Probability in the Engineering and Informational Sciences, **15**, 2001, 15–33. Printed in the U.S.A.

THRESHOLD POLICIES FOR A SINGLE-SERVER QUEUING NETWORK

P. S. ANSELL AND K. D. GLAZEBROOK
Department of Statistics
University of Newcastle upon Tyne
Newcastle upon Tyne NE1 7RU, United Kingdom
E-mail: {p.s.ansell; kevin.glazebrook}@newcastle.ac.uk

I. MITRANI
Department of Computing Science
University of Newcastle upon Tyne
Newcastle upon Tyne NE1 7RU, United Kingdom
E-mail: isi.mitrani@newcastle.ac.uk

We consider a single-server queuing system with two job classes under service policies of threshold type. The server switches from type 1 to type 2 when either the former queue is empty or the latter reaches size T; it switches from type 2 to type 1 when the former queue size drops below T and the latter is not empty. The joint queue-length distribution is determined for preemptive and nonpreemptive implementations using both analytic techniques and the power series algorithm.

1. INTRODUCTION

An important problem that must be addressed in current telecommunication networks is the provision of different levels of service to different types of traffic. The standard mechanism for achieving such discrimination relies on the use of priority scheduling. Thus, the short-delay requirements of real-time packets (e.g., voice) can be met by assigning a higher priority to them than to other traffic (e.g., data). However, pure priority policies, whether preemptive or nonpreemptive, have the disadvantage that the lower-priority traffic is very heavily penalized in terms of the quality of service it receives. It is, therefore, desirable to devise and implement policies that satisfy the requirements of higher-priority traffic while offering acceptable performance for the lower-priority one.

 0269-9648/01 $12.50

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

In order to make some progress toward the above objective, we define and analyze a class of threshold policies for a single server with two types of demand: type 1 has high priority, except when the queue size of type 2 exceeds a certain level. The idea is to "soften" the effect of assigning strict priorities. This contrasts with the introduction of thresholds in the context of polling policies (Lee and Sengupta [6], Boxma et al. [3]), where the intention is to "enhance" the effect of servicing each queue exhaustively.

Both the preemptive and the nonpreemptive versions of the threshold policies are examined. In each case, the aim is to determine the joint distribution of the two-dimensional queue size process. Two solution methods are presented: one using generating functions and one based on the power series algorithm (PSA) (Blanc [2]). The former approach is efficient for small threshold values but suffers from numerical difficulties when the threshold is large. That solution, in the case of preemptive thresholds, was used by Ansell et al. [1] to study an optimization problem subject to variance constraints. The PSA is applicable to a wider range of models, particularly when used in conjunction with techniques for improving its convergence.

The model is described in Section 2. The analysis of the preemptive and nonpreemptive threshold policies by means of generating functions is presented in Sections 3 and 4, respectively. Section 5 deals with the power series solution and with the epsilon and conformal mapping convergence enhancing techniques. Section 6 contains the results of several numerical experiments, including an additional examination of the problem introduced in [1].

2. THE MODEL

We consider a model of two $M/M/1$ queues served by a single server. The service policy is assumed to be nonanticipative and work-conserving. Arrivals to queue k form independent Poisson streams with parameters λ_k and have exponentially distributed service times with mean $1/\mu_k$. If the load $\rho = \lambda_1/\mu_1 + \lambda_2/\mu_2$ is assumed to be less than unity, then we satisfy the ergodicity condition. Figure 1 illustrates the system.

We shall study two service policies, which are of the threshold type: one preemptive and the other nonpreemptive. In the preemptive case, the service policy depends on an integer threshold T. $N_k(t)$ denotes the number of customers in queue

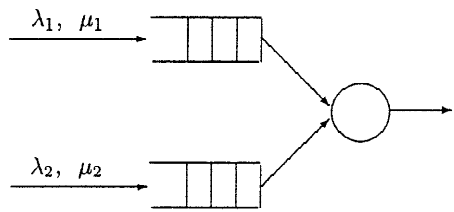

FIGURE 1. Two unbounded Markovian queues served by a single server.

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

k (waiting and in service) at time t. If $N_1(t) > 0$ and $N_2(t) < T$, then a type 1 customer is served; if $N_2(t) \geq T$ or $N_1(t) = 0$, a type 2 customer is served. More simply, the server switches from type 1 to type 2 when either the former queue is empty or the latter reaches size T. It switches from type 2 to type 1 when the former queue size drops below T and the latter is not empty. Preemptions and server reallocations do not result in any delay or any penalty cost. Note that when $T = 1$, the policy gives preemptive priority to type 2 customers, and when $T = \infty$, it gives preemptive priority to type 1 customers. The nonpreemptive case depends on the same integer threshold parameter T, but prohibits preemptions. Thus, the server can only switch at a service completion.

Consider a linear cost function $C = c_1E(N_1) + c_2E(N_2)$ where the $E(N_k)$, $k = 1,2$, are the expected queue lengths and the c_k, $k = 1,2$, are holding costs. The policy which minimizes C over all admissible preemptive service policies is one which gives strict preemptive priority to the class with the larger value of $c_k\mu_k$: the so-called $c\mu$ rule. Similarly, the policy which minimizes the cost function over all nonpreemptive service policies is one which gives "head of the line" priority according to the $c\mu$ rule (see Gelenbe and Mitrani [4]). Although these policies are optimal for the linear holding cost problems, they are often unacceptable in practice because of the heavy penalties imposed on the low-priority customers. These customers suffer from large queue lengths and, perhaps more significantly, large variances in these quantities. Thus, our ultimate aim is to obtain a readily implementable family of service policies which simultaneously minimize the linear cost function and satisfy prescribed constraints on, say, the second moments or the variances of the queue lengths (see Ansell et al. [1]). It is such problems which motivated this study of our class of threshold policies and the discussion in Section 6 gives grounds for believing that these policies perform well.

3. ANALYTIC SOLUTION OF THE PREEMPTIVE MODEL

Our aim in this section is to determine the joint steady-state distribution

$$p_{i,j} = \lim_{t\to\infty} P[N_1(t) = i, N_2(t) = j]$$

for the preemptive version of our model. These probabilities satisfy the balance equations

$$\begin{aligned}[\lambda_1 + \lambda_2 + \mu_1\delta(i > 0, j < T) + \mu_2\delta(j \geq T) + \mu_2\delta(i = 0, 0 < j < T)]p_{i,j} \\ = \lambda_1 p_{i-1,j} + \lambda_2 p_{i,j-1} + \mu_1\delta(j < T)p_{i+1,j} + \mu_2\delta(j \geq T-1)p_{i,j+1} \\ + \mu_2\delta(i = 0, j < T-1)p_{i,j+1}, \qquad \textbf{(1)}\end{aligned}$$

where probabilities with negative indices are 0 by definition and where $\delta(A) = 1$ if the condition A holds, and 0 otherwise. In addition, we have the normalizing equation

$$\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} p_{i,j} = 1. \qquad \textbf{(2)}$$

