Queueing Systems 42, 91–101, 2002

Optimal Policies for Multi-server Non-preemptive Priority Queues

EROL A. PEKÖZ
School of Management, Boston University, Boston, MA 02215, USA

Received 30 July 2001; Revised 28 February 2002

Abstract. We consider a multi-server non-preemptive queue with high and low priority customers, and a decision maker who decides when waiting customers may enter service. The goal is to minimize the mean waiting time for high-priority customers while keeping the queue stable. We use a linear programming approach to find and evaluate the performance of an asymptotically optimal policy in the setting of exponential service and inter-arrival times.

1. Introduction

In this article we consider a decision-maker who controls a k-server non-preemptive queue with Poisson rate λ arrivals and independent exponential rate μ service times, $\lambda < k\mu$. Each arriving customer independently is high priority with probability p, and otherwise is low-priority. The decision-maker must decide when waiting customers should enter service with the goal of minimizing the mean waiting time for high-priority customers, while keeping the queue stable. No service can be interrupted, and the decision-maker may start any waiting customer in service whenever she chooses if a server is available. Here we consider only the class of policies, which we refer to as *admissible* policies, where all decisions are non-anticipating and the resulting queue length process is stationary, ergodic, and stable in the sense that the rate of arrivals equals the long-run rate of departures. We do not restrict our policies to be work-conserving, where the server is never idle while customers are waiting. Work-conserving policies are optimal in the single-server setting, but as we show below they may not be optimal in the multi-server setting.

This model has applications in ambulance dispatching, hospital bed management, and internet traffic routing. In the ambulance dispatching setting, there are emergencies calls and non-emergency requests for transfers between hospitals. A reasonable policy in practice is to dispatch for non-emergency calls only when there are sufficient idle ambulances available to cover potential emergencies. In the internet traffic setting, there are high-priority transmission such as voice or video, and lower priority transmissions such as email.

This model was previously considered in [9,11], where the following "cutoff" policy was proposed:

Policy 1.1. Serve high-priority customers as soon as possible, and start serving low-priority customers only when less than n servers are busy.

This policy represents the sensible idea of keeping some "strategic reserve" of idle servers even while low-priority customers are waiting in order to help avoid delays for high priority customers who might arrive later. In this article we show that policy 1.1 is not optimal in general, but the following type of policy can perform arbitrarily close to optimal using an appropriate n and f:

Policy 1.2. Place a "mark" on each arriving low-priority customer independently with probability f. Then serve high-priority customers as soon as possible, serve marked low-priority customers when exactly n servers are busy, and serve unmarked low-priority customers when exactly $n-1$ servers are busy.

In other words, the fraction f of low-priority customers are served while n servers are busy, and the remaining are served while $n-1$ servers are busy. Note that policy 1.2 makes sense only if $0 \leqslant n < k$ and $0 \leqslant f \leqslant 1$. Here we allow $n = 0$ and $f = 1$, with the interpretation in this case that low-priority customers start service only when the system is empty.

We show that using appropriate n and f, policy 1.2 is admissible and can give mean high-priority waiting times arbitrarily close to the infimum over all admissible policies. We also give a formula for the appropriate n, and for a sequence of values for f which gives performance converging to the infimum. Our approach is interesting in that the policy we give is quite straightforward to implement, though we must use a linear programming formulation to justify it. This sharp bound we obtain on performance is useful as a benchmark by which to evaluate more practical policies.

It is also interesting to note, as we will show below, that in the single-server case policy 1.2 with $n = 0$ and $f = 1$ is optimal. This corresponds to the work-conserving policy of starting low-priority customers whenever the server is free. It is clear that policy 1.2 is not work-conserving for $n > 0$.

This paper makes a contribution to the literature in several ways. Though there is quite a large literature on optimal policies for preemptive single-server queues, there seems to be little corresponding study of multi-server non-preemptive queues. In previous approaches to this problem [2,7,9,11,12] reasonable policies are studied and performance is evaluated. Our approach is different in that we find policies which are, in some sense, optimal. This fundamental issue of optimality is natural to study, though it does not seem to have been addressed in the literature for this setting.

The recent Glazebrook and Nino-Mora [4] study this problem with linear holding costs in the setting of a preemptive work-conserving service discipline, and Federgruen and Groenevelt [1] study a non-preemptive work-conserving service discipline with general service times. Our approach, though restricted to the case of exponential service times, expands the scope of study beyond work-conserving policies.

