

## ON THE STRUCTURE OF VALUE FUNCTIONS FOR THRESHOLD POLICIES IN QUEUEING MODELS

SANDJAI BHULAI\* AND  
GER KOOLE,\*\* *Vrije Universiteit Amsterdam*

### Abstract

We study the multiserver queue with Poisson arrivals and identical independent servers with exponentially distributed service times. Customers arriving at the system are admitted or rejected according to a fixed threshold policy. Moreover, the system is subject to holding, waiting, and rejection costs. We give a closed-form expression for the average costs and the value function for this multiserver queue. The result will then be used in a single step of policy iteration in the model where a controller has to route to several finite-buffer queues with multiple servers. We numerically show that the improved policy has a close to optimal value.

*Keywords:* Difference equation; multiserver queue; one-step policy improvement; threshold policy; value function

AMS 2000 Subject Classification: Primary 93E25  
Secondary 60K25

### 1. Introduction

The application of Markov decision theory to the control of queueing networks often leads to models with enormous state and action spaces. Hence, direct computation of optimal policies with standard techniques and algorithms is almost impossible for most practical models. This phenomenon is also known as ‘the curse of dimensionality’. Consequently, there is a need for other exact or good approximation methods that avoid this problem.

Bertsekas and Tsitsiklis [1] discuss reinforcement learning. This approximation method constructs approximations to the value function based on a certain functional form. However, choosing the initial functional form such that the resulting approximations are good is difficult. It requires a great deal of insight into the system under study, and in particular to the dynamic programming optimality equations.

Ott and Krishnan [6] introduce the idea of applying one-step policy improvement. In this case, we are required to obtain an explicit solution to the optimality equations for a fixed policy. The result will then be used in one step of the policy iteration algorithm from Markov decision theory in order to obtain an improved policy.

In this paper we study the multiserver queue with Poisson arrivals and identical independent servers with exponentially distributed service times. Customers arriving at the system are admitted or rejected according to a fixed threshold policy. The motivation for studying threshold policies stems from the fact that threshold policies are optimal or close to optimal in many

---

Received 17 May 2001; revision received 27 November 2002.

\* Current address: Bell Laboratories, Lucent Technologies, 700 Mountain Avenue, Room 2C-313, Murray Hill, NJ 07974-0636, USA. Email address: sbhulai@research.bell-labs.com

\*\* Postal address: Vrije Universiteit, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands.

queueing systems (see e.g. [8]). Hence, we can expect to obtain improved policies which are good approximations by applying reinforcement learning or one-step policy improvement.

Additionally, the system is subject to holding, waiting, and rejection costs. Note that this model can also be viewed as a multiserver queue with a finite buffer where no control is applied (the standard M/M/s/c queue). Our main result is an explicit solution to the optimality equations under the threshold policy. From this expression we can also obtain the solution to the optimality equations of the single- and infinite-server models.

The main contribution of our result with respect to reinforcement learning is the structure of the value function for the single-, multiple-, and infinite-server queues. It provides a theoretical foundation for choices of the functional equation. The solution to the optimality equations of the three models can readily be used in the one-step policy improvement method. This is illustrated in Section 4 where a routing problem for parallel queues is studied. We numerically show that the improved policy has a close to optimal value.

### 2. Multiserver queue

Consider a queueing system with one queue and  $s$  identical independent servers. The arrivals are determined by a Poisson process with parameter  $\lambda$ . The service times are exponentially distributed with parameter  $\mu$ . Let state  $x$  denote the number of customers in the system. A controller decides to admit or reject customers arriving at the system according to a threshold policy with threshold level  $c \in \mathbb{N}_0 = \{0, 1, \dots\}$ . Thus, if a customer arrives while  $x < c$ , the controller will decide to admit the customer. The controller rejects the customer when  $x \geq c$ . Hence, when starting with an empty system, the states are limited to  $x \in \{0, \dots, c\}$ . Note that a threshold level of  $c = 0$  rejects every customer, whereas the limiting case  $c \rightarrow \infty$  admits every customer.