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

The solution of these equations requires the following generating functions:

$$g_j(x) = \sum_{i=0}^{\infty} p_{i,j}\, x^i, \quad j = 0,1,\ldots,T-1, \tag{3}$$

$$g(x,y) = \sum_{i=0}^{\infty} \sum_{j=T}^{\infty} p_{i,j}\, x^i y^{j-T}. \tag{4}$$

The first step in our method of solution is to transform (1), for $j < T-1$, into the following set of recurrence relations between the functions $g_j(x)$:

$$b(x)g_j(x) = \lambda_2 x g_{j-1}(x) - [\mu_1(1-x) + \mu_2 x\delta(j>0)]p_{0,j} + \mu_2 x p_{0,j+1}, \quad j = 0,1,\ldots,T-2, \tag{5}$$

where all functions with negative indices are, by definition, zero, and

$$b(x) = \lambda_1 x(1-x) + \lambda_2 x - \mu_1(1-x).$$

Using (5), we can then determine the functions $g_0(x), g_1(x), \ldots, g_{T-2}(x)$, in terms of the constants $p_{0,0}, p_{0,1}, \ldots, p_{0,T-1}$.

The probability of an empty system $p_{0,0}$ is equal to $1-\rho_1-\rho_2$. This can be established via the balance and normalizing equations, or more simply by direct application of Little's result. This means that we have $T-1$ unknown probabilities to be determined. Note that the quadratic $b(x)$, which appears in the left-hand side of (5) is negative at $x=0$, positive at $x=1$, and negative in the limit as $x \to \infty$. Hence, $b(x)$ has exactly one zero, x_0, in the interval (0,1) and one zero, x_1, in the interval (1,∞). Since the functions $g_j(x)$ are finite at $x = x_0$, the right-hand side of (5) must vanish at that point for every $j = 0,1,\ldots,T-2$. Setting $x = x_0$ in (5), for $j = 0$, and equating the right-hand side to zero determines the value of $p_{0,1}$. Since both sides of (5) now divide by $x - x_0$, the function $g_0(x)$ is of the form

$$g_0(x) = \frac{a_{0,0}}{x_1 - x},$$

where $a_{0,0}$ is a known constant and x_1 is the second zero of $b(x)$. Further, setting $x = x_0$ in (5) for $j = 1$ and equating the right-hand side to zero determines $p_{0,2}$. We can now express the function $g_1(x)$ in terms of elementary fractions,

$$g_1(x) = \frac{a_{1,0}}{x_1 - x} + \frac{a_{1,1}}{(x_1 - x)^2},$$

where $a_{1,0}$ and $a_{1,1}$ are known constants. By iterating this process, we can establish values for all of the unknown constants $p_{0,j}, j = 1,\ldots,T-1$, and thus obtain the $g_j(x), j = 0,1,\ldots,T-2$. The functions $g_j(x)$ can be written as a sum of elementary fractions:

$$g_j(x) = \sum_{k=0}^{j} \frac{a_{j,k}}{(x_1 - x)^{k+1}}, \quad j = 0,1,\ldots,T-2. \tag{6}$$

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

This simple form of the generating functions implies that when $j < T-1$, the probabilities $p_{i,j}$ are given by

$$p_{i,j} = \sum_{k=0}^{j} a_{j,k} \binom{i+k}{i} x_1^{-(i+k+1)}, \quad i = 0,1,\ldots, j = 0,1,\ldots, T-2.$$

The only generating functions now left to determine are $g_{T-1}(x)$ and $g(x,y)$. To do this, we use the balance equations, (1), for $j = T-1$ and $j \geq T$, leading to the following relations:

$$b(x)g_{T-1}(x) = \lambda_2 x g_{T-2}(x) + \mu_2 x g(x,0) - [\mu_1(1-x) + \mu_2 x]p_{0,T-1}, \tag{7}$$

$$k(x,y)g(x,y) = \lambda_2 y g_{T-1}(x) - \mu_2 g(x,0), \tag{8}$$

where

$$k(x,y) = \lambda_1 y(1-x) + \lambda_2 y(1-y) - \mu_2(1-y).$$

Again, we note that for every x in the interval $[0,1]$, there is exactly one value of y in the same interval, $y = \beta(x)$, such that $k(x,\beta(x)) = 0$. Since $g(x,\beta(x))$ is finite, the right-hand side of (7) vanishes when $y = \beta(x)$. This allows us to eliminate $g(x,0)$ from (7) and (8) to obtain

$$g_{T-1}(x) = \frac{\lambda_2 x g_{T-2}(x) - [\mu_1(1-x) + \mu_2 x]p_{0,T-1}}{b(x) - \lambda_2 x \beta(x)}, \tag{9}$$

$$g(x,y) = \frac{\lambda_2[y - \beta(x)]g_{T-1}(x)}{k(x,y)}. \tag{10}$$

The denominator in (9) is zero at $x = 1$, but so is the numerator. It can be shown that when the ergodicity condition holds, the function $g_{T-1}(x)$ has no singularities in the unit disk. This remark also applies to (10).

We now have all the unknowns specified and can evaluate the performance measures $E(N_1)$, $E(N_2)$, $\mathrm{Var}(N_1)$, and $\mathrm{Var}(N_2)$ from the above generating functions in the usual way.

The derivatives of the generating functions at $x = 1$, $y = 1$ involve indeterminacies of the type $0/0$ which are resolved by L'Hôpital's rule. In this problem, the unknown quantities are found by successive substitutions rather than by solving a set of simultaneous equations. The computational complexity of this solution is therefore $O(T^2)$, rather than $O(T^3)$.

4. ANALYTIC SOLUTION OF THE NONPREEMPTIVE MODEL

In this section, we prohibit preemptions; thus, switching decisions are only made at service completions. Let $S(t)$ be a random variable equal to k if the server is processing queue k at time t. We introduce the steady-state probabilities

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

$$p^1_{i,j} = \lim_{t\to\infty} P(N_1(t) = i, N_2(t) = j, S(t) = 1), \quad i \geq 1, j \geq 0,$$

$$p^2_{i,j} = \lim_{t\to\infty} P(N_1(t) = i, N_2(t) = j, S(t) = 2), \quad i \geq 0, j \geq 1,$$

$$p_{00} = \lim_{t\to\infty} P(N_1(t) = 0, N_2(t) = 0)$$

where $p^1_{0,j} = 0$, $j \geq 1$, and $p^2_{i,0} = 0$, $i \geq 1$. These probabilities satisfy the following balance equations:

$$\begin{aligned}(\lambda_1 + \lambda_2 + \mu_1)p^1_{i,j} = {} & \lambda_1 p^1_{i-1,j} + \lambda_2 p^1_{i,j-1} + \mu_1 \delta(j \leq T-1) p^1_{i+1,j} \\ & + \mu_2 \delta(j \leq T-1) p^2_{i,j+1} + \lambda_1 \delta(i = 1, j = 0) p_{00}, \\ & i > 0, j \geq 0, \quad \textbf{(11)}\end{aligned}$$