Finally, our linear programming approach to a solution for a priority queue may be of interest itself. The usual dynamic-programming approach to solving such control problems seems difficult here, as the state space is very large. The decision to start a customer in service may depend on things such as the number of customers waiting, the past history of the process, and exogenous randomization. In the method we employ, a linear program used in the proof is analytically solved but, interestingly, does not actually appear in the policy itself. In theory using a larger state space, different service rates and more than two classes could be handled, though the details would be more complicated. See [5] for a survey of the use of linear programming methods in other more general Markov decision problems.

The paper is organized as follows. In section 2 we give the main result, and in section 3 we give the proof.

2. Main result

We consider a k-server non-preemptive queue with Poisson rate λ arrivals and independent exponential rate μ service times, $\lambda < k\mu$. Each arriving customer independently is high-priority with probability p, and otherwise is low-priority. Our goal is to decide when waiting customers may enter service in order to minimize the mean waiting time for high-priority customers, while keeping the queue stable. We allow only policies, which we call *admissible*, where all decisions are non-anticipating and the resulting queue length process is stationary, ergodic, and stable in the sense that the rate of arrivals equals the rate of departures. The policy may use the past history of the system and any exogenous randomization in deciding when to start service on any number of high or low priority customers who are waiting.

We now present our main theorem, which states that policy 1.2 with an appropriate n and f is admissible and can give performance arbitrarily close to the infimum over all admissible policies. It also gives a formula for evaluating this infimum. We state our result in terms of the stationary number of customers X in a standard $M/M/k$ queue with rate λp arrivals, service rate μ. We write its tail probability (see [3, p. 43]), for $0 \leqslant n \leqslant k$, as

$$\begin{aligned} H_n &= P(X \geqslant n) \\ &= \left(\frac{(\lambda p/\mu)^k/k!}{1-\lambda p/(k\mu)} + \sum_{i=n}^{k-1}\frac{(\lambda p/\mu)^i}{i!}\right)\Bigg/\left(\frac{(\lambda p/\mu)^k/k!}{1-\lambda p/(k\mu)} + \sum_{i=0}^{k-1}\frac{(\lambda p/\mu)^i}{i!}\right). \end{aligned}$$

Theorem 2.1. Let $h_i = pH_i - H_{i+1}$, and let

$$m = \min\{i\colon\ h_i \geqslant 0\}.$$

(a) For the above problem, the infimum of mean stationary high-priority waiting times over all admissible policies equals

$$\left(\frac{H_k}{k\mu - \lambda p}\right)\left(\frac{\lambda - m\mu}{\lambda p H_m - m\mu H_{m+1}}\right). \tag{1}$$

(b) If $m > 0$, then policy 1.2 using

$$n = \min\{i\colon h_i > 0\}$$

and

$$f = \frac{n\mu h_{n-1}}{\lambda p h_n + n\mu h_{n-1}} + \varepsilon$$

is admissible and achieves mean stationary high-priority waiting times arbitrarily close to (1) for sufficiently small $\varepsilon > 0$. If $m = 0$ then policy 1.2 using $n = 0$ and $f = 1$ is admissible and achieves (1).

In other words, no admissible policy can perform better than (1), and the performance of policy 1.2 can be made arbitrarily close to this bound by using appropriate values for n, f. We thus say policy 1.2 is “asymptotically optimal.”

Note. The quantity (1) has a simple interpretation in the special case where $p = H_{m+1}/H_m$ for some $m > 0$. Consider a stationary $M/M/k$ queue with arrival rate λp and service rate μ, and let X, Y respectively be the number of customers in the system and the waiting time (until the start of service) for a randomly arriving customer. In this case the performance (1) reduces to

$$(k\mu - \lambda p)^{-1}\frac{H_k}{H_{m+1}} = E[Y \mid X > m].$$

In other words, to high-priority customers the queue looks like a standard $M/M/k$ (without the low-priority customers) given more than m servers are busy.

3. The proof

Our approach to the proof is the following. The performance of any policy turns out to depend (linearly) only on the long run rate of transitions between states in a suitably defined state space. For stability these rates must satisfy certain linear constraints, and these together define a linear program. We solve the linear program analytically to get a sharp bound on possible performance, and we use the solution to characterize a sequence of policies which asymptotically achieves this bound.

