In order to see the analogy, in this section we formulate as a Markov decision process (MDP) both the admission control problem without delay and its counterpart with one-period delayed observation.

### 2.1 Admission Control with No Delay

First we formulate as an MDP the no-delay admission control problem. Let  $X(t)$  be the state process, denoting the *queue length* (including jobs in service, if any) at time period  $t$ . If  $a(t)$  denotes the action process, then the queue's gatekeeper must choose at time period  $t$  between shutting the entry gate ( $a(t) = 1$ ) and opening it ( $a(t) = 0$ ). The MDP elements are as follows:

- The *action space* is denoted by  $\mathbb{A} \triangleq \{0, 1\}$ .
- The *state space* is  $\mathbb{X} \triangleq \{0, 1, \dots, n\}$ , where state  $i \in \mathbb{X}$  represents the number of jobs in the system.
- Writing  $\zeta \triangleq \lambda(1 - \mu)$ ,  $\eta \triangleq \mu(1 - \lambda)$ , and  $\varepsilon \triangleq 1 - \zeta - \eta$ , the *one-period transition probabilities*  $p_{ij}^a \triangleq \mathbb{P}[X(t) = j | X(t-1) = i, a(t-1) = a]$  from state  $1 \leq i \leq n-1$  to state  $j$  under action  $a$  are

$$p_{ij}^0 = \begin{cases} \eta & \text{if } j = i - 1 \\ \varepsilon & \text{if } j = i \\ \zeta & \text{if } j = i + 1 \end{cases} \quad p_{ij}^1 = \begin{cases} \mu & \text{if } j = i - 1 \\ 1 - \mu & \text{if } j = i; \end{cases} \quad (5)$$

for the boundary cases,  $p_{00}^1 = 1$ , and

$$p_{0j}^0 = \begin{cases} 1 - \zeta & \text{if } j = 0 \\ \zeta & \text{if } j = 1 \end{cases} \quad p_{nj}^a = \begin{cases} \mu & \text{if } j = n - 1 \\ 1 - \mu & \text{if } j = n \end{cases} \quad (6)$$

All other transition probabilities are zero.

- If the queue length is  $i \in \mathbb{X}$ , then *one-period holding cost*  $C_i$  is incurred. We assume  $C_i$  to be *convex nondecreasing in  $i$* . At the same time, imagining that the gatekeeper performs work when he rejects an arrival, the one-period quantity of work expended is the expected number of rejected jobs during the current period, given by

$$Q_i^1 \triangleq \lambda, \quad Q_i^0 \triangleq \begin{cases} \lambda & \text{if } i = n \\ 0 & \text{otherwise.} \end{cases}$$

## 2.2 Admission Control with Delay

In the admission control problem with delay, the decision at period  $t$  is based on the *augmented state*  $Y(t) \triangleq (a(t-1), X(t-1))$ . Thus,  $Y(t)$  is the observed state at time period  $t$ , while  $X(t)$  is the actual (hidden) queue length process. The MDP elements of the admission control problem with delay are as follows:

- The *action space* is  $\mathbb{A}$  as in the no-delay problem.
- First note that states  $(0, n)$  and  $(1, n)$  in the problem with delay are indistinguishable, so we will merge them into a unique state denoted by  $(*, n)$ . We therefore define the *augmented state space* as

$$\mathbb{Y} \triangleq (\mathbb{A} \times \{0, 1, \dots, n-1\}) \cup \{(*, n)\}.$$

- The *one-period transition probabilities* are
- $$p_{(a,i),(b,j)}^{a'} \triangleq \mathbb{P}[Y(t+1) = (b, j) | Y(t) = (a, i), a(t) = a'] \\ = p_{ij}^a \cdot \mathbf{1}\{a' = b\},$$

where the  $p_{ij}^a$  are as in the no-delay model above. For the merged state  $(*, n)$ , we have

$$p_{(a,i),(*,n)}^{a'} \triangleq p_{in}^a.$$

- If the current-period augmented state is  $(a, i)$ , then the *one-period holding cost* is taken to be  $C_{(a,i)} \triangleq C_i$ . Similarly, the gatekeeper's *one-period work* is defined as the expected number of rejected jobs during the previous period,

$$Q_{(1,i)} \triangleq \lambda \quad Q_{(0,i)} \triangleq \begin{cases} \lambda & \text{if } i = n \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

To evaluate an admission control policy  $\pi$  under the discounted criterion relative to a given discount factor  $0 < \beta < 1$ , we consider the following two measures. Let

$$g_{(a,i)}^\pi \triangleq \mathbb{E}_{(a,i)}^\pi \left[ \sum_{t=0}^{\infty} Q_{Y(t)} \beta^t \right]. \quad (8)$$

be the expected total discounted work expended (i.e., the expected total discounted number of rejected jobs) if starting

from state  $Y(0) = (a, i)$  under policy  $\pi$ . The corresponding expected total discounted holding cost incurred is

$$f_{(a,i)}^\pi \triangleq \mathbb{E}_{(a,i)}^\pi \left[ \sum_{t=0}^{\infty} C_{Y(t)} \beta^t \right]. \quad (9)$$

If the rejection cost  $\nu$  is interpreted as the wage paid to gatekeeper per rejected job, then the objective is to solve the following  *$\nu$ -wage problem* for each  $\nu \in \mathbb{R}$ , where one seeks to minimize the sum of holding and labor costs:

$$\min_{\pi \in \Pi} f_{(a,i)}^\pi + \nu g_{(a,i)}^\pi, \quad (10)$$

where  $\Pi$  is the class of all admission control policies that use state and action information up to and including the previous period.