$$\begin{aligned}(\lambda_1 + \lambda_2 + \mu_2)p^2_{i,j} = {} & \lambda_1 p^2_{i-1,j} + \lambda_2 p^2_{i,j-1} + \mu_2 \delta(i = 0, j \leq T-1) p^2_{i,j+1} \\ & + \mu_2 \delta(j \geq T-1) p^2_{i,j+1} + \mu_1 \delta(j \geq T) p^1_{i+1,j} \\ & + \lambda_2 \delta(i = 0, j = 1) p_{00} + \mu_1 \delta(j \leq T-1) p^1_{1,j}, \\ & i \geq 0, j > 0, \quad \textbf{(12)}\end{aligned}$$

$$(\lambda_1 + \lambda_2)p_{00} = \mu_1 p^1_{1,0} + \mu_2 p^2_{0,1}, \quad \textbf{(13)}$$

where, once again, $\delta(A) = 1$ if the condition A holds, and 0 otherwise. In addition, the following normalizing equation holds:

$$p_{00} + \sum_{i=1}^{\infty}\sum_{j=0}^{\infty} p^1_{i,j} + \sum_{i=0}^{\infty}\sum_{j=1}^{\infty} p^2_{i,j} = 1. \quad \textbf{(14)}$$

To solve this problem, we introduce the generating functions

$$g_1(x, y) = \sum_{i=1}^{\infty}\sum_{j=T}^{\infty} p^1_{i,j} x^{i-1} y^{j-T},$$

$$g_2(x, y) = \sum_{i=0}^{\infty}\sum_{j=T}^{\infty} p^2_{i,j} x^{i} y^{j-T},$$

$$g^1_j(x) = \sum_{i=1}^{\infty} p^1_{i,j} x^{i-1}, \quad 0 \leq j \leq T-1,$$

$$g^2_j(x) = \sum_{i=0}^{\infty} p^2_{i,j} x^{i}, \quad 1 \leq j \leq T-1.$$

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

Equations (11) and (12) are then transformed into

$$\begin{aligned} b(x)g_j^1(x) &= \lambda_2 x g_{j-1}^1(x) - \mu_1 g_j^1(0) + \lambda_1 x p_{00}\delta(j=0) \\ &\quad + \mu_2[g_{j+1}^2(x) - g_{j+1}^2(0)]\delta(j<T-1) \\ &\quad + \mu_2[g_2(x,0) - g_2(0,0)]\delta(j=T-1), \quad 0 \le j \le T-1, \end{aligned} \tag{15}$$

$$\begin{aligned} c(x)g_j^2(x) &= \lambda_2 g_{j-1}^2(x) + \mu_2 g_2(0,0)\delta(j=T-1) + \mu_2 g_{j+1}^2(0)\delta(j<T-1) \\ &\quad + \lambda_2 p_{00}\delta(j=1) + \mu_1 g_j^1(0), \quad 1 \le j \le T-1, \end{aligned} \tag{16}$$

$$d(x,y)g_1(x,y) = \lambda_2 g_{T-1}^1(x) \tag{17}$$

$$k(x,y)g_2(x,y) = -u_2 g_2(x,0) + u_1 y g_1(x,y) + \lambda_2 y g_{T-1}^2(x), \tag{18}$$

where $g_{-1}^m(x) = 0$, $m = 1,2$, by definition, and the functions $b(x)$, $c(x)$, $d(x)$, and $k(x,y)$ are defined as

$$\begin{aligned} b(x) &= \lambda_1 x(1-x) + \lambda_2 x - \mu_1(1-x), \\ c(x) &= \lambda_1(1-x) + \lambda_2 + \mu_2, \\ d(x,y) &= \lambda_1(1-x) + \lambda_2(1-y) + \mu_1, \\ k(x,y) &= \lambda_1 y(1-x) + \lambda_2 y(1-y) - \mu_2(1-y). \end{aligned}$$

We note that we need to evaluate $2T$ unknown constants in order to fully determine the generating functions. To do this, we use recurrence relation (15) for $g_j^1(x)$, $j \le T-1$, giving

$$\begin{aligned} b(x)^{T-1} g_{T-2}^1(x) &= (\lambda_2 x)^{T-2}\{-\mu_1 g_0^1(0) + \lambda_1 x p_{00} + \mu_2[g_1^2(x) - g_1^2(0)]\} \\ &\quad + \sum_{i=1}^{T-2} b(x)^i (\lambda_2 x)^{T-2-i} \\ &\quad \times \{-\mu_1 g_i^1(0) + \mu_2[g_{i+1}^2(x) - g_{i+1}^2(0)]\}. \end{aligned} \tag{19}$$

Further to this, we substitute (16) involving the $g_{i+1}^2(x)$, $1 \le j \le T-1$, into (19). Note that the function $b(x)$ has one zero $x_0 \in (0,1)$ and the other zero $x_1 \in (1,\infty)$. The functions $g_j^1(x)$, $j < T-1$, are finite in the unit disk. At the point x_0, the left-hand side of (19) vanishes, which, in turn, implies that at x_0, the right-hand side of (19) must also vanish. Note also that the first $T-2$ derivatives of the left-hand side of (19) are also zero at x_0, so the first $T-2$ derivatives of the right-hand side of (19) must also vanish at this point. These observations lead to $T-1$ equations. A further $T-1$ equations are obtained by substituting $x = 0$ in (16). This leaves us two equations to find. The first of these is obtained from the balance equation (13) relating to the empty system, namely

$$(\lambda_1 + \lambda_2)p_{00} = \mu_1 g_0^1(0) + \mu_1 g_1^2(0), \tag{20}$$

and the following argument gives the final equation. We note that for every x in the interval $[0,1]$, there is exactly one value of y in the same interval, $y = \alpha(x)$, such that

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

$k(x,\alpha(x)) = 0$. Since $g_2(x,\alpha(x))$ is finite, the right-hand side of (18) must also vanish. Thus, we have

$$-\mu_2 g_2(x,0) + \mu_1 \alpha(x) g_1(x,y) + \lambda_2 \alpha(x) g^2_{T-1}(x) = 0. \quad \textbf{(21)}$$

Moreover, the equation for $g_1(x,y)$ is directly related to the function $g^1_{T-1}(x)$, and so at the point $x = 0$, we have that

$$-\mu_2 g_2(0,0) + \frac{\mu_1 \alpha(0) \lambda_2 g^1_{T-1}(0)}{d(0,\alpha(0))} + \lambda_2 \alpha(0) g^2_{T-1}(0) = 0. \quad \textbf{(22)}$$

We now have $2T$ equations in terms of the $2T$ unknown constants and, therefore, can fully determine the generating functions $g^1_j(x)$, $j = 0,\ldots,T-2$ and $g^2_j(x)$, $j = 1,\ldots,T-1$. This leaves us with three functions to evaluate, namely $g_1(x,y)$, $g_2(x,y)$, and $g^1_{T-1}(x)$. To do this, we solve the remaining equations, (21), (17), and (15) (for $j = T-1$). Having done this, we have all the generating functions specified and can proceed to calculate the moments.