To carry out this proof we first present some preliminary propositions. The first one states that when looking for optimal policies, it suffices to consider only ones which serve high-priority customers as soon as possible.

Proposition 3.1. Given any policy there exists another with mean high-priority waiting times no greater, where high-priority customers are served as soon as possible.

Proof. First we claim that given any policy there exists another with high-priority mean waiting times no greater, where low priority customers are never served while high-priority customers are waiting. We use a simple interchange argument for this. Consider two coupled systems having the same arrival process and service times. Start with any policy running in the first system. Use the same policy in the second system except whenever a low priority customer is to be served ahead of some waiting high priority customer, interchange the service times and start service on the high-priority customer. The number of high-priority customers in the second system will never be greater than in the first system. This establishes the claim.

Now suppose we are given a policy where low-priority customers are never served while high-priority customers are waiting. Since the order of service within a priority class does not affect the mean waiting time for the class, we can assume there is a FIFO service discipline within the high-priority class of customers. Then we can imagine high-priority customers are assigned an idle server as soon as possible and block the server from use by any other customer until the decision-maker allows that high-priority customer to begin service. Since we assume low-priority customers are never served ahead of high-priority customers and there is FIFO among high-priority customers, this blocking will not affect the dynamics of the queue length process. We then create a coupled second system as follows. Suppose at some time in the first system there is a high-priority customer waiting who will require a service time of x time units after keeping some server blocked for y time units. In the second system start service immediately with service time of duration x followed by a blocking period of duration y, but continue to make all decisions based on the state of the first system. At any point in time the number of high-priority customers in the second system is never greater than the number in the first system. The results of these two paragraphs together give the proposition. □

The next proposition shows that the mean high-priority waiting time depends essentially only on the fraction of time all servers are busy. Thus it suffices to look for policies which minimize this fraction which, due to the Poisson arrival stream, also equals the probability an arriving high-priority customer finds all servers busy.

Proposition 3.2. Let W and N respectively denote the waiting time and the number of busy servers seen by a randomly arriving high-priority customer. Any policy which serves high-priority customers as soon as possible must have

$$E[W] = \frac{P(N = k)}{k\mu - \lambda p}.$$

Note that in particular this mean waiting time is linear in $P(N = k)$ and depends on any decisions involving low-priority customers only through this value. This property is the key to the linear programming formulation we use below.

Proof of proposition 3.2. Note that $W > 0$ if and only if $N = k$, and while $N = k$ the number high-priority customers waiting in queue evolves like an $M/M/1$ queue with service rate $k\mu$ and arrival rate λp. Thus at a random arrival epoch when $N = k$, the queue looks like a stationary $M/M/1$ queue and so $E[W|N = k] = 1/(k\mu - \lambda p)$. Unconditioning gives the proposition. □

For a given policy which serves high-priority customers as soon as possible, let $S(t)$ be the number of high-priority customers in the system plus the number of low-priority customers in service at time t. We refer to $S(t)$ as the "state" of the system at time t. Note that we do not count low-priority customers unless they are actually in service.

Next we focus on low-priority customers and let $N_i(t)$ denote the number of low-priority customers who enter service by time t and by doing so bring the state of the system from i to $i+1$. By ergodicity we can define the "state i low-priority unconditional service start rate"

$$r_i = \lim_{t\to\infty} \frac{N_i(t)}{t}, \quad i = 0, \dots, k-1.$$

If the policy prescribes that at some time more than one waiting customer is to start service simultaneously, these events are assumed to occur sequentially so only transitions between adjacent states are possible. For example, if a policy requires that two customers simultaneously enter service at time t while the system is in state i, the customers are counted in both $N_i(t)$ and $N_{i+1}(t)$.

By ergodicity the limiting rate of transitions from state i to state $i+1$ must equal the rate of transitions from state $i+1$ to state i. Since customers depart at rate $(i+1)\mu$ while $i+1$ servers are busy, and high-priority customers arrive at rate λp while in state i, the rate balance equations

$$r_i + \lambda p P_i = (i+1)\mu P_{i+1}, \quad i = 0, \dots, k-1, \tag{2}$$

and

$$\lambda p P_i = k\mu P_{i+1}, \quad i \geqslant k, \tag{3}$$

must hold, where

$$P_i = \lim_{t\to\infty} P\bigl(S(t) = i\bigr)$$

denotes the fraction of time the system is in state i. It can be easily verified by substitution that the solution to the rate balance equations (2) and (3) is

$$P_n = \left(x + \sum_{i=0}^{n\wedge k-1} x_i\right) f_n, \quad n \geqslant 0, \tag{4}$$

where we make the change of variables

$$x_i = \frac{r_i}{\lambda p f_i}, \quad i \geqslant 0, \quad \text{and} \quad x = P_0, \tag{5}$$

and we use the notation

$$f_i = \frac{(\lambda p)^i}{\mu_1 \cdots \mu_i},$$

where

$$\mu_i = \begin{cases} i\mu & 1 \leqslant i \leqslant k, \\ k\mu & i \geqslant k. \end{cases}$$