## 3. RESTLESS BANDIT INDEXATION

In the previous section we have formulated the admission control problem with delay as a binary-action Markov decision process (MDP), i.e., a *restless bandit*, where shutting the entry gate corresponds to the active action, and opening it as the passive action.

We next address such a problem by deploying a restless bandit indexation approach, following the seminal idea introduced in [15] and developed by the second author, in work surveyed in [12]. We focus on the finite-buffer problem under the discounted criterion. The solution to the problem under the time-average criterion is treated in Section 4.3.

MDP theory ensures existence of an optimal policy that is stationary, deterministic and independent of the initial state. We represent a stationary deterministic policy in terms of an *active set*  $S \subseteq \mathbb{Y}$ , i.e., the set of states in which it prescribes to shut the gate. The problem to find an optimal admission control policy is thus reduced to finding an optimal active set,

$$\min_{S \subseteq \mathbb{Y}} f_{(a,i)}^S + \nu g_{(a,i)}^S. \quad (11)$$

Now, let us say that the  $\nu$ -wage problem (11) is *indexable* if, as the parametric rejection cost  $\nu$  is increased from  $-\infty$  to  $\infty$  the set of states where it is optimal to take the active action (shut the gate) decreases monotonically from  $\mathbb{Y}$  to  $\emptyset$ . Such an indexability property is equivalent to existence of break-even values  $\nu_{(a,i)}^*$  of the rejection cost  $\nu$  attached to augmented states  $(a, i) \in \mathbb{Y}$ , which characterize the optimal policies for (11) as follows: it is optimal to take the active action when the system occupies augmented state  $(a, i)$  iff  $\nu_{(a,i)}^* \geq \nu$ . We will refer to index  $\nu_{(a,i)}^*$  as the *marginal productivity index* (MPI), after its economic interpretation as the marginal productivity of work at state  $(a, i)$ , as elucidated in [8, 10].

### 3.1 Exploiting Special Structure

While one could test numerically whether a given instance is indexable and calculate the MPI  $\nu_{(a,i)}^*$ , we aim instead to establish *analytically* the indexability of the admission control problem with delay. This will further allow us to achieve our objective of obtaining a fast way of computing the indices. In this subsection we show how to exploit special structure for such a purpose.

The key is to be able to identify a family  $\mathcal{F} \subseteq 2^{\mathbb{Y}}$  of active sets, satisfying certain connectivity conditions (see [12, Sec. 4.2] for the details), among which an optimal policy (active

set) for  $\nu$ -wage problem (11) exists for every value of  $\nu \in \mathbb{R}$ . We will thus aim to establish indexability relative to such a family, or  $\mathcal{F}$ -indexability.

Before presenting the appropriate family  $\mathcal{F}$  to use for the admission control problem with delay, we next the sufficient conditions we will deploy to establish  $\mathcal{F}$ -indexability, based on satisfaction of *partial conservation laws* (PCLs), introduced in [7, 8].

Denote by  $\langle a, S \rangle$  the policy that takes action  $a$  in the initial period and adopts the  $S$ -active policy afterwards. Define the *marginal work measure* by

$$w_{(a,i)}^S \triangleq g_{(a,i)}^{(1,S)} - g_{(a,i)}^{(0,S)}, \quad (12)$$

i.e., as the marginal increment in work expended (recall that work is performed by the gatekeeper when he rejects an arrival) that results from shutting the gate instead of opening it at the initial period, provided that the  $S$ -active policy is followed on subsequent periods. Analogously, define the *marginal cost measure*,

$$c_{(a,i)}^S \triangleq f_{(a,i)}^{(0,S)} - f_{(a,i)}^{(1,S)}, \quad (13)$$

as the corresponding decrease in holding costs incurred. Finally, define the *marginal productivity rate*

$$\nu_{(a,i)}^S \triangleq \frac{c_{(a,i)}^S}{w_{(a,i)}^S},$$

provided that the denominator does not vanish. As we will see,  $w_{(a,i)}^S$  is positive for the present model under active sets  $S$  taken from the active-set family  $\mathcal{F}$  to be given below.

**Table 1: Adaptive-greedy Index Algorithm  $\text{AG}_{\mathcal{F}}$ .**

<p><b>Output:</b> <math>\{(a_k, i_k), \nu_{(a_k, i_k)}^*\}_{k=1}^{2n+1}</math>  <math>S_0 \triangleq \mathbb{Y}</math>;  <b>for</b> <math>k = 1</math> <b>to</b> <math>2n + 1</math> <b>do</b>      <b>pick</b> <math>(a_k, i_k) \in \arg \min \left\{ \nu_{(a,i)}^{S_{k-1}} : (a, i) \in S_{k-1} \right.</math>          <b>and</b> <math>S_{k-1} \setminus \{(a, i)\} \in \mathcal{F}</math>;      <math>\nu_{(a_k, i_k)}^* := \nu_{(a_k, i_k)}^{S_{k-1}}</math>;      <math>S_k \triangleq S_{k-1} \setminus \{(a_k, i_k)\}</math>;  <b>end {for}</b>;</p>
---

We will further refer to the *adaptive-greedy index algorithm*  $\text{AG}_{\mathcal{F}}$  described in Table 1, which computes both an augmented state string  $(a_k, i_k)$  spanning  $\mathbb{Y}$  (note that the size of the augmented state space is  $|\mathbb{Y}| = 2n + 1$ ) along with index values  $\nu_{(a_k, i_k)}^*$ .

