Queueing Systems 44, 137–160, 2003

Stability Conditions of the MMAP$[K]/G[K]/1$/LCFS Preemptive Repeat Queue

QI-MING HE Qi-Ming.He@dal.ca
Department of Industrial Engineering, Dalhousie University, Halifax, NS, Canada B3J 2X4

HUI LI hli@msvu1.msvu.ca
Department of Mathematics, Mount Saint Vincent University, Halifax, NS, Canada B3M 2J6

Abstract. In this paper, we study the stability conditions of the MMAP$[K]/G[K]/1$/LCFS preemptive repeat queue. We introduce an embedded Markov chain of matrix $M/G/1$ type with a tree structure and identify conditions for the Markov chain to be ergodic. First, we present three conventional methods for the stability problem of the queueing system of interest. These methods are either computationally demanding or do not provide accurate information for system stability. Then we introduce a novel approach that develops two linear programs whose solutions provide sufficient conditions for stability or instability of the queueing system. The new approach is numerically efficient. The advantages and disadvantages of the methods introduced in this paper are analyzed both theoretically and numerically.

Keywords: queueing system, stability, Markov process of matrix $M/G/1$ type with a tree structure, matrix analytic methods, linear programming

1. Introduction

Queueing systems with multiple types of customers have broad applications in the manufacturing, service, and telecommunication industries. A special class of such queueing systems consists of queues with a last-come-first-served (LCFS) preemptive repeat service discipline with equal priority to all customers. This class is special because these queueing systems can be non-work conserving. To our knowledge, little has been done on the stability conditions of such queueing systems. In this paper, we consider the MMAP$[K]/G[K]/1$/LCFS preemptive repeat queue and introduce a novel approach for solving its system stability problem.

For queueing systems with K types of customers, let $\lambda_1, \lambda_2, \ldots$, and λ_K be the arrival rates of type 1, type 2, ..., and type K customers, respectively. Let $\mu_1, \mu_2, \ldots$, and μ_K be the service rates of type 1, type 2, ..., and type K customers, respectively. If the service discipline is work conserving (e.g., FIFS or LCFS preemptive resume), under fairly general conditions, the quantity $\rho = \lambda_1/\mu_1 + \cdots + \lambda_K/\mu_K$ provides information for a complete classification of the Markov chain and stability of the queueing system (see [3,14]). More specifically, it has been proved that a queueing system with a work conserving service discipline is stable if and only if $\rho < 1$. Unfortunately, that conclusion is not always true for non-work conserving queueing systems. Since many queueing systems with practical applications are non-work conserving, there is a need to develop efficient methods to solve their system stability problem.

For queueing networks with Poisson arrival processes and service times, some important progress has been made in recent years. In [13], a linear programming approach was developed for finding out whether or not a queueing network can reach steady state. In [4] and [2], a fluid model approach was developed to find necessary and sufficient conditions for system stability. For queueing systems with a single server, some stability conditions were found for simple cases with priority service disciplines such as the priority queue with three types of customers and Poisson arrivals in [3]. Using matrix analytic methods [16,17], necessary and sufficient conditions (Neuts' conditions) that can be applied to many queueing models were found. For a queueing system with Poisson arrival processes and a FCLS preemptive repeat service discipline, He [8] found a simple and explicit condition for a complete classification of the queueing system (see equation (4.4) in this paper). However, for non-Poisson arrival queues, none of these methods or results can be applied directly to find system stability conditions.

In this paper, we consider queueing systems with arrival process MMAP[K] [1,12], general service times, and a LCFS preemptive repeat service discipline. We shall describe three conventional methods and also develop a new approach to find sufficient conditions for system stability. We shall compare the four approaches analytically and numerically.

The methodology used in this paper to find conditions for system stability was developed in [8–10]. Basically, we consider $M/G/1$ type Markov chains with a tree structure [18,20]. According to the results obtained in [8,9], the Perron–Frobenius eigenvalue of a nonnegative matrix provides information for a complete classification of the Markov chain of interest. We call this approach the *PFE* method. For the queueing system of interest, the queueing process can be formulated as a Markov chain with a tree structure. Thus, we can apply the PFE method to solve system stability problems. In [11], the PFE method was applied to queueing systems with multiple types of customers with an LCFS-GPR (general preemptive resume) service discipline. In [19], the PFE method was applied to evaluate a telecommunication system. While these applications show the usefulness of the PFE method, they also raise questions about the computational efficiency of that method. The reason is that the PFE method requires the calculation of a fixed point of a set of nonlinear equations. Therefore, we introduce a novel and computationally efficient approach for sufficient conditions for system stability or instability, based on two sufficient conditions obtained in [9]. Both analytical and numerical results will show that the methods developed in this paper can be useful in practice.

An interesting observation from the numerical examples given in section 7 is that, even if the classical traffic intensity ρ is significantly larger than 1, the LCFS queue introduced in section 3 (non-work conserving) can still be stable. This observation is counterintuitive but useful in the design of queueing systems.

The rest of the paper is organized as follows. In section 2, we introduce $M/G/1$ type Markov chains with a tree structure and some results about the ergodicity of such Markov processes. In section 3, we introduce the MMAP[K]/G[K]/1/LCFS preemptive repeat queue. Then we present three conventional methods for system stability in section 4. In section 5, a novel approach is developed for system stability. Two linear programs are introduced for sufficient conditions for system stability or instability. In section 6, some computational details are dealt with. Finally, in section 7, some numerical examples are given to gain insight into the methods developed in this paper and to draw general conclusions about the usefulness of these methods.

2. The Markov chain of matrix $M/G/1$ type with a tree structure

In this section, we first introduce the $M/G/1$ type Markov chain with a tree structure. Then we summarize some existing results about the ergodicity of such Markov chains.

The following discrete time Markov process of matrix $M/G/1$ type with a tree structure was first introduced in [18]. Consider a discrete time two-dimensional Markov chain $\{(C_n, \eta_n),\ n \geqslant 0\}$ in which the values of C_n are represented by the nodes of a K-ary tree, and η_n takes integer values between 1 and m, where m is a positive integer. C_n is referred to as the variable for the node and η_n is referred to as the auxiliary variable of the Markov chain at time n. Next, we give a full description of the Markov chain $\{(C_n, \eta_n),\ n \geqslant 0\}$.

The K-ary tree of interest is a tree for which each node, except the root node, has a parent and K children. The root node is denoted as 0. No boundary node is linked to the root node. Strings of integers between 1 and K are used to represent nodes of the tree. For instance, the kth child of the root node is represented by k, the lth child of node k is represented by kl, and so on.

Let $\aleph = \{J\colon\ J = k_1k_2\ldots k_n,\ 1 \leqslant k_i \leqslant K,\ 1 \leqslant i \leqslant K,\ n > 0\} \cup \{0\}$. Any string $J \in \aleph$ is a node in the K-ary tree. The length of a string J is defined to be the number of integers in the string and is denoted by $|J|$. When $J = 0$, $|J| = 0$. Let $N(J, k)$ be the number of appearances of integer k in the string J. The following two operations are defined for strings in $\aleph$:

- *Addition operation*: given $J = k_1 \ldots k_n \in \aleph$ and $H = h_1 \ldots h_i \in \aleph$, then $J + H = k_1 \ldots k_n h_1 \ldots h_i \in \aleph$;
- *Subtraction operation*: given $J = k_1 \ldots k_n \in \aleph$, $H = k_i \ldots k_n \in \aleph$, $i > 0$, then $J - H = k_1 \ldots k_{i-1} \in \aleph$.

The Markov chain (C_n, η_n) takes values in $\aleph \times \{1, 2, \ldots, m\}$. If (C_n, η_n) transits at each step from the current node only to its parent node or a descendent of its parent node, then it is called a homogenous Markov chain of matrix $M/G/1$ type with a tree structure. All possible transitions of (C_n, η_n) and their corresponding transition probabilities are given as follows. If $(C_n, \eta_n) = (0, i)$,

1. $(C_{n+1}, \eta_{n+1}) = (J, i')$ with probability $b_{(i,i')}(J)$ for $J \in \aleph$.

If $(C_n, \eta_n) = (H + k, i)$ for $k > 0$ and $1 \leqslant i, i' \leqslant m$,

2. $(C_{n+1}, \eta_{n+1}) = (H + J, i')$ with probability $a_{(i,i')}(k, J)$ for $J \in \aleph$.

Note that the transition probability depends only on the last integer in the string representing the current node $H + k$. In matrix form, transition probabilities are represented by $A(k, J)$ and $B(J)$, where

3. $B(J)$ is an $m \times m$ matrix with elements $b_{(i,i')}(J)$ for $J \in \aleph$;
4. $A(k, J)$ is an $m \times m$ matrix with elements $a_{(i,i')}(k, J)$, $1 \leqslant k \leqslant K$, for $J \in \aleph$.

For convenience, we shall occasionally use $A(H, J)$ to denote the transition probability matrix from node H to node J for any $H, J \in \aleph$. Let

$$\begin{aligned} A(k) &= \sum_{J\in\aleph} A(k, J), & 1 \leqslant k \leqslant K; \\ B^{*(1)}(k) &= \sum_{J\in\aleph} B(J)N(J, k), & 1 \leqslant k \leqslant K. \end{aligned} \tag{2.1}$$

Then we must have $A(k)\mathbf{e} = \mathbf{e}$, $1 \leqslant k \leqslant K$, and $\sum_{J\in\aleph} B(J)\mathbf{e} = \mathbf{e}$, where $\mathbf{e}$ is the vector with all components being one.