An admissible policy implies the overall rate at which low priority arrivals enter service must equal the overall arrival rate of low-priority customers, or $\sum_{i=0}^{k-1} r_i = (1-p)\lambda$. This using (5) is equivalent to

$$\sum_{i=0}^{k-1} f_i x_i = \frac{1-p}{p}. \tag{6}$$

The limiting probabilities must also satisfy $\sum_{i=0}^{\infty} P_i = 1$, which using (4) is equivalent to

$$G_0 x + \sum_{i=0}^{k-1} G_{i+1} x_i = 1, \tag{7}$$

where

$$G_n = \sum_{i=n}^{\infty} f_i.$$

Finally, the chance that all servers are busy equals

$$\sum_{i \geqslant k} P_i = \left(x + \sum_{i=0}^{k-1} x_i \right) G_k. \tag{8}$$

By proposition 3.2 the search for an optimal policy can thus be reduced to finding non-negative values $x, x_0, x_1, \ldots, x_{k-1}$ which minimize (8) subject to (6) and (7). Clearly this is a linear program with $k+1$ variables, $k+1$ non-negativity constraints, and two equality constraints (6) and (7). It follows that at the solution at least $k+1$ constraints will be binding and, since (6) and (7) must be binding, no more than two of the variables $x, x_0, x_1, \ldots, x_{k-1}$ will be nonzero.

The next proposition gives the solution to this linear program.

Proposition 3.3. Let $n = \min\{i \geqslant 0\colon\ p \geqslant G_{i+1}/G_i\}$. The solution to the linear program

$$\text{minimize (8)} \quad \text{subject to (6), (7), and } x, x_0, x_1, \ldots, x_{k-1} \geqslant 0$$

falls into two cases:

- Case 1: If $n = 0$ then the solution is $0 = x_1 = x_2 = \cdots = x_{k-1}$, $x_0 = (1 - p)/p$ and $x = (1 - G_1 x_0)/G_0$.
- Case 2: If $n > 0$ then the solution is $0 = x = x_0 = x_1 = \cdots = x_{n-2} = x_{n+1} = \cdots = x_{k-1}$,
$$x_n = \frac{G_n/p - G_{n-1}}{f_n G_n - f_{n-1} G_{n+1}},$$
and
$$x_{n-1} = \frac{1 - G_{n+1} x_n}{G_n}.$$

In either case the minimal value of the objective function equals
$$\frac{G_k(\lambda - n\mu)}{\lambda p G_n - n\mu G_{n+1}}.$$

Comments. We can interpret the solution in each case as follows:

- Case 1. Since only x, x_0 can be nonzero, only r_0 out of $r_0, \ldots, r_{k-1}$ can be nonzero. This corresponds to starting low-priority customers in service only when all servers are free. It is clear that policy 1.2 with $n = 0$ and $f = 1$ achieves this and, if it is admissible, the policy will be optimal. We argue admissibility below.
- Case 2. Here only r_n and r_{n-1} can be nonzero, and this corresponds to starting low-priority customers in service only while n or $n - 1$ servers are busy. Since $x = 0$ we also have $P_0 = 0$, and this may not necessarily be possible to achieve in an admissible policy. We therefore seek a sequence of admissible policies (each having $P_0 > 0$) with performance converging to this infimum performance given above. We argue below that the sequence of values f given in the theorem accomplishes this.

Proof of proposition 3.3. We start with case 2. Our argument is in three steps:

- Step 1. First we will suppose that for some a all except x_n, x_{n-1} and x_a are zero, and we will show that the objective function is a nondecreasing function of x_a.
- Step 2. Then we will show that $x_n, x_{n-1} \geqslant 0$.
- Step 3. Finally we will suppose all variables except x_n, x_{n-1} and x are zero, and we will show that the objective function is a nondecreasing function of x, and this will complete the proof for case 2.