**DEFINITION 1.** *We say that problem (11) is  $\text{PCL}(\mathcal{F})$ -indexable if:*

- (i) *for every active set  $S \in \mathcal{F}$  and augmented state  $(a, i) \in \mathbb{Y}$ ,  $w_{(a,i)}^S > 0$ ;*

*and either of the following conditions holds:*

- (ii) *algorithm  $\text{AG}_{\mathcal{F}}$  computes index  $\nu_{(a_k, i_k)}^*$  in nondecreasing order; or*

- (ii') *for every rejection cost  $\nu \in \mathbb{R}$ , there exists an optimal active set  $S \in \mathcal{F}$ .*

The concept of  $\text{PCL}(\mathcal{F})$ -indexability was introduced in [7, 8] in terms of conditions (i) and (ii) in Definition 1. The more recent equivalent reformulation in terms of conditions (i) and (ii') is announced in [12, Sec. 4.2] and [11]. The usefulness of such a concept is made clear by the following result.

**THEOREM 1.** *If problem (11) is  $\text{PCL}(\mathcal{F})$ -indexable, then it is  $\mathcal{F}$ -indexable, and the index  $\nu_{(a_k, i_k)}^*$  computed by algorithm  $\text{AG}_{\mathcal{F}}$  is its MPI.*

### 3.2 Choosing the Active-Set Family

To choose the appropriate active-set family  $\mathcal{F}$  we draw on results in [1], which establish the optimality of bi-threshold policies for the problem of concern. A bi-threshold active-set policy with open-gate threshold  $m_0$  and shut-gate threshold  $m_1$  will be denoted by

$$S^{(m_0, m_1)} \triangleq \{(0, m_0), (0, m_0 + 1), \dots, (0, n)\} \cup \{(1, m_1), (1, m_1 + 1), \dots, (1, n)\}, \quad (14)$$

which is well-defined for all  $0 \leq m_0, m_1 \leq n + 1$  except the active sets  $S^{n+1, n}$  and  $S^{n, n+1}$ , because states  $(1, n)$  and  $(0, n)$  are duplicates, and hence either both or none of them can belong to  $S^{(m_0, m_1)}$ .

Thus, the  $S^{(m_0, m_1)}$ -active policy proceeds as follows. If the gate was open (resp. shut) at the previous period, the policy prescribes to shut the gate if the queue length at the previous period was greater than or equal to the open-gate threshold  $m_0$  (resp. the shut-gate threshold  $m_1$ ).

Intuitively, if an active set  $S^{(m_0, m_1)}$  is optimal for some rejection cost  $\nu$ , then it must be  $m_0 \leq m_1$ . Indeed, for a given previous-period queue length, we should be less prone to shut the gate if it was previously shut than if it was previously open, because the queue length cannot get larger under a shut gate. Indeed, we can draw on [1, Theorem 3.1] to show that it suffices to use such active sets to solve the problem of concern, further restricting attention to thresholds that satisfy  $m_1 \leq m_0 + 1$ . Therefore, the appropriate family of optimal active sets to choose for the admission control problem with delay is

$$\mathcal{F} \triangleq \{S^{(m, m)} : m = 0, 1, \dots, n + 1\} \cup \{S^{(m, m+1)} : m = 0, 1, \dots, n - 1\}, \quad (15)$$

where  $S^{(0, 0)} = \mathbb{Y}$  and  $S^{(n+1, n+1)} = \emptyset$ .

Although the result in [1, Theorem 3.1] was shown for the problem with infinite buffer, it directly applies to the finite-buffer variant of concern herein. Note that if a bi-threshold policy is optimal for the infinite-buffer problem, then it is also optimal for all problems with buffer size equal to or larger than both thresholds.

For the active-set family  $\mathcal{F}$  given in (15), picking  $(a_k, i_k)$  in the adaptive-greedy algorithm becomes trivial, because there is a unique feasible augmented state at each step. Thus, at step  $k = 1$ , only state  $(1, 0)$  both belongs in  $S_0$  and  $S_0 \setminus \{(1, 0)\} = S^{(0, 1)} \in \mathcal{F}$ , since  $S_0 \triangleq \mathbb{Y} = S^{(0, 0)}$ . Similarly, at step  $k = 2$ , only state  $(0, 0)$  both belongs in  $S_1$  and  $S_1 \setminus \{(0, 0)\} = S^{(1, 1)} \in \mathcal{F}$ . In general,  $(a_k, i_k) = (0, (k/2) - 1)$  for every even  $1 \leq k \leq 2n$ , and  $(a_k, i_k) = (1, (k - 1)/2)$  for

every odd  $1 \leq k \leq 2n$ . Finally, at step  $k = 2n + 1$ , the chosen state is  $(*, n)$ .