Ergodicity conditions. The following ergodicity conditions were obtained in [9] for the Markov chain $\{(C_n, \eta_n),\ n \geqslant 0\}$. Let $\mathbf{X} = \{X_1, \ldots, X_K\}$, where $X_1, X_2, \ldots,$ and X_K are $m \times m$ stochastic matrices, i.e., $X_k\mathbf{e} = \mathbf{e}$, $1 \leqslant k \leqslant K$. Let $\Re$ be a set consisting of $\mathbf{X}$ for which $\{X_1, X_2, \ldots, X_K\}$ are stochastic matrices and satisfy the following equations, $1 \leqslant k \leqslant K$ $(J = k_1 \ldots k_{|J|})$

$$X_k = A(k, 0) + \sum_{J\in\aleph, J\neq 0} A(k, J)X_{k_{|J|}}X_{k_{|J|-1}} \cdots X_{k_1}. \tag{2.2}$$

Using the well-known Brouwer's fixed point theorem [7], it was shown in He [9] that the set $\Re$ is nonempty. For any fixed point $\mathbf{X} = \{X_1, \ldots, X_K\} \in \Re$, define $X^{(J)} = X_{k_{|J|}}X_{k_{|J|-1}} \cdots X_{k_1}$ for all $J = k_1 \ldots k_{|J|} \in \aleph$ and define the following $m \times m$ matrices, for $J = k_1 \ldots k_{|J|} \in \aleph$,

$$\begin{aligned} N(0, j, \mathbf{X}) &= 0, & 1 \leqslant j \leqslant K; \\ N(J, j, \mathbf{X}) &= I\delta(k_{|J|}, j) + \sum_{n=1}^{|J|-1} X_{k_{|J|}}X_{k_{|J|-1}} \cdots X_{k_{n+1}}\delta(k_n, j), & 1 \leqslant j \leqslant K; \end{aligned} \tag{2.3}$$

$$p(k, j, \mathbf{X}) = \sum_{J\in\aleph} A(k, J)N(J, j, \mathbf{X}), \quad 1 \leqslant k,\ j \leqslant K,$$

where $\delta(k, j) = 1$, if $k = j$; $\delta(k, j) = 0$, otherwise, and I is the identity matrix. Note that $N(J, j, \mathbf{X})$ counts the number of appearances of integer j in the string J, while keeping track of the phase changes in the transition process. The matrix $p(k, j, \mathbf{X})$ can be interpreted as the average number of appearances of integer j in the next transition, given that the Markov chain is currently in a node $H+k$ for $H \in \aleph$. Define an $(mK) \times (mK)$ matrix $P(\mathbf{X})$ by

$$P(\mathbf{X}) = \begin{pmatrix} p(1,1,\mathbf{X}) & \dots & p(1,K,\mathbf{X}) \\ \vdots & \vdots & \vdots \\ p(K,1,\mathbf{X}) & \dots & p(K,K,\mathbf{X}) \end{pmatrix}. \tag{2.4}$$

Let $sp(P(\mathbf{X}))$ be the Perron–Frobenius eigenvalue of the matrix $P(\mathbf{X})$ (i.e., the eigenvalue with the largest real part, which is real and nonnegative for nonnegative matrices [6]). The following result was proved in [9].

Theorem 2.1 [9, theorem 3.2]. Assume that the Markov chain $\{(C_n, \eta_n),\ n \geqslant 0\}$ is irreducible and aperiodic and that $B^{*(1)}(k)$ is finite for $1 \leqslant k \leqslant K$. For any set $\mathbf{X} \in \Re$, if the matrix $P(\mathbf{X})$ is irreducible, then the Markov chain of matrix $M/G/1$ type with a tree structure $\{(C_n, \eta_n),\ n \geqslant 0\}$ is

(1) positive recurrent if and only if $sp(P(\mathbf{X})) < 1$;

(2) null recurrent if and only if $sp(P(\mathbf{X})) = 1$;

(3) transient if and only if $sp(P(\mathbf{X})) > 1$.

When $m = 1$, $\mathbf{X}$ is reduced to $\mathbf{X} = \{1, 1, \dots, 1\}$. Then theorem 2.1 gives an explicit ergodicity condition [8, theorem 3.2]. When $m > 1$, since the matrix set $\mathbf{X}$ has to be calculated in order to construct the matrix $P(\mathbf{X})$, the usefulness of theorem 2.1 is compromised. Thus, it is desirable to find ergodicity conditions not depending on $\mathbf{X}$.

Let $\mathbf{z} = (z_1, \dots, z_K)$, $z_k \geqslant 0$, $1 \leqslant k \leqslant K$. Define, for $1 \leqslant k \leqslant K$,

$$A^*(k, \mathbf{z}) = \sum_{J \in \aleph} \mathbf{z}^{(J)} A(k, J), \qquad B^*(\mathbf{z}) = \sum_{J \in \aleph} \mathbf{z}^{(J)} B(J), \tag{2.5}$$

where $\mathbf{z}^{(J)} = z_{k_{|J|}} \cdots z_{k_1}$ for $J = k_1 k_2 \dots k_{|J|} \in \aleph$ and $J \neq 0$, and $\mathbf{z}^{(J)} = 1$ if $J = 0$.

Lemma 2.2 [9, lemma 6.1]. Assume that the Markov chain $\{(C_n, \eta_n),\ n \geqslant 0\}$ is irreducible and aperiodic and that $B^{*(1)}(k)$ is finite for $1 \leqslant k \leqslant K$. If there exists an $m \times 1$ positive vector $\mathbf{u}$ such that $A^*(k, \mathbf{z})\mathbf{u} < z_k \mathbf{u}$ (i.e., every element of $A^*(k, \mathbf{z})\mathbf{u}$ is strictly smaller than its counterpart in $z_k \mathbf{u}$), and $B^*(\mathbf{z})\mathbf{u} < \infty$ for some $\mathbf{z}$ satisfying $1 < z_k < \infty$, $1 \leqslant k \leqslant K$, then the Markov chain $\{(C_n, \eta_n),\ n \geqslant 0\}$ is positive recurrent.

Lemma 2.3 [9, lemma 6.2]. Assume that the Markov chain $\{(C_n, \eta_n),\ n \geqslant 0\}$ is irreducible and aperiodic and that $B^{*(1)}(k)$ is finite for $1 \leqslant k \leqslant K$. If there exists an $m \times 1$ positive vector $\mathbf{u}$ such that $A^*(k, \mathbf{z})\mathbf{u} \leqslant z_k \mathbf{u}$ for some $\mathbf{z}$ satisfying $0 < z_k < 1$ for at least one k, $1 \leqslant k \leqslant K$, then the Markov chain $\{(C_n, \eta_n),\ n \geqslant 0\}$ is transient.

Lemmas 2.2 and 2.3 can be proved using the mean-drift method or, more specifically, the Foster criteria [5,15]. In [9], lemmas 2.2 and 2.3 were applied to queueing systems to find system stability conditions. In section 5, we present another interesting application of these lemmas.

3. The continuous time MMAP$[K]/G[K]/1$/LCFS preemptive repeat queue

In this section, we introduce the queueing model of interest. First, we introduce the customer arrival process MMAP$[K]$ [1,12].

The Markov arrival process with marked arrivals (MMAP$[K]$) is represented by a set of $m \times m$ matrices $\{D_J,\ J \in \aleph\}$, where D_0 is a matrix with negative diagonal elements and nonnegative off-diagonal elements, and $\{D_J,\ J \neq 0,\ J \in \aleph\}$ are nonnegative matrices. For $J = k_1 k_2 \ldots k_{|J|} \in \aleph$, D_J is the (matrix) arrival rate of the batch J, which has $|J|$ customers and the first customer in the batch is of type k_1, second type $k_2, \ldots$, and the last type $k_{|J|}$. The matrix $D = \sum_{J \in \aleph} D_J$ is an infinitesimal generator of the underlying Markov process $\eta(t)$. For $\mathbf{z} = (z_1, z_2, \ldots, z_K)$, let

$$D^*(\mathbf{z}) = \sum_{J \in \aleph} \mathbf{z}^{(J)} D_J. \tag{3.1}$$

We assume that D is irreducible and the matrix $D^*(\mathbf{z})$ is analytic at $\mathbf{z} = \mathbf{e}^{\mathrm{T}}$, where the superscript $^{\mathrm{T}}$ indicates the transpose of matrix. Let $\boldsymbol{\theta}$ be the stationary distribution of the matrix D, i.e., $\boldsymbol{\theta} D = 0$ and $\boldsymbol{\theta}\mathbf{e} = 1$. The arrival rate of type k customers is given by

$$\lambda_k = \boldsymbol{\theta} \sum_{n=1}^{\infty} n \left(\sum_{J:\ N(J,k)=n} D_J \right) \mathbf{e} = \boldsymbol{\theta} \sum_{J \in \aleph} D_J N(J,k) \mathbf{e}, \quad 1 \leqslant k \leqslant K. \tag{3.2}$$

Recall that $N(J,k)$ is the number of appearance of the integer k in the string J (section 2). The service time of a type k customer has a distribution function $F_k(x)$ with Laplace–Stieltjes transform $f_k^*(s)$ and mean $1/\mu_k$. The customer arrival process and the service process are independent.

All the customers are served on a last-come-first-served preemptive repeat basis. That is, when a new customer arrives, it pushes the customer in the server (if any) out of the server and starts its own service immediately. In case of a batch arrival, the last customer in the batch enters the server. All other customers in the batch join the queue, according to their order in the batch, after the customer who is pushed out of service. When the server becomes available to customers in the queue, the newest customer reenters the server and restarts its service as a new customer.

The nth customer in a batch $J = k_1 k_2 \ldots k_{|J|} \in \aleph$ is a type k_n customer, $1 \leqslant k_n \leqslant K$, $1 \leqslant n \leqslant |J|$. Thus, the queue of such a queueing system can be represented by a string of integers representing the type of customer in each position. Let $q(t)$ be the queue string at time t. Then, for example, $q(t) = 122$ means that there are 3 customers in the system at time t: the customer who arrived first is of type 1; the customer who arrived second is of type 2; and the customer who arrived last is of type 2. The server is serving the last customer (type 2). When a batch J of customers arrives, the queue string becomes $q(t) = 122 + J$ and the server starts to serve the customer $k_{|J|}$.

We observe the queueing system at arrival and departure epochs. Let q_n be the queue string right after the nth arrival or departure epoch. Let η_n be the phase of the underlying Markov process $\eta(t)$ of the Markov arrival process right after the nth arrival or departure epoch. Then $\{(q_n, \eta_n),\ n \geqslant 0\}$ is a Markov chain of matrix $M/G/1$ type with a tree structure. It can be verified that the Markov chain $\{(q_n, \eta_n),\ n \geqslant 0\}$ is irreducible and aperiodic. Transition probabilities (matrices) of the Markov chain are given by, for $1 \leqslant k \leqslant K$,

$$\begin{aligned} A(k, 0) &= \int_0^\infty F_k(\mathrm{d}x) \exp\{x D_0\} = f_k^*(D_0); \\ A(k, k+J) &= \int_0^\infty F_k(\mathrm{d}x) \int_0^x \exp\{t D_0\}\, \mathrm{d}t\, D_J \\ &= \big(I - f_k^*(D_0)\big)\big(-D_0^{-1} D_J\big), \quad J \in \aleph,\ J \neq 0; \\ A(k, k) &= 0; \qquad A(k, j+J) = 0, \qquad j \neq k,\ 1 \leqslant j \leqslant K,\ J \in \aleph. \end{aligned} \tag{3.3}$$

Transition matrices $\{B(J),\ J \in \aleph\}$ associated with the root node are not given explicitly since they are not used in the sequel. For later use, we introduce the following functions, for $1 \leqslant k \leqslant K$,

$$A^*(k, \mathbf{z}) \equiv \sum_{J \in \aleph} A(k, J) \mathbf{z}^{(J)} = f_k^*(D_0) + \big(I - f_k^*(D_0)\big)\big(-D_0^{-1}\big)\big(D^*(\mathbf{z}) - D_0\big) z_k. \tag{3.4}$$

With the Markov chain $\{(q_n, \eta_n),\ n \geqslant 0\}$, we are ready to address the stability problem of the queueing system of interest. We say that the queueing system is stable if the Markov chain $\{(q_n, \eta_n),\ n \geqslant 0\}$ is ergodic; unstable, otherwise.