Additionally, assume that the system is subject to holding, waiting, and rejection costs. Let  $u_t^c(i)$  denote the total expected costs up to time  $t$  when the system starts in state  $i$  under the threshold policy with level  $c$ . The Markov chain satisfies the unichain assumption and, due to Proposition 8.2.1 of [7], the average costs  $\varphi_c = \lim_{t \rightarrow \infty} u_t^c(i)/t$  are independent of the initial state  $i$ . By the same proposition we see that the result also holds for the limiting case  $c \rightarrow \infty$  under the assumption that  $\rho = \lambda/s\mu < 1$ . The dynamic programming optimality equations for the system are thus given by

$$\begin{aligned} \varphi_c + \lambda V_c(0) &= \lambda V_c(1), \\ \varphi_c + (\lambda + x\mu)V_c(x) &= hx + \lambda V_c(x + 1) + x\mu V_c(x - 1), \quad x = 1, \dots, s - 1, \\ \varphi_c + (\lambda + s\mu)V_c(x) &= hx + \lambda w(x - s + 1) + \lambda V_c(x + 1) + s\mu V_c(x - 1), \quad (1) \\ &\quad x = s, \dots, c - 1, \\ \varphi_c + s\mu V_c(c) &= hc + \lambda r + s\mu V_c(c - 1). \end{aligned}$$

In this set of equations the constants  $h$ ,  $w$ , and  $r$  denote the holding, waiting, and rejection costs respectively. Note that we use  $w(x - s + 1)$  instead of  $w(x - s)$ , since this term is conditioned on an arriving customer. The function  $V_c(x)$  is called the (relative) value function, and is the quantity that we are interested in. The function  $V_c(x)$  can be interpreted as the asymptotic difference in total reward that results from starting the process in state  $x$  instead of some reference state  $y$ . Without loss of generality, we take the reference state to be  $y = 0$ . Observe that due to linearity the value function can be decomposed as  $V_c(x) = V_c^h(x) + V_c^w(x) + V_c^r(x)$ ,

the summands of which are due to holding, waiting, and rejection costs respectively. In the same way, the average cost can be decomposed as  $\varphi_c = \varphi_c^h + \varphi_c^w + \varphi_c^r$ .

We adopt the following approach to solving the optimality equations. We first consider the optimality equations for  $x = 0, \dots, s - 1$ . Since by definition  $V_c(0) = 0$ , the equations have a unique solution expressed in terms of  $\varphi_c$ . The solution also holds for  $x = s$  when considering holding and rejection costs. Then the optimality equations for  $x = s, \dots, c - 1$  are solved. In this case  $V_c(s - 1)$  or  $V_c(s)$  is known and again guarantees a unique solution expressed in terms of  $\varphi_c$ . Finally, the equation for  $x = c$  is considered. This equation provides an expression for  $\varphi_c$ , which solves the complete system explicitly.

The optimality equations are so-called linear, inhomogeneous, second-order difference equations. Mickens [5] gives a good overview of the theory of difference equations. The expression (3.112) is especially helpful since it provides the structure of the solution to second-order difference equations when one solution is known. Before solving the optimality equations, first define the hypergeometric function  $F(x)$  by

$$F(x) = \sum_{k=0}^{x-1} \frac{\Gamma(x)}{\Gamma(x-k)} \left(\frac{\mu}{\lambda}\right)^k,$$

with  $\Gamma(x) = (x - 1)!$  when  $x$  is integer. Then the first step of our approach is given by the following theorem.

**Theorem 1.** *Let  $k \in \{h, w, r\}$ , and consider the optimality equations for states  $x = 0, \dots, s$  when  $k \in \{h, r\}$ , and  $x = 0, \dots, s - 1$  when  $k = w$ . The unique solution to this set of equations is given by*

$$V_c^k(x) = \frac{\varphi_c^k}{\lambda} \sum_{i=1}^x F(i) - \frac{\alpha(k)}{\lambda} \sum_{i=1}^x (i - 1)F(i - 1), \tag{2}$$

with  $\alpha(h) = h$ ,  $\alpha(w) = 0$ , and  $\alpha(r) = 0$ .

*Proof.* Let  $\Delta V(x) = V(x) - V(x - 1)$  and let  $k \in \{h, w, r\}$ . Then the optimality equations for  $x = 0, \dots, s$  when  $k \in \{h, r\}$ , and  $x = 0, \dots, s - 1$  when  $k = w$  can be written as

$$\varphi_c^k + x\mu\Delta V_c^k(x) - \alpha(k)x - \lambda\Delta V_c^k(x + 1) = 0,$$

with  $\Delta V_c^k(0) = 0$  by definition. We check that the solution (2) satisfies this expression by substitution into this expression. First observe that  $(\lambda/\mu)F(x + 1) = \lambda/\mu + xF(x)$ , and thus