To summarize, the sequence of active sets  $S_k$  generated by algorithm  $\text{AG}_{\mathcal{F}}$  under such an active-set family  $\mathcal{F}$  is uniquely determined, being given by

$$\begin{aligned} S_0 &= \mathbb{Y} = S^{(0,0)}, S_1 = S^{(0,1)}, S_2 = S^{(1,1)}, S_3 = S^{(1,2)}, \\ S_4 &= S^{(2,2)}, \dots, S_{2n-1} = S^{(n-1,n)}, S_{2n} = S^{(n,n)}, \\ S_{2n+1} &= S^{(n+1,n+1)} = \emptyset, \end{aligned} \quad (16)$$

and the generated sequence of augmented states  $(a_k, i_k)$  is

$$\begin{aligned} (a_1, i_1) &= (1, 0), (a_2, i_2) = (0, 0), (a_3, i_3) = (1, 1), \\ (a_4, i_4) &= (0, 1), \dots, (a_{2n-1}, i_{2n-1}) = (1, n-1), \\ (a_{2n}, i_{2n}) &= (0, n-1), (a_{2n+1}, i_{2n+1}) = (*, n). \end{aligned}$$

In light of the above results, Table 2 shows the resultant simplified version of algorithm  $\text{AG}_{\mathcal{F}}$  as it applies to the postulated family  $\mathcal{F}$  given in (15). Note however that the computational complexity remains at the same level since the main difficulty lies in the calculation of  $\nu_{(a_k, i_k)}^{S_{k-1}}$ , for which no computational details are given.

**Table 2: Simplified Version of Algorithm  $\text{AG}_{\mathcal{F}}$  under the Active-set Family  $\mathcal{F}$  in (15).**

```

Output:      {ν*(a,i)}_{(a,i)∈Y}
for m = 1 to n do
  ν*(1,m-1) := ν_{(1,m-1)}^{S^{(m-1,m-1)}};
  ν*(0,m-1) := ν_{(0,m-1)}^{S^{(m-1,m)}};
end {for};
ν*(*,n) := ν_{(*,n)}^{S^{(n,n)}};

```

## 4. RESULTS

In this section we focus on the admission control problem with delay with a finite buffer of size  $n \geq 2$  under the discounted criterion. The case  $n = 1$  is treated in Section 4.2. The results under the time-average criterion are summarized in Section 4.3.

Our main results are twofold. First, we show that the problem is  $\text{PCL}(\mathcal{F})$ -indexable (cf. Definition 1) under the active-set family  $\mathcal{F}$  given in (15), which allows us to invoke Theorem 1 to ensure that algorithm  $\text{AG}_{\mathcal{F}}$  can be used to compute the MPI. Second, we dramatically reduce the complexity of such an algorithm, which has generally a cubic complexity  $O(n^3)$  as shown in [11], obtaining an implementation that performs only  $O(n)$  arithmetic operations.

Let us introduce a more compact notation. For a variable of the form  $x_{(a,i)}$ , we will use the backward difference operator in the first dimension, i.e., the *action-difference operator*,

$$\Delta_1 x_{(1,i)} \triangleq x_{(1,i)} - x_{(0,i)} \quad (17)$$

and in the second dimension, i.e., the *state-difference operator*,

$$\Delta_2 x_{(a,i)} \triangleq x_{(a,i)} - x_{(a,i-1)} \quad (18)$$

whenever the right-hand side expressions are defined. We further write  $\Delta_2 x_{(a,0)} \triangleq 0$  for  $a \in \mathbb{A}$ . Note that from such definitions we obtain the following auxiliary identity:

$$\Delta_2 x_{(1,i)} - \Delta_2 x_{(0,i)} = \Delta_1 x_{(1,i)} - \Delta_1 x_{(1,i-1)}. \quad (19)$$

We list our main results, drawing on the technical analysis of work measures which is omitted due to space constraints.

**PROPOSITION 1.** *The admission control problem with delay (11) is  $\text{PCL}(\mathcal{F})$ -indexable, and therefore it is indexable and algorithm  $\text{AG}_{\mathcal{F}}$  computes its MPI.*

### 4.1 A Fast Index Algorithm

In order to avoid unnecessary technical complications, we henceforth restrict attention to the *linear holding cost* case, i.e.,  $C_i \triangleq ci$  for some  $c > 0$ . We develop an implementation of the above adaptive-greedy algorithm that computes the MPI performing only  $O(n)$  arithmetic operations, which is two orders of magnitude faster than the best general implementation of the algorithm, which performs  $O(n^3)$  operations.

Note that algorithm  $\text{AG}_{\mathcal{F}}$  in Table 2 is formulated in its *bottom-up* version, as it calculates the MPI values in non-decreasing order (cf. Definition 1(ii)). One can similarly consider the algorithm's reformulation in its equivalent, *top-down* version, starting with the empty set and calculating the indices in nonincreasing order.

In other words, while the bottom-up version of algorithm  $\text{AG}_{\mathcal{F}}$  traverses the active-set family  $\mathcal{F}$  in the order (cf. 16)

$$S^{(0,0)}, S^{(0,1)}, S^{(1,1)}, S^{(1,2)}, \dots, S^{(n-1,n)}, S^{(n,n)}, S^{(n+1,n+1)},$$

the top-down version does it in the reverse order

$$S^{(n+1,n+1)}, S^{(n,n)}, S^{(n-1,n)}, \dots, S^{(1,2)}, S^{(1,1)}, S^{(0,1)}, S^{(0,0)}.$$

For instance, the MPI value  $\nu_{(1,0)}^*$  is calculated as the marginal productivity rate  $\nu_{(1,0)}^{S^{(0,0)}}$  in the bottom-up version, while the same value is calculated as the marginal productivity rate  $\nu_{(1,0)}^{S^{(0,1)}}$  in the top-down version. In fact, [8, Theorem 6.4(b)] implies that  $\nu_{(a_k, i_k)}^{S_{k-1}} = \nu_{(a_k, i_k)}^{S_k}$ , using the notation in Table 2. Therefore, we can reformulate the algorithm in terms of marginal productivity rates under active sets  $S^{(m,m)}$  only. The resultant simplified algorithm is presented in Table 3.