4. Three conventional approaches

4.1. *The PFE method*

The ergodicity problem of the Markov chain $\{(q_n, \eta_n),\ n \geqslant 0\}$ can be completely solved by applying theorem 2.1. We call this approach the *Perron–Frobenius eigenvalue* (PFE) method. First, since the matrices $\{A(k, J),\ 1 \leqslant k \leqslant K,\ J \in \aleph\}$ are given explicitly in equation (3.3), equation (2.2) becomes

$$X_k = f_k^*(D_0) + \big[I - f_k^*(D_0)\big](-D_0)^{-1}\big(D^*(\mathbf{X}) - D_0\big) X_k, \quad 1 \leqslant k \leqslant K, \tag{4.1}$$

which does not involve the matrices $\{A(k, J)\}$ directly. Similarly, the matrix $p(k, j, \mathbf{X})$ has a simpler expression, for $1 \leqslant k, j \leqslant K$,

$$\begin{aligned} p(k, j, \mathbf{X}) &= \big[I - f_k^*(D_0)\big](-D_0)^{-1}\Bigg(\sum_{J \neq 0, J \in \aleph} D_J N(k+J, j, \mathbf{X})\Bigg) \\ &= \big[I - f_k^*(D_0)\big](-D_0)^{-1}\Bigg(\sum_{J \neq 0, J \in \aleph} D_J N(J, j, \mathbf{X}) + \delta(k, j) \sum_{J \neq 0, J \in \aleph} D_J \mathbf{X}^{(J)}\Bigg). \end{aligned} \tag{4.2}$$

The matrix $P(\mathbf{X})$ becomes

$$P(\mathbf{X}) = \begin{pmatrix} [I - f_1^*(D_0)](-D_0)^{-1} \\ \vdots \\ [I - f_K^*(D_0)](-D_0)^{-1} \end{pmatrix} \times \left(\sum_{J\neq 0, J\in\aleph} D_J N(J,1,\mathbf{X}) \quad \ldots \quad \sum_{J\neq 0, J\in\aleph} D_J N(J,K,\mathbf{X}) \right) + \begin{pmatrix} [I - f_1^*(D_0)](-D_0)^{-1} \sum_{J\neq 0, J\in\aleph} D_J \mathbf{X}^{(J)} & & \\ & \ddots & \\ & & [I - f_K^*(D_0)](-D_0)^{-1} \sum_{J\neq 0, J\in\aleph} D_J \mathbf{X}^{(J)} \end{pmatrix}. \tag{4.3}$$

Since the underlying Markov process D is irreducible, it is easy to see that $P(\mathbf{X})$ is irreducible. We can find whether or not the queueing system is stable as follows:

(1) calculate $f_k^*(D_0)$;

(2) use equation (4.1) for computing $\mathbf{X} = \{X_k,\ 1 \leqslant k \leqslant K\}$;

(3) construct the matrix $P(\mathbf{X})$ using equation (4.3); and

(4) calculate $sp(P(\mathbf{X}))$ and apply theorem 2.1.

Of these four steps, (1) and (3) are not straightforward. While these two steps can be completed using existing methods, we shall give a simple method for step (1) for a special case (i.e., queueing systems with PH-distributed service times). In fact, step (1) is critical for all the methods given in this paper. Therefore, we shall come back to it in section 6.

The PFE method for stability can be difficult to use because we have to compute the matrices $\{X_k,\ 1 \leqslant k \leqslant K\}$, which can be numerically challenging when the dimension m becomes large. Thus, we still need to explore other simpler approaches for the stability problem.

4.2. The FPP method

The commonly used matrices $\{G(k),\ 1 \leqslant k \leqslant K\}$ in matrix analytic methods can also be used for finding out whether or not the queueing system is stable. We call this approach the *first-passage probability* (FPP) method. Let $G(k)_{i,j}$ be the probability that the Markov chain $\{(q_n, \eta_n),\ n \geqslant 0\}$ reaches the node J in (J, j), given that it started in $(J + k, i)$. Let $G(k)$ be an $m \times m$ matrix with (i, j)th element $G(k)_{i,j}$. It can be shown that $\{G(k),\ 1 \leqslant k \leqslant K\}$ are the minimal nonnegative solutions to equation (4.1). According to Takine et al. [18], $\mathbf{G} = \{G(k),\ 1 \leqslant k \leqslant K\}$ are all stochastic matrices if and only if the Markov chain is recurrent. If the Markov chain is recurrent, $\mathbf{G} \in \Re$ and is the only stochastic solution to equation (4.1), i.e., $\mathbf{G} = \mathbf{X}$. If not all matrices in $\mathbf{G}$ are stochastic, i.e., $\mathbf{G} \notin \Re$ and $\mathbf{G} \neq \mathbf{X}$, the Markov chain is transient. Therefore, we can use $\mathbf{G}$ to check whether or not the queueing system is stable. However, we cannot use $\mathbf{G}$ to distinguish the positive recurrent and null recurrent cases. Furthermore, we encounter the same computational difficulty as in the PFE method when m is large.

4.3. *The PTI method*

The third approach to system stability is to find some single measurement similar to the usual traffic intensity ρ (defined in section 1). We call this approach the *pseudo-traffic intensity* (PTI) method. If $m = 1$, let

$$\rho_1 = \sum_{k=1}^{K} \frac{\lambda_k(1 - f_k^*(D_0))}{(-D_0) f_k^*(D_0)}. \tag{4.4}$$

He [8] proved that ρ_1 provides information for a complete classification of the ergodicity of the Markov chain, i.e., $\rho_1 < 1$ for positive recurrent, $\rho_1 = 1$ for null recurrent, and $\rho_1 > 1$ for transient. In fact, if $m = 1$, there is a close relationship between ρ_1, $sp(P(\mathbf{X}))$, and ρ. We summarize the results in the following property.

Property 4.1. Consider the continuous time MMAP[K]/G[K]/1/LCFS preemptive repeat queue defined in section 3. We assume that $m = 1$, i.e., the underlying Markov process of the arrival process has a single phase.

(1) The quantity ρ_1 defined in equation (4.4) is consistent with $sp(P(\mathbf{X}))$ with respect to the stability of the queueing system of interest. Furthermore, if $\rho_1 < 1$, then $\rho_1 < sp(P(\mathbf{X})) < 1$; if $\rho_1 = 1$, then $\rho_1 = sp(P(\mathbf{X})) = 1$; and if $\rho_1 > 1$, then $\rho_1 > sp(P(\mathbf{X})) > 1$.

(2) If all the service times are exponentially distributed, then $\rho_1 = \rho$.

Proof. First, note that $X_1 = X_2 = \cdots = X_K = 1$ if $m = 1$. He [8] proved that both ρ_1 and $sp(P(\mathbf{X}))$ provide information for a complete classification of the Markov chain with a tree structure. Here we present another proof. First, a simpler expression of $P(\mathbf{X})$ can be derived from equation (4.3) if $m = 1$. Define a vector $\boldsymbol{\beta} = (\lambda_1/f_1^*(D_0), \lambda_2/f_2^*(D_0), \ldots, \lambda_K/f_K^*(D_0))$. It is easy to verify that $\boldsymbol{\beta} P(\mathbf{X}) = (\rho_1 - 1)\boldsymbol{\lambda} + \boldsymbol{\beta}$, where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_K)$. Since $P(\mathbf{X})$ is irreducible, there exists a positive vector $\mathbf{u}$ such that $P(\mathbf{X})\mathbf{u} = sp(P(\mathbf{X}))\mathbf{u}$ and $\boldsymbol{\beta}\mathbf{u} = 1$. Then we have $\boldsymbol{\beta} P(\mathbf{X})\mathbf{u} = (\rho_1 - 1)\boldsymbol{\lambda}\mathbf{u} + \boldsymbol{\beta}\mathbf{u}$, which leads to $sp(P(\mathbf{X})) - 1 = (\rho_1 - 1)\boldsymbol{\lambda}\mathbf{u}$. Now, it is clear that ρ_1 and $sp(P(\mathbf{X}))$ are consistent with respect to the stability of the queueing system. Since $0 < f_k^*(D_0) < 1$, for $1 \leqslant k \leqslant K$, it is easy to see that $0 < \boldsymbol{\lambda}\mathbf{u} < \boldsymbol{\beta}\mathbf{u} = 1$. Then the relationship between ρ_1 and $sp(P(\mathbf{X}))$ follows immediately. This proves part (1).

Part (2) follows from the fact that $f_K^*(D_0) = \mu_k(\mu_k - D_0)^{-1}$, if the service times of type k customers have an exponential distribution with mean $1/\mu_k$, $1 \leqslant k \leqslant K$. Part (2) is also intuitive since the service times are memoryless so that the queueing system is equivalent to a work conserving queue. □

Note. Properties 4.1, 4.2 and corollary 5.4 deal with the special cases with $m = 1$ or exponential service times. These results are interesting because they unveil some fundamental relationships between all the four methods presented in this paper for system stability, they are extremely useful in debugging and validating computational programs, and they shed light on generalizations from $m = 1$ to $m > 1$. In fact, many of the results and relationships obtained analytically for queues with $m = 1$ hold approximately for queues with $m > 1$. We will discuss this further in section 7.

By property 4.1, ρ_1 can be considered as the traffic intensity for the queueing system of interest with $m = 1$. The reason that an explicit formula is obtained for ergodicity for this case is that there is only one phase for the arrival process (i.e., $m = 1$), so no phase transition has to be considered (i.e., $X_k = 1$, $1 \leqslant k \leqslant K$).