$$\begin{aligned} & \varphi_c^k + x\mu\Delta V_c^k(x) - \alpha(k)x - \lambda\Delta V_c^k(x + 1) \\ &= \varphi_c^k + \frac{x\mu\varphi_c^k}{\lambda} F(x) - \frac{x\mu\alpha(k)}{\lambda} (x - 1)F(x - 1) - \alpha(k)x - \lambda\Delta V_c^k(x + 1) \\ &= \varphi_c^k + \frac{x\mu\varphi_c^k}{\lambda} F(x) - \alpha(k)x F(x) + \alpha(k)x - \alpha(k)x - \lambda\Delta V_c^k(x + 1) \\ &= \varphi_c^k + \frac{x\mu\varphi_c^k}{\lambda} F(x) - \alpha(k)x F(x) - \varphi_c^k F(x + 1) + \alpha(k)x F(x) \\ &= \varphi_c^k + \varphi_c^k F(x + 1) - \varphi_c^k - \varphi_c^k F(x + 1) \\ &= 0. \end{aligned}$$

Note that, since  $V_c^k(0) = 0$ , the solution is also unique.

The first term in the value function (2) is the particular solution to  $\varphi_c^k$  in the optimality equation. Therefore, this term appears in  $V_c^k(x)$  for all  $k \in \{h, w, r\}$ . The second term is the particular solution to the costs. Since in this case no waiting or rejections occur, the second term is zero in  $V_c^w(x)$  and  $V_c^r(x)$ . The terms are rather complicated due to the fact that the rates in the optimality equation are dependent on the state. This does not occur when the rates are constant. The following theorem shows this for the solution of the optimality equations for  $x = s, \dots, c - 1$ .

**Theorem 2.** Consider the optimality equations for  $x = s, \dots, c - 1$ . Let  $\rho = \lambda/s\mu$  and  $\Delta V_c^k(x) = V_c^k(x) - V_c^k(x - 1)$ . Then the unique solution to this set of equations is given by

$$\begin{aligned} V_c^k(x) = & -\frac{(x - \sigma(k))\rho}{1 - \rho} \frac{\varphi_c^k}{\lambda} \\ & + \left[ \frac{(x - \sigma(k))(x - \sigma(k) + 1)\rho}{2(1 - \rho)} + \frac{(x - \sigma(k))\rho(\rho + \gamma(k))}{(1 - \rho)^2} \right] \frac{\beta(k)}{\lambda} \\ & + V_c^k(\sigma(k)) \\ & + \frac{(1/\rho)^{x - \sigma(k)} - 1}{1 - \rho} \left[ \frac{\rho}{1 - \rho} \frac{\varphi_c^k}{\lambda} + \frac{\sigma(k)}{s} \Delta V_c^k(\sigma(k)) - \frac{\rho(\rho + \gamma(k))}{(1 - \rho)^2} \frac{\beta(k)}{\lambda} \right] \quad (3) \end{aligned}$$

for  $k \in \{h, w, r\}$ , with  $\sigma(k) = s - \mathbf{1}_{\{k=w\}}$ ,  $\gamma(k) = s(1 - \rho) \mathbf{1}_{\{k \neq w\}}$ ,  $\beta(h) = h$ ,  $\beta(w) = \lambda w$ , and  $\beta(r) = 0$ .

*Proof.* Note that the optimality equations for  $k \in \{h, w, r\}$  and  $x = s, \dots, c - 1$  can be written as

$$\varphi_c^k + s\mu \Delta V_c^k(x) - \left[ x - \sigma(k) + \frac{\gamma(k)}{1 - \rho} \right] \beta(k) - \lambda \Delta V_c^k(x + 1) = 0.$$

The first term in the solution (3) is the particular solution to  $\varphi_c^k$  in the optimality equations. Similarly, the second term is the particular solution to the inhomogeneous term  $[x - \sigma(k) + \gamma(k)/(1 - \rho)]\beta(k)$ . This fact follows directly by substitution into the optimality equations, as follows:

$$\begin{aligned} & \varphi_c^k + s\mu \Delta V_c^k(x) - \left[ x - \sigma(k) + \frac{\gamma(k)}{1 - \rho} \right] \beta(k) - \lambda \Delta V_c^k(x + 1) \\ &= \varphi_c^k - \frac{1}{1 - \rho} \varphi_c^k + \left[ \frac{x - \sigma(k)}{1 - \rho} + \frac{\rho + \gamma(k)}{(1 - \rho)^2} \right] \beta(k) \\ & \quad - \left[ x - \sigma(k) + \frac{\gamma(k)}{1 - \rho} \right] \beta(k) - \lambda \Delta V_c^k(x + 1) \\ &= -\frac{\rho}{1 - \rho} \varphi_c^k + \left[ \frac{(x - \sigma(k))\rho}{1 - \rho} + \frac{(1 + \gamma(k))\rho}{(1 - \rho)^2} \right] \beta(k) - \lambda \Delta V_c^k(x + 1) \\ &= \left[ \frac{(x - \sigma(k))\rho}{1 - \rho} + \frac{(1 + \gamma(k))\rho}{(1 - \rho)^2} \right] \beta(k) - \left[ \frac{((x + 1) - \sigma(k))\rho}{1 - \rho} + \frac{(\rho + \gamma(k))\rho}{(1 - \rho)^2} \right] \beta(k) \\ &= 0. \end{aligned}$$

The optimality equations also have two homogeneous solutions, which are given by  $a_1$  and  $a_2((1/\rho)^{x-\sigma(k)} - 1)/(1 - \rho)$ , where  $a_1$  and  $a_2$  are constants. This can be seen from Theorem 4.2 of [5]. Since  $V_c^k(x)$  equals the value function of Theorem 1 at  $x = s$  in case  $k \in \{h, r\}$ , and at  $x = s - 1$  when  $k = w$ , we have  $a_1 = V_c^k(\sigma(k))$ . The coefficient  $a_2$  easily follows from solving the equation on the boundary  $\varphi_c^k + \gamma(k)\mu\Delta V_c^k(\gamma(k)) - \alpha(k)x - \lambda\Delta V_c^k(\gamma(k) + 1) = 0$ . These coefficients fix the value of  $V_c^k(\gamma(k))$  and  $V_c^k(\gamma(k) + 1)$ . Hence, the solution is also unique.

Theorem 1 and Theorem 2 fully characterize the solution to the dynamic programming optimality equations expressed in  $\varphi_c^k$ . The optimality equation for state  $x = c$  can now be used to explicitly determine  $\varphi_c^k$ . This will also explicitly determine the solutions to the complete set of optimality equations. The results are given by the following theorem.

**Theorem 3.** *The average costs  $\varphi_c^k$  for  $k \in \{h, w, r\}$  are given by*

$$\begin{aligned} \varphi_c^h &= \left[ \frac{1}{\rho} F(s) + \frac{1 - \rho^{c-s+1}}{1 - \rho} \right]^{-1} \\ &\quad \times \left[ sF(s) + c\rho^{c-s} + \frac{s\rho}{1 - \rho} + \frac{\rho}{(1 - \rho)^2} - \frac{\rho^{c-s+1}}{(1 - \rho)^2} - \frac{c\rho^{c-s}}{1 - \rho} \right] h, \\ \varphi_c^w &= \left[ \frac{s-1}{s\rho} F(s-1) + \frac{1 - \rho^{c-s+2}}{1 - \rho} \right]^{-1} \left[ \frac{\rho}{(1 - \rho)^2} - \frac{(c-s+1)\rho^{c-s+1}}{1 - \rho} - \frac{\rho^{c-s+2}}{(1 - \rho)^2} \right] \lambda w, \\ \varphi_c^r &= \left[ \frac{1}{\rho} F(s) + \frac{1 - \rho^{c-s+1}}{1 - \rho} \right]^{-1} \rho^{c-s} \lambda r. \end{aligned}$$

*Proof.* The optimality equations for  $x = c$  and  $k \in \{h, w, r\}$  can be written as

$$\varphi_c^k + s\mu\Delta V_c^k(c) - \mathbf{1}_{\{k=h\}} hc - \mathbf{1}_{\{k=r\}} \lambda r = 0.$$

After substitution of (3) from Theorem 2 we get an equation in  $\varphi_c^k$  only. With some calculus, it can be shown that the solution is indeed as stated in the theorem.

The set of optimality equations is now solved, and we have derived an explicit solution expressed in the parameters  $\lambda$ ,  $\mu$ ,  $s$ , and  $c$ . Note that we did not require any restriction on the parameters. However, when we consider the limiting case  $c = \infty$ , we require stability of the queueing system, i.e.  $\rho < 1$ . Assuming that the stability condition holds, we can directly obtain that

$$\varphi_\infty^r = 0, \tag{4}$$

$$\frac{\varphi_\infty^h}{h} = \frac{\varphi_\infty^w}{ws\mu} + \frac{\lambda}{\mu}. \tag{5}$$

The first equation (4) follows directly from Theorem 3 when taking the limit, since we assumed that  $\rho < 1$ . Indeed, when all customers are admitted, the costs due to rejection are zero. The equation (5) is more involved and is explained as follows. The mean time spent waiting in the queue is obtained when  $w = 1/\lambda s\mu$ . Adding the mean service time  $1/\mu$  to it gives the mean sojourn time. Applying Little's formula gives the mean queue length, and thus explains (5).

