On the bias vector of a two-class preemptive priority queue

Robin Groenevelt[†] Ger Koole[‡] Philippe Nain[†]
† INRIA, B.P. 93, 06902 Sophia Antipolis, France
‡ Vrije Universiteit, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

December 10, 2001

Abstract

We give a closed-form expression for the long-run average cost and the bias vector in a two-class exponential preemptive resume priority queue with holding and switching costs. The bias vector is the sum of a quadratic function of the number of customers in each priority class and an exponential function of the number of customers in the high priority class. We use this result to perform a single step of the policy iteration algorithm in the model where the switches of the server from one priority class to the other can be controlled. It is numerically shown that the policy resulting from the application of a single step of the policy iteration algorithm is close to the optimal policy.

1 Introduction

Finding optimal controls in queueing networks is a challenging problem. Few analytical results exist, therefore numerical solutions are sought. However, straightforward methods consisting of executing standard Markov decision algorithms such as value or policy iteration are often impossible because of the *curse of dimensionality*. We therefore have to rely on other exact or approximation methods that do not suffer from the dimensionality of the problem. Two different approximation methods are *one-step optimization* and *reinforcement learning*. For both methods it is necessary to have some insight in the dynamic programming optimality equation. For one-step optimization we need to have an explicit solution to the optimality equation for one fixed policy; for reinforcement learning we need the structure of the optimality equation (e.g., quadratic in the queue lengths) for all policies. In this paper we study the average cost for a specific two-dimensional control problem, namely a single server serving two customer classes with holding and switching costs. The objective is to change class in such a way as to minimize the sum of expected long-run average holding and switching costs. We concentrate ourselves on the one-step optimization method. We first solve the optimality equation for the preemptive priority rule. It appears to be almost quadratic in the queue lengths, with a single exponential term. Then we apply one-step optimization, starting from the priority rule. Due to the low dimensionality of the example problem, we can compute the optimal policy and compare the results. In principle this method can also be used for problems of a higher

dimension (of course, without being able to compare results with the optimal policy). Currently we are doing research to extend the results to more than two dimensions and general service time distributions, both for the average and discounted cost criterion. With respect to reinforcement learning, our results give some theoretical back-up to the often made assumption that the bias vector is quadratic.

This work is the average cost counterpart of Koole & Nain [10]. The average cost can be seen as the limit of the discounted cost as the discounting goes to 0, but calculating this limit appeared to be a non-trivial task. Therefore we use a direct argument. The idea of using one step of value iteration was introduced in Ott & Krishnan [12], and used in many papers since then. A different approach to finding good policies for the same two-dimensional model is described in Koole [9]. Computational methods for obtaining the stationary distribution of the resulting threshold policies are developed in Boxma et al. [5] and Boxma & Down [4]. Background on reinforcement learning or neuro-dynamic programming can be found in Bertsekas & Tsitsiklis [3]. A paper where quadratic value functions are used is Marbach et al. [11].

This paper is organized as follows. In the next section we introduce the preemptive priority queue and its value function. In Section 3 we derive the main result of this paper, which is the closed form expression of the solution of the value function. In the last section we use this result to study the two-dimensional control problem with holding and switching costs using one-step optimization.

2   The model and the optimality equation

We consider a single-server queue equipped with an infinite capacity buffer and fed by two classes of customers: customers of class i ($i = 1, 2$) arrive to the queue according to a Poisson process with rate $\lambda_i > 0$ and require independent and identically distributed service times with a common exponential distribution with mean $1/\mu_i$. We assume that the arrival and the service time processes are mutually independent processes. We set $\lambda := \lambda_1 + \lambda_2$.

There are holding costs and switching costs. We denote by c_i ($i = 1, 2$) the cost of holding a customer of class i in the system during one unit of time and by s_{12} (resp. s_{21}) the cost of switching from serving a customer of class 1 to a customer of class 2 (resp. from a customer of class 2 to a customer of class 1). We will say that the server is in position $i = 1, 2$ at time t if it is attending customers of class i at time t. We assume that the server remains at its current position when the system empties; in this case a switch will occur at the next arrival only if the class of the arriving customer is different from the class of the last customer which has left the system. Server switches are assumed to be instantaneous.

We shall assume for the time being that customers are served according to the preemptive resume service discipline [6], where customers of class 1 have priority over customers of class 2. More general service disciplines will be considered in Section 4.

Under the preemptive resume service discipline a customer of class 1 is always attended by the server as long as there are customers of this class in the system, regardless how many customers of class 2 are waiting. A preemption occurs when an arriving customer of class 1 finds a customer

of class 2 in the server; in this case the customer of class 2 is preempted at once and the server switches from position 2 to position 1. The service of the preempted customer will be resumed.

The server switches from position 1 to position 2 whenever a customer of class 1 departs the system and leaves behind him no customer of his class and at least one customer of class 2.

Since customers of class 1 are not affected by the presence of customers of class 2, the number of customers of class 1 in this system at any time is nothing but the number of customers in an ordinary M/M/1 queue with arrival intensity λ_1 and service intensity μ_1 (provided the initial states in both systems are the same).

From now on we shall assume that the condition $\sum_{i=1}^{2} \lambda_i / \mu_i < 1$ holds. This condition is the stability condition under the preemptive resume priority discipline [6]. More generally, it is the stability condition under any work-conserving service discipline and, in particular, under all policies to be considered in Section 4.