Unfortunately, for the $m > 1$ case, it seems difficult to obtain an explicit formula for system stability. As an natural extension of ρ_1, for $m > 1$, we define the following pseudo-traffic intensity

$$\rho_2 = \sum_{k=1}^{K} \frac{\lambda_k}{(-\boldsymbol{\theta} D_0)(I - f_k^*(D_0))^{-1} f_k^*(D_0)\mathbf{e}}. \tag{4.5}$$

Let $\boldsymbol{\theta}(k) = -r_k \boldsymbol{\theta} D_0 (I - f_k^*(D_0))^{-1}$ and $r_k = -[\boldsymbol{\theta} D_0 (I - f_k^*(D_0))^{-1}\mathbf{e}]^{-1}$. Note that $-\boldsymbol{\theta} D_0 \geqslant 0$ so that $\boldsymbol{\theta}(k)$ is nonzero and nonnegative. Then the service rate of type k customers with a probability distribution $\boldsymbol{\theta}(k)$ is given by $\boldsymbol{\theta} D_0 (I - f_k^*(D_0))^{-1} f_k^*(D_0)\mathbf{e}$. Thus, intuitively, equation (4.5) can be interpreted as the ratio of arrival rates to service rates. Unfortunately, the phase distribution of the underlying Markov process of the arrival process may not be $\boldsymbol{\theta}(k)$ in steady state. Therefore, the service rate estimate may not be accurate. Consequently, the quantity ρ_2 defined in equation (4.5) may not provide accurate information for the classification of the queueing system. Counterexamples will be given in section 7. Nonetheless, the following property implies that ρ_2 is useful since it does provide correct information for the stability of many special cases of the queueing system of interest.

Property 4.2. Consider the continuous time MMAP$[K]/G[K]/1/$LCFS preemptive repeat queue defined in section 3. (1) If $m = 1$, then $\rho_2 = \rho_1$. (2) If all the service times are exponentially distributed (for $m \geqslant 1$), then $\rho_2 = \rho$. For these two cases, ρ_2 provides information for a complete classification of the queueing system of interest.

Proof. If $m = 1$, $\rho_2 = \rho_1$ by their definitions. If all the service times are exponentially distributed with mean $1/\mu_k$, we have $f_K^*(D_0) = \mu_k(\mu_k I - D_0)^{-1}$, $1 \leqslant k \leqslant K$. Then ρ_2 is reduced to ρ. □

Property 4.2 shows that ρ_2 does provide accurate information for system stability for many queueing systems of interest. In fact, numerical examples in section 7 will show that ρ_2 and $sp(P(\mathbf{X}))$ tend to be consistent with respect to system stability, though counterexamples can be easily found. The relationship between ρ_2 and $sp(P(\mathbf{X}))$ is similar to, but not the same as, that of ρ_1 and $sp(P(\mathbf{X}))$ if $m = 1$.

In summary, the above three approaches for system stability are either numerically difficult or may not provide correct information. Since it seems difficult, if not impossible, to obtain a simple performance measure for system stability of the queueing system of interest, in Section 5, we explore a novel approach to find sufficient conditions for the queueing system of interest to be stable or unstable.

5. New sufficient conditions for system stability and instability

In this section, we use lemmas 2.2 and 2.3 to find sufficient conditions for system stability or system instability by solving linear systems. We call this approach the *linear system* (LS) method. We shall first introduce two linear systems whose solutions provide information for system stability or instability. Then we propose a linear programming approach to solve the linear systems and so to find out whether or not the system is stable or unstable.

Let $\boldsymbol{\delta} = (\delta_1, \delta_2, \ldots, \delta_K)$ and $\mathbf{v} = (v_1, v_2, \ldots, v_m)$. Define a linear system with variables $(\boldsymbol{\delta}, \mathbf{v}, \varepsilon)$ by

$$\sum_{j=1}^{K} \delta_j \mathbf{d}(k, j) + \big(I - f_k^*(D_0)\big)\big(-D_0^{-1}\big)D\mathbf{v} + \varepsilon\mathbf{e} \leqslant 0, \quad 1 \leqslant k \leqslant K, \tag{5.1}$$

where $\boldsymbol{\delta} \geqslant 0$, $-\infty < v_j < \infty$, $1 \leqslant j \leqslant m$, $\varepsilon \geqslant 0$, and

$$\mathbf{d}(k, j) = \begin{cases} \big(I - f_k^*(D_0)\big)(-D_0)^{-1}\Big(\sum_{J\neq 0, J\in\aleph} D_J N(J, j)\Big)\mathbf{e}, \\ \quad 1 \leqslant j,\ k \leqslant K,\ j \neq k; \\ \big(I - f_k^*(D_0)\big)(-D_0)^{-1}\Big(\sum_{J\neq 0, J\in\aleph} D_J N(J, j)\Big)\mathbf{e} - f_k^*(D_0)\mathbf{e}, \\ \quad 1 \leqslant j = k \leqslant K. \end{cases} \tag{5.2}$$

Lemma 5.1. Consider the continuous time MMAP[K]/G[K]/1/LCFS preemptive repeat queue defined in section 3. If the linear system (5.1) has a solution $(\boldsymbol{\delta}, \mathbf{v}, \varepsilon)$ with positive δ_k, $1 \leqslant k \leqslant K$, and positive ε, then the Markov chain $\{(q_n, \eta_n),\ n \geqslant 0\}$ is ergodic and the queueing system is stable.

Proof. We use lemma 2.2 to prove lemma 5.1. First, we note that the boundary conditions (i.e., $B^{*(1)}(k)$ is finite for $1 \leqslant k \leqslant K$) are satisfied since all average arrival rates are finite.

The idea of the proof of lemma 5.1 is to choose a direction $\boldsymbol{\delta} = (\delta_1, \delta_2, \dots, \delta_K)$ with $\delta_k \geqslant 0$, $1 \leqslant k \leqslant K$, such that, when $z_k(t) = 1+\delta_k t$, $1 \leqslant k \leqslant K$ (i.e., $\mathbf{z}(t) = \mathbf{e}+\boldsymbol{\delta} t$) and $\mathbf{u}(t) = \mathbf{e} + t\mathbf{v}$, we have

$$A^*\big(k, \mathbf{z}(t)\big)\mathbf{u}(t) \leqslant z_k(t)\mathbf{u}(t) - t\varepsilon\mathbf{e}, \quad 1 \leqslant k \leqslant K, \tag{5.3}$$

with positive $\mathbf{u}(t)$ and $\boldsymbol{\delta}\mathbf{e} = 1$ for some positive ε and positive t. By using the Taylor expansion of $A^*(k, \mathbf{z}(t))$ with respect to t, the problem can be transformed into the linear program.

First, by the explicit expression (3.1) of $D^*(\mathbf{z})$, the following Taylor expansion holds:

$$D^*\big(\mathbf{z}(t)\big) = \sum_{J\in\aleph} \mathbf{z}(t)^{(J)} D_J = D + t\sum_{j=1}^{K} \delta_j \sum_{J\neq 0, J\in\aleph} D_J N(J, j) + \mathrm{o}(t). \tag{5.4}$$

By the explicit expression (3.3) of $A^*(k, \mathbf{z})$, both sides of inequality (5.3) can be expanded and evaluated in the following way: for $t > 0$ and $1 \leqslant k \leqslant K$,

$$\begin{aligned}
&\Bigg[f_k^*(D_0) + \big(I - f_k^*(D_0)\big)\big(-D_0^{-1}\big)\Bigg(\sum_{J\neq 0, J\in\aleph} D_J + t\Bigg(\sum_{j=1}^{K}\delta_j \sum_{J\neq 0, J\in\aleph} D_J N(J, j)\Bigg)\Bigg)\\
&\quad\times (1+\delta_k t)\Bigg](\mathbf{e} + t\mathbf{v}) + \mathrm{o}(t)\\
&= f_k^*(D_0) + \big(I - f_k^*(D_0)\big)\big(-D_0^{-1}\big)\Bigg(\sum_{J\neq 0, J\in\aleph} D_J\Bigg)\mathbf{e}\\
&\quad + t\Bigg[f_k^*(D_0)\mathbf{v} + \big(I - f_k^*(D_0)\big)\big(-D_0^{-1}\big)\\
&\quad\times\Bigg(\sum_{J\neq 0, J\in\aleph} D_J(\delta_k\mathbf{e} + \mathbf{v}) + \sum_{j=1}^{K}\delta_j \sum_{J\neq 0, J\in\aleph} D_J N(J, j)\mathbf{e}\Bigg)\Bigg] + \mathrm{o}(t)\\
&= \mathbf{e} + t\Bigg[f_k^*(D_0)\mathbf{v} + \big(I - f_k^*(D_0)\big)\big(-D_0^{-1}\big)\\
&\quad\times\Bigg(\sum_{J\neq 0, J\in\aleph} D_J(\delta_k\mathbf{e} + \mathbf{v}) + \sum_{j=1}^{K}\delta_j \sum_{J\neq 0, J\in\aleph} D_J N(J, j)\mathbf{e}\Bigg)\Bigg] + \mathrm{o}(t)\\
&\leqslant (1+\delta_k t)(\mathbf{e} + t\mathbf{v}) - t\varepsilon\mathbf{e}\\
&= \mathbf{e} + t(\delta_k\mathbf{e} + \mathbf{v} - \varepsilon\mathbf{e}) + \mathrm{o}(t),
\end{aligned} \tag{5.5}$$

where we have used $(-D_0)^{-1}(\sum_{J\neq 0} D_J)\mathbf{e} = \mathbf{e}$. Canceling the vector $\mathbf{e}$, dividing by t, letting $t \to 0$ on both sides of the inequality in equation (5.5), yields

$$f_k^*(D_0)\mathbf{v} + \big(I - f_k^*(D_0)\big)\big(-D_0^{-1}\big)\left(\sum_{J\neq 0, J\in\aleph} D_J(\delta_k\mathbf{e} + \mathbf{v}) + \sum_{j=1}^{K}\delta_j \sum_{J\neq 0, J\in\aleph} D_J N(J, j)\mathbf{e}\right)$$
$$\leqslant \delta_k\mathbf{e} + \mathbf{v} - \varepsilon\mathbf{e}. \tag{5.6}$$

After some algebra, equation (5.6) leads to

$$\big(I - f_k^*(D_0)\big)\big(-D_0^{-1}\big)\left(\sum_{j=1}^{K}\delta_j \sum_{J\neq 0, J\in\aleph} D_J N(J, j)\right)\mathbf{e} - f_k^*(D_0)\mathbf{e}\delta_k$$
$$+ \big(I - f_k^*(D_0)\big)\big(-D_0^{-1}\big)D\mathbf{v} + \varepsilon\mathbf{e} \leqslant 0. \tag{5.7}$$

It is easy to see that inequality (5.7) is equivalent to the linear system (5.1). If the linear system (5.1) has a solution $(\boldsymbol{\delta}, \mathbf{v}, \varepsilon)$ with positive δ_k, $1 \leqslant k \leqslant K$, and positive ε, then equation (5.3) holds for small enough positive t and for some positive $\varepsilon' < \varepsilon$. Note that we can choose a small enough positive t to ensure that inequality (5.3) is satisfied, $z_k(t) > 1$ for $1 \leqslant k \leqslant K$, and $\mathbf{u}(t)$ is positive. By lemma 2.2, the Markov chain is ergodic. Consequently, the queueing system is stable. □

Next, we turn our attention to system instability. Define the following linear system for $(\boldsymbol{\delta}, \mathbf{v}, \varepsilon)$:

$$-\sum_{j=1}^{K}\delta_j\mathbf{d}(k, j) + \big(I - f_k^*(D_0)\big)\big(-D_0^{-1}\big)D\mathbf{v} + \varepsilon\mathbf{e} \leqslant 0, \quad 1 \leqslant k \leqslant K, \tag{5.8}$$

with $\boldsymbol{\delta} \geqslant 0$, $-\infty < v_j < \infty$, $1 \leqslant j \leqslant m$, and $\varepsilon \geqslant 0$.