Unlike the method used in the preemptive case, here we have to solve the system of simultaneous equations; thus, the computational complexity of the solution of $O(T^3)$.

5. POWER SERIES ALGORITHM

The methods proposed in Sections 3 and 4 are computationally expensive, especially in the nonpreemptive case, because of the large number of derivatives and limits that need to be evaluated. This means that obtaining the moments of the queue lengths becomes increasingly difficult as T gets larger (>5, say). Moreover, the methods are not easily extended to models with three or more queues. The power series algorithm (PSA) is a numerical method for evaluating performance measures for multidimensional Markov processes (see Blanc [2]). It approximates the steady-state distribution of a general Markov process by computing the coefficients of a simple recursion, which is obtained as a result of introducing an artificial parameter χ.

We consider first the preemptive model and introduce an artificial parameter χ by replacing the arrival rate λ_i by $\lambda_i \chi$. The service parameters are left unchanged. If we express the balance equations (1) in terms of the artificial parameter, we obtain

$$\begin{aligned}[(\lambda_1+\lambda_2)\chi + \mu_1\delta(i>0, j<T) + \mu_2\delta(j\geq T) + \mu_2\delta(i=0, 0<j<T)]p_{i,j} \\ = [\lambda_1 p_{i-1,j} + \lambda_2 p_{i,j-1}]\chi + \mu_1\delta(j<T)p_{i+1,j} + \mu_2\delta(j\geq T-1)p_{i,j+1} \\ + \mu_2\delta(i=0, j<T-1)p_{i,j+1}. \quad \textbf{(23)}\end{aligned}$$

We now write

$$p_{i,j} = \chi^{i+j} \sum_{k=0}^{\infty} \hat{p}_{k,i,j} \chi^k. \quad \textbf{(24)}$$

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

Now, if we replace the $p_{i,j}$ according to (24) in (23), eliminate the factor χ^{i+j}, and equate terms with equal powers of χ, we obtain the following recursion for the coefficients $\hat{p}_{k,i,j}$:

$$\begin{aligned}[\mu_1\delta(i>0,j<T)+\mu_2\delta(j\geq T)+\mu_2\delta(i=0,0<j<T)]\hat{p}_{k,i,j}\\ =-(\lambda_1+\lambda_2)\delta(k>0)\hat{p}_{k-1,i,j}+\lambda_1\hat{p}_{k,i-1,j}+\lambda_2\hat{p}_{k,i,j-1}\\ +\mu_1\delta(j<T,k>0)\hat{p}_{k-1,i+1,j}\\ +\mu_2\delta(j\geq T-1,k>0)\hat{p}_{k-1,i,j+1}\\ +\mu_2\delta(i=0,j<T-1,k>0)\hat{p}_{k-1,i,j+1}. \end{aligned} \tag{25}$$

To determine the coefficients, we also need to use the normalizing equation, which we express as

$$\sum_{k=0}^{\infty}\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\hat{p}_{k,i,j}\chi^{k+i+j}=1.$$

From this, we obtain

$$\hat{p}_{000}=1 \tag{26}$$

and

$$\sum_{i+j+k=l}\hat{p}_{k,i,j}=0,\quad l\geq 1. \tag{27}$$

We can now calculate the coefficients from (25)–(27). By setting χ to 1, we return to the original formulation. To obtain an approximation, we truncate the number of coefficients to those with a power of χ less than a value K.

The nonpreemptive case is analyzed in an analogous manner. Again, we replace λ_i by $\lambda_i\chi$ and write

$$p_{i,j}^{m}=\chi^{i+j}\sum_{k=0}^{\infty}\hat{p}_{k,i,j}^{m}\chi^{k},\quad m=1,2, \tag{28}$$

to obtain the following recursions:

$$\begin{aligned}\mu_1\hat{p}_{k,i,j}^{1}=&-(\lambda_1+\lambda_2)\delta(k>0)\hat{p}_{k-1,i,j}^{1}+\lambda_1\hat{p}_{k,i-1,j}^{1}+\lambda_2\hat{p}_{k,i,j-1}^{1}\\ &+\mu_1\delta(j\leq T-1,k>0)\hat{p}_{k-1,i+1,j}^{1}+\mu_2\delta(j\leq T-1,k>0)\hat{p}_{k-1,i,j+1}^{2}\\ &+\lambda_1\delta(i=1,j=0)\hat{p}_{k00},\quad i>0,j\geq 0,\\ \mu_2\hat{p}_{k,i,j}^{2}=&-(\lambda_1+\lambda_2)\delta(k>0)\hat{p}_{k-1,i,j}^{2}+\lambda_1\hat{p}_{k,i-1,j}^{2}+\lambda_2\hat{p}_{k,i,j-1}^{2}\\ &+\mu_2\delta(i=0,j\leq T-1,k>0)\hat{p}_{k-1,i,j+1}^{2}\\ &+\mu_2\delta(j\geq T-1,k>0)\hat{p}_{k-1,i,j+1}^{2}\\ &+\mu_1\delta(j\geq T,k>0)\hat{p}_{k-1,i+1,j}^{1}+\lambda_2\hat{p}_{k00}\delta(i=0,j=1)\\ &+\mu_1\delta(j\leq T-1,k>0)\hat{p}_{k-1,1,j}^{1},\quad i\geq 0,j>0.\end{aligned}$$

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

The normalizing equation is again used to completely determine the coefficients.

5.1. Convergence

The procedure just described does not in itself guarantee convergence of the power series obtained by the recursion. In fact, in both models, the method only converges for values of ρ less than about 0.6 (i.e., lightly loaded systems). To overcome this problem, we employ two methods:

- Conformal Mapping Technique
- Epsilon Algorithm.

5.2. Conformal Mapping Technique

In (24) and (28), we expand the steady-state probabilities about zero. The radius of convergence for such a power series is the distance between the origin and the nearest singularity. To enlarge the radius of convergence, we must move the singularities further away from the origin. One method of doing this is to use origin-preserving bilinear mapping:

$$\theta = \Gamma_G(\chi) = \frac{(1+G)\chi}{1+G\chi}, \qquad \chi = \Gamma_G^{-1}(\theta) = \frac{\theta}{1+G-G\theta}, \quad G \geq 0.$$

We obtain another recursive computational scheme by expanding the steady-state probabilities as a function of θ:

$$p_{i,j} = \theta^{i+j} \sum_{k=0}^{\infty} \theta^k \hat{p}_{k,i,j} \quad \text{(preemptive)}.$$

$$p_{i,j}^m = \theta^{i+j} \sum_{k=0}^{\infty} \theta^k \hat{p}_{k,i,j}^m, \quad m = 1,2 \text{ (nonpreemptive)}.$$

The choice of G is by rule of thumb. In much of the literature, a value of 1.5 is recommended; however, in some cases, we found a slightly larger value to be a better choice.

