RAIRO-Oper. Res. 53 (2019) 767–786
https://doi.org/10.1051/ro/2017059

RAIRO Operations Research
www.rairo-ro.org

NUMERICAL SOLUTION FOR THE PERFORMANCE CHARACTERISTICS OF THE M/M/C/K RETRIAL QUEUE WITH NEGATIVE CUSTOMERS AND EXPONENTIAL ABANDONMENTS BY USING VALUE EXTRAPOLATION METHOD

Zidani Nesrine[1,*], Spiteri Pierre[2] and Djellab Natalia[1]

Abstract. This paper deals with a retrial queueing system M/M/C/K with exponential abandonment at which positive and negative primary customers arrive according to Poisson processes. This model is of practical interest: it permits to analyze the performance in call centers or multiprocessor computer systems. For model under study, we find the ergodicity condition and also the approximate solution by applying Value Extrapolation method which includes solving of some algebraic system of equations. To this end, we have resolved the algebraic system in question by different numerical methods. We present also numerical results to analyze the system performance.

Mathematics Subject Classification. 60K25, 90B22, 68M20.

Received October 22, 2016. Accepted July 23, 2017.

1. Introduction: Model description

Retrial queueing systems or systems with repeated attempts are characterized by the requirement that customers finding the service area busy must join the retrial group (orbit) and retry for service at random intervals. A review of the main results on this topic can be found in [5, 15, 16]. The modeling of repeated attempts has been also the subject matter of many researches dealing with influence of the retrial phenomenon in the quality of service of real communication systems such as: call centers, cellular mobile networks, Internet, radio cognitive networks, ... [2, 9, 11, 34]. Thus, there is a practical need for the analysis of M/M/C retrial queues (in particular those including some specific phenomena: waiting area, abandonment, negative arrivals).

The most important characteristic of M/M/C retrial queues is the spatial heterogeneity of underlying Markov chains and infinite state spaces, caused by the retrial feature (in general, the retrials are governed by the classical retrial policy where the total retrial rate depends on the number of customers in the orbit). Thereby, the exact results have not been obtained except for some special cases [20] and many attempts have been made to develop a number of efficient algorithmic methods [1, 4, 10]. However, many researchers resort to numerically tractable approximations, which are often based on finite truncated models. This approximation replaces the original

Keywords. Multiserver retrial queue, abandonment, negative customer, ergodicity condition, value extrapolation, algebraic linear system of equations, invertible matrix, numerical method.

[1] Department of Mathematics University of Annaba, BP.12, Annaba, 23000 Algeria.

[2] IRIT, site ENSEEIHT, INP Toulouse, BP. 7122, 31071 Toulouse, France.

*Corresponding author: zidaninesrine@hotmail.com

infinite state space by a finite one. Stepanov (1999) presented an efficient algorithm of optimal calculation of the steady state probabilities of the system states. It is based on the truncation of the state space at a sufficiently large level L related to the number of customers in the orbit. This truncation level was obtained after preliminary estimation and ignoring the states with negligible probabilities. However, at high levels of congestion, this direct truncation method is computationally very demanding because the truncation level L becomes very large. To overcome this drawback, several approximate models, generalized truncated models, are proposed in the literature. These approximations replace the initial infinite state space by another infinite but solvable state space. In [29], the main M/M/C retrial queue was approximated by a multiserver retrial queue with constant total retrial rate (that does not depend on the number of customers in the orbit) as long as the number of orbiting customers is greater than a specified value. With this assumption, the steady state probabilities of the system states were calculated by a matrix-geometric method. For computing the same probabilities, a numerically stable schema was also developed in [8]. Domenech−Benlloch *et al.* [13] proposed some generalized truncated methods (inspired in the model given in [29]) that are able to effectively solve multiserver retrial queues. The performances of the finite truncated models and the generalized truncated models were compared in [5, 7], and the convergence problem of truncated models was discussed in [14].

An interesting approximation, based on the homogenization of the state space of M/M/C retrial queues, permits us to analyze a retrial queue as a quasi-birth-death process, the steady state probabilities of which can be obtained by using the matrix-geometric method [6, 23, 40], or by a finite truncation method [33]. In [24], this approach provided a solution to the steady state system state distribution of the M/M/C/N+C feedback queue with constant retrial rate. But it presented the state explosion problem when the queueing capacity was large. This problem was resolved by using the spectral expansion method based on the eigenvalues and eigenvecteurs of the characteristic matrix polynomial associated with balance equation [10]. In [12], the approach using quasi-birth-death process, permitted to provide the exact expression of the conditional mean number of customers in the orbit. Recent contributions on this topic include also the papers of Phung−Duc [31, 32].

The above mentioned methods give the numerical solution for the steady state distribution of the Continuous Time Markov Chains. In [26], an alternative approach for calculating the performance measures of infinite state space Markov processes (Value Extrapolation) has been introduced. This type of approximation considers the queueing system in its Markov Decision Process setting and solves the expected value (representing a performance measure) from the Howard equations written for a truncated space. However, instead of a simple truncation, the relative values of states just outside the truncated state space are estimated by polynomial extrapolation based on the states inside. This leads to a closed system of equations and permits us to obtain accurate results with small truncated spaces. In [19], the performance of this approximation (in terms of accuracy and computation time) was compared with that of the approximations based on finite truncated and generalized truncated models. As a result, the Value Extrapolation method is recommended for solving multiserver retrial queues.