We call an *event time* any arrival or departure time of either class of customers. Let $0 = T_1 < T_2 < \cdots < T_n < \cdots <$ be the successive event times. Let X_n (resp. Y_n) be the number of customers of class 1 (resp. class 2) in the system just *after* the n-th event time, and let $Z_n \in \{1, 2\}$ be the class which is attended by the server just *after* the n-th event time, including the server switch if applicable.

At time T_n, the state of the system will be represented by the triple $U_n := (X_n, Y_n, Z_n)$. The subsequent analysis will heavily rely on the fact that under the preemptive resume priority policy, the stochastic sequence $\{U_n, n \geq 1\}$ constitutes a discrete-time aperiodic and irreducible Markov chain on the state-space $\mathbf{S} = \mathbf{N}^2 \times \{1, 2\} - \{(x, 0, 2), x \geq 1\} - \{(0, y, 1), y \geq 1\}$. We will denote by $p_{u,u'}$ the one-step transition probability from state u to state u'.

Throughout $\mathbf{E}_u$ will denote the expectation operator conditioned on $U_1 = u$. Given that the system is in state $u = (x, y, z) \in \mathbf{S}$ at time T_n, the total expected cost $c(u)$ incurred in $[T_n, T_{n+1})$ is given by

$$
\begin{aligned}
c(u) &= (c_1 x + c_2 y) \, \mathbf{E}_u[T_{n+1} - T_n] \\
&\quad + s_{12} \, \mathbf{E}_u[\mathbf{1}(\theta_n = a_2, x = y = 0, z = 1) + \mathbf{1}(\theta_n = d_1, x = 1, y \geq 1)] \\
&\quad + s_{21} \, \mathbf{E}_u[\mathbf{1}(\theta_n = a_1, z = 2)]
\end{aligned}
\tag{1}
$$

where $\theta_n \in \{a_1, a_2, d_1, d_2\}$ is the type of the n-th event, with $\theta_n = a_i$ (resp. $\theta_n = d_i$) if a customer of class $i = 1, 2$ arrives (resp. leaves) at time T_n. Straightforward algebra yields

$$
\begin{aligned}
c(u) &= (c_1 x + c_2 y) \left(\frac{1}{\lambda + \mu_1} \mathbf{1}(x \geq 1) + \frac{1}{\lambda + \mu_2} \mathbf{1}(x = 0, y \geq 1) + \frac{1}{\lambda} \mathbf{1}(x = y = 0) \right) \\
&\quad + s_{12} \left(\frac{\lambda_2}{\lambda} \mathbf{1}(x = y = 0, z = 1) + \frac{\mu_1}{\lambda + \mu_1} \mathbf{1}(x = 1, y \geq 1) \right) \\
&\quad + s_{21} \left(\frac{\lambda_1}{\lambda + \mu_2} \mathbf{1}(x = 0, y \geq 1, z = 2) + \frac{\lambda_1}{\lambda} \mathbf{1}(x = y = 0, z = 2) \right).
\end{aligned}
\tag{2}
$$

3

The long-run average total expected cost $g(u)$ starting in state u at time 0 is defined as

$$g(u) = \liminf_{n \to \infty} \frac{\mathbf{E}_u\left[\sum_{i=1}^n c(U_n)\right]}{\mathbf{E}_u[T_{n+1}]}, \quad u \in \mathbf{S}. \tag{3}$$

Since the Markov chain $\{U_n,\, n \geq 1\}$ has a single ergodic class, it is known [13] that $g(u)$ is a constant function of $u \in \mathbf{S}$, namely, there exists a non-negative constant g such that $g(u) \equiv g$ for all $u \in \mathbf{S}$.

In order to determine the constant g, we need to solve the dynamic programming optimality equation – also known as the Poisson equation – given by [13, Eq. (11.4.17)]

$$g\mathbf{E}_u[T_2] + h(u) = c(u) + \sum_{u' \in \mathbf{S}} p_{u,u'}\, h(u'), \quad u \in \mathbf{S}. \tag{4}$$

When the state-space $\mathbf{S}$ is finite and costs are bounded, then it is known [13] that there exist a unique constant g and a unique function h (up to an additive constant for h) solution to (4). When $\mathbf{S}$ is countable and costs are unbounded, which is the case here, then an argument similar to that developed in [14] can be used to show the existence of a constant g and of a function $h : \mathbf{S} \to \mathbb{R}$ satisfying (4). Uniqueness of the pair (g, h) can also be shown (cf. Spieksma [15]) within a certain normed space. The constant g of this unique normed solution corresponds to the long-run average total expected cost (3). These existence and uniqueness issues are part of ongoing research; giving a full proof here would fall outside the scope of this paper.

Our objective is to determine, in explicit form, both the constant g and the function h, usually referred to as the *bias vector*. The bias in a state u can be considered as the total difference in costs between starting in u and in some fixed reference state. As reference state we take $(0, 0, 1)$.