Step 1. First suppose that for some a all variables except x_n, x_{n-1} and x_a are zero. The objective function in (8) becomes
$$Y = (x_a + x_n + x_{n-1})G_k \tag{9}$$
and the constraints (6) and (7) become
$$G_{a+1}x_a + G_{n+1}x_n + G_n x_{n-1} = 1 \tag{10}$$

and

$$f_a x_a + f_n x_n + f_{n-1} x_{n-1} = \frac{1-p}{p}. \tag{11}$$

Solving for x_n, x_{n-1} in terms of x_a in (10) and (11), inserting into (9) and taking the derivative gives

$$\frac{\partial (Y/G_k)}{\partial x_a} = 1 - \frac{\lambda p G_a - \mu_n G_{a+1}}{\lambda p G_n - \mu_n G_{n+1}}, \tag{12}$$

where we do some simplification using the facts $G_n = G_{n+1} + f_n$ and $f_n = f_{n-1}\lambda p/\mu_n$. The numerator and denominator in (12) are both positive since, for $j < k$,

$$\lambda p G_j = \sum_{i=j}^{\infty} \lambda p f_i = \sum_{i=j}^{\infty} \mu_{i+1} f_{i+1} > \sum_{i=j}^{\infty} \mu_j f_{i+1} = \mu_j G_{j+1}. \tag{13}$$

Next we claim

$$\lambda p G_a - \mu_n G_{a+1} \leqslant \lambda p G_n - \mu_n G_{n+1}, \tag{14}$$

which together with (13) implies (12) is non-negative. To establish (14) we consider the cases $a < n-1$ and $a > n$ separately. When $a < n-1$ we have

$$\mu_n (G_{a+1} - G_{n+1}) = \sum_{i=a+1}^{n} \mu_n f_i \geqslant \sum_{i=a+1}^{n} \mu_i f_i = \sum_{i=a+1}^{n} \lambda p f_{i-1} = \lambda p (G_a - G_n),$$

and then (14) holds. Similarly when $a > n$ we have

$$\mu_n (G_{n+1} - G_{a+1}) = \sum_{i=n+1}^{a} \mu_n f_i \leqslant \sum_{i=n+1}^{a} \mu_i f_i = \sum_{i=n+1}^{a} \lambda p f_{i-1} = \lambda p (G_n - G_a),$$

and again (14) holds. This completes step 1.

Step 2. Next we will show that $x_n, x_{n-1} \geqslant 0$ when $n = \min\{i \geqslant 0\colon\ p \geqslant G_{i+1}/G_i\}$ and $x_a = 0$. Some algebra with (10) and (11) shows that

$$x_{n-1} = \frac{G_n - G_{n+1}/p}{f_n G_n - f_{n-1} G_{n+1}} \tag{15}$$

and

$$x_n = \frac{G_n/p - G_{n-1}}{f_n G_n - f_{n-1} G_{n+1}}. \tag{16}$$

The denominator in both cases is non-negative, since

$$f_{n-1} G_{n+1} = \frac{f_n}{\lambda p} \sum_{i>n} \mu_n f_i \leqslant \frac{f_n}{\lambda p} \sum_{i>n} \mu_i f_i = f_n \sum_{i>n} f_{i-1} = f_n G_n.$$

Finally, by the definition of n the numerator in both cases is also positive. This completes step 2. Essentially the same argument holds for step 3, and case 1 holds in a similar manner.

Combining (15) and (16) and simplifying gives

$$x_n + x_{n-1} = \frac{\lambda - \mu_n}{\lambda p G_n - \mu_n G_{n+1}},$$

and thus the minimal objective function given is obtained. □

We are finally ready to assemble the pieces to give a proof for theorem 2.1.

Proof of theorem 2.1. First note that after some simplification, $H_i = G_i/G_0$. Thus using proposition 3.2 along with the two cases of proposition 3.3 establishes that (1) is a lower bound on the infimum mean stationary waiting time over all admissible policies. We next show that policy 1.2 is admissible and can perform arbitrarily close to this bound.