This paper deals with a retrial queueing system M/M/C/K with exponential abandonments and negative arrivals. Queueing models with negative arrivals, G-queues, were first introduced by Gelenbe (1989) [17] to model the neural networks. In its simplest version, a negative arrival has the effect of deleting a positive (ordinary) customer from the system according to some strategy, for example: arrival of a negative customer eliminates all the customers in the system, or removes the customer from the head of queue (including the one in service), or deletes the customer at the end of queue. For a detailed study in queueing models with negative arrivals, see [3, 18]. The negative customers can be interpreted as viruses or inhibit signals. They can also represent some additional behaviors in communication systems such as breakdowns, killing signals or call losses. For example [39]:

-: In computer networks, if a virus enters a node, one or more files may be infected and the system manager may have to do through a number of backups to recover the infected files.
-: In multiple resource system with requests for service (positive customers), the decisions to cancel some requests represent the negative customers.

-: In local area networks, the communication between the two users is realized by the channel which may be breakdown during the transmission due to a hackers attack or some viruses. Therefore, the packet in transmission will be lost. A breakdown at the server can be represented by the arrival of a negative customer which causes the customer being in service to be lost when the server is busy. There is the case when the negative arrival deletes only the customers in service.

In this work, we consider a queueing system with C servers and $K-C$ waiting positions at which positive and negative primary customers arrive according to Poisson processes with rates $\lambda > 0$ and $\nu > 0$, respectively. An arriving positive customer receives immediate service if there is an idle server, otherwise we have two possibilities depending on whether the number of customers in the waiting space is equal to $K-C$ or less than $K-C$. In the first case, he leaves the service area temporarily to join the retrial group (orbit); in the second one, he is kept to wait in the waiting space. Any orbiting customer will repeatedly retry until the time at which he finds an idle server or an idle waiting position. The retrial times are exponentially distributed with rate $\theta > 0$. There is a classical retrial policy where the total retrial rate depends on the number of customers in the orbit. The waiting positive customers will abandon the system if their patience threshold is exceeded and will not retry. We assume that only waiting customers abandon with exponential rate $\gamma > 0$.

A negative arrival has the effect of removing some number of customers from the service. Let q_g be the probability of deleting $1 \leq g \leq C$ customers from service when a negative arrival occurs. We also define the deleting rate $\eta_g = \nu\ q_g$.

The service times follow an exponential distribution having finite mean $(1/\mu)$. Finally, we admit the hypothesis of mutual independence between all random variables defined above.

The state of the system at time t can be described by means of the process

$$\{C(t), N_o(t), t \geq 0\}, \tag{1.1}$$

where $C(t)$ is the number of customers in the service area (being served and waiting) and $N_o(t)$ is the number of customers in the orbit at time t. Under the above assumptions, this bi-variate process is a Markovian one with state space $S = \{0, 1, \ldots, C, C+1, \ldots, K-1, K\} \times \mathbb{N}$. Its infinitesimal transition rates $q_{ij}(k,l)$ are as follows:

$$\begin{cases}
\text{1. For } 0 \leq i \leq C-1 \\
q_{ij}(k,l) = \begin{cases} \lambda & \text{if} \quad (k,l) = (i+1, j) \\ i\mu & \text{if} \quad (k,l) = (i-1, j) \\ \eta_g & \text{if} \quad (k,l) = (i-g, j), 1 \leq g \leq i \\ j\theta & \text{if} \quad (k,l) = (i+1, j-1) \\ -(\lambda + i\mu + j\theta + \eta_g) & \text{if} \quad (k,l) = (i,j) \\ 0 & \text{otherwise;} \end{cases} \\
\text{2. For } C \leq i \leq K-1 \\
q_{ij}(k,l) = \begin{cases} \lambda & \text{if} \quad (k,l) = (i+1, j) \\ C\mu + (i-C)\gamma & \text{if} \quad (k,l) = (i-1, j) \\ \eta_g & \text{if} \quad (k,l) = (i-g, j), 1 \leq g \leq C \\ j\theta & \text{if} \quad (k,l) = (i+1, j-1) \\ -(\lambda + C\mu + (i-C)\gamma + j\theta + \eta_g) & \text{if} \quad (k,l) = (i,j) \\ 0 & \text{otherwise;} \end{cases} \\
\text{3. For } i = K \\
q_{Kj}(k,l) = \begin{cases} \lambda & \text{if} \quad (k,l) = (K, j+1) \\ C\mu + (K-C)\gamma & \text{if} \quad (k,l) = (K-1, j) \\ -(\lambda + C\mu + (K-C)\gamma) & \text{if} \quad (k,l) = (K,j) \\ 0 & \text{otherwise.} \end{cases}
\end{cases} \tag{1.2}$$

This model is of practical interest: it can be used for analyzing the performance in call centers and multi-processor computer systems. Indeed, consider a call center at which customer calls arrive and are served by an agent if one is available (there is a finite number of agents). If not, the arriving calls join the buffer space having finite capacity where they are assigned a random life span (so some customers are impatient to wait for an agent to be available and then abandon after waiting for certain time). Since the number of buffers is finite, a call may be balked when all buffers are occupied and retries to access the call center later. The system can be subject to service interruptions or breakdowns. In this case, one or more calls will be deleted from the service. Consider also a computer system consisting of a group of processors. It is often exposed to virus infection. When a virus enters the system (negative arrival), it can destroy the message under transmission and forces the system manager to make a reset of the system.