From the definition of the Markov chain $\{U_n,\, n \geq 1\}$ we may rewrite equations (4) in terms of the model parameters as follows:

$$
\begin{aligned}
(\lambda + \mu_1)\, h(x, y, 1) + g &= c_1 x + c_2 y + s_{1,2}\mu_1 \mathbf{1}(x = 1, y \geq 1) + \lambda_1 h(x + 1, y, 1) \\
&\quad + \lambda_2 h(x, y + 1, 1) + \mu_1 h(x - 1, y, 1)\mathbf{1}(x \geq 2) \\
&\quad + \mu_1 h(0, y, 2)\mathbf{1}(x = 1, y \geq 1), \quad x \geq 1, y \geq 0; \tag{5} \\
(\lambda + \mu_2)h(0, y, 2) + g &= c_2 y + s_{2,1}\lambda_1 + \lambda_1 h(1, y, 1) + \lambda_2 h(0, y + 1, 2) \\
&\quad + \mu_2 h(0, y - 1, 2), \quad y \geq 1; \tag{6} \\
\lambda h(0, 0, z) + g &= s_{2,1}\lambda_1 \mathbf{1}(z = 2) + s_{1,2}\lambda_2 \mathbf{1}(z = 1) + \lambda_1 h(1, 0, 1) \\
&\quad + \lambda_2 h(0, 1, 2), \quad z = 1, 2. \tag{7}
\end{aligned}
$$

We can extend the definition of h to the entire set $\mathbb{N}^2 \times \{1, 2\}$ by setting $h(0, y, 1) = s_1 + h(0, y, 2)$ for all $y \geq 1$ and $h(x, y, 2) = s_2 + h(x, y, 1)$ for all $x \geq 1$, $y \geq 0$. These additional equations correspond to server switches immediately after a departure or after an arrival epoch. With this extension, equations (5)–(7) can be transformed into the following set of equivalent equations on $\mathbf{S}$

$$(\lambda + \mu_1)h(x, y, 1) + g = c_1 x + c_2 y + \lambda_1 h(x + 1, y, 1) + \lambda_2 h(x, y + 1, 1)$$

$$+\mu_1 h(x-1, y, 1), \quad x \geq 1, y \geq 0; \tag{8}$$

$$h(0, y, 1) = s_1 + h(0, y, 2), \quad y \geq 1; \tag{9}$$

$$h(x, y, 2) = s_2 + h(x, y, 1), \quad x \geq 1, \ y \geq 0; \tag{10}$$

$$(\lambda + \mu_2) h(0, y, 2) + g = c_2 y + \lambda_1 h(1, y, 2) + \lambda_2 h(0, y+1, 2)$$

$$+\mu_2 h(0, y-1, 2), \quad y \geq 1; \tag{11}$$

$$(\lambda) h(0, 0, z) + g = \lambda_1 h(1, 0, z) + \lambda_2 h(0, 1, z), \quad z = 1, 2. \tag{12}$$

In the next section we will construct an explicit solution (g, h) to the set of equations (12).

3 Solution of the optimality equation

The theorem below gives a closed-form expression for the bias function h and for the cost g.

Theorem 3.1 *The relative differential equation $h(x, y, z)$ and the average cost g of a two-class M/M/1 queue with a preemptive priority discipline are given by*

$$h(0, 0, 1) = 0, \tag{13}$$

$$h(0, y, 1) = s_1 + h(0, y, 2), \quad y \geq 1; \tag{14}$$

$$h(0, y, 2) = (b_2 + b_2')y + b_2 y^2 + b_4' - s_1, \quad y \geq 0; \tag{15}$$

$$h(x, 0, 1) = (b_1 + b_1')x + b_1 x^2 + b_4'[1 - z^x(\lambda_2)], \quad x \geq 1; \tag{16}$$

$$h(x, y, 2) = s_2 + h(x, y, 1), \quad x \geq 1, y \geq 0; \tag{17}$$

$$h(x, y, 1) = (b_1 + b_1')x + b_1 x^2$$

$$+(b_2 + b_2')y + b_2 y^2 + b_3 xy + b_4', \quad x, y \geq 1, \tag{18}$$

$$g = \lambda_1 \left(2b_1 + b_1' + b_4'[1 - z(\lambda_2)] \right) + \lambda_2 \left(2b_2 + b_2' + b_4' \right) \tag{19}$$

if the stability condition $\sum_{i=1}^{2} \lambda_i / \mu_i < 1$ is met. Here

$$b_1 = \frac{1}{2} \left(\frac{1}{\mu_1 - \lambda_1} \right) \left(c_1 + c_2 \cdot \frac{\lambda_2 \mu_2}{(\mu_1 - \lambda_1)(\mu_2 - \lambda_2) - \lambda_1 \lambda_2} \right); \tag{20}$$

$$b_1' = (s_1 + s_2) \left(\frac{\lambda_1}{\mu_1} \right) \left(\frac{\lambda_1 z(\lambda_2)}{\lambda} - 1 \right); \tag{21}$$

$$b_2 = \frac{1}{2} \cdot \frac{\mu_1 c_2}{(\mu_1 - \lambda_1)(\mu_2 - \lambda_2) - \lambda_1 \lambda_2}; \tag{22}$$

$$b_2' = (s_1 + s_2) \left(\frac{\lambda_1}{\mu_2} \right) \left(\frac{\lambda_1 z(\lambda_2)}{\lambda} \right); \tag{23}$$

$$b_3 = \frac{\mu_2 c_2}{(\mu_1 - \lambda_1)(\mu_2 - \lambda_2) - \lambda_1 \lambda_2}; \tag{24}$$

$$b_4' = \frac{\lambda_1(s_1 + s_2)}{\lambda} \tag{25}$$

and $z(\lambda_2)$ is the unique root in $z \in (0, 1)$ of

$$\lambda_1 z^2 - (\lambda + \mu_1)z + \mu_1 = 0. \tag{26}$$

Before giving a formal proof of this theorem, let us first try to explain the presence of some terms through simple queueing arguments.