5.3. Epsilon Algorithm

The aim of the epsilon algorithm is to accelerate the convergence of a slowly converging sequence. To do this, the epsilon algorithm converts a polynomial into quotients of two polynomials. The following scheme is used:

$$\epsilon_{\kappa+1}^{(m)} = \epsilon_{\kappa-1}^{(m+1)} + [\epsilon_{\kappa-1}^{(m+1)} - \epsilon_{\kappa}^{(m)}]^{-1}, \qquad \epsilon_{-1}^{(m)} = 0, \qquad \epsilon_0^{(m)} = \sum_{k=0}^{m} c_k \beta^k,$$

where the $c_k, k = 0,1,2,\ldots$, stand for coefficients of a series such as the ones defined in (24) and (28). The even sequences $\{\epsilon_{2\kappa}^{(m)}, m = 0,1,\ldots\}, k = 1,2,\ldots$, may converge

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

faster to a limit than the initial sequence. The odd sequences are intermediate steps in the calculation. For further details, see van den Hout [7].

6. NUMERICAL RESULTS

In this section, we present numerical results for two examples. The first of these is used to confirm that the PSA provides us with accurate approximations of the performance measures of interest. In the second example, attention is focused on the stochastic optimization problem alluded to in Section 2.

Note that we could also define a *dual* family of policies with parameter T' by placing the threshold on type 1 jobs instead of on type 2; setting $T' = 1$ in this family would give priority to type 1 and $T' = \infty$ would lead to priority being given to type 2. The analysis of this policy is carried out simply by swapping λ_1 with λ_2 and μ_1 with μ_2 and then using the techniques described in Sections 3–5. This policy will be used in the second example.

Example 1: In this example, we present the results of the model when the arrival rates are both 1 and the service rates are both 3. The load on the system is fairly low with $\rho = \frac{2}{3}$. For this example, we restricted the computation to all coefficients $\hat{p}_{k,i,j}$ such that $k + i + j \leq 200$. The value of G used was 1.5 and the epsilon algorithm was invoked to increase the convergence properties of the resultant power series. To ensure the accuracy of the numerical method, we computed coefficients until the sum of the steady-state probabilities was sufficiently close to 1 (six decimal places were deemed enough). Tables 1 and 2 contain the expected queue lengths and queue length variances for the preemptive and nonpreemptive variants of the threshold policy. These were obtained using the PSA. To check the accuracy of our results, we compared them with the exact results obtained by implementing (using Maple) the analytical techniques of Sections 3 and 4. Because of the computational demands of the exact methods, only a relatively small number of policies can be compared in this way. In all cases, the PSA approximated all performance measures (both mean queue lengths and variances) to at least three decimal places.

Figure 2 shows how the achievable variance pairs behave as the value of the threshold parameter increases. Both preemptive and nonpreemptive results are illustrated. If, as has been conjectured, the true boundary of variance pairs is a convex curve passing through the variance pairs of the priority policies ($T = 1$ and $T = \infty$) and the FIFO policy, which is (2,2) in this case, then this picture gives us some encouragement to believe that the variances achievable by threshold policies might be quite close to this boundary.

Example 2: Consider now the model with parameters $\boldsymbol{\lambda} = (1,5)$ and $\boldsymbol{\mu} = (3,12)$. This system can be thought of as being unbalanced in the sense that the type 1 jobs arrive, on average, five times less often than type 2 jobs but their service requirements are, on average, four times longer. We employed the conformal mapping technique of Section 5 with the value of G equal to 2.5. The epsilon algorithm enhanced the convergence properties of the resulting power series.

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

Table 1. Queue Lengths for the Models with $\boldsymbol{\lambda} = (1,1)$, $\boldsymbol{\mu} = (3,3)$ for Example 1

T	$E_P(N_1)$	$E_P(N_2)$	$E_{NP}(N_1)$	$E_{NP}(N_2)$
1	1.500	0.500	1.333	0.667
2	1.151	0.849	1.101	0.899
3	0.923	1.077	0.949	1.051
4	0.776	1.224	0.850	1.150
5	0.680	1.320	0.787	1.213
6	0.618	1.382	0.745	1.258
7	0.577	1.423	0.718	1.282
8	0.551	1.450	0.701	1.300
9	0.534	1.467	0.689	1.311
10	0.522	1.478	0.681	1.319
20	0.500	1.500	0.667	0.333
∞	0.500	1.500	0.667	1.333

Table 2. Variance of Queue Lengths for the Models with $\boldsymbol{\lambda} = (1,1)$, $\boldsymbol{\mu} = (3,3)$ for Example 1

T	$\mathrm{Var}_P(N_1)$	$\mathrm{Var}_P(N_2)$	$\mathrm{Var}_{NP}(N_1)$	$\mathrm{Var}_{NP}(N_2)$
1	4.500	0.750	3.944	1.000
2	3.492	0.977	3.067	1.256
3	2.670	1.450	2.414	1.667
4	2.063	1.997	1.953	2.109
5	1.636	2.518	1.638	2.510
6	1.344	2.971	1.424	2.850
7	1.146	3.344	1.282	3.123
8	1.014	3.638	1.187	3.337
9	0.925	3.866	1.124	3.499
10	0.866	4.037	1.082	3.621
20	0.753	4.490	1.001	3.934
∞	0.750	4.500	1.000	3.944

Tables 3 and 4 contain the expected queue lengths and variance pairs for both the preemptive and nonpreemptive variants of the threshold policy. Figure 3 illustrates the property that as the expected queue length of type i increases so does its variance (we only plot the threshold policies with odd values of T).

Consider now the problem (alluded to at the end of Section 2) of minimizing the linear cost function C subject to some prescribed variance constraints expressed as

$$\mathrm{Var}(N_1) \leq B_1, \tag{29}$$

$$\mathrm{Var}(N_2) \leq B_2. \tag{30}$$

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

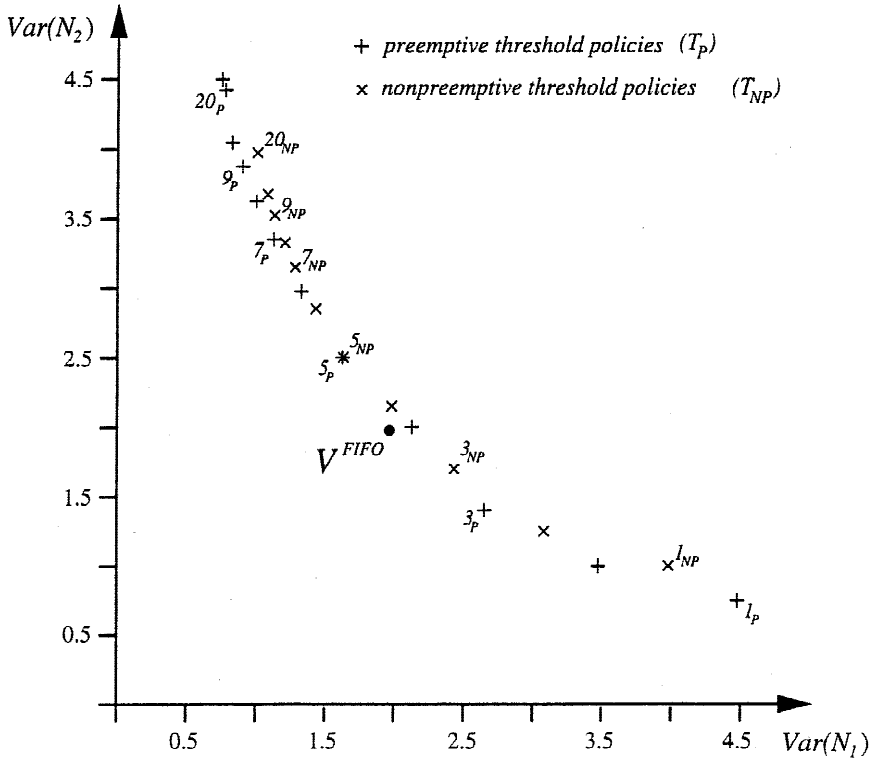

FIGURE 2. Achievable variance pairs for Example 1.