By lemma 2.3, the following sufficient condition is obtained for system instability.

Lemma 5.2. Consider the continuous time MMAP[K]/G[K]/1/LCFS preemptive repeat queue defined in section 3. If the linear system (5.8) has a solution $(\boldsymbol{\delta}, \mathbf{v}, \varepsilon)$ with nonzero nonnegative vector $\boldsymbol{\delta}$ and positive ε, then the Markov chain $\{(q_n, \eta_n),\ n \geqslant 0\}$ is transient and the queueing system is unstable.

Proof. The proof is similar to that of lemma 5.1. Choose $z_k(t) = 1 - \delta_k t$, $1 \leqslant k \leqslant K$, i.e., $\mathbf{z}(t) = \mathbf{e} - \boldsymbol{\delta}t$, and $\mathbf{u}(t) = \mathbf{e} + t\mathbf{v}$. By lemma 2.3, we need to find $\boldsymbol{\delta}$ and $\mathbf{v}$ such that $A^*(k, \mathbf{z}(t))\mathbf{u}(t) \leqslant z_k(t)\mathbf{u}(t)$ for some positive t. As in the proof of lemma 5.1, we expand $A^*(k, \mathbf{z}(t))\mathbf{u}(t) \leqslant z_k(t)\mathbf{u}(t)$ and evaluate the expanded expressions:

$$\left[f_k^*(D_0) + \big(I - f_k^*(D_0)\big)\big(-D_0^{-1}\big)\left(\sum_{J\neq 0, J\in\aleph} D_J - t\left(\sum_{j=1}^{K}\delta_j \sum_{J\neq 0, J\in\aleph} D_J N(J, j)\right)\right)\right.$$
$$\left.\times\ (1 - \delta_k t)\right](\mathbf{e} + t\mathbf{v}) + \mathrm{o}(t)$$

$$
\begin{aligned}
&= \mathbf{e} + t\Bigg[f_k^*(D_0)\mathbf{v} + \big(I - f_k^*(D_0)\big)\big(-D_0^{-1}\big) \\
&\quad \times \Bigg(\sum_{J \neq 0, J \in \aleph} D_J(\mathbf{v} - \delta_k \mathbf{e}) - \sum_{j=1}^{K} \delta_j \sum_{J \neq 0, J \in \aleph} D_J N(J, j)\mathbf{e} \Bigg) \Bigg] + \mathrm{o}(t) \\
&\leqslant (1 - \delta_k t)(\mathbf{e} + t\mathbf{v}) \\
&= \mathbf{e} + t(\mathbf{v} - \delta_k \mathbf{e}) + \mathrm{o}(t).
\end{aligned} \tag{5.9}
$$

Canceling the vector $\mathbf{e}$, dividing by t, and letting $t \to 0$ on both sides of the inequality in equation (5.9), yields

$$
\begin{aligned}
&-\big(I - f_k^*(D_0)\big)\big(-D_0^{-1}\big)\Bigg(\sum_{j=1}^{K} \delta_j \sum_{J \neq 0, J \in \aleph} D_J N(J, j) \Bigg)\mathbf{e} + \big(I - f_k^*(D_0)\big)\big(-D_0^{-1}\big)D\mathbf{v} \\
&\leqslant -\delta_k f_k^*(D_0)\mathbf{e}.
\end{aligned} \tag{5.10}
$$

We add a vector $\varepsilon\mathbf{e}$ to the left-hand side of equation (5.10) to obtain equation (5.8). If the linear system (5.8) has a solution $(\boldsymbol{\delta}, \mathbf{v}, \varepsilon)$ with a nonzero nonnegative vector $\boldsymbol{\delta}$ and positive ε, inequality (5.10) is satisfied in the strict sense. That implies that inequality (5.9) holds for small enough positive t. Thus the conditions given in lemma 2.3 are satisfied for small enough positive t. Therefore, the queueing system is stable. □

To use lemmas 5.1 and 5.2, we need to calculate the matrices $\{f_k^*(D_0),\ 1 \leqslant k \leqslant K\}$ and the vectors $\{\mathbf{d}(k, j),\ 1 \leqslant k,\ j \leqslant K\}$ in order to setup the linear systems (5.1) and (5.8). Then we need to solve the corresponding linear systems. It is easy to see that the idea of applying lemmas 5.1 and 5.2 is to show the existence of the required solutions to the linear systems. For that purpose, we introduce the following linear programs. First, we define a linear program from equation (5.1) to give a sufficient condition for a stable queueing system (LP-SQ):

$$
\begin{aligned}
&\xi_1 = \max\{\varepsilon\} \\
&\text{s.t.} \quad \begin{pmatrix}
\mathbf{d}(1,1) & \dots & \mathbf{d}(1,K) & \big(I - f_1^*(D_0)\big)\big(-D_0^{-1}\big)D & \mathbf{e} \\
\mathbf{d}(2,1) & \dots & \mathbf{d}(2,K) & \big(I - f_2^*(D_0)\big)\big(-D_0^{-1}\big)D & \mathbf{e} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\mathbf{d}(K,1) & \dots & \mathbf{d}(K,K) & \big(I - f_K^*(D_0)\big)\big(-D_0^{-1}\big)D & \mathbf{e} \\
1 & \dots & 1 & 0 \ \dots \ 0 & 0
\end{pmatrix}
\begin{pmatrix} \boldsymbol{\delta} \\ \mathbf{v} \\ \varepsilon \end{pmatrix}
\leqslant
\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \\
&\boldsymbol{\delta} \geqslant 0, \quad \varepsilon \geqslant 0, \quad -\infty < v_j < \infty, \quad 1 \leqslant j \leqslant m.
\end{aligned} \tag{5.11}
$$

Note that the constraints of the above linear program are from linear system (5.1), except that the constraint $\boldsymbol{\delta}\mathbf{e} = 1$ is added to ensure a finite optimal solution (which may not be unique). Next, we define a linear program from equation (5.8) to give a sufficient condition for a unstable queueing system (LP-USQ):

$$\xi_2 = \max\{\varepsilon\}$$

$$\text{s.t.} \quad \begin{pmatrix} -\mathbf{d}(1,1) & \dots & -\mathbf{d}(1,K) & \big(I - f_1^*(D_0)\big)\big(-D_0^{-1}\big)D & \mathbf{e} \\ -\mathbf{d}(2,1) & \dots & -\mathbf{d}(2,K) & \big(I - f_2^*(D_0)\big)\big(-D_0^{-1}\big)D & \mathbf{e} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -\mathbf{d}(K,1) & \dots & -\mathbf{d}(K,K) & \big(I - f_K^*(D_0)\big)\big(-D_0^{-1}\big)D & \mathbf{e} \\ 1 & \dots & 1 & 0 \ \dots \ 0 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta} \\ \mathbf{v} \\ \varepsilon \end{pmatrix} \leqslant \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix},$$

$$\boldsymbol{\delta} \geqslant 0, \quad \varepsilon \geqslant 0, \quad -\infty < v_j < \infty, \quad 1 \leqslant j \leqslant m. \tag{5.12}$$

Now, we are ready to present the main theorem of this paper.

Theorem 5.3. Consider the continuous time MMAP[K]/G[K]/1/LCFS preemptive repeat queue defined in section 3. The linear system (5.1) has a solution with positive $(\boldsymbol{\delta}, \varepsilon)$ if and only if $\xi_1 > 0$, i.e., the objective function of the optimal solution of LP-SQ (5.11) is positive. The linear system (5.8) has a solution with nonzero $\boldsymbol{\delta}$ and positive ε if and only if $\xi_2 > 0$, i.e., the objective function of the optimal solution of LP-SQ (5.11) is positive. Consequently, if $\xi_1 > 0$, the queueing system is stable, and if $\xi_2 > 0$, the queueing system is unstable.

Proof. First, we note that both (5.11) and (5.12) have a feasible solution $(\boldsymbol{\delta}, \mathbf{v}, \varepsilon) = (0, 0, 0)$. Therefore, an optimal solution exists for both linear programs.

If the linear system (5.1) has a solution with positive $(\boldsymbol{\delta}, \varepsilon)$, then we have $\xi_1 > \varepsilon > 0$. If the objective function of the optimal solution of (5.11) is positive, then the linear system (5.1) has a solution $(\boldsymbol{\delta}, \mathbf{v}, \varepsilon)$ with positive ε. Next, we show that $\boldsymbol{\delta}$ is also positive. If $\boldsymbol{\delta} = 0$, then the constraints in equation (5.11) (except the last line) become $(I - f_k^*(D_0))(-D_0)^{-1}D\mathbf{v} + \varepsilon\mathbf{e} \leqslant 0$, $1 \leqslant k \leqslant K$. Multiplying both sides of these inequalities by the nonnegative and nonzero vector $\boldsymbol{\theta}(-D_0)(I - f_k^*(D_0))^{-1}$, yields $\varepsilon\boldsymbol{\theta}(-D_0)(I - f_k^*(D_0))^{-1}\mathbf{e} \leqslant 0$, $1 \leqslant k \leqslant K$, which is a contradiction. (Note that $\boldsymbol{\theta}D = 0$ and $\boldsymbol{\theta}(-D_0) \geqslant 0$ and $\boldsymbol{\theta}(-D_0) \neq 0$.) Therefore, $\boldsymbol{\delta}$ is nonzero. If the vector $\boldsymbol{\delta}$ is not positive, $\exists k$ such that $\delta_k = 0$. Then we multiply $\boldsymbol{\theta}(I - f_k^*(D_0))^{-1}(-D_0)$ on both sides of the kth inequality in (5.11) to obtain

$$\sum_{j=1, j\neq k}^{K} \delta_j \boldsymbol{\theta}(-D_0)\big(I - f_k^*(D_0)\big)^{-1}\mathbf{d}(k,j) + \varepsilon\boldsymbol{\theta}(-D_0)\big(I - f_k^*(D_0)\big)^{-1}\mathbf{e} \leqslant 0. \tag{5.13}$$

Since all of the components of the above vectors are nonnegative and some of them are positive ($\varepsilon > 0$), the inequality cannot hold. Therefore, the vector $\boldsymbol{\delta}$ is positive. By lemma 5.1, the queueing system is stable.