Recall that $h(x, y, z)$ is the difference in total cost between starting in state (x, y, z) and starting in the reference state $(0, 0, 1)$.

Let us first focus on holding costs. Note that the presence or absence of class-2 customers does not influence the costs incurred by class-1 customers.

Assume that there are $x \geq 1$ class-1 customers in the system at time $t = 0$. The time until the number of class-1 customers has decreased to $x - 1$ is the same as the duration of a busy period in an M/M/1 queue with arrival rate λ_1 and service rate μ_1, which is equal to $1/(\mu_1 - \lambda_1)$ [7]. During the first busy period the cost per unit time incurred by class-1 customers present in the system at time $t = 0$ is $c_1 x$; during the second busy period this cost reduces to $c_1(x - 1)$, etc. It is therefore expected that the contribution of the holding cost of class-1 customers to $h(x, y, 1)$ for $y \geq 0$ is given by $c_1 \sum_{i=0}^{x-1} (x - i)/(\mu_1 - \lambda_1) = c_1 x(x + 1)/(2(\mu_1 - \lambda_1))$. The reader can check that this term corresponds to the coefficient of c_1 in both equations (16) and (18).

Now consider holding costs incurred by class-2 customers when there are no class-1 customers in the system at time $t = 0$, a situation encountered in (15). Starting from $y \geq 1$ class-2 customers, the time C_2 needed to decrease to $y - 1$ class-2 customers can be obtained by considering that the service of the class-2 customer that has entered the server at time $t = 0$ is interrupted each time a class-1 or a class-2 customer joins the system, and is resumed when all customers which have arrived after time $t = 0$ have been served. In other words, C_2 is the duration of a busy period of an M/G/1 queue with arrival rate λ (recall that $\lambda = \lambda_1 + \lambda_2$), mean service time $(\lambda_1/\mu_1 + \lambda_2/\mu_2)/\lambda$, and with an initial workload equal to the service time of a class-2 customer. Standard queueing arguments show [6, pp. 64-65] that $\mathbf{E}[C_2] = \mu_1/[(\mu_1 - \lambda_1)(\mu_2 - \lambda_2) - \lambda_1 \lambda_2]$. Therefore, the contribution of the holding cost of class-2 customers to $h(0, y, 2)$ for $y \geq 1$ should be given by $c_2 \sum_{i=0}^{x-1} \mathbf{E}[C_2](x - i) = c_2 \mathbf{E}[C_2]y(y + 1)/2$. The reader can check that this term corresponds to $b_2 y + b_2 y^2$ in (15).

Consider now the situation when customers of both classes are present in the system at time $t = 0$ (equation (18)). In this case, all initial class-2 customers have to wait for class-1 customers including all arrivals (class-1 and class-2 customers) before being served. For each class-1 customer this corresponds to the time needed to empty an M/G/1 queue with initially one class-1 customer, arrival rate λ and mean service time $(\lambda_1/\mu_1 + \lambda_2/\mu_2)/\lambda$. Denote this time with C_1; as for C_2 it is readily seen that $\mathbf{E}[C_1] = \mu_2/[(\mu_1 - \lambda_1)(\mu_2 - \lambda_2) - \lambda_1 \lambda_2]$. For each class-1 customer initially present there are holding costs $c_2 \mathbf{E}[C_1] = b_3$ per class-2 customer that is initially present. This explains the xy term in (18).

6

What remains, concerning the holding costs, are the additional costs related to newly arriving class-2 customers whose service is delayed compared to the initially empty situation because of the presence of class-1 customers. The term related to these additional costs is the coefficient of c_2 in b_1. As the number of delayed class-2 customers is related to the length of a class-1 busy period, and as the time serve the additional class-2 customers is equal to C_2, we expect indeed to find both $\mu_1 - \lambda_1$ and $(\mu_1 - \lambda_1)(\mu_2 - \lambda_2) - \lambda_1\lambda_2$ in the denominator. Unfortunately we could not find a simple explanation of the numerator.

Remains the expression for the average holding costs. It can be derived using standard queueing theory on priority queueing (see Kleinrock [8, pp. 124–125]).

With this we have for a large part explained the expression for the holding costs. For the switching costs we were not able to derive the average cost in a straightforward way. By numerical experimentation we found out that the average switching cost is almost, but not quite, quadratic. For finding the additional non-quadratic term we could use the equivalent result with discounting, from Koole and Nain [10]. To explain this, consider the initial state $(1, 0, 1)$. Before reaching the empty state, two things may happen: one or more arrivals of class-2 occur, or no class-2 arrival occurs. The probability of the last event happening is $\mathbf{E}[\exp(-\lambda_2 B)]$, where B is the length of the busy period of an M/M/1 queue with arrival rate and service rate λ_1 and μ_1, respectively. The quantity $\mathbf{E}[\exp(-\lambda_2 B)]$ is equal to the Laplace-Stieltjes transform of B at the point λ_2, denoted as $z(\lambda_2)$. If no arrival in class 2 occurs, then the server stays at class 1. Otherwise the server switches on emptying queue 1 to class 2.