We shall discuss this for the preemptive version of our model. If $c_1\mu_1 > c_2\mu_2$, we would most naturally impose thresholds on type 2 jobs and find the optimal threshold policy by identifying the policy with the largest value of T which satisfies both constraints (29) and (30). Alternatively, if $c_2\mu_2 > c_1\mu_1$, we would use the *dual* family of threshold policies and find the optimal threshold policy by identifying the policy which has the largest value of T' satisfying (29) and (30). By this route, the minimized cost available from the class of threshold policies could then be found by substituting the respective expected queue lengths into the cost function C.

To assess the quality of threshold policies for the variance constrained problem, we compare them with a more commonly studied family of policies. This latter family, which we shall call a mixed-priority family, is dependent on a single parameter α: At the beginning of each busy period, a random decision is made giving preemptive priority to type 1 jobs with probability α, and to type 2 with probability $1 - \alpha$. Plainly, as α ranges from 0 to 1, the mixed-priority policies range from strict priority to type 2, to strict priority to type 1: The corresponding set of expected queue

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

Table 3. Queue Lengths for the Models with $\boldsymbol{\lambda} = (1,5)$, $\boldsymbol{\mu} = (3,12)$ for Example 2

T	$E_P(N_1)$	$E_P(N_2)$	$E_{NP}(N_1)$	$E_{NP}(N_2)$
1	1.571	0.714	1.333	1.667
2	1.442	1.230	1.240	2.039
3	1.330	1.681	1.160	2.359
4	1.232	2.072	1.091	2.635
5	1.147	2.412	1.031	2.876
10	0.857	3.571	0.821	3.716
20	0.616	4.535	0.641	4.437
30	0.539	4.843	0.582	4.671
40	0.514	4.946	0.563	4.750
50	0.505	4.981	0.556	4.777
∞	0.500	5.000	0.552	4.792

Table 4. Variance of Queue Lengths for the Models with $\boldsymbol{\lambda} = (1,5)$, $\boldsymbol{\mu} = (3.12)$ for Example 2

T	$\mathrm{Var}_P(N_1)$	$\mathrm{Var}_P(N_2)$	$\mathrm{Var}_{NP}(N_1)$	$\mathrm{Var}_{NP}(N_2)$
1	4.157	1.224	3.587	4.444
2	3.945	1.474	3.366	5.210
3	3.720	2.159	3.154	6.284
4	3.492	3.207	2.953	7.602
5	3.268	4.548	2.763	9.112
10	2.323	13.734	2.008	18.079
20	1.317	32.460	1.238	34.478
30	0.950	44.221	0.959	44.274
40	0.821	50.182	0.861	49.136
50	0.775	52.931	0.826	51.347
∞	0.750	55.000	0.801	53.012

length pairs coincides with the complete set of performances achievable by all admissible policies. The first two moments of the number of type 1 jobs in the system under the mixed policy with parameter α are given by

$$\begin{aligned} E_\alpha(N_1) &= \alpha E_{1,2}(N_1) + (1-\alpha)E_{2,1}(N_1), \\ E_\alpha(N_1^2) &= \alpha E_{1,2}(N_1^2) + (1-\alpha)E_{2,1}(N_1^2), \end{aligned} \tag{31}$$

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

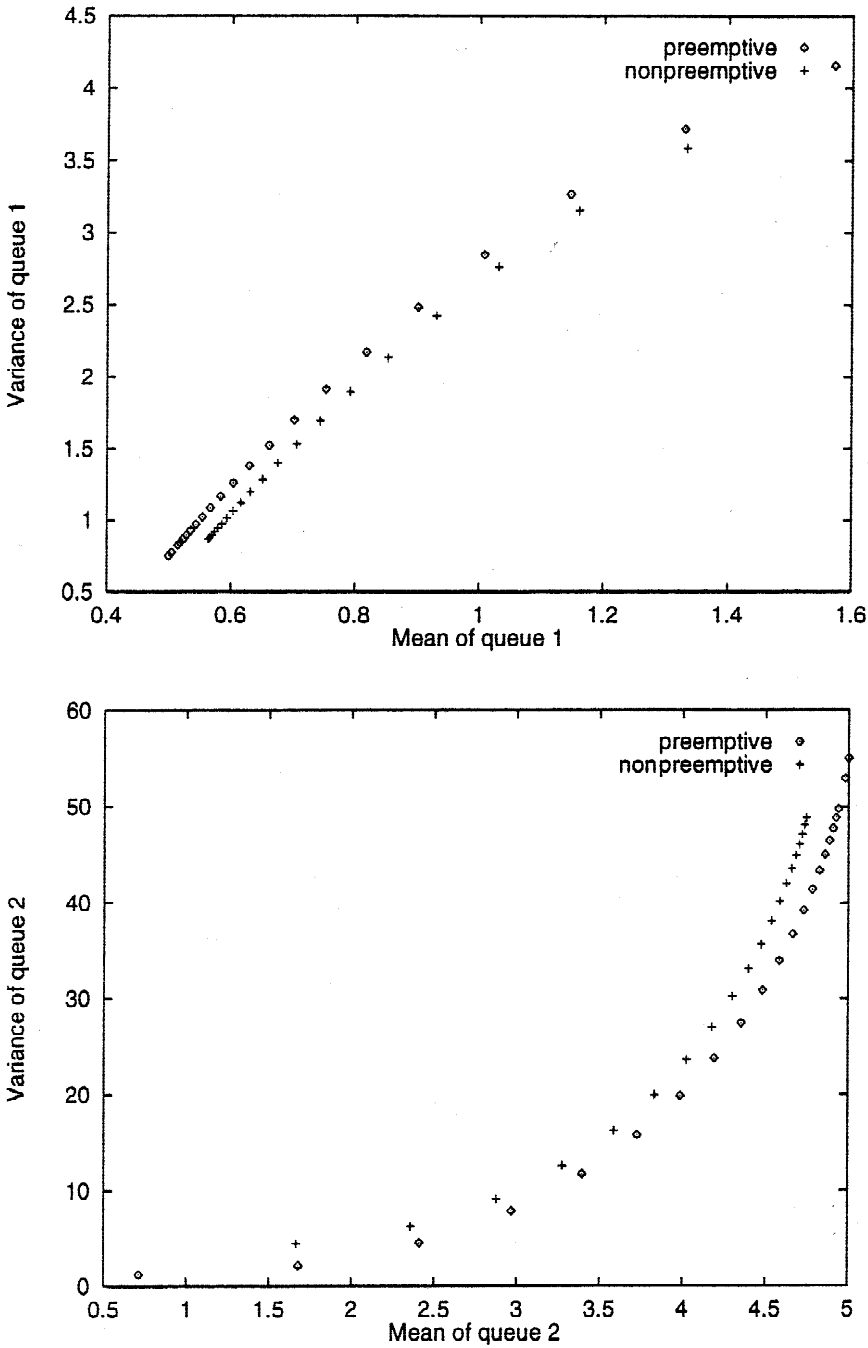

FIGURE 3. Mean of queue *i* against variance of queue *i* for Example 2.