**Table 3: Reformulation of Algorithm  $\text{AG}_{\mathcal{F}}$  in terms of Active Sets  $S^{(m,m)}$  only.**

```

Output:      {ν*(a,i)}_{(a,i)∈Y}
ν*(1,0) := ν_{(1,0)}^{S^{(0,0)}};
for m = 1 to n - 1 do
  ν*(0,m-1) := ν_{(0,m-1)}^{S^{(m,m)}}; ν*(1,m) := ν_{(1,m)}^{S^{(m,m)}};
end {for};
ν*(0,n-1) := ν_{(0,n-1)}^{S^{(n,n)}}; ν*(*,n) := ν_{(*,n)}^{S^{(n+1,n+1)}};

```

We next develop an efficient implementation of the algorithmic scheme  $\text{AG}_{\mathcal{F}}$ , termed the *fast index* (FI) algorithm, which is described in Table 4. Algorithm FI is two orders

of magnitude faster than the best existing general implementation of the algorithm AG<sub>g</sub>. We characterize the MPI calculated as indicated in Table 3 in terms of closed-form expressions of pivot state-differences.

**PROPOSITION 2.** *The FI algorithm in Table 4 computes the MPI for problem (11) under the discounted criterion, performing  $O(n)$  arithmetic operations.*

Once the MPI is available is known, the optimal thresholds for a given rejection cost  $\nu$  are readily obtained. The optimal open-gate threshold is

$$m_0 = \min\{i \in \mathbb{X}: \nu_{(0,i)}^* \geq \nu\},$$

and the optimal shut-gate threshold is

$$m_1 = \min\{i \in \mathbb{X}: \nu_{(1,i)}^* \geq \nu\}.$$

If the rejection charge is high enough, in particular if  $\nu > \nu_{(*,n)}^*$ , then  $m_0 \triangleq n + 1$  and  $m_1 \triangleq n + 1$ , meaning that it is optimal to always shut the gate.

Note that the indices calculated in the FI algorithm's "loop" are *independent* of the buffer length  $n$  (only the indices of boundary states  $(0, n - 1)$  and  $(*, n)$  depend on  $n$ ). Therefore, the algorithm can be used to obtain the MPI for the model extension with infinite buffer space, although in such a case the algorithm's loop would never stop.

## 4.2 The case of a single buffer space

In this section we solve the admission control problem with delay for  $n = 1$ , i.e., no job is allowed to be queued, except for the one in service. While this problem may not be of intrinsic interest, its solution given next serves as a basis for the corresponding problem of admission control and/or routing to parallel queues where each queue has a single buffer space.

**PROPOSITION 3.** *The MPI of state  $(a, i) \in \mathbb{Y}$  in case  $n = 1$  is state-independent and equals*

$$\nu_{(a,i)}^* := \frac{\zeta\beta C}{\lambda} = \frac{c\lambda\beta(1-\mu)}{1-\beta(1-\mu)}.$$

These indices can be obtained in the same way as the general case  $n \geq 2$ . The state-independent marginal productivity index means that, given a rejection cost  $\nu$ , it is optimal either to admit always, or to reject always, regardless of the previous-period state and previous-period action, i.e., regardless of information available.

## 4.3 The Time-Average Criterion

The above results extend directly to the admission control problem with delay under the time-average criterion.

**PROPOSITION 4.** *By setting  $\beta = 1$ , the FI algorithm in Table 4 computes the MPI under the time-average criterion.*

## 4.4 Further Remarks

If in state  $(1, 0)$ , the buffer is empty, because it was empty a period ago and the gate has been closed since then. Therefore, one could expect that the index of state  $(1, 0)$  is the same as the index of state  $0$  in the no-delay problem, which is in fact true. Moreover, there is a simple interpretation of that expression.

If the buffer is empty, the expected total  $\beta$ -discounted holding cost is

$$\zeta\beta c [1 + \beta(1 - \mu) + (\beta(1 - \mu))^2 + \dots] = \frac{\beta\zeta c}{1 - \beta + \beta\mu},$$

because  $\zeta$  is the probability that the job remains in the buffer for more than a period. The above expression is equal to  $\lambda\nu_{(1,0)}^*$ , the expected (total  $\beta$ -discounted) rejection cost is if the rejection cost  $\nu = \nu_{(1,0)}^*$ . Thus, in state  $(1, 0)$  it is optimal to close the gate if the expected rejection cost is lower than the expected discounted total holding cost of an admitted job. Further, in state  $(1, 0)$  it is optimal to let the gate open if the expected rejection cost is greater than the expected discounted total holding cost of an admitted job. If the two expected costs are equal, both closing and opening is optimal. It is also clear that under the former condition it is optimal to close the gate in *any* state, and therefore the indices of all states must not be smaller than  $\nu_{(1,0)}^*$ .