Thus, in state $(x, 0, 1)$, we expect a term of the form $bz(\lambda_2)^x$ occurring in some of the terms. This is indeed what we have found. Note that it is well-known (see Jaiswal [6]) that $z(\lambda_2)$ is the unique root in $z \in (0, 1)$ of

$$\lambda_1 z^2 - (\lambda + \mu_1)z + \mu_1 = 0.$$

Proof of Theorem 3.1. The proof consists in checking that equations (13)–(19) satisfy the optimality equation (8)–(12). Note that all expressions are finite due to the stability condition. We verify (8)–(12) for all possible states.

Verification for state $(0, 0, 1)$. From (19), and (14), (15), and (16) it follows that

$$g \;=\; \lambda_1 h(1, 0, 1) + \lambda_2 h(0, 1, 1). \tag{27}$$

From this (12) for $z = 1$ follows immediately.

Verification for state $(0, 0, 2)$. For state $i = (0, 0, 2)$ the optimality equation looks as follows:

$$h(0, 0, 2) + \frac{g}{\lambda} \;=\; \frac{\lambda_1}{\lambda_1 + \lambda_2} h(1, 0, 2) + \frac{\lambda_2}{\lambda} h(0, 1, 2) \quad \Longrightarrow$$

$$\lambda_1 s_2 - \lambda_2 s_1 + g \;=\; \lambda_1 \left[s_2 + h(1, 0, 1) \right] + \lambda_2 \left[-s_1 + h(0, 1, 1) \right] \quad \Longrightarrow \quad (27).$$

This shows the validity of (12).

Validition of states $(0, y, 1)$ with $y \geq 1$. This case is trivial due to the fact that equation (9) is equal to (14).

7

Validition of states $(0, y, 2)$ *with* $y \geq 1$. Inserting (15), (17) and (18) in (11) and some rewriting gives

$$
\begin{aligned}
g \; = \; & c_2 y + \lambda_1 \left[s_2 + s_1 + b_1 + b_1' + b_1 + b_3 y + h(0, y, 2) \right] \\
& + \lambda_2 \left[b_2 + b_2' + (2y + 1)b_2 + h(0, y, 2) \right] \\
& + \mu_2 \left[-(b_2 + b_2') - (2y - 1)b_2 + h(0, y, 2) \right] - (\mu_2 + \lambda)h(0, y, 2) \quad \Longrightarrow \\
g \; = \; & y \left[c_2 + \lambda_1 b_3 - 2(\mu_2 - \lambda_2)b_2 \right] + \left[\lambda_1(2b_1 + b_1') - \mu_2 b_2' + \lambda_1 b_4' \right] \\
& \left[\lambda_2(2b_2 + b_2') + \lambda_1(s_1 + s_2) - \lambda_1 b_4' \right] \quad \Longrightarrow \\
g \; = \; & \lambda_1 \left[2b_1 + b_1' - b_4' \left(z(\lambda_2) - 1 \right) \right] + \lambda_2 \left[2b_2 + b_2' + b_4' \right].
\end{aligned}
$$

The last step follows due to the y-terms summing up to zero and by definition of b_2, b_2', and b_3. Since this equation is equal to (19), the optimality equation holds.

Verification for states $(x, y, 2)$ *with* $x \geq 1$. This case is trivial as equations (10) and (17) are equal.

Verification for states $(x, 0, 1)$ *with* $x \geq 1$. The optimality equation for state $(x, 0, 1)$ can be rewritten to produce, after inserting the right values from (16) and (18):

$$
\begin{aligned}
g = \; & c_1 x + \lambda_1 \left[b_1 + b_1' + (2x + 1)b_1 - b_4' \left[z^{x+1}(\lambda_2) - z^x(\lambda_2) \right] + h(x, 0, 1) \right] \\
& + \lambda_2 \left[2b_2 + b_2' + b_3 x + s_1 + b_4' + b_4' \left[z^x(\lambda_2) - 1 \right] + h(x, 0, 1) \right] \\
& + \mu_1 \left[-b_1 - b_1' - (2x - 1)b_1 - b_4' \left[z^{x-1}(\lambda_2) - z^x(\lambda_2) \right] + h(x, 0, 1) \right] \\
& - (\lambda + \mu_1)h(x, 0, 1) \Longrightarrow \\
g = \; & x \left[c_1 - 2(\mu_1 - \lambda_1)b_1 + \lambda_2 b_3 \right] + \lambda_1 [2b_1 + b_1' - \frac{\mu_1}{\lambda_1} b_1' - \frac{\lambda_2}{\lambda_1} b_4'] \\
& + \lambda_2 \left[2b_2 + b_2' + b_4' \right] \\
& - \left[b_4' z^{x-1} [\lambda_1 z^2(\lambda_2) - (\lambda + \mu_1)z(\lambda_2) + \mu_1] \right] \Longrightarrow \\
g = \; & \lambda_1 \left[2b_1 + b_1' - b_4' \left[z(\lambda_2) - 1 \right] \right] + \lambda_2 \left[2b_2 + b_2' + b_4' \right].
\end{aligned}
$$

The last step follows due to x-terms summing to 0 and by the definition of $z(\lambda_2)$ as being the unique root of equation (26).

Verification for states $(x, y, 1)$ *and* $(x, y, 1)$ *with* $x \geq 1$ *and* $y \geq 1$. The validation of states $(x, y, 1)$ and $(x, y, 2)$ with $x \geq 1$ and $y \geq 1$ is done in a similar manner.