For the model under study, we obtain the ergodicity condition and the approximate solution by applying the Value Extrapolation method (VE method). We do not examine the efficiency and effectiveness of this method by comparing with other approaches (especially those based on the direct truncation of the infinite state space) existing in the literature and briefly described above. In [19], this comparison is already carried out, in the case of the main M/M/C retrial queue. It was concluded that VE method greatly outperforms the other methods not only in terms of accuracy but also in terms of computation cost. In this work, we have another objective. Indeed, the application of VE method leads to the resolution of a certain algebraic linear system of equations $AX = B$. We have asked the following question: Is (in the case of the complex system studied in this paper) the matrix A, having few properties to be invertible, is really invertible? Given the impossibility of providing a rigorous mathematical demonstration, we use numerical methods. In the first time, we discuss the applicability of some existing numerical methods, and then we solve the algebraic system in question with the help of five methods. We obtain a lot of numerical results to show that VE method can be used to find the solution for the considered model, and examine the system performance.

The rest of the paper is organized as follows. In the next section, we obtain the ergodicity condition for our model. Section 2 deals with solving method: Value Extrapolation. In Section 3, we present pertinent numerical aspects of our application. Finally, Section 4 contains the numerical results and conclusions.

2. Ergodicity condition

The next question to be investigated is the ergodicity of the Markov process $\{C(t), N_o(t), t \geq 0\}$. To this end, we give the following theorem.

Theorem 2.1. *The Markov process $\{C(t), N_o(t), t \geq 0\}$ is ergodic if and only if $\lambda < C\mu + (K - C)\gamma$.*

Proof. **Sufficient condition for ergodicity**

We use a criterion based on mean drifts (theory of Lyapunov functions [30]), in particular a result (named Tweedie's theorem in [16]) [38], which defines the assumptions for a Markov process to be regular and ergodic [21, 22]. Consequently, we consider the following test function $\varphi(s) \equiv \varphi(i, j)=ai + j$, where a is some unknown parameter and $s \in S$. The mean drifts $y_s \equiv y_{ij}$ are

$$y_{ij} = \begin{cases} \lambda a + i\mu(-a) + g\eta_g(-a) + j\theta(a-1), & \text{if} \quad 0 \leq i \leq C-1 \\ \lambda a + (C\mu + (i-C)\gamma)(-a) + g\eta_g(-a) + j\theta(a-1), & \text{if} \quad C \leq i \leq K-1 \\ \lambda + (C\mu + (K-C)\gamma)(-a), & \text{if} \quad i = K. \end{cases}$$

Since $\lim\limits_{j\to\infty} y_{ij}$ exists for all $i = 0, 1, \ldots, K$, the assumptions of Tweedie's theorem take place if and only if all variables L_i, representing the mean drifts from states i, are negative. Thereby,

$$\lim_{j\to\infty} y_{ij} = L_i = \begin{cases} (a-1).\infty, & \text{if} \quad 0 \leq i \leq C-1 \\ (a-1).\infty, & \text{if} \quad C \leq i \leq K-1 \\ \lambda - a\left[C\mu + (K-C)\gamma\right], & \text{if} \quad i = K; \end{cases}$$

$$a - 1 < 0;$$
$$a - 1 < 0;$$
$$\lambda - a\,[C\mu + (K - C)\,\gamma] < 0.$$

This set of linear inequalities can be rewritten as

$$\frac{\lambda}{[C\mu + (K - C)\,\gamma]} < a < 1.$$

Thereafter, the unknown parameter a can be found if and only if the interval $((\lambda/(C\mu + (K - C)\gamma)),1)$ is not empty or, in other words,

$$\lambda < C\mu + (K - C)\gamma. \tag{2.1}$$

The expression (2.1) is a sufficient condition for ergodicity of the model under investigation.

Necessary condition for ergodicity

To prove that (2.1) is also a necessary condition, we apply the following theorem [27].

Theorem 2.2. *Let ζ_n be a Markov chain with state space S and one-step transition probabilities r_{sp}. Assume that there exist*:

- *A nonnegative function $\varphi(s)$, $s \in S$, such that for some d we have*
$$r_{sp} \neq 0 \Rightarrow |\varphi(s) - \varphi(p)| \leq d;$$
- *A positive number b such that*:
  (a) *set $A_b \equiv \{s \in S/\varphi(s) > b\} \neq 0$;*
  (b) $\inf_{s\in A_b} x_s \equiv \inf_{s\in A_b} E(\varphi(\zeta_{n+1}) - \varphi(s)/\zeta_n = s)$ *is nonnegative (or positive).*

Then the chain $\{\zeta_n\}$ is nonergodic.

Consider $\{\zeta_n\}$ an embedded Markov chain for the above defined Markov process $\{C(t), N_o(t), t \geq 0\}$. Its one step transition probabilities are given by

1. For $0 \leq i \leq C - 1$

$$r_{ij}(k,l) = \begin{cases} \dfrac{\lambda}{\lambda + i\mu + j\theta + \eta_g} & \text{if} \quad (k,l) = (i+1, j) \\ \dfrac{i\mu}{\lambda + i\mu + j\theta + \eta_g} & \text{if} \quad (k,l) = (i-1, j) \\ \dfrac{\eta_g}{\lambda + i\mu + j\theta + \eta_g} & \text{if} \quad (k,l) = (i-g, j),\ 1 \leq g \leq i \\ \dfrac{j\theta}{\lambda + i\mu + j\theta + \eta_g} & \text{if} \quad (k,l) = (i+1, j-1) \\ 0 & \text{otherwise} \end{cases};$$