Figure 1 shows the indices for a number of instances of the admission control problem with delay. An extensive simulation study we have performed suggests a convergence of the indices:

$$\begin{aligned} \nu_{(1,i)}^* &\rightarrow \nu_{(0,i)}^* && \text{as } \lambda \rightarrow 0, \\ \nu_{(1,i)}^* &\rightarrow \nu_{(0,i-1)}^* && \text{as } \zeta \rightarrow 1, \\ \nu_{(0,i)}^* &\rightarrow \frac{\beta c}{1 - \beta} && \text{as } i \rightarrow \infty, \\ \nu_{(1,i)}^* &\rightarrow \frac{\beta c}{1 - \beta} && \text{as } i \rightarrow \infty. \end{aligned}$$

The convergence of the marginal productivity indices to  $\beta c / (1 - \beta)$  is intuitive. If the buffer is almost full (say, the previous-period queue length is  $n - 2$ ), then admitting a job means to increase the overall holding cost by  $c$  at least in the following  $n - 2$  periods, because the admitted job cannot leave the system earlier than the previous  $n - 2$  jobs. Therefore, the expected total  $\beta$ -discounted holding cost is at least

$$\beta c [1 + \beta + \beta^2 + \dots + \beta^{n-2}] = \frac{\beta c (1 - \beta^{n-1})}{1 - \beta}.$$

On the other hand, it is not greater than the expected holding cost of remaining in the buffer forever, which is

$$\beta c [1 + \beta + \beta^2 + \dots] = \frac{\beta c}{1 - \beta}.$$

Now it is clear that the marginal productivity indices converge to  $\beta c / (1 - \beta)$  as  $n \rightarrow \infty$ .

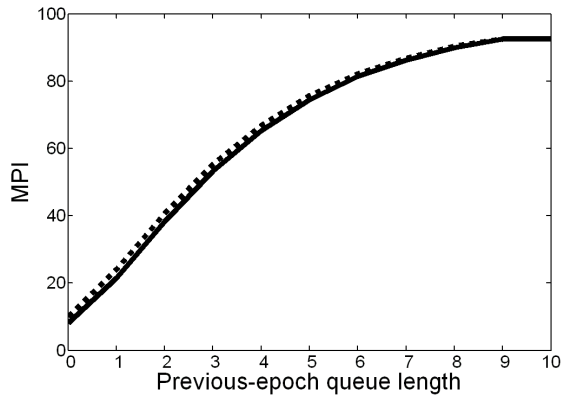
## 5. ACKNOWLEDGMENTS

This research is part of work carried out towards the completion of graduate student P. Jacko's doctorate under the direction of J. Niño-Mora, who acknowledges funding support by the Spanish Ministry of Education and Science under projects MTM2004-02334, MTM2007-63140, and an I3 faculty endowment grant, by the European Union's Network of Excellence Euro-FGI, and by the Autonomous Community of Madrid under grant CCG07-UC3M/ESP-3389. P. Jacko's work has been supported by the Spanish Ministry of Education and Science under predoctoral research training grant BES-2005-7026, attached to project MTM2004-02334.

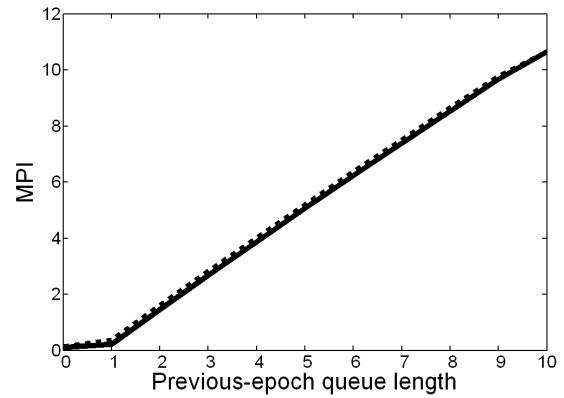
**Table 4: The Fast Index Algorithm FI.**