Let us now compute the expression for  $\varphi_\infty^h$ . For simplicity we assume that  $h = 1$ . Then from Theorem 3 it follows that

$$\begin{aligned} \varphi_\infty^h &= \lim_{c \rightarrow \infty} \varphi_c^h \\ &= \left[ \frac{1}{\rho} F(s) + \frac{1}{1-\rho} \right]^{-1} \left[ sF(s) + \frac{s\rho}{1-\rho} + \frac{\rho}{(1-\rho)^2} \right] \\ &= \frac{\rho}{(1-\rho)^2} \left[ \frac{1}{\rho} F(s) + \frac{1}{1-\rho} \right]^{-1} + s\rho \\ &= \frac{(s\rho)^s \rho}{s!(1-\rho)^2} \left[ \frac{(s\rho)^{s-1}}{\Gamma(s)} F(s) + \frac{(s\rho)^s}{s!(1-\rho)} \right]^{-1} + s\rho \\ &= \frac{(s\rho)^s \rho}{s!(1-\rho)^2} \left[ \sum_{n=0}^{s-1} \frac{(s\rho)^n}{n!} + \frac{(s\rho)^s}{s!(1-\rho)} \right]^{-1} + s\rho. \end{aligned}$$

Note that we have derived in an alternative way the well-known expression for the average queue length in a multiserver queueing system with infinite buffer (see e.g. Section 2.3 of [2]).

In the literature it is usual to try to derive the value function for a specific policy (see e.g. [3] and [6]). However, the results of Theorems 1–3 concern a class of policies, in contrast to a specific policy. The results can therefore be used for finding the best threshold policy within the class. Observe that optimizing with respect to the threshold level  $c$  is not difficult.

Koole and Spieksma [4] obtain expressions for deviation matrices for birth–death processes, and in particular for the M/M/s/c queue. The deviation matrix is independent of the cost structure. Hence, it enables us to compute the average costs and the relative value function for various cost structures (depending on the state only) by evaluating a sum involving entries of the deviation matrix. However, the expressions they derive for the deviation matrix are very complicated. Therefore, evaluating the sum is not easy in many situations. The method adopted in this paper shows the benefit of working with costs integrated into the problem formulation. The expressions are simpler and are easier to obtain in contrast to either working with deviation matrices or equilibrium probabilities.

The results also explicitly depict the structure of the value function. This information can be fruitfully used in reinforcement learning to guess the structure or even the values of value functions for other queueing models. As will become clear in Section 3, the value function of the infinite-buffer single-server queue is a sum of linear and quadratic terms. However, the finite-buffer single-server queue also contains exponential terms. When the rates in the optimality equation depend on the states (as in the infinite-server queue), then hypergeometric terms also appear.

The results can also be directly applied to one-step policy improvement. In this setting, a policy is chosen with the property that the value function and the average costs can be computed. Then this policy will be used in one step of the policy iteration algorithm from Markov decision theory resulting in an improved policy. The improved policy will in general be sufficiently complicated to render another step of policy iteration impossible. In Section 4, we will discuss routing to several parallel finite-buffer queues. Before doing that, we first consider special cases of the multiserver queue.

### 3. Special cases

In this section, we consider special cases of the multiserver queue. The case where  $s = 1$  results in the single-server queue. The case with  $s = c$  results in the infinite-server queue. We will discuss both the finite-buffer ( $c$  finite) and the infinite-buffer ( $c = \infty$  and  $\rho < 1$ ) model. We first start with the treatment of the single-server queue.