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

where the subscripts {1,2} and {2,1} mean "under the preemptive priority policy giving top priority to type 1" and "under the preemptive priority policy giving top priority to type 2," respectively. The moments on the right-hand side of (31) are obtained from the known solution to the $M/M/1$ priority queue (see Jaiswal [5]). Related expressions exist for type 2 jobs.

Using the Maple package, it is a fairly simple task to show the following, using the exact algebraic forms of $\mathrm{Var}_\alpha(N_1)$ and $\mathrm{Var}_\alpha(N_2)$, for $0 \leq \alpha \leq 1$,

1. $\mathrm{Var}_\alpha(N_1)$ is decreasing in α.
2. $\mathrm{Var}_\alpha(N_2)$ is increasing in α.

Thus, substituting (31) into (29) and solving for α yields an inequality $\alpha \geq \alpha_1$, where $0 \leq \alpha_1 \leq 1$, if there are mixed policies that satisfy (29). Similarly, (30) implies $\alpha \leq \alpha_2$, where $0 \leq \alpha_2 \leq 1$, if there are mixed-priority policies that satisfy (30). If $\alpha_1 \leq \alpha_2$, then both variance constraints can be satisfied by a mixed-priority policy and one of the two extreme values provides the best policy for the variance constrained problem from this family.

More specifically, let the variance constraints be, say,

$$\mathrm{Var}(N_1) \leq 3.268, \tag{32}$$

$$\mathrm{Var}(N_2) \leq 32.460, \tag{33}$$

where the constraints are chosen to correspond to specific threshold policies. The variance pairs achievable by the preemptive threshold policies, dual threshold policies, and the mixed-priority policies are shown in Figures 4 and 5. Figures 4 and 5 also illustrate how the variance constraints (32) and (33) restrict the set of achievable policies. It is clear that there is a narrow range of performance pairs achievable by the randomized policies, and a much wider one achievable by the threshold policies. Indeed, there will be pairs of constraints that cannot be achieved by any mixed priority policy that can be achieved by one or more threshold policies.

Let $\mathbf{c} = (10,1)$. Note that $c_1\mu_1 > c_2\mu_2$ and so we concentrate on the original type of threshold policy for this problem. We find that the best policy has $T = 20$ (see Fig. 4). The cost of employing this policy is 10.695. The best mixed-priority policy has an optimal cost of 13.243, 24% worse than the cost of the optimal threshold policy. However, if $c_1\mu_1 < c_2\mu_2$ [e.g., if $\mathbf{c} = (1,1)$], then we concentrate on the dual family of threshold policies. The best of these has $T' = 6$ (see Fig. 5) and a cost of 2.577. The cost of the best mixed-priority policy has an optimal cost of 3.366. Thus, the best threshold policy is doing more than 30% better than the best mixed-priority policy in cost terms. For this system, without constraints, the set of achievable performance pairs is a line segment defined by the conservation law and inequalities

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

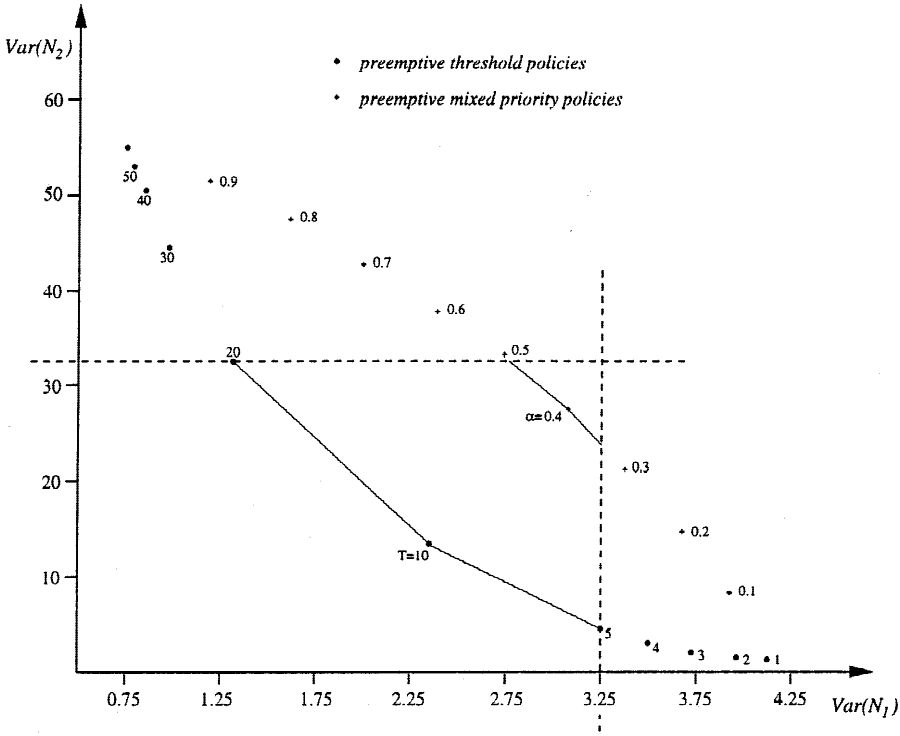

FIGURE 4. Achievable variance pairs for Example 2 using the threshold family of policies.

$$\frac{1}{\mu_1}E(N_1)+\frac{1}{\mu_2}E(N_2)=\frac{\rho_1\mu_1^{-1}+\rho_2\mu_2^{-1}}{1-\rho_1-\rho_2},$$

$$E(N_1)\geq\frac{\rho_1}{1-\rho_1},$$

$$E(N_2)\geq\frac{\rho_2}{1-\rho_2},$$

where μ_k and ρ_k are the service rate and traffic intensity for type k ($k=1,2$). The extreme points of the line segment are the performance pairs P_{12} and P_{21}, corresponding to the policies which give strict preemptive priority to type 1 and type 2 jobs, respectively (see Gelenbe and Mitrani [4]). Figure 6 illustrates the results given above in relation to the performance region for mean pairs. Section A is the set of expected queue lengths achievable by the threshold policies satisfying the constraints, whereas Section B is the set of expected queue lengths achievable by mixed-priority policies. We have looked at many different parameter values and imposed a