<p><b>Input:</b> <math>n, \lambda, \mu, c, \beta</math> ;</p> <p><b>Output:</b> <math>\{\nu_{(a,i)}^*\}_{(a,i) \in \mathbb{Y}}</math></p> <p><b>{ Initialization }</b></p> <p><math>\zeta := \lambda(1 - \mu)</math>; <math>\eta := \mu(1 - \lambda)</math>; <math>\varepsilon := 1 - \zeta - \eta</math>;</p> <p><math>A_0 := 0</math>; <math>A'_0 := \beta\zeta</math>; <math>B := \beta\mu/(1 - \beta + \beta\mu)</math>; <math>B' := \beta\zeta B + \beta(\mu - \eta)</math>; <math>C := c/(1 - \beta + \beta\mu)</math>; <math>D_0 := 0</math>;</p> <p><math>\nu_{(1,0)}^* := \zeta\beta C/\lambda</math>;</p> <p><math>\nu_{(0,0)}^* := \frac{\zeta}{\lambda} \beta C_1 \frac{(1 - \beta + \beta\mu)(1 + \beta\zeta) + \beta\zeta(\mu + \beta\mu + \beta\zeta) + (1 - \beta + \beta\mu)(1 + \beta)(\mu - \eta)}{(1 - \beta + \beta\mu)(1 + \beta\zeta) + \beta\zeta\beta\zeta(1 - B) - \beta\zeta\beta(\mu - \eta)}</math>;</p> <p><b>{ Loop }</b></p> <p><b>for</b> <math>m := 1</math> <b>to</b> <math>n - 1</math> <b>do</b></p> <p><math>A_m := \beta\zeta/[1 - \beta + \beta\zeta + \beta\eta(1 - A_{m-1})]</math>; <math>A'_m := \beta\zeta + \beta(\mu - \eta)A_m</math>; <math>D_m := (c + \beta\eta D_{m-1}) A_m/(\beta\zeta)</math>;</p> <p><math>f^0 := -\frac{\frac{\beta\zeta}{A_m} D_m + \beta\zeta(c + \beta\mu B D_{m-1}) + [c - \beta(\mu - \eta)\beta D_{m-1}] B'}{\frac{A'_m}{A_m} + \beta A'_{m-1} B' + \beta\zeta\beta\mu(1 - B A_{m-1})}</math>;</p> <p><math>f^1 := -\frac{\frac{\beta\zeta}{A_m} D_m + c\beta\zeta B A_{m-1} + [\beta\mu\beta\zeta + (1 - \beta)\beta(\mu - \eta)] D_{m-1} + A'_{m-1}(c - \beta\zeta\beta C)}{\frac{A'_m}{A_m} + \beta A'_{m-1} B' + \beta\zeta\beta\mu(1 - B A_{m-1})}</math>;</p> <p><math>g^0 := \frac{\beta\lambda(1 + B')}{\frac{A'_m}{A_m} + \beta A'_{m-1} B' + \beta\zeta\beta\mu(1 - B A_{m-1})}</math>; <math>g^1 := \frac{1 + A'_{m-1}}{1 + B'} g^0</math>;</p> <p><b>if</b> <math>m &gt; 1</math> <b>then</b></p> <p><math>\nu_{(0,m-1)}^* := \frac{[\beta(\mu - \eta)(D_{m-1} - c) + \beta\eta\beta\zeta D_{m-1} + \beta\zeta\beta\zeta C] - [\beta\eta A_{m-1} A'_{m-2} + \beta\varepsilon A'_{m-1}] f^0 - \beta\zeta B' f^1}{\beta\lambda - [\beta\eta A_{m-1} A'_{m-2} + \beta\varepsilon A'_{m-1}] g^0 - \beta\zeta B' g^1}</math>;</p> <p><b>end if</b>;</p> <p><math>\nu_{(1,m)}^* := \frac{[\beta(1 - \mu)\beta\zeta C + \beta\mu\beta(\mu - \eta)D_{m-1}] - \beta\mu A'_{m-1} f^0 - \beta(1 - \mu)B' f^1}{\beta\lambda - \beta\mu A'_{m-1} g^0 - \beta(1 - \mu)B' g^1}</math>;</p> <p><b>end for</b>;</p> <p><b>{ Termination }</b></p> <p><math>A_n := \beta\zeta/[1 - \beta + \beta\zeta + \beta\eta(1 - A_{n-1})]</math>; <math>A'_n := \beta\zeta + \beta(\mu - \eta)A_n</math>; <math>D_n := (c + \beta\eta D_{n-1}) A_n/(\beta\zeta)</math>;</p> <p><math>f^0 := -\frac{\frac{\beta\zeta}{A_n} D_n - \beta(\mu - \eta)\beta\mu D_{n-1}}{\frac{A'_n}{A_n} + \beta\mu A'_{n-1}}</math>; <math>g^0 := \frac{\lambda(1 + \beta\mu)}{\frac{A'_n}{A_n} + \beta\mu A'_{n-1}}</math>;</p> <p><math>\nu_{(0,n-1)}^* := \frac{[\beta(\mu - \eta)(D_{n-1} - c) + \beta\eta\beta\zeta D_{n-1}] - [\beta\eta A_{n-1} A'_{n-2} + \beta\varepsilon A'_{n-1}] f^0}{\beta(1 - \zeta)\lambda - [\beta\eta A_{n-1} A'_{n-2} + \beta\varepsilon A'_{n-1}] g^0}</math>;</p> <p><math>\nu_{(*,n)}^* := \frac{\beta(\mu - \eta)D_{n-1} A'_n + \beta\zeta D_n A'_{n-1}}{\lambda A'_n - \lambda A_n A'_{n-1}}</math></p>
--