As $|z(\lambda_2)| < 1$, it is readily seen that h is bounded by a second-order polynomial in x and y. For this reason it has a finite norm in the sense of Spieksma [14]. This shows that we have indeed found the unique solution with bounded norm. ∎

4   Application to a controlled polling system

The server assignment discipline studied so far is that of a fixed priority discipline, namely the preemptive priority resume. See Table 1 for a description of this policy if there are x class-1

customers and y class-2 customers present in the system. The dot represents the situation in which the server remains in its current position, a number means that in this state the server should switch to this queue if not yet present at that queue.

However, the average cost can be lowered by introducing more freedom at the moment where the server can change class. This leads to the model where the server can change at any instant; the question now becomes what the optimal policy is. In the rest of this section we assume $c_1 > 0$ and $c_2 > 0$. In the case that the switching costs are identical to zero the optimal policy is equal to the well-known μc rule (see e.g., Baras et al. [1]). That is, priority is given to a class-i customer based on the value of $\mu_i c_i$; priority is given to the queue with higher value of $\mu_i c_i$. When switching costs are greater than zero, the optimal policy is not known and a numerical method such as policy or value iteration has to be used (see e.g., Puterman [13]).

Being iterative procedures, finding the optimaly policy can be numerically demanding. As the dimension increases, the curse of dimensionality (Bellman [2]) starts playing a role and alternative methods for policy or value iteration have to be used. One alternative used in the literature for different models is performing a single step of policy iteration. This means that, using the value function of a known policy, we choose the minimizing actions. This corresponds, in time, to choosing the minimizing action given that the known initial policy is used for all future decisions. Using this new policy all the time gives an improvement over the initial policy, as can easily be shown in an iterative manner.

As initial policy we take the μc rule and in the optimization step we use the expressions found in Theorem 3.1 for the relative values and the average cost. This implies that the first step of policy improvement can be done by straightforward calculations. Define $\mu = \max\{\mu_1, \mu_2\}$, $\gamma = \lambda + \mu$, e_a is the ath unity vector, and I the indicator function. In order to derive a closed form expression for the one-step improved policy, define for some fixed x and y

$$
\begin{aligned}
z_{k,l} \;=\; & s_k \mathbf{1}\{k \neq l\} + c_1 x + c_2 y + \frac{\lambda_1}{\gamma} h(x+1, y, l) + \frac{\lambda_2}{\gamma} h(x, y+1, l) \\
& + \frac{\mu_l}{\gamma} h(((x,y) - e_l)^+, l) + \frac{\mu - \mu_l}{\gamma} h(x, y, l),
\end{aligned}
$$

as the expected bias if the server immediately switches from position k to position l and uses the preemptive priority rule afterwards. The one-step improved policy is simply the policy that minimizes for each (x, y, k) the expression $\min_a\{z_{k,a}\}$. Define $a_{x,y,k} = \arg\min_a\{z_{k,a}\}$. If we assume that s_1, $s_2 \geq 0$, then it is easily seen that either $a_{x,y,1} = a_{x,y,2}$ or $a_{x,y,k} = k$ for $k = 1, 2$. Define $a_{x,y} = a_{x,y,1}$ if $a_{x,y,1} = a_{x,y,2}$ and $a_{x,y} = 0$ if $a_{x,y,k} = k$ for $k = 1, 2$. Thus $a_{x,y} = 0$ means that the server remains where it is; if $a_{x,y} > 0$ then the server goes to class $a_{x,y}$, irrespective of its current class. An alternative way to write $a_{x,y}$ is as follows:

$$
a_{x,y} = \mathbf{1}\left(z_{1,1} < z_{1,2}\right) \cdot \mathbf{1}\left(z_{2,1} < z_{2,2}\right) + 2 \cdot \mathbf{1}\left(z_{1,1} \geq z_{1,2}\right) \cdot \mathbf{1}\left(z_{2,1} \geq z_{2,2}\right), \quad x, y \in I\!N.
$$

We will simplify this expression for several choices of x and y.

The one-step improved policy for $x = 0$ and $y \geq 1$ is derived as follows.

$$
a_{x,y} \;=\; \mathbf{1}\left(\frac{c_2 y}{\gamma} + \frac{\lambda_1}{\gamma} h(1, y, 1) + \frac{\lambda_2}{\gamma} h(0, y+1, 1) + \frac{\mu}{\gamma} h(0, y, 1) < s_1 + \frac{c_2 y}{\gamma}\right.
$$

$$+\frac{\lambda_1}{\gamma}h(1,y,2) + \frac{\lambda_2}{\gamma}h(0,y+1,2) + \frac{\mu_2}{\gamma}h(0,y-1,2) + \frac{\mu-\mu_2}{\gamma}h(0,y,2)\Big)$$

$$\cdot \mathbf{1}\Big(s_2 + \frac{c_2 y}{\gamma} + \frac{\lambda_1}{\gamma}h(1,y,1) + \frac{\lambda_2}{\gamma}h(0,y+1,1) + \frac{\mu}{\gamma}h(0,y,1) < \frac{c_2 y}{\gamma} + \frac{\lambda_1}{\gamma}h(1,y,2)$$

$$+\frac{\lambda_2}{\gamma}h(0,y+1,2) + \frac{\mu_2}{\gamma}h(0,y-1,2) + \frac{\mu-\mu_2}{\gamma}h(0,y,2)\Big)$$