#### 3.1. Single-server queue

The single-server queue can be obtained by considering the multiserver queue with one server only, i.e.  $s = 1$ . In this case the optimality equations (1) become simpler:

$$\begin{aligned} \varphi_c + \lambda V_c(0) &= \lambda V_c(1), \\ \varphi_c + (\lambda + \mu)V_c(x) &= hx + \lambda wx + \lambda V_c(x + 1) + \mu V_c(x - 1), \quad x = 1, \dots, c - 1, \\ \varphi_c + \mu V_c(c) &= hc + \lambda r + \mu V_c(c - 1). \end{aligned}$$

The solution to this set of equations is given by Theorems 2 and 3 with  $s = 1$ . After some tedious calculus, the value function can be derived as

$$V_c^k(x) = a_1^k \frac{(1/\rho)^x - 1}{1 - \rho} + \frac{x(x + 1)}{2\mu(1 - \rho)} \beta(k) - a_1^k x$$

for  $k \in \{h, w, r\}$ , with  $\beta(h) = h$ ,  $\beta(w) = \lambda w$ , and  $\beta(r) = 0$ . The constants  $a_1^k$  for  $k \in \{h, w, r\}$  and the average costs are given by

$$\begin{aligned} a_1^h &= -\frac{(c + 1)\rho}{\mu((1/\rho)^c - \rho)(1 - \rho)} h, & \varphi_c^h &= \left[ 1 - \frac{(c + 1)(1 - \rho)}{(1/\rho)^c - \rho} \right] \frac{\rho}{1 - \rho} h, \\ a_1^w &= -\frac{(c + \rho)\rho}{((1/\rho)^c - \rho)(1 - \rho)} w, & \varphi_c^w &= \left[ 1 - \frac{(c + \rho)(1 - \rho)}{(1/\rho)^{c-1} - \rho^2} \right] \frac{\rho}{1 - \rho} \lambda w, \\ a_1^r &= \frac{\rho}{(1/\rho)^c - \rho} r, & \varphi_c^r &= \frac{1 - \rho}{(1/\rho)^c - \rho} \lambda r. \end{aligned}$$

The average costs and the value function for the single-server queue with an infinite buffer, or, equivalently, with no rejections, are given by

$$\varphi_\infty^k = \lim_{c \rightarrow \infty} \varphi_c^k = \frac{\rho}{1 - \rho} \beta(k) \quad \text{and} \quad V_\infty^k(x) = \lim_{c \rightarrow \infty} V_c^k(x) = \frac{x(x + 1)}{2\mu(1 - \rho)} \beta(k).$$

The value function of the infinite-buffer single-server queue is thus the sum of linear and quadratic terms. However, exponential terms appear in the value function when working with a finite buffer.

#### 3.2. Infinite-server queue

The infinite-server queue is obtained by considering the multiserver queue with  $s = c$ . The optimality equations (1) then become

$$\begin{aligned} \varphi_c + \lambda V_c(0) &= \lambda V_c(1), \\ \varphi_c + (\lambda + x\mu)V_c(x) &= hx + \lambda V_c(x + 1) + x\mu V_c(x - 1), \quad x = 1, \dots, c - 1, \\ \varphi_c + c\mu V_c(c) &= hc + \lambda r + c\mu V_c(c - 1). \end{aligned}$$

The solution to these equations is in fact already given in Theorem 1. However, the average costs do differ and are given by

$$\varphi_c^h = \left[ 1 - \frac{1}{F(c+1)} \right] \rho h, \quad \varphi_c^w = 0, \quad \varphi_c^r = \frac{1}{F(c+1)} \lambda r,$$

where now  $\rho = \lambda/\mu$ . Hence, the average costs and the value function of the infinite server queue with no rejections are given by

$$\varphi_\infty^k = \lim_{c \rightarrow \infty} \varphi_c^k = \rho \alpha(k) \quad \text{and} \quad V_\infty^k(x) = \lim_{c \rightarrow \infty} V_c^k(x) = \frac{x}{\mu} \alpha(k),$$

with  $\alpha(h) = h$ ,  $\alpha(w) = 0$ , and  $\alpha(r) = 0$ . In this case, we observe that the infinite-server model with no rejections has a linear value function. However, rejections cause hypergeometric terms to appear in the value function. This is due to the fact that the rates in the optimality equation depend on the state.

### 4. Application to routing problems

In this section, we illustrate the one-step policy improvement method by studying a routing problem for parallel queues. The general idea is to start with a policy such that each queue behaves as a multiserver queue. In this way, the value function and the average costs can be determined from the results of the previous sections. Finally, one step of policy iteration can be applied to obtain a better policy without having to compute the value function in an iterative way.