The second part of the theorem about equation (5.12) can be proved similarly. However, the vector $\boldsymbol{\delta}$ only has to be nonzero in this case. □

We note that, if $m = 1$, information provided by the solutions of (5.11) and (5.12) for system stability is sufficient and necessary. In fact, we have the following interesting results.

Corollary 5.4. Consider the continuous time MMAP$[K]/G[K]/1$/LCFS preemptive repeat queue defined in section 3. Assume $m = 1$. Then the queueing system is stable if and only if LP-SQ (5.11) has a positive optimal objective value. The queueing system is unstable if and only if LP-USQ (5.12) has a positive optimal objective value. Consequently, if neither LP-SQ nor LP-USQ has a positive solution, then the Markov chain of interest is null recurrent.

Proof. First, we prove the conclusion for the stable case. If $m = 1$, it is easy to see that $D = 0$. Thus, the vector $\mathbf{v}$ is not involved. Then the constraints of (5.11) are reduced to $P(\mathbf{X})\boldsymbol{\delta} - \boldsymbol{\delta} + \varepsilon\mathbf{e} \leqslant 0$ and $\boldsymbol{\delta}\mathbf{e} \leqslant 1$. If (5.11) has a positive optimal solution $(\boldsymbol{\delta}, \varepsilon)$, then $P(\mathbf{X})\boldsymbol{\delta} < \boldsymbol{\delta}$, which implies that $sp(P(\mathbf{X})) < 1$. Thus, the queueing system is stable. On the other hand, if the queueing system is stable, then we must have $sp(P(\mathbf{X})) < 1$, which implies that $P(\mathbf{X})\boldsymbol{\delta} - \boldsymbol{\delta} + \varepsilon\mathbf{e} \leqslant 0$ holds for some positive $(\boldsymbol{\delta}, \varepsilon)$. In fact, we can choose

$$\delta_k = \frac{1}{f_k^*(D_0)}, \quad 1 \leqslant k \leqslant K. \tag{5.14}$$

Therefore, equation (5.11) has a positive solution $(\boldsymbol{\delta}, \varepsilon)$ and $\xi_1 > 0$.

The unstable queue case can be proved similarly. This completes the proof of corollary 5.4. □

Unfortunately, it is not easy to check how accurate the information provided by equations (5.11) and (5.12) is if $m > 1$. Nonetheless, the result for the case $m = 1$ implies that (5.11) and (5.12) may provide useful information for system stability when $m > 1$. In section 7, a numerical analysis on this issue shall be carried out to analyze the usefulness of theorem 5.3.

The sufficient conditions given by theorem 5.3 are closely related to that of $sp(P(\mathbf{X}))$. To see the connection, let $\mathbf{v} = 0$. Then the constraints of equation (5.11) become $P(\mathbf{X})\boldsymbol{\Delta} - \boldsymbol{\Delta} + \varepsilon\mathbf{e} \leqslant 0$, where $\boldsymbol{\Delta} = (\delta_1\mathbf{e}^{\mathrm{T}}, \delta_2\mathbf{e}^{\mathrm{T}}, \ldots, \delta_K\mathbf{e}^{\mathrm{T}})^{\mathrm{T}}$. Thus, finding a solution to (5.11) is equivalent to finding a special type of subinvariant measure of $P(\mathbf{X})$. However, $P(\mathbf{X})$ may not have such a subinvariant vector. Thus, theorem 5.3 may fail to provide accurate information about system stability for some cases. Comparing the four methods, we observe that the PFE and FPP methods take *the change of the phase of the underlying Markov process* into consideration, while PTI and LS do not. Thus PTI and LS only provide partial information for the classification of the Markov chain of interest.

Compared to the methods introduced in section 4, the LS approach has a larger matrix to calculate (the constraints of equations (5.11) and (5.12)). In fact, the space complexity of the PFE method is $\mathrm{O}(Km^2)$ and the space complexity of the LS method is $\mathrm{O}(K + m + 1)(Km + 1) = \mathrm{O}(K^2m + Km^2)$. If m is much larger than K, then the space complexity of the two methods is more or less the same. On the other hand, the matrix iterations for matrices $\mathbf{X} = \{X_1, \ldots, X_K\}$ that are necessary for the PFE and FPP methods are avoided so that time complexity can be reduced and numerical precision can be ensured. Furthermore, there are well developed algorithms and software for solving linear programming problems. Therefore, the LS approach does have advantages over other methods.

A theoretical issue related to LP-SQ (5.11) and LP-USQ (5.12) is whether or not they are consistent, i.e., they do not provide conflicting information about system stability. That question is answered by the following property.

Property 5.5. Consider LP-SQ (5.11) and LP-USQ (5.12). If $\xi_1 > 0$, then $\xi_2 = 0$ and the queueing system is stable. On the other hand, if $\xi_2 > 0$, then $\xi_1 = 0$ and the queueing system is unstable. It is possible that $\xi_1 = \xi_2 = 0$. In this case, LP-SQ (5.11) and LP-USQ (5.12) provide no information about the stability of the queueing system.

Proof. We only have to show that it is impossible to have $\xi_1 > 0$ and $\xi_2 > 0$. If $\xi_1 > 0$ and $\xi_2 > 0$, then the optimal solutions $(\boldsymbol{\delta}_1, \mathbf{v}_1, \xi_1)$ and $(\boldsymbol{\delta}_2, \mathbf{v}_2, \xi_2)$ of (5.11) and (5.12), respectively, must have positive $\boldsymbol{\delta}_1$, ξ_1, and ξ_2 (see the proof of theorem 5.3). Furthermore, $\boldsymbol{\delta}_2$ must be nonnegative and nonzero. Thus we can choose a positive t such that $\boldsymbol{\delta}_1 - t\boldsymbol{\delta}_2$ is a positive vector, since every component of $\boldsymbol{\delta}_2$ is finite. Then it is easy to verify that $(\boldsymbol{\delta}_1 - t\boldsymbol{\delta}_2, \mathbf{v}_1 + t\mathbf{v}_2, \xi_1 + t\xi_2)$ is a feasible solution to (5.11). But $\xi_1 + t\xi_2 > \xi_1$, which contradicts the optimality of ξ_1. □

To end this section, we propose a computational scheme to check whether or not the queueing system defined in section 3 is stable or unstable.

Step 1. Calculate $\{f_k^*(D_0),\ 1 \leqslant k \leqslant K\}$.

Step 2. Calculate $\{\mathbf{d}(k, j),\ 1 \leqslant k, j \leqslant K\}$ by equation (5.2).

Step 3. Solve LP-SQ (5.11) and LP-USQ (5.12).

Step 4. If neither (5.11) nor (5.12) provides information about system stability, use the PFE method given in section 4.1.

6. Computational details and PH-distributed service times

In sections 4 and 5, we introduced four methods for checking system stability. For all four methods, computation is straightforward except for the matrices $\{f_k^*(D_0),\ 1 \leqslant k \leqslant K\}$. In this section, we address that issue when the service times have either PH-distributions or mixed PH-distributions.

By definition (3.3), the computation of the matrices $\{f_k^*(D_0), 1 \leqslant k \leqslant K\}$ can be done using standard methods of integration. Unfortunately, these general methods are not always efficient and may cause significant errors in computation. To overcome such a problem, at least partially, we consider a special class of distributions, PH-distributions, for the service times. We derive an explicit formula for computing the matrices $\{f_k^*(D_0),$

$1 \leqslant k \leqslant K\}$. Note that it is well known that PH-distributions can be used to approximate any probability distribution. Therefore, the assumption about the service times is not too restrictive.

Let $(\boldsymbol{\alpha}_k, T_k, T_k^0)$ be the matrix representation of the PH-distribution for the service times of type k customers, for $1 \leqslant k \leqslant K$, where $\boldsymbol{\alpha}_k$ is a nonnegative row vector of the size m_k with $\boldsymbol{\alpha}_k\mathbf{e} = 1$, T_k is an $m_k \times m_k$ matrix with negative diagonal elements and nonnegative off-diagonal elements, and T_k^0 is a nonnegative column vector of the size m_k given by, $T_k^0 = -T_k\mathbf{e}$. We refer to [16] for more details about PH-distributions. It is well known that $F_k(x) = 1 - \boldsymbol{\alpha}_k \exp\{T_k x\}\mathbf{e}$ and $f_k^*(s) = \boldsymbol{\alpha}_k(sI - T_k)^{-1}T_k^0$.

Lemma 6.1. If $F_k(x)$ has a PH-distribution with matrix representation $(\boldsymbol{\alpha}_k, T_k, T_k^0)$, then $f_k^*(D_0)$ defined by equation (3.3) is given explicitly as, for $1 \leqslant k \leqslant K$,

$$f_k^*(D_0) = (\boldsymbol{\alpha}_k \otimes I_m)\phi^{-1}\big(\phi(-I_{m_k m})\big(T_k^{\mathrm{T}} \otimes I_{m_k m^2} + I_{m_k^2 m} \otimes D_0\big)^{-1}\big)\big(T_k^0 \otimes I_m\big), \quad (6.1)$$

where $\otimes$ is the *Kronecker product* of matrices, $\phi(\cdot)$ is the *direct sum* of matrices, and $\phi^{-1}(\cdot)$ is the inverse transform of $\phi(\cdot)$ (the definition of $\phi(\cdot)$ is given right after equation (6.4)).