$$+2\cdot\mathbf{1}\Big(\frac{c_2 y}{\gamma} + \frac{\lambda_1}{\gamma}h(1,y,1) + \frac{\lambda_2}{\gamma}h(0,y+1,1) + \frac{\mu}{\gamma}h(0,y,1) > s_1 + \frac{c_2 y}{\gamma}$$

$$+\frac{\lambda_1}{\gamma}h(1,y,2) + \frac{\lambda_2}{\gamma}h(0,y+1,2) + \frac{\mu_2}{\gamma}h(0,y-1,2) + \frac{\mu-\mu_2}{\gamma}h(0,y,2)\Big)$$

$$\cdot\mathbf{1}\Big(s_2 + \frac{c_2 y}{\gamma} + \frac{\lambda_1}{\gamma}h(1,y,1) + \frac{\lambda_2}{\gamma}h(0,y+1,1) + \frac{\mu}{\gamma}h(0,y,1) > \frac{c_2 y}{\gamma} + \frac{\lambda_1}{\gamma}h(1,y,2)$$

$$+\frac{\lambda_2}{\gamma}h(0,y+1,2) + \frac{\mu_2}{\gamma}h(0,y-1,2) + \frac{\mu-\mu_2}{\gamma}h(0,y,2)\Big),$$

$$= \mathbf{1}\Big(s_1\gamma + \lambda_1 s_2 - \lambda_2 s_1 - \mu s_1 - \mu_2(2b_2 y + b_2') > 0\Big)$$

$$\cdot\mathbf{1}\Big(-s_2\gamma + \lambda_1 s_2 - \lambda_2 s_1 - \mu s_1 - \mu_2(2b_2 y + b_2') > 0\Big)$$

$$+2\cdot\mathbf{1}\Big(s_1\gamma + \lambda_1 s_2 - \lambda_2 s_1 - \mu s_1 - \mu_2(2b_2 y + b_2') < 0\Big)$$

$$\cdot\mathbf{1}\Big(-s_2\gamma + \lambda_1 s_2 - \lambda_2 s_1 - \mu s_1 - \mu_2(2b_2 y + b_2') < 0\Big),$$

$$= \mathbf{1}\Big(\mu_2(2b_2 y + b_2') - \lambda_1(s_1 + s_2) < 0\Big)$$

$$\cdot\mathbf{1}\Big(\mu_2(2b_2 y + b_2') + (\mu + \lambda_2)(s_1 + s_2) < 0\Big)$$

$$+2\cdot\mathbf{1}\Big(\mu_2(2b_2 y + b_2') - \lambda_1(s_1 + s_2) > 0\Big)$$

$$\cdot\mathbf{1}\Big(\mu_2(2b_2 y + b_2') + (\mu + \lambda_2)(s_1 + s_2) > 0\Big),$$

$$= 2\cdot\mathbf{1}\Big(\frac{c_2 y}{1 - \lambda_1/\mu_1 - \lambda_2/\mu_2} > (s_1 + s_2)\Big(\frac{\lambda_1}{\lambda}\Big)\Big(\lambda_1(1 - z(\lambda_2)) + \lambda_2\Big)\Big).$$

The last step follows due to $y, \lambda_2, \mu, \mu_2, b_2, b_2', s_1$, and s_2 all being greater than zero causing the first, second, and forth term to disappear. This result tells us that for $x = 0$ and $y \geq 1$ the server will *never* switch to position one with one-step improvement of the μc rule, and that if there are no switching costs, the server will *always* switch to position two, if not already there. When switching costs are positive, the decision is based upon a weighted comparison of the holding and switching costs.

Derivation of the one-step improved policy for the other three parts is done in a similar manner.

For $x \geq 1$ and $y \geq 1$ the result is

$$
\begin{aligned}
a_{x,y} \;=\; & \mathbf{1}\left((\mu_1 b_1 - \mu_2 b_3)x > (s_1 + s_2)\left(\lambda_1 + \left(\frac{\lambda_1 z^x(\lambda_2)}{\lambda_1 + \lambda_2} \right) \cdot \mathbf{1}(y=1) \right) \right) \\
& + 2 \cdot \mathbf{1}\left((\mu_1 b_1 - \mu_2 b_3)x < (s_1 + s_2)\left(-\mu - \lambda_2 + \left(\frac{\lambda_1 z^x(\lambda_2)}{\lambda} \right) \cdot \mathbf{1}(y=1) \right) \right),
\end{aligned}
$$

whereas for $x = 0$ and $y = 0$ the outcome will always be $a_{0,0} = 0$. For $x > 0$ and $y = 0$ the outcome will always be $a_{x,0} = 1$. If the switching costs are equal to zero, then the one-step policy improvement of the μc rule results in the μc rule again. This proves in an alternative manner that the μc rule is the optimal policy which minimizes the long-term average cost if there are no switching costs. In Table 2 the one-step improved policy is given for a certain choice of parameters.

The policy depicted in Table 2 is identical to the one found in Koole & Nain [10] for the discounted costs for the same choice of parameters. The average cost resulting from using a single step of policy iteration is 3.09895. This is a reduction of the costs by 14.6% as compared to using the μc-rule where the average cost is 3.62894. By using policy iteration to find the optimal policy the results hardly improve; the average cost is at lowest 3.09261. Surprisingly enough, the optimal policy is found in two steps of policy iteration, see Tables 3 and 4. The fast convergence of the policy iteration algorithm is not coincidental; the average cost of the policies generated by policy iteration converge at least exponentially fast to the minimum cost, with the greatest improvement in the first few steps (see Tijms [16], page 194).