Consider two parallel finite-buffer queues. For  $i = 1, 2$ , queue  $i$  has a buffer size of  $c_i$  customers and has its own set of  $s_i$  dedicated servers, each working at rate  $\mu_i$ . Furthermore, queue  $i$  has holding, waiting, and rejection costs of  $h_i$ ,  $w_i$ , and  $r_i$  respectively. An arriving customer can be either sent to queue 1 or to queue 2. The objective is to minimize the average costs. The optimality equation for this system is given by

$$\begin{aligned} \varphi + (\lambda + (x \wedge s_1)\mu_1 + (y \wedge s_2)\mu_2)V(x, y) \\ = h_1x_1 + h_2x_2 + \lambda(F(x, y) \wedge G(x, y)) \\ + (x \wedge s_1)\mu_1V([x - 1]^+, y) + (y \wedge s_2)\mu_2V(x, [y - 1]^+), \end{aligned}$$

with  $[k]^+ = \max\{k, 0\}$ ,  $(k \wedge l) = \min\{k, l\}$ ,

$$\begin{aligned} F(x, y) &= \mathbf{1}_{\{x < c_1\}}[x - s_1 + 1]^+ w_1 + \mathbf{1}_{\{x = c_1\}} r_1 + V((x + 1 \wedge c_1), y), \\ G(x, y) &= \mathbf{1}_{\{y < c_2\}}[y - s_2 + 1]^+ w_2 + \mathbf{1}_{\{y = c_2\}} r_2 + V(x, (y + 1 \wedge c_2)), \end{aligned}$$

and where  $x, y$  are the numbers of customers in queue 1 and 2 respectively.

Consider the policy that splits the arrival stream into two streams such that there are arrivals to queue 1 at rate  $\eta\lambda$  and to queue 2 at rate  $(1 - \eta)\lambda$  with  $\eta \in [0, 1]$ . We call this policy a Bernoulli policy with parameter  $\eta$ . The optimality equation under the Bernoulli policy is obtained by changing  $\lambda(F(x, y) \wedge G(x, y))$  into  $\eta\lambda F(x, y) + (1 - \eta)\lambda G(x, y)$ . Hence, we can see that the two queues behave independently as a multiserver queue. Therefore, the corresponding value function becomes

$$V_B(x, y) = V_{(\eta\lambda, \mu_1, s_1, c_1, h_1, w_1, r_1)}^{\text{MS}}(x) + V_{((1-\eta)\lambda, \mu_2, s_2, c_2, h_2, w_2, r_2)}^{\text{MS}}(y),$$



TABLE 1: Relevance of rejection costs.

		One step improved policy									
		x									
y		0	1	2	3	4	5	6	7	8	9
9		2	2	2	2	2	2	2	2	2	1
8		1	1	1	1	1	2	2	2	1	1
7		1	1	1	1	1	1	1	1	1	1
6		1	1	1	1	1	1	1	1	1	1
5		1	1	1	1	1	1	1	2	1	1
4		1	1	1	1	1	2	2	2	1	1
3		1	1	1	1	2	2	2	2	2	1
2		1	1	1	2	2	2	2	2	2	1
1		2	2	2	2	2	2	2	2	2	1
0		2	2	2	2	2	2	2	2	2	1

  

		Optimal policy									
		x									
y		0	1	2	3	4	5	6	7	8	9
9		2	2	2	2	2	2	2	2	2	1
8		1	1	1	2	2	2	2	2	1	1
7		1	1	1	1	1	2	2	1	1	1
6		1	1	1	1	1	1	1	1	1	1
5		1	1	1	1	1	1	1	1	1	1
4		1	1	1	1	1	2	2	1	1	1
3		1	1	1	1	2	2	2	2	1	1
2		1	1	1	2	2	2	2	2	2	1
1		2	2	2	2	2	2	2	2	2	1
0		2	2	2	2	2	2	2	2	2	1

with  $V^{MS}$  the value function of the multiserver queue of Section 2 with the corresponding parameters. Similarly, the average cost is expressed as

$$\varphi_B = \varphi_{(\eta\lambda, \mu_1, s_1, c_1, h_1, w_1, r_1)}^{MS} + \varphi_{((1-\eta)\lambda, \mu_2, s_2, c_2, h_2, w_2, r_2)}^{MS},$$

with  $\varphi^{MS}$  the average costs for the multiserver queue. From numerical experiments, it follows that not all parameters of the Bernoulli policy result in an improved policy which is close to the optimal value. Therefore, we will use the optimal Bernoulli policy for deriving the improved policy in the sequel. Note that, for a specified set of parameters, the maximization with respect to  $\eta$  is not difficult. The one-step policy improvement step now follows from the minimizing action in  $(F(x, y) \wedge G(x, y))$ .