Proof. First, note that for any four matrices X, Y, Z, and W, we have $(XY) \otimes (ZW) = (X \otimes Z)(Y \otimes W)$, provided that all multiplications are well-defined (see [6] for more details about the Kronecker product). Then

$$\begin{aligned} f_k^*(D_0) &= \int_0^\infty F_k(\mathrm{d}x)\exp\{xD_0\} = \int_0^\infty \boldsymbol{\alpha}_k \exp\{T_k x\}T_k^0\,\mathrm{d}x\,\exp\{xD_0\} \\ &= \int_0^\infty \big(\boldsymbol{\alpha}_k \exp\{T_k x\}T_k^0\big) \otimes \big(\exp\{xD_0\}\big)\,\mathrm{d}x \\ &= \int_0^\infty (\boldsymbol{\alpha}_k \otimes I_m)\big(\exp\{T_k x\} \otimes \exp\{xD_0\}\big)\big(T_k^0 \otimes I_m\big)\,\mathrm{d}x \\ &= (\boldsymbol{\alpha}_k \otimes I_m)\int_0^\infty \big(\exp\{T_k x\} \otimes \exp\{xD_0\}\big)\,\mathrm{d}x\big(T_k^0 \otimes I_m\big). \end{aligned} \quad (6.2)$$

Note that the subscript of I represents the size of that matrix. Using the definition of the exponential functions $\exp\{T_k x\}$ and $\exp\{D_0 x\}$, we have

$$\begin{aligned} \exp\{T_k x\} \otimes \exp\{xD_0\} &= \sum_{n,w\geqslant 0} \frac{x^{n+w}}{n!w!} T_k^n \otimes D_0^w = \sum_{n,w\geqslant 0} \frac{x^{n+w}}{n!w!}\big(T_k^n I_{m_k}^w\big) \otimes \big(I_m^n D_0^w\big) \\ &= \sum_{n,w\geqslant 0} \frac{x^{n+w}}{n!w!}(T_k \otimes I_m)^n (I_{m_k} \otimes D_0)^w \\ &= \exp\big\{(T_k \otimes I_m)x\big\}\exp\big\{(I_{m_k} \otimes D_0)x\big\}. \end{aligned} \quad (6.3)$$

Partial integration leads to

$$
\begin{aligned}
L_k &\equiv \int_0^\infty \exp\{(T_k \otimes I_m)x\} \exp\{(I_{m_k} \otimes D_0)x\}\,\mathrm{d}x \\
&= -(I_{m_k} \otimes D_0)^{-1} \\
&\quad - (T_k \otimes I_m) \int_0^\infty \exp\{(T_k \otimes I_m)x\} \exp\{(I_{m_k} \otimes D_0)x\}\,\mathrm{d}x (I_{m_k} \otimes D_0)^{-1} \\
&= -(I_{m_k} \otimes D_0)^{-1} - (T_k \otimes I_m) L_k (I_{m_k} \otimes D_0)^{-1}. \qquad (6.4)
\end{aligned}
$$

By routine algebra, (6.4) yields $(T_k \otimes I_m)L_k + L_k(I_{m_k} \otimes D_0) = -I_{m_k m}$. Denote by $\phi(X)$ the *direct-sum* of the matrix X, i.e., put all rows of X from top to bottom into a single row vector from left to right. Note that for any three matrices X, Y, and Z, $\phi(XYZ) = \phi(Y)(X^{\mathrm{T}} \otimes Z)$, providing that the matrix multiplications are well-defined (see [16] for more about the direct sum). Then we have

$$
\phi(L_k)\big[(T_k \otimes I_m)^{\mathrm{T}} \otimes I_{m_k m}\big] + \phi(L_k)[I_{m_k m} \otimes I_{m_k} \otimes D_0] = \phi(-I_{m_k m}), \qquad (6.5)
$$

which leads to

$$
\phi(L_k) = \phi(-I_{m_k m})\big((T_k \otimes I_m)^{\mathrm{T}} \otimes I_{m_k m} + I_{m_k m} \otimes I_{m_k} \otimes D_0\big)^{-1}. \qquad (6.6)
$$

The inverse matrix in equation (6.6) exists since T_k and D_0 are invertible.

Finally, combining equations (6.2) and (6.6) yields (6.1). This completes the proof of lemma 6.1. □

Once the matrices $\{f_k^*(D_0),\ 1 \leqslant k \leqslant K\}$ are obtained, we are ready to apply any of the four methods for system stability. In fact, applying the PTI and LS methods is now straightforward, though the latter requires an algorithm to solve linear programming problems. As for the PFE and FPP methods, we have all the basic matrices needed for further calculations.

To end this section, we briefly consider the mixed PH-distribution case. We assume that the service times have the following *mixture of PH-distributions*, $1 \leqslant k \leqslant K$,

$$
F_k(x) = \sum_{n=1}^{S_k} p_{k,n}\big[1 - \boldsymbol{\alpha}_k(n) \exp\{T_k(n)x\}\mathbf{e}\big], \qquad (6.7)
$$

where $\sum_{n=1}^{S_k} p_{k,n} = 1$, $S_k \geqslant 1$, and the dimension of $\boldsymbol{\alpha}_k(n)$ and $T_k(n)$ is $m_{k,n}$. Using lemma 6.1, it can be proved that

$$
\begin{aligned}
f_k^*(D_0) = \sum_{n=1}^{S_k} p_{k,n} \big(\boldsymbol{\alpha}_k(n) \otimes I_m\big)\phi^{-1}\big(\phi(-I_{m_{k,n}m})\big(T_k(n)^{\mathrm{T}} \otimes I_{m_{k,n}m^2} \\
+ I_{m_{k,n}^2 m} \otimes D_0\big)^{-1}\big)\big(T_k^0(n) \otimes I_m\big), \qquad (6.8)
\end{aligned}
$$

where $T_k^0(n) = -T_k(n)\mathbf{e}$. Details are omitted. The mixed PH-distribution is interesting since it includes the mixed Erlang distributions and many other commonly used distributions as its special cases.

7. Numerical examples

In this section, we compare the four approaches for system stability and show, numerically, the usefulness of the sufficient conditions given in section 5. We shall mainly compare the following methods:

1. *Perron–Frobenius Eigenvalue* (PFE). Use theorem 2.1 with the matrix $P(\mathbf{X})$, where $\mathbf{X}$ is the stochastic fixed point of equation (4.1).
2. *First Passage Probability* (FPP). Use the matrices in the minimal solution $\mathbf{G}$.
3. *Pseudo-Traffic Intensity* (PTI). Use the function ρ_2 defined in equation (4.5).
4. *Linear System* (LS). Use LP-SQ (5.11) and LP-USQ (5.12).

While information obtained from PFE and FPP accurately predicts the stability of the system, information provided by PTI and LS does not. Since the fixed points of equation (4.1) are not involved, the PTI and LS methods are computationally much more efficient than the PFE and FPP methods, especially when m is large. Thus, each of the four methods has some advantages and some disadvantages. Since PFE and FPP are consistent, we shall mainly compare the PFE, PTI, and LS methods.

First, as we promised, we show in example 7.1 that the PFE, PTI, and LS methods may provide different information about system stability. The implication is that the PTI and LS methods are inaccurate. Nonetheless, in example 7.3, we shall show how good the PTI and LS methods are by summarizing the results from a large number of randomly chosen examples.

Example 7.1. First, we consider a queueing system defined in section 3 with two types of customers. System parameters are given as follows: $K = m = 2$,

$$\begin{aligned}
&D_0 = \begin{pmatrix} -10 & 0 \\ 0 & -1 \end{pmatrix}, && D_1 = \begin{pmatrix} 4.5 & 0.5 \\ 0 & 0 \end{pmatrix}, && D_2 = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}, \\
&D_{12} = \begin{pmatrix} 2 & 0 \\ 1 & 0 \end{pmatrix}, && D_{22} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}; && \\
&m_1 = 2, && \boldsymbol{\alpha}_1 = (0.2, 0.8), && T_1 = \begin{pmatrix} -0.4 & 0 \\ 2 & -24.2 \end{pmatrix}; \\
&m_2 = 2, && \boldsymbol{\alpha}_2 = (0.3, 0.7), && T_2 = \begin{pmatrix} -20 & 0.5 \\ 1 & -11 \end{pmatrix}.
\end{aligned} \tag{7.1}$$

All other matrices are zero. For this queueing system, we have both single arrivals and batch arrivals. The average arrival rates are $\lambda_1 = 6.66667$ and $\lambda_2 = 2.6667$. The average service rates are $\mu_1 = 1.4319$ and $\mu_2 = 12.0936$. Then it is easy to obtain $\rho = 4.8761$. The matrices $\{f_k^*(D_0),\ 1 \leqslant k \leqslant 2\}$ are obtained as follows:

$$f_1^*(D_0) = \begin{pmatrix} 0.5288 & 0.0 \\ 0.0 & 0.7800 \end{pmatrix}, \qquad f_2^*(D_0) = \begin{pmatrix} 0.5528 & 0.0 \\ 0.0 & 0.9238 \end{pmatrix}. \tag{7.2}$$

We have $\rho_2 = 0.9865$. After calculating $\{G_1, G_2\}$ and $\{X_1, X_2\}$, we find that $\{G_1, G_2\}$ are clearly nonstochastic and $sp(P(\mathbf{X})) = 1.0107$. Thus, the queueing system is unstable and the PTI method does not provide accurate information about it.

It is interesting to point out that $\rho = 4.8761$ is far larger than $sp(P(\mathbf{X}))$ because the service times have the *decreasing failure rate* property. Thus, it is likely that new customer may have a shorter remaining service time than an old one. This example shows that, unlike the work conserving queues, the stability of non work conserving queues is not intuitive. Example 7.2 shows that even when $\rho > 4$, the LCFS queue introduced in section 3 can be stable. This fact is useful in the design of queueing systems.

Now, we turn to the LS methods. For this example, both ξ_1 and ξ_2 are zero. Therefore, we cannot apply theorem 5.3 to find whether or not the system is stable. In fact, PFE and FPP are the only methods that work for this example. Why do both PTI and LS methods fail to provide useful information for this case? It seems that the queueing system is close to the border line of being stable or unstable as suggested by the value of $sp(P(\mathbf{X}))$. The following example makes this point clearer.

Example 7.2. Consider the same queueing system as in example 7.1 except that the service times of type 1 customers have the following PH-distribution:

$$m_1 = 2, \qquad \boldsymbol{\alpha}_1 = (0.2, 0.8), \qquad T_1 = \begin{pmatrix} -0.4 & 0 \\ 2 & -\mu \end{pmatrix}, \tag{7.3}$$

where $\mu \geqslant 2$. Letting μ go from 15 to 35, we plot ρ_2, $sp(P(\mathbf{X}))$, ξ_1, and ξ_2 for the corresponding queueing systems in figure 1. We point out that, in figure 1, $sp(P(\mathbf{X}))$ is the only measure that always provides correct information about system stability.

Note that in figure 1, the horizontal line crossing the vertical axis at 1 is introduced to see the difference between the PFE and PTI methods clearly.

As shown in figure 1, when $sp(P(\mathbf{X}))$ is close to one around $\mu = 24$ ($20 < \mu < 27$), ρ_2, ξ_1 and ξ_2 does not provide correct information for system stability. If $\mu < 23$ or $\mu > 26$, ρ_2 provides correct information for system stability. When $20 < \mu < 27$, ξ_1 and ξ_2 do not provide any information for system stability since they are both zero. If $\mu < 23$, ξ_2 provides information for system instability. If $\mu > 26$, ξ_1 provides information for system stability. This and many other examples show that ρ_2 is more accurate than ξ_1 and ξ_2 with respect to predicting system stability. However, ρ_2 is an approximation and information from ρ_2 is unreliable. On the other hand, when ξ_1 and ξ_2 are not both zero, the information about system stability is correct.