We claim that any policy of the form of policy 1.2 where $P_0 > 0$ is admissible. We first consider the single server case. Here $P_0 > 0$ implies the system becomes empty infinitely often and, due to the Poisson arrival stream and the independent marks placed, the system regenerates at these epochs. This implies the resulting queue length process will be stationary and ergodic, the rate of departures will equal the rate of arrivals, and the policy will be admissible. With $n > 0$ servers, $P_0 > 0$ implies $P_{n-1} > 0$ by (4), and this implies that infinitely often there are no low-priority customers at all in the system. These times form regeneration epochs, and again this implies the policy will be admissible.

We next claim that $P_0 > 0$ whenever $\varepsilon > 0$ in the policy given in the theorem. For a given ε the policy in the theorem satisfies

$$\frac{r_n}{r_n + r_{n-1}} = f + \varepsilon$$

and

$$r_n + r_{n-1} = \lambda(1 - p).$$

A straightforward calculation using these shows that

$$\begin{aligned} x_{n-1} &= \frac{G_n - G_{n+1}/p}{f_n G_n - f_{n-1} G_{n+1}} - \varepsilon\left(\frac{1-p}{p f_{n-1}}\right), \\ x_n &= \frac{G_n/p - G_{n-1}}{f_n G_n - f_{n-1} G_{n+1}} + \varepsilon\left(\frac{1-p}{p f_n}\right), \end{aligned}$$

and by (7) we must have

$$P_0 = x = \frac{1 - G_n x_{n-1} - G_{n+1} x_n}{G_0}. \tag{17}$$

The definition of n implies we have $x_n, x_{n-1} \geqslant 0$ for sufficiently small ε, and by using (13) the quantity (17) can be easily shown to be strictly positive.

It is clear that as $\varepsilon \to 0$ we obtain the limiting values (15) and (16), and hence the optimal objective function (1). Thus the policy given in the theorem is admissible and asymptotically optimal. □

4. Summary

We have given a sharp lower bound on achievable mean high-priority waiting times for a stable multi-server non-preemptive priority queue, and have given a sequence of policies which asymptotically achieves this bound. The bound itself is also useful as a benchmark by which to judge policies which are more practically implemented in applications. The problem here does not appear amenable to the usual dynamic programming approach; our solution employs an analytically solved linear program which, interestingly, does not actually appear in the final sequence of policies.

Acknowledgements

I would like to thank two anonymous referees for their insightful comments which have improved the presentation of this paper significantly. I would also like to thank one of the referees for providing a simplified argument for proposition 3.2.

References

[1] A. Federgruen and H. Groenevelt, $M/G/c$ queueing systems with multiple customer classes: characterization and control of achievable performance under nonpreemptive priority rules, Managm. Sci. 34(9) (1988) 1121–1138.

[2] H.R. Gail, S.L. Hantler and B.A. Taylor, Analysis of a nonpreemptive priority multiserver queue, Adv. in Appl. Probab. 20(4) (1988) 852–879.

[3] E. Gelenbe and G. Pujolle, *Introduction to Queueing Networks*, 2nd ed. (Wiley, New York, 1999).

[4] K.D. Glazebrook and J. Niño-Mora, Parallel scheduling of multiclass $M/M/m$ queues: Approximate and heavy-traffic optimization of achievable performance, Oper. Res. 49(4) (2001) 609–623.

[5] L.C.M. Kallenberg, Survey of linear programming for standard and nonstandard Markovian control problems. I. Theory, Z. Oper. Res. 40(1) (1994) 1–42.

[6] L.C.M. Kallenberg, Survey of linear programming for standard and nonstandard Markovian control problems. II. Applications, Z. Oper. Res. 40(2) (1994) 127–143.

[7] O. Kella and U. Yechiali, Waiting times in the nonpreemptive priority $M/M/c$ queue, Comm. Statist. Stochastic Models 1(2) (1985) 257–262.

[8] S.M. Ross, *Stochastic Processes*, 2nd ed. (Wiley, New York, 1996).

[9] C. Schaack and R.C. Larson, An N-server cutoff priority queue, Oper. Res. 34(2) (1986) 257–266.

[10] T. Takine, The nonpreemptive priority $MAP/G/1$ queue, Oper. Res. 47(6) (1999) 917–927.

[11] I.D.S. Taylor and J.G.C. Templeton, Waiting time in a multiserver cutoff-priority queue, and its application to an urban ambulance service, Oper. Res. 28(5) (1980) 1168–1188.

[12] T.M. Williams, Nonpreemptive multiserver priority queues, J. Oper. Res. Soc. 31(12) (1980) 1105–1107.