Acknowledgement

References

[1] J.S. Baras, D.-J. Ma, and A.M. Makowski. *K* competing queues with geometric service requirements and linear costs: The μc-rule is always optimal. *Systems and Control Letters*, 6:173–180, 1985.

[2] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[3] D.P. Bertsekas and J.N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[4] O.J. Boxma and D.G. Down. Dynamic server assignment in a two-queue model. *European Journal of Operational Research*, 103:595–609, 1997.

[5] O.J. Boxma, G.M. Koole, and I. Mitrani. Polling models with threshold switching. In F. Baccelli, A. Jean-Marie, and I. Mitrani, editors, *Quantitative Methods in Parallel Systems*, pages 129–140. Springer-Verlag, 1995. Esprit Basic Research Series.

[6] N.K. Jaiswal. *Priority Queues*. Academic Press, 1968.

[7] L. Kleinrock. *Queueing Systems, Volume I: Theory*. Wiley, 1975.

[8] L. Kleinrock. *Queueing Systems, Volume II: Computer Applications*. Wiley, 1976.

[9] G.M. Koole. Assigning a single server to inhomogeneous queues with switching costs. *Theoretical Computer Science*, 182:203–216, 1997.

[10] G.M. Koole and P. Nain. On the value function of a priority queue with an application to a controlled polling model. *Queueing Systems*, 34:199–214, 2000.

[11] P. Marbach, O. Mihatsch, and J.N. Tsitsiklis. Call admission control and routing in integrated service networks using neuro-dynamic programming. *IEEE Journal on Selected Areas in Communications*, 18:197–208, 2000.

[12] T.J. Ott and K.R. Krishnan. Separable routing: A scheme for state-dependent routing of circuit switched telephone traffic. *Annals of Operations Research*, 35:43–68, 1992.

[13] M.L. Puterman. *Markov Decision Processes*. Wiley, 1994.

[14] F.M. Spieksma. The existence of sensitive optimal policies in two multi-dimensional queueing models. *Annals of Operations Research*, 28:273–296, 1991.

[15] F.M. Spieksma. Personal communication. 2001.

[16] H.C. Tijms. *Stochastic Models. An Algorithmic Approach*. Wiley, 1986.

y	x=0	1	2	3	4	5	6	7	8	9	10
10	2	1	1	1	1	1	1	1	1	1	1
9	2	1	1	1	1	1	1	1	1	1	1
8	2	1	1	1	1	1	1	1	1	1	1
7	2	1	1	1	1	1	1	1	1	1	1
6	2	1	1	1	1	1	1	1	1	1	1
5	2	1	1	1	1	1	1	1	1	1	1
4	2	1	1	1	1	1	1	1	1	1	1
3	2	1	1	1	1	1	1	1	1	1	1
2	2	1	1	1	1	1	1	1	1	1	1
1	2	1	1	1	1	1	1	1	1	1	1
0	·	1	1	1	1	1	1	1	1	1	1

Table 1: Class the server goes to under the preemptive priority rule

y	x=0	1	2	3	4	5	6	7	8	9	10
10	2	·	·	1	1	1	1	1	1	1	1
9	2	·	·	1	1	1	1	1	1	1	1
8	2	·	·	1	1	1	1	1	1	1	1
7	2	·	·	1	1	1	1	1	1	1	1
6	2	·	·	1	1	1	1	1	1	1	1
5	2	·	·	1	1	1	1	1	1	1	1
4	2	·	·	1	1	1	1	1	1	1	1
3	2	·	·	1	1	1	1	1	1	1	1
2	2	·	·	1	1	1	1	1	1	1	1
1	·	·	·	·	1	1	1	1	1	1	1
0	·	1	1	1	1	1	1	1	1	1	1

Table 2: One step improvement for $\lambda_1 = \lambda_2 = 1$, $\mu_1 = 6$, $\mu_2 = 3$, $c_1 = 2$, $c_2 = 1$, $s_1 = s_2 = 2$. Here $g = 3.09895$.

Iteration	Average cost	Comment
0	3.62894	μc rule
1	3.09895	One-step improvement
2	3.09261	Optimal policy

Table 3: Policy iteration results

y	x=0	1	2	3	4	5	6	7	8	9	10
10	2	·	1	1	1	1	1	1	1	1	1
9	2	·	1	1	1	1	1	1	1	1	1
8	2	·	1	1	1	1	1	1	1	1	1
7	2	·	1	1	1	1	1	1	1	1	1
6	2	·	1	1	1	1	1	1	1	1	1
5	2	·	1	1	1	1	1	1	1	1	1
4	2	·	1	1	1	1	1	1	1	1	1
3	2	·	1	1	1	1	1	1	1	1	1
2	2	·	·	1	1	1	1	1	1	1	1
1	2	·	·	·	1	1	1	1	1	1	1
0	·	1	1	1	1	1	1	1	1	1	1

Table 4: Optimal policy for $\lambda_1 = \lambda_2 = 1, \mu_1 = 6, \mu_2 = 3, c_1 = 2, c_2 = 1, s_1 = s_2 = 2$. Here $g = 3.09261$.