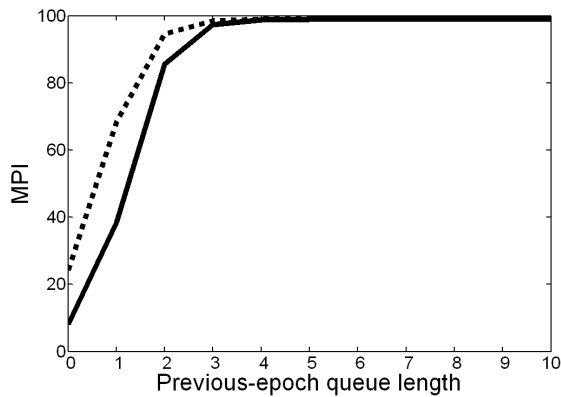




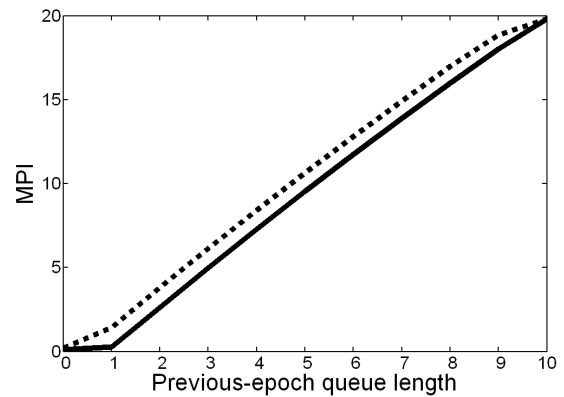
(a) if  $\lambda = 0.1, \mu = 0.1$



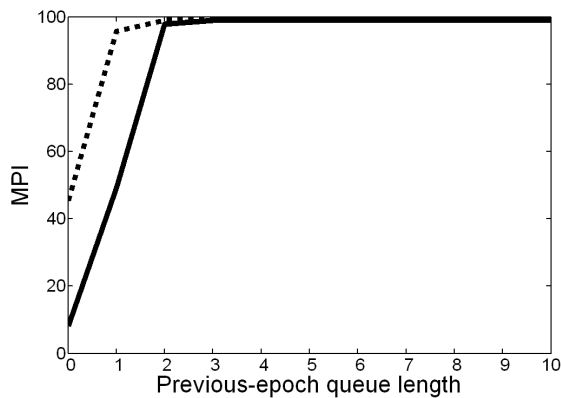
(b) if  $\lambda = 0.1, \mu = 0.9$



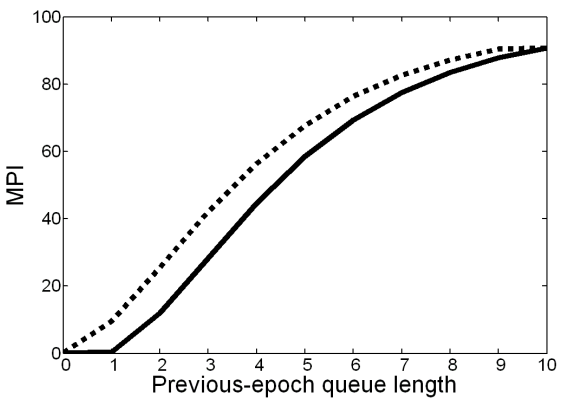
(c) if  $\lambda = 0.5, \mu = 0.1$



(d) if  $\lambda = 0.5, \mu = 0.9$



(e) if  $\lambda = 0.9, \mu = 0.1$



(f) if  $\lambda = 0.9, \mu = 0.9$

Figure 1: MPIs for the admission control problem with delay with parameters  $n = 10, c = 1, \beta = 0.99$ . The solid line shows the index  $\nu_{(1,i)}^*$  and the dotted line shows  $\nu_{(0,i)}^*$ .

## 6. REFERENCES

- [1] E. Altman and P. Nain. Closed-loop control with delayed information. *Performance Evaluation Rev.*, 20:193–204, 1992.
- [2] D. Artiges. Optimal routing into two heterogeneous service stations with delayed information. *IEEE Trans. Automat. Control*, 40:1234–1236, 1995.
- [3] B. Hajek. Optimal control of two interacting service stations. *IEEE Trans. Automat. Control*, AC-29:491–499, 1984.
- [4] A. Hordijk and G. Koole. On the optimality of the generalized shortest queue policy. *Probab. Eng. Inf. Sci.*, 4:477–487, 1990.
- [5] D. J. Houck. Comparison of policies for routing customers to parallel queueing systems. *Oper. Res.*, 35:306–310, 1987.
- [6] J. Kuri and A. Kumar. Optimal control of arrivals to queues with delayed queue length information. *IEEE Trans. Automat. Control*, 40:1444–1450, 1995.
- [7] J. Niño-Mora. Restless bandits, partial conservation laws and indexability. *Adv. Appl. Probab.*, 33:76–98, 2001.
- [8] J. Niño-Mora. Dynamic allocation indices for restless projects and queueing admission control: a polyhedral approach. *Math. Program.*, 93:361–413, 2002.
- [9] J. Niño-Mora. Marginal productivity index policies for scheduling a multiclass delay-/loss-sensitive queue. *Queueing Syst.*, 54:281–312, 2006.
- [10] J. Niño-Mora. Restless bandit marginal productivity indices, diminishing returns and optimal control of make-to-order/make-to-stock  $M/G/1$  queues. *Math. Oper. Res.*, 31:50–84, 2006.
- [11] J. Niño-Mora. Characterization and computation of restless bandit marginal productivity indices. In *ValueTools '07: Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools (Nantes, France)*, ACM International Conference Proceedings Series, vol. 321. ICST, Brussels, Belgium, 2007. Published online in the ACM Digital Library.
- [12] J. Niño-Mora. Dynamic priority allocation via restless bandit marginal productivity indices. *TOP*, 15:161–198, 2007. Followed by six discussions by I. J. B. F. Adan and O. J. Boxma, E. Altman, O. Hernández-Lerma, R. Weber, P. Whittle, and D. D. Yao.
- [13] J. Niño-Mora. Marginal productivity index policies for admission control and routing to parallel multi-server loss queues with reneging. In *Proceedings of NET-COOP 2007, the 1st Euro-FGI Conference on Network Control and Optimization (Avignon, France)*, *Lecture Notes in Computer Science*, vol. 4465, pp. 138–149. Springer, Berlin, 2007.
- [14] J. Niño-Mora. Marginal productivity index policies for scheduling multiclass delay-/loss-sensitive traffic with delayed state observation. In *Proceedings of NGI 2007, the 3rd Euro-NGI Conference on Next Generation Internet Networks: Design and Engineering for Heterogeneity (Trondheim, Norway)*, pp. 209–217. IEEE, Piscataway, NJ, 2007.
- [15] P. Whittle. Restless bandits: Activity allocation in a changing world. In J. Gani, editor, *A Celebration of Applied Probability*, volume 25A of *J. Appl. Probab.*, pp. 287–298. Applied Probability Trust, Sheffield, UK, 1988.
- [16] W. Winston. Optimality of the shortest line discipline. *J. Appl. Probab.*, 14:181–189, 1977.