2. For $C \leq i \leq K - 1$

$$r_{ij}(k,l) = \begin{cases} \dfrac{\lambda}{\lambda + C\mu + (i-C)\gamma + j\theta + \eta_g} & \text{if} \quad (k,l) = (i+1, j) \\ \dfrac{C\mu + (i-C)\gamma}{\lambda + C\mu + (i-C)\gamma + j\theta + \eta_g} & \text{if} \quad (k,l) = (i-1, j) \\ \dfrac{\eta_g}{\lambda + C\mu + (i-C)\gamma + j\theta + \eta_g} & \text{if} \quad (k,l) = (i-g, j),\ 1 \leq g \leq C \\ \dfrac{j\theta}{\lambda + C\mu + (i-C)\gamma + j\theta + \eta_g} & \text{if} \quad (k,l) = (i+1, j-1) \\ 0 & \text{otherwise} \end{cases};$$

3. For $i = K$

$$r_{Kj}(k,l) = \begin{cases} \dfrac{\lambda}{\lambda + C\mu + (K-C)\gamma} & \text{if} \quad (k,l) = (K, j+1) \\ \dfrac{C\mu + (K-C)\gamma}{\lambda + C\mu + (K-C)\gamma} & \text{if} \quad (k,l) = (K-1, j) \\ 0 & \text{otherwise} \end{cases}.$$

Assume that $\lambda \geq C\mu + (K-C)\gamma$ and consider the test function $\varphi(i,j) = i + j$.

Then,

- If $r_{ij}(n,m) \neq 0$, then $|i-n| \leq 1$, $|j-m| \leq 1$. So $|\varphi(i,j) - \varphi(n,m)| = |i-n+j-m| \leq |i-n| + |j-m| \leq 2$.
- Choose for b any positive number. Indeed,
  (a) $A_b = \{(i,j)/i = 0, 1, \ldots, K; j > \max(b-i, 0)\} \neq 0$;
  (b)

$$x_{ij} = \begin{cases} \dfrac{\lambda - i}{\lambda + i\mu + j\theta + \eta_g} & \text{if} \quad 0 \leq i \leq C-1 \\ \dfrac{\lambda - i}{\lambda + C\mu + (i-C)\gamma + j\theta + \eta_g} & \text{if} \quad C \leq i \leq K-1 \\ \dfrac{\lambda - K}{\lambda + C\mu + (K-C)\gamma} & \text{if} \quad i = K \end{cases}.$$

Thus, the variable x_{ij} is always nonnegative and moreover $x_{ij} \geq 0$ for $(i,j) \in A_b$.
Therefore, the Markov process $\{C(t), N_o(t); t \geq 0\}$ is nonergodic.

At present, consider the case where $\lambda > C\mu + (K-C)\gamma$ and the test function is $\varphi(i,j) = ai + j$ with $a \in (1, (\lambda/(C\mu + (K-C)\gamma)))$. Application of the previous theorem gives:

- If $r_{ij}(n,m) \neq 0$ then $|i-n| \leq 1$,$|j-m| \leq 1$. So $|\varphi(i,j) - \varphi(n,m)| = |a(i-n) + j - m| \leq a\,|i-n| + |j-m| \leq a+1$, *i.e.*, $d = a+1$.
- The set A_b is nonempty for any $b > 0$. So we must choose b in such a way that $\inf\limits_{(i,j)\in A_b} x_{ij} > 0$. But for $i = K$, $x_{Kj} = \frac{\lambda - aK}{\lambda + C\mu + (K-C)\gamma} > 0$ (for all j), and for $0 \leq i \leq K-1$, $\lim\limits_{j\to\infty} x_{ij} = a - 1 > 0$.

Therefore, for any $i = 0, 1, \ldots K$ there exist N_i such that $x_{ij} > \frac{a-1}{2}$ for all N_i. Here, N_i is the number of i-customers in the orbit. If $\epsilon = \min\{\frac{\lambda - aK}{\lambda + C\mu + (K-C)\gamma}, \frac{a-1}{2}\}$, then $x_{ij} \geq \epsilon$ for all $(i,j) \in S$ except for (i,j) such that $i = 0, 1, \ldots, K-1$ and $0 \leq j \leq N_i - 1$. Suppose that b is large enough so that the set A_b does not contain these states. Then $x_{ij} \geq \epsilon$, for all $(i,j) \in A_b$, or $\inf\limits_{(i,j)\in A_b} x_{ij} \geq \epsilon > 0$. Consequently, the continuous time Markov chain $\{C(t), N_o(t), t \geq 0\}$ is transient. □

Remark 2.3. Since only customers in service area can be deleted when a negative arrival occurs, the deleting rate η_g does not appear in the obtained ergodicity condition.

3. Value extrapolation method

Assume that the ergodicity condition $\lambda < C\mu + (K-C)\gamma$ is satisfied. We now can define the steady state joint distribution of the number of customers in the service and waiting areas and the number of customers in the orbit