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

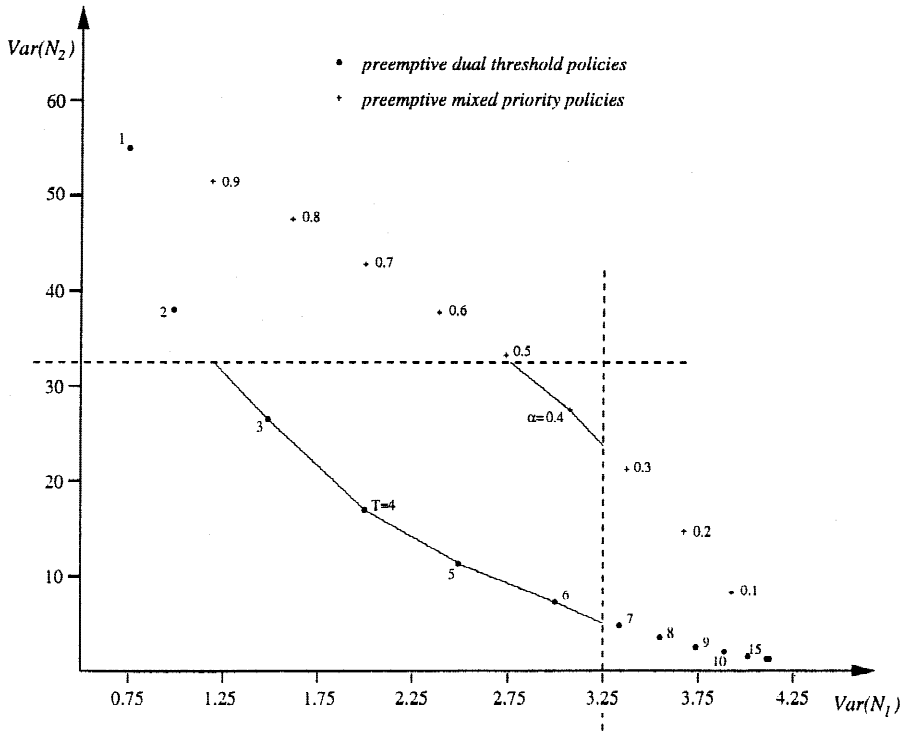

FIGURE 5. Achievable variance pairs for Example 2 using the dual family of threshold policies.

wide variety of variance constraints. The example presented here is typical of the results obtained.

We can carry out similar calculations and obtain similar results when considering nonpreemptive service policies.

7. CONCLUSION

We have provided analyses of both preemptive and nonpreemptive threshold policies. These seem to offer a realistic means of scheduling jobs in a way that mitigates the effect of excessive and unpredictable queue lengths. Such considerations are important if they are to be implemented in a real system. Moreover, we have used the power series algorithm to solve both the models. This will be advantageous when extending the approach to more than two queues. Computational evidence is given to show that the family of threshold policies outperforms a family of mixed-priority policies for a stochastic optimization problem in which policies must satisfy constraints on the variances of the queue lengths.

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press

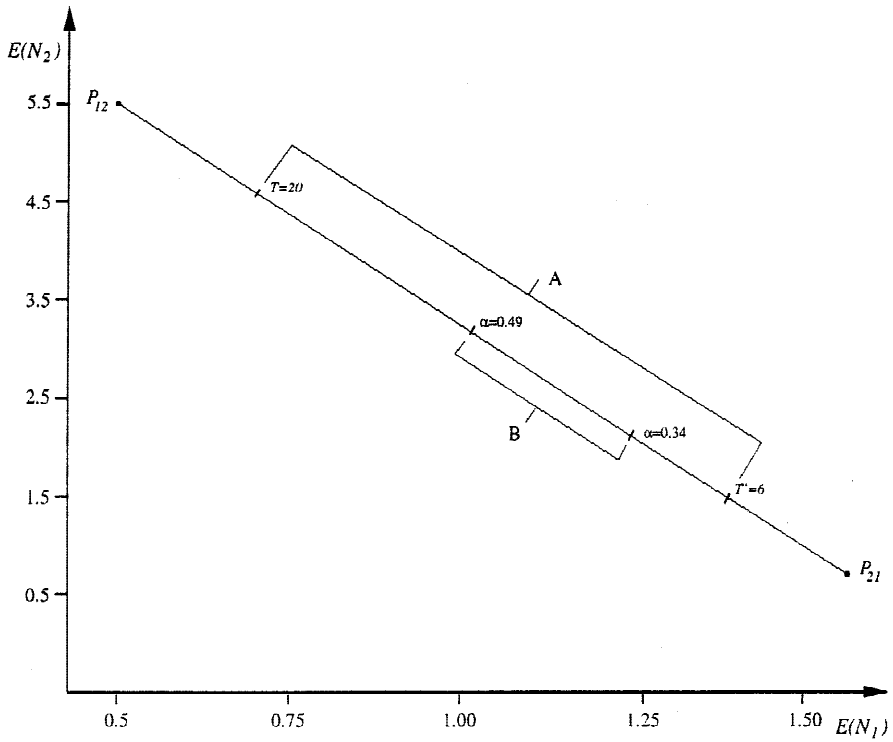

FIGURE 6. Achievable performance pairs for Example 2.

Acknowledgment

We express our appreciation to the Engineering and Physical Sciences Research Council for supporting the work of the first author by means of a research studentship and the second author through the award of grants GR/K03043 and GR/M09308.

References

1. Ansell, P.S., Glazebrook, K.D., & Mitrani, I. (1996). Server allocation subject to variance constraints. *Performance Evaluation* 27/28: 147–158.
2. Blanc, J.P.C. (1993). Performance analysis and optimization with the power-series algorithm. In L. Donatiello & R. Nelson (eds.), *Performance evaluation of computer and communication systems.* Lecture Notes in Computer Science 729, Berlin: Springer-Verlag, pp. 53–80.
3. Boxma, O.J., Koole, G.M., & Mitrani, I. (1995). A two-queue polling model with a threshold service policy. In P. Dowd & E. Gelenbe (eds.), *Proceedings MASCOTS '95.* Los Alamitos, CA: IEEE Computer Society Press, pp. 84–89.
4. Gelenbe, E. & Mitrani, I. (1980). *Analysis and synthesis of computer systems.* London: Academic.
5. Jaiswal, N.K. (1968). *Priority queues.* New York: Academic.
6. Lee, D-S. & Sengupta, B. (1993). Queueing analysis of a threshold based priority scheme for ATM networks. *IEEE/ACM Transactions on Networking* 1: 709–717.
7. van den Hout, W.B. (1996). The power series algorithm. Unpublished Ph.D. thesis, Centre for Economic Research, Tilburg University, Tilburg, The Netherlands.

https://doi.org/10.1017/S0269964801151028 Published online by Cambridge University Press