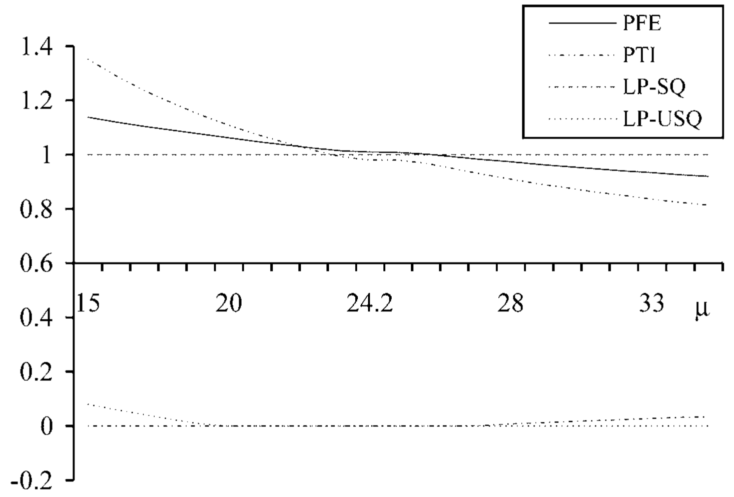

Figure 1. Plot of $sp(P(\mathbf{X}))$, ρ_2, ξ_1, and ξ_2 for example 7.2.

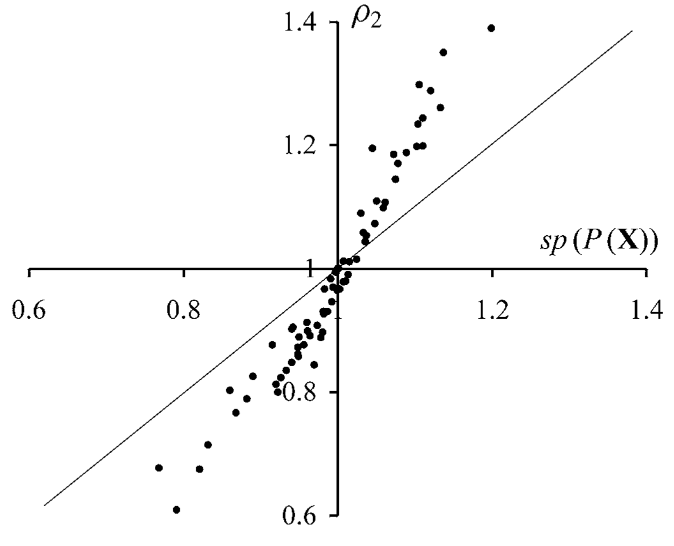

Figure 2. Plot of $sp(P(\mathbf{X}))$ and ρ_2.

Example 7.3. In this example, we plot $sp(P(\mathbf{X}))$ against ρ_2, ξ_1 and ξ_2 for a large number of randomly chosen examples. In figure 2, we plot $(sp(P(\mathbf{X})), \rho_2)$. In figure 3, we plot $(sp(P(\mathbf{X})), \xi_1)$ and $(sp(P(\mathbf{X})), \xi_2)$.

Figure 2 shows that ρ_2 and $sp(P(\mathbf{X}))$ are consistent for most of the examples. But there are a significant number of exceptions around $sp(P(\mathbf{X})) = 1$. Thus, if the queueing system is on the border of being stable or unstable, ρ_2 cannot be used to predict stability of the queueing system. From property 4.1, if $m = 1$, we expect to see that $sp(P(\mathbf{X})) > \rho_2$ if $sp(P(\mathbf{X})) < 1$ and $sp(P(\mathbf{X})) < \rho_2$ if $sp(P(\mathbf{X})) > 1$. Figure 1

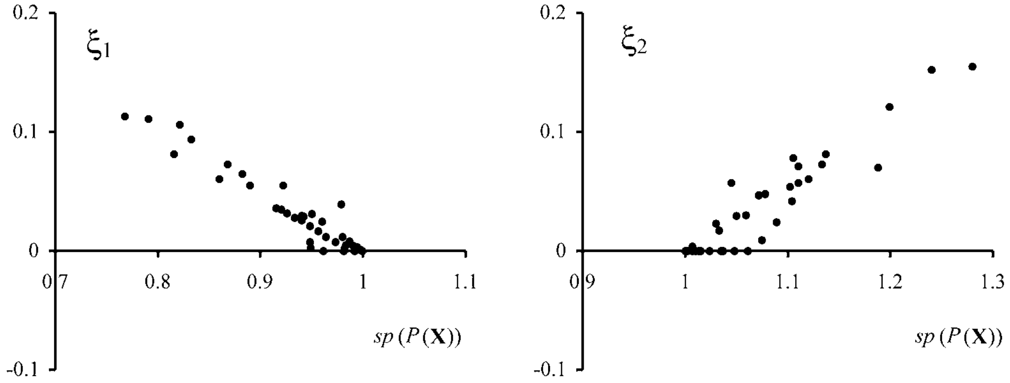

Figure 3. Plots of $(sp(P(\mathbf{X})), \xi_1)$ and $(sp(P(\mathbf{X})), \xi_2)$.

shows that that relationship holds for most of the examples. Again, if $sp(P(\mathbf{X})) \approx 1$, a significant number of exceptions occur. In conclusion, ρ_2 can be used for system stability if its value is not close to 1.

By theorem 5.3, ξ_1 is completely useless when the queue is unstable and ξ_2 is completely useless when the queue is stable. Therefore, we plot $(sp(P(\mathbf{X})), \xi_1)$ only for $sp(P(\mathbf{X})) < 1$ and $(sp(P(\mathbf{X})), \xi_2)$ only for $sp(P(\mathbf{X})) > 1$. Figure 3 shows that both ξ_1 and ξ_2 fail to provide information about queueing stability when $sp(P(\mathbf{X})) \approx 1$ $(0.95 < sp(P(\mathbf{X})) < 1.05)$. Nonetheless, figure 3 does show that ξ_1 starts to provide information when $sp(P(\mathbf{X}))$ goes below 0.95 and ξ_1 starts to provide information when $sp(P(\mathbf{X}))$ goes above 1.05. Figure 3 also shows that ξ_1 and ξ_2 may provide useful information even when $sp(P(\mathbf{X}))$ is close to one. It is reasonable to conclude that if both LP-SQ and LP-USQ do not provide information for system stability, i.e., $\xi_1 = \xi_2 = 0$, then the queueing system is on the border between being stable and unstable.

In conclusion, the PTI approach provides accurate information about system stability for many of the queueing systems of interest. But it can fail for others, especially when ρ_2 is close to one. The LS methods do provide correct information about system stability but may not work when the queueing system is on the border between being stable or unstable. It is then reasonable to believe that the information provided by the PTI and LS methods can be used to classify our queueing system as stable, not sure but close to unstable, not sure but close to stable, and unstable. In summary, the following scheme can be used for system stability, without using the PFE and FPP methods.

- If $\xi_1 > 0$, the queueing system is stable.
- If $\xi_2 > 0$, the queueing system is unstable.
- If $\xi_1 = \xi_2 = 0$ and ρ_2 is close to one, the queueing system is on the border of being stable or unstable.
- Otherwise, it is unsure.

Acknowledgements

The authors would like to thank Dr. Susan Seager for proofreading the manuscript. This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through operating grants.

References

[1] S. Asmussen and G. Koole, Marked point processes as limits of Markovian arrival streams, J. Appl. Probab. 30 (1993) 365–372.
[2] H. Chen and H. Zhang, Stabiliy of multiclass queueing networks under priority service disciplines, Oper. Res. 48 (2000) 26–37.
[3] J.W. Cohen, *The Single Server Queue* (North-Holland, Amsterdam, 1982).
[4] J. Dai and S. Meyn, Stability and convergence of moments for multiclass queueing networks via fluid limit models, IEEE Trans. Automat. Control 40 (1995) 1889–1904.
[5] G. Fayolle, V.A Malyshev and M.V. Menshikov, *Topics in the Constructive Theory of Countable Markov Chains* (Cambridge Univ. Press, Cambridge, 1995).
[6] F.R. Gantmacher, *The Theory of Matrices* (Chelsea, New York, 1959).
[7] K. Goebel and W.A. Kirk, *Topics in Metric Fixed Point Theory* (Cambridge Univ. Press, Cambridge, 1990).
[8] Q.-M. He, Classification of Markov processes of $M/G/1$ type with a tree structure and its applications to queueing systems, Oper. Res. Lett. 26 (2000) 67–80.
[9] Q.-M. He, Classification of Markov processes of matrix $M/G/1$ type with a tree structure and its applications to the MMAP$[K]/PH[K]/1$ queue, Stochastic Models 16 (2000) 407–433.
[10] Q.-M. He, A fixed point approach to the classification of Markov chains with a tree stucture, Stochastic Models 19(1) (2003) 76–114.
[11] Q.-M. He and A.S. Alfa, The discrete time MMAP$[K]/PH[K]/1$/LCFS-GPR queues and its variants, in: *Advances in Algorithmic Methods for Stochastic Models – Proc. of the 3rd Internat. Conf. on Matrix Analytic Methods*, eds. G. Latouche and P.G. Taylor (Notable Publications, New Jersey, 2000) pp. 167–190.
[12] Q.-M. He and M.F. Neuts, Markov arrival processes with marked transitions, Stochastic Process. Appl. 74(1) (1998) 37–52.
[13] P.R. Kumar and S.P. Meyn, Duality and linear programs for stability and performance analysis of queueing networks and scheduling policies, IEEE Trans. Automat. Control 41(1) (1996) 4–17.
[14] R. Loynes, The stability of a queue with non-independent inter-arrival and service times, Proc. Cambridge Philos. Soc. 58 (1962) 497–520.
[15] S.P. Meyn and R. Tweedie, *Markov Chains and Stochastic Stability* (Springer, Berlin, 1993).
[16] M.F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach* (Johns Hopkins Univ. Press, Baltimore, MD, 1981).
[17] M.F. Neuts, *Structured Stochastic Matrices of $M/G/1$ Type and Their Applications* (Marcel Dekker, New York, 1989).
[18] T. Takine, B. Sengupta and R.W. Yeung, A generalization of the matrix $M/G/1$ paradigm for Markov chains with a tree structure, Stochastic Models 11 (1995) 411–421.
[19] B. Van Houdt and C. Blondia, Stability and performance of stack algorithms for random access communication modeled as a tree structured QBD Markov chain, Stochastic Models 17(3) (2001) 1–28.
[20] R.W. Yeung and B. Sengupta, Matrix product-form solutions for Markov chains with a tree structure, Adv. in Appl. Probab. 26(4) (1994) 965–987.