$P_{ij} = \lim\limits_{t\to\infty} P(C(t) = i, N_o(t) = j), (i,j) \in S$. The Kolmogorov equations for P_{ij} are given by

$$(\lambda + i\mu + j\theta + \eta_g)P_{ij} = \lambda P_{i-1,j} + (j+1)\theta P_{i-1,j+1} + ((i+1)\mu + \eta_g)P_{i+1,j}, 0 \leq i \leq C-1 \qquad \text{and} \qquad 1 \leq g \leq i;$$

$$(\lambda + C\mu + (i-C)\gamma + j\theta + \eta_g)P_{ij} = \lambda P_{i-1,j} + (j+1)\theta P_{i-1,j+1} + (C\mu + (i+1-C)\gamma + \eta_g)P_{i+1,j},$$
$$C \le i \le K-2 \qquad \text{and} \qquad 1 \le g \le C;$$

$$(\lambda + C\mu + (i-C)\gamma + j\theta + \eta_g)P_{ij} = \lambda P_{i-1,j} + (j+1)\theta P_{i-1,j+1} + (C\mu + (i+1-C)\gamma)P_{i+1,j},$$
$$i = K-1 \qquad \text{and} \qquad 1 \le g \le C;$$

$$(\lambda + C\mu + (K-C)\gamma)P_{K,j} = \lambda P_{K-1,j} + (j+1)\theta P_{K-1,j+1} + \lambda P_{K,j-1}, i = K$$

and the normalizing condition is

$$\sum_{i=0}^{K}\sum_{j=0}^{\infty} P_{ij} = 1.$$

The steady state distribution of the Markov process $\{C(t), N_o(t), t \ge 0\}$ contributes to obtaining some formulae for the main performance measures: the steady state blocking probability $P_K = \lim_{t\to\infty} P(C(t) = K)$; the mean number of customers in the orbit $\overline{N}_o = \lim_{t\to\infty} E[N_o(t)]$; the mean number of customers in the service and waiting areas $\overline{C} = \lim_{t\to\infty} E[C(t)]$.

The essential characteristics of the state space of our model are its infinite dimension (due to the unlimited orbit) and the non-homogeneity along it (this non-homogeneity is produced by using the classical retrial policy). So the exact analytical solution for the steady state distribution P_{ij} cannot be obtained. Thus, we apply Value Extrapolation method based on the tools of the theory of Markov Decision Processes (MDP_s).

Markov Decision Process [35] is a stochastic control process. At each time step, it is in some state s, and the decision maker may choose any action a that is available in state s. The process responds at the next time step by randomly moving into a new state s' and giving a corresponding reward $R(s, s')$. The probability that the process moves into its new state is influenced by the chosen action. This is given by the state transition function $P(s, s')$. Consequently, a MDP can be defined as $\{S, A, P, R\}$, where S, A, P and R are a set of states, a set of actions, a state transition function and a revenue function, respectively. A revenue function must be defined as a function of the system state, $r(s)$. Following the definition of the revenue function for every state, we will find a mean revenue rate of the entire process, $\bar{r} = \int r(s).P(s)ds$. The latter is the performance measure which we want to compute. After performing an action in state s, the system will collect a revenue $r(s)$. However, as the number of transitions increases, the average collected revenue converges to $\bar{r}$. The relative state value $v(s)$ shows the difference between the total revenue cumulated when the system starts at state s and the total revenue cumulated in a system for which the mean revenue rate at all states is $\bar{r}$:

$$v(s) = E\left[\int_{t=0}^{\infty} (r(S(t)) - \bar{r})\, \mathrm{d}t | S(0) = s\right].$$

The Howard equations relate revenues $r(s)$, relative state values $v(s)$ and transition rates $q_{ss'}$ in the following manner:

$$r(s) - \bar{r} + \sum_{s'} q_{ss'}(v(s') - v(s)) = 0, \quad \forall s.$$

In the case of our system, by using the transition rates (1.2) given in Section 1, we obtain the following Howard equations:

$$\begin{gathered} r(i,j) - \bar{r} + \lambda(v(i+1,j) - v(i,j)) + i\mu(v(i-1,j) - v(i,j)) \\ +j\theta(v(i+1,j-1) - v(i,j)) + \eta_1(v(i-1,j) - v(i,j)) = 0, \\ 0 \le i \le C-1; \end{gathered} \tag{3.1}$$

$$r(i,j) - \bar{r} + \lambda(v(i+1,j) - v(i,j)) + (C\mu + (i-C)\gamma)(v(i-1,j)$$
$$-v(i,j)) + j\theta(v(i+1,j-1) - v(i,j)) + \eta_1(v(i-1,j) - v(i,j)) = 0,$$
$$C \leq i \leq K-1; \tag{3.2}$$

$$r(K,j) - \bar{r} + (C\mu + (K-C)\gamma)(v(K-1,j) - v(K,j))$$
$$+\lambda(v(K,j+1) - v(K,j)) = 0,$$
$$i = K. \tag{3.3}$$

Here, without loss of generality, we have supposed that an arriving negative customer can delete only one customer in service (η_g=η_1). This is quite realistic: in a call center, it is more commonly observed that only one telephone line (allocated to an agent) suffers the service interruption (breakdown).

Since the state space S of the process $\{C(t), N_o(t), t \geq 0\}$ is infinite, we will perform a truncation procedure to obtain
$\hat{S} := \{s = (i,j) : 0 \leq i \leq K, 0 \leq j \leq L\}$. The number of Howard equations corresponds to the number of states, $\left|\hat{S}\right|$, and the number of unknowns is defined by $\left|\hat{S}\right|$ relative state values plus expected revenue $\bar{r}$. However, as only the differences in the relative values appear in the Howard equations, we can set $v(0) = 0$. Thereby, we have a linear system of equations with the same number of equations as unknowns.

The next problem to study is the determination of an extrapolating function $f(s)$ that interpolates some points $(s, v(s))$ for $s \in \hat{S}$ so that it approximates also $(s, v(s))$ for $s \notin \hat{S}$. From [19], we have that the most appropriate, in this type of applications, extrapolating functions are the polynomials. We have two types of extrapolating procedures depending on whether we use all pairs $(s, v(s))$ of the state space or a subset of them, S_f. The choice of an extrapolating function and also of a subset S_f is governed by the fact that the number of different pairs $(s, v(s))$ in the subset in question has to be equal or greater than the number of coefficients in the polynomial. Moreover, the choice of S_f also depends on the relative state value we want to extrapolate.