We start by demonstrating the relevance of rejection costs when working with finite-buffer queues. Set  $h_1 = h_2 = 1, w_1 = w_2 = r_1 = r_2 = 0, \lambda = 5, \mu_1 = 2, \mu_2 = 3, s_1 = 3, s_2 = 2,$  and  $c_1 = c_2 = 9$ . Thus, we study two parallel finite buffer queues with holding costs only. The first queue has more dedicated servers than the second, but they work at a lower rate.

The optimal Bernoulli policy yields a value of  $\varphi_B = 2.351414$ , the one-step improved policy  $\varphi' = 1.993648$ , and the optimal policy  $\varphi^* = 1.993563$ . Table 1 shows the routing policy for the values of  $x$  and  $y$  under the one-step improved policy and the optimal policy. We would

TABLE 2: Numerical results.

$\lambda$	$\mu_1$	$\mu_2$	$s_1$	$s_2$	$c_1$	$c_2$	$h_1$	$h_2$	$w_1$	$w_2$	$r_1$	$r_2$	$\varphi_B$	$\varphi'$	$\varphi^*$
10	2	2	3	3	10	10	0	0	0	0	1	1	0.390401	0.082642	0.082642
10	2	2	3	3	10	5	0	0	0	0	1	1	0.836706	0.253959	0.226499
10	3	2	2	3	10	10	0	0	0	0	1	1	0.367001	0.072194	0.071396
8	2	2	3	3	10	10	0	0	1	1	1	1	8.807790	3.595779	3.531940
8	2	2	3	3	10	5	0	0	1	1	1	1	4.662343	1.917528	1.911727
8	3	2	2	3	10	10	0	0	1	1	1	1	9.945102	4.081310	3.921034
8	2	2	3	3	10	10	1	1	0	0	1	1	5.491495	4.606377	4.599034
8	2	2	3	3	10	5	1	1	0	0	1	1	4.999463	4.454041	4.425574
8	3	2	2	3	10	10	1	1	0	0	1	1	5.024346	3.950910	3.914964
8	2	2	3	3	10	10	1	1	1	1	1	1	14.228695	8.182282	8.092028
8	4	2	2	3	10	5	1	1	1	1	1	1	7.654585	4.386521	4.200002

expect an increasing switching curve. However, when one of the queues becomes congested, lack of rejection costs results in routing to that queue, and so rejections occur.

In the previous example, the one-step improved policy had a value close to the optimal value. Table 2 shows that this also holds for other parameter values. The greatest proportional extra cost  $(\varphi' - \varphi^*)/\varphi^*$  over all experiments in Table 2 is 0.12 (the second experiment). For all the other experiments, the proportional extra cost lies within the range of about 0.01 to 0.04. Note that this method can easily be used for more than two queues. In this section, we restricted ourselves to two queues, since the computation of the optimal policy becomes numerically difficult for more than two queues. For  $N$  stations of M/M/s/c queues, the number of states is equal to  $(c + 1)^N$ ; thus the complexity is exponential in the number of queues. However, a single step of policy iteration has linear complexity.

### Acknowledgements

The first author would like to thank Jan Sanders for bringing to his attention the work of Mickens [5]. He also gratefully acknowledges René Swarttouw for many valuable discussions.

### References

- [1] BERTSEKAS, D. P. AND TSITSIKLIS, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.
- [2] GROSS, D. AND HARRIS, C. M. (1985). *Fundamentals of Queueing Theory*, 2nd edn. John Wiley, New York.
- [3] KOOLE, G. M. AND NAIN, P. (2000). On the value function of a priority queue with an application to a controlled polling model. *Queueing Systems* **34**, 199–214.
- [4] KOOLE, G. M. AND SPIEKMA, F. (2001). On deviation matrices for birth–death processes. *Prob. Eng. Inf. Sci.* **15**, 239–258.
- [5] MICKENS, R. E. (1990). *Difference Equations: Theory and Applications*, 2nd edn. Van Nostrand Reinhold, New York.
- [6] OTT, T. J. AND KRISHNAN, K. R. (1992). Separable routing: a scheme for state-dependent routing of circuit switched telephone traffic. *Ann. Operat. Res.* **35**, 43–68.
- [7] PUTERMAN, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley, New York.
- [8] STIDHAM JR, S. AND WEBER, R. R. (1993). A survey of Markov decision models for control of networks of queues. *Queueing Systems* **13**, 291–314.