Thus, we have an interpolation problem. The polynomial, in its Lagrange setting, is $L(w) = \prod_{j=0}^{n-1} v(w_j) l_j(w)$, where

$$l_j(w) = \prod_{i=0 j\neq i}^{n-1} \frac{w - w_i}{w_j - w_i} = \frac{w - w_0}{w_j - w_0} \cdots \frac{w - w_{j-1}}{w_j - w_{j-1}} \frac{w - w_{j+1}}{w_j - w_{j+1}} \cdots \frac{w - w_{n-1}}{w_j - w_{n-1}} \cdot$$

For the truncation problem in interest, we will have a Howard equation (6) in which appears the state value $v(K, L+1)$ of the state $(K, L+1)$ that does not belong to $\widehat{S}$. Therefore, the $v(K, L+1)$ will be approximated by using some relative state values of the states belonging to $\widehat{S}$. It is important to emphasize that the extrapolation of $v(K, L+1)$ will be performed by using the states having form $s = (K, j)$ (with different values of j). With this choice, it is possible to define the mapping w as $w = w((K,j)) = j$. Moreover, we use an $(n-1)$-th degree polynomial to interpolate the n points in $S_f := \{s_i = (K, L-i) | i = 0, \ldots, n-1\}$. Consequently, $w(S_f) = \{w_i = L - i | i = 0, \ldots, n-1\}$:

$$w_0 = L \longrightarrow v(w_0) = v(K, L);$$

$$w_1 = L - 1 \longrightarrow v(w_1) = v(K, L-1);$$

$$\vdots$$

$$w_{n-1} = L - (n-1) \longrightarrow v(w_{n-1}) = v(K, L-(n-1)).$$

In this way, the general expression of the extrapolated relative state value corresponding to the state in question is

$$v^{(n)}(K, L+1) = L^{(n)}(L+1) = \sum_{j=0}^{n-1} v(K, L-j) l_j(L+1).$$

For example, in the case of linear extrapolation with $n = 2$, we use $(L, v(K, L))$ and $(L-1, v(K, L-1))$, so

$$\begin{aligned} v^{(2)}(K, L+1) &= L^{(2)}(L+1) = v(K, L) l_0(L+1) + v(K, L-1) l_1(L+1) \\ &= v(K, L) \frac{(L+1)-(L-1)}{(L-(L-1))} + v(K, L-1) \frac{((L+1)-L)}{((L-1)-L)} \\ &= 2v(K, L) - v(K, L-1). \end{aligned}$$

For $(n-1)$-th degree polynomial, by using the Lagrange basis to reduce the complexity of the procedure, we obtain the following closed form expression for the extrapolated relative state value: $v^{(n)}(K, L+1) = \sum_{k=0}^{n-1} (-1)^k C_n^{k+1} v(K, L-k)$, where n is the number of coefficients taken for Lagrange polynomial. For model under study, we will only have to replace $v(K, L+1)$ by its approximate value in the Howard equation that corresponds to the state $v(K, L)$. For example, if we use linear extrapolation ($n = 2$), the equation becomes

$$r(K, L) - \bar{r} + v(K, L)(-\lambda - C\mu - (K-C)\gamma) + (C\mu + (K-C)\gamma) v(K-1, L) + \lambda v(K, L+1) = 0;$$

$$r(K, L) - \bar{r} + v(K, L)(\lambda - C\mu - (K-C)\gamma) + (C\mu + (K-C)\gamma) v(K-1, L) - \lambda v(K, L-1) = 0.$$

As $v(K, L+1)$ no longer appears in the Howard equations, we have a linear system of $(K+1)(L+1)$ equations with the same number of unknowns. This system can be expressed in the following matrix form $AX = B$, where X is a vector having $(K+1)(L+1)$ unknowns and B contains the negative revenue rates:

$$X = [\bar{r}, v(0,1), \ldots, v(0, L), v(1, 0), \ldots, v(C, 0), \ldots, v(C, L), \ldots, v(K, 0), \ldots, v(K, L)]^t;$$

$$B = [-r(0,0), -r(0,1), \ldots, -r(C,0), -r(C,1), \ldots, -r(K, L)]^t.$$

The matrix A is composed of the submatrices and constructed making all the elements in the first column of A' equal to -1. Intially, the matrix A' is given by

$$A' = \begin{bmatrix} D_1^0 & D_2^1 & O & \ldots & O & O & O \\ D_0^0 & D_1^1 & D_2^2 & \ldots & O & O & O \\ O & D_0^1 & \ddots & \ddots & O & O & O \\ \vdots & \vdots & \ddots & D_1^C & \ddots & \vdots & \vdots \\ O & O & O & D_0^C & \ddots & \ddots & O \\ O & O & O & \ldots & \ddots & D_1^{K-1} & D_2^K \\ O & O & O & \ldots & O & D_0^{K-1} & (D_1^K)^t, \end{bmatrix}$$

where each submatrix has the dimension $(L+1) \times (L+1)$ and O is the zero matrix.

$$D_0^i = ((i+1)\mu + \eta_1) I, \qquad \text{for} \qquad 0 \le i \le C-1;$$

$$D_0^i = (C\mu + (i+1-C)\gamma + \eta_1) I, \qquad \text{for} \qquad C \le i \le K-2;$$

$$D_0^i = (C\mu + (i+1-C)\gamma) I, \qquad \text{for} \qquad i = K-1;$$

$$D_2^i = \begin{bmatrix} \lambda & 0 & 0 & \dots & 0 & 0 \\ \theta & \lambda & \vdots & \dots & 0 & 0 \\ \vdots & 2\theta & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \lambda & 0 \\ 0 & 0 & 0 & \dots & L\theta & \lambda \end{bmatrix}, \quad \text{for} \quad 1 \leq i \leq K;$$

$$D_1^i = \begin{bmatrix} \alpha & 0 & 0 & \dots & 0 \\ 0 & \alpha-\theta & 0 & \dots & 0 \\ 0 & 0 & \alpha-2\theta & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \alpha-L\theta \end{bmatrix}, \quad \text{for} \quad 0 \leq i \leq C-1 \quad \text{and} \quad \alpha = -\lambda - i\mu - \eta_1;$$

$$D_1^i = \begin{bmatrix} \alpha & 0 & 0 & \dots & 0 \\ 0 & \alpha-\theta & 0 & \dots & 0 \\ 0 & 0 & \alpha-2\theta & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \alpha-L\theta \end{bmatrix}, \quad \text{for} \quad C \leq i \leq K-1 \quad \text{and} \quad \alpha = -\lambda - C\mu - (i-C)\gamma - \eta_1.$$

When $i = K$, using the linear extrapolation ($n = 2$), we obtain,

$$D_1^K = \begin{bmatrix} \beta & 0 & \dots & 0 & 0 \\ \lambda & \beta & \dots & 0 & 0 \\ 0 & \lambda & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & \beta & -\lambda \\ 0 & 0 & \dots & \lambda & \lambda - C\mu - (K-C)\gamma \end{bmatrix}$$

where $\beta = -\lambda - C\mu - (K-C)\gamma$. So, if the extrapolation is done with $n \leq L+1$ points, the matrix D_1^K will be defined as

$$D_1^K = \begin{bmatrix} \beta & 0 & \dots & 0 & c_L^{(n)} \\ \lambda & \beta & \dots & 0 & c_{L-1}^{(n)} \\ 0 & \lambda & \dots & 0 & c_{L-2}^{(n)} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & \lambda c_2^{(n)} \\ 0 & 0 & \dots & \beta & \lambda c_1^{(n)} \\ 0 & 0 & \dots & \lambda & -\lambda - C\mu - (K-C)\gamma + \lambda c_0^{(n)} \end{bmatrix},$$

where

$$c_l^{(n)} = \begin{cases} (-1)^l C_n^{l+1}, & \text{if} \quad l < n \\ 0, & \text{if} \quad l \geq n. \end{cases}$$

We underline that the size of matrix A does not depend on the degree of the polynomial used to perform the extrapolation; only the last column in the matrix D_1^K depends on the polynomial adjustment. It must be noted that the value extrapolation method permits us to compute one performance measure each time we solve the system. So choosing another performance measure to solve will only affect to the values in B.

The next point to clarify is how the revenue rate can be set in order to obtain the performance measures for our model. In the case of the blocking probability, $r(i,j) = 1$ for $i = K$ and all $0 \leq j \leq L$, otherwise

$r(i,j) = 0$; when we wish to calculate the mean number of customers in the orbit, $r(i,j) = j$ for all $0 \le i \le K$ and $0 \le j \le L$, otherwise $r(i,j) = 0$; and finally in the case of the mean number of customers in the service and waiting areas, $r(i,j) = i$ for all $0 \le i \le K$ and $0 \le j \le L$, $r(i,j) = 0$ otherwise.

The solution of the system $AX = B$ can be obtained by using one of the numerical methods presented below in the next section.

4. Numerical aspects

In fact, the solving of the problem under investigation leads to solve an algebraic linear system

$$AX = B \tag{4.1}$$

where X and B are two vectors, $X, B \in R^N$, with $N = (K+1)(L+1)$ as the dimension of the matrix A. The numerical solution of (4.1) is not *a priori* easy and can be delicat to achieve. In our application the matrix A is sparse, of tridiagonal structure and has, possibly, large dimension. Moreover, A is not a symmetric definite positive matrix or a strictly or diagonally dominant matrix. However A is an irreducible matrix; from a mathematical point of view this property can be stated by using one the following results (see [25, 36]).

Theorem 4.1. *If a matrix is irreducible, then there exists on each row and on each column, at least one nonzero off-diagonal entry.*

In fact Theorem 3.1. gives a sufficient condition of irreducibility. We can also use a more general result to prove the irreducibility of the matrix A:

Theorem 4.2. *A matrix $A \in L(R^N)$ is irreducible if and only if, for any two indices $1 \le i, j \le N$, there is a sequence of nonzero elements of A of the form $a_{i,i1}, a_{i1,i2}, \ldots, a_{im,j}$.*

For the matrix A it is possible, as an exercise, to write down explicitly a chain of nonzero elements for any i, j, but the indexing can become cumbersome. It is more transparent to return to the considered application; then the irreducibility of A follows immediately and obviously from the fact that all states of S are communicating, *i.e.*, the transition matrix is formed of a single recurent class of states; therefore the irreducibility of A reflects the fact that any state is reachable from any other state, and reciprocally. Nevertheless irreducibility of A implies nothing in general about its invertibility. Unfortunately, as previously said, the matrix A is not diagonally dominant matrixÂ ; so once again, we have to come back to the application in order to prove the invertibility of A. Such previous property of invertibility follows from the fact that the Markov chain (1.1) is ergodic if $\lambda < C\mu + (K-C)\gamma$: irreducible, apperiodic and obviously positive since the elements of the transition matrix are transition probabilities. Consequently, when the ergodicity condition $\lambda < C\mu + (K-C)\gamma$ is verified, the matrix A is invertible [38].

For the numerical solution of the algebraic system (4.1), we can use several and various methods. Usually there are two main classes of numerical methods:

- the direct methods, such as the classical and optimal elimination Gauss method whose complexity is of order of $2/3N^3$ arithmetic operations;
- the iterative methods, that are best suited when the matrix A is sparse, due to the fact that the number of arithmetic operations is low; nevertheless, for such iterative methods the convergence and the asymptotic rate of convergence must be studied.

Among the classical iterative methods are included on one hand the relaxation methods, such that the Jacobi and Gauss-Seidel methods and on the other hand the classical conjugate gradient method. The Jacobi method will diverge; indeed if $\varphi(J)$ denotes the spectral radius of the Jacobi's matrix J, defined by

$$J = Id - H^{-1}.A,$$

where H denotes the diagonal of A, due to the fact that

$$\varphi(J) \leq \|J\|_\infty$$

where $\|J\|_\infty$ is the maximum value of the sum of the moduli of the elements of J, then at least, $\|J\|_\infty = (N-1)$ since the first column of J is formed of $(N-1)$ terms equal to (-1), the diagonal term of this first column is zero; so, at worst, regardless of what may be happening on the other columns, $\|J\|_\infty = (N-1)$ and the Jacobi method is likely to diverge. It seems more difficult to predict the convergence or the divergence of the Gauss-Seidel, but taking account of the divergent behavior of the Jacobi method, we discard the relaxation methods.

The conjugate gradient method is not applicable in our case, since the matrix A is not symmetric positive definite. Note that there exist many variants of the conjugate gradient method when the matrix A is not symmetric positive definite.

One of the most used methods is the projection method on Krylov subspaces $\left\{\mathrm{e}^0, A\mathrm{e}^0, A^2\mathrm{e}^0, A^m\mathrm{e}^0\right\}$; the different versions of Krylov subspaces methods arise from the different choice of subspaces. Note also that on such subspaces vectors defining the basis are orthonormalized by the Gram$-$Schmidt process. This leads to the FOM (Full Orthogonalization Method) and the GMRES method (Generalised Minimum Residual Method), the last one being the best known. The GMRES method is constructed such that convergence is ensured but with the disadvantage that the storage of information can be very high, which is why we discarded this algorithm. Since the GMRES algorithm becomes impractical when the dimension of A is large, becaue of the growth of memory and computational requirements, variants of the GMRES method is to consider either a restarted method or a truncated method of GMRES (see [36]).

Among the methods very easy to code when the matrix is not symmetric positive definite, we will retain: the CGNR method, or Conjugate Gradient applied on the Normal Equations (see [25] in p. 536), the biconjugate gradient method (see [28] in p. 371), the BiCG conjugate gradient squared method (see [28] in p. 379), the BiCGSTAB method (see [28] in p. 379).

These four last algorithms will be briefly summarized and described in the sequel.

We also do not know if the matrix is well conditioned, *i.e.*, if the spread of rounding errors can damage the calculated result. Since A is not symmetric, the condition number is given by [25]

$$Z(A) = \frac{\mu_{\max}}{\mu_{\min}},$$

where $\mu_{\max}$ and $\mu_{\min}$ represent the largest and smallest singular values of the matrix A. This computation is particularly important when using the Gauss method that includes a relatively large number of arithmetic operations but can be useful to measure the speed of convergence of considered iterative methods.

When iterative methods are used, for the definition of the stopping criterion, the normwise backward error (see [36]) defined by

$$\tau_{A,B}(X) = \frac{\|e\|}{(\|A\| \,.\, \|X\| + \|B\|)}$$

is usually used, where $e = AX - B$ is the residue. The backward error analysis is a powerful concept for analyzing the quality of an approximate solution; indeed such backward error analysis is independent of the details of round-off propagation, since the errors introduced during the computation are interpreted in terms of perturbations of the initial guess X^0, and the computed solution is considered as exact for the perturbed problem; moreover round-off errors are seen as data perturbations and can be compared to numerical approximation or to physical uncertainties measurements. So, in the presented iterative methods, the iterative process is stopped when

$$\tau_{A,B}(X) < \epsilon$$

where ϵ is a given threshold.

As previously said, and among the used iterative methods, we will describe the previous algorithms.

- The CGNR method is based on technique for converting a non-symmetric algebraic linear system into a symmetric one. Such technique solves the equivalent linear system $A^tAX = A^tB$ called the normal equations, instead of (3.1). If A is nonsingular, as we have assumed, then A^tA is symmetric positive definite and the classical conjugate gradient method can be applied. This CGNR algorithm can be summarized in the following form:

Starting from any X^0, let $\mathrm{e}^0 = B - AX^0$, $p^0 = A^t\mathrm{e}^0$

For $k = 0, 1, \ldots$ **until** convergence **Do**

$$\alpha^k = \frac{\left\|A^t\mathrm{e}^k\right\|^2}{\left\|Ap^k\right\|^2},$$

$$X^{k+1} = X^k + \alpha^k p^k,$$

$$\mathrm{e}^{k+1} = \mathrm{e}^k - \alpha^k Ap^k,$$

$$\phi^{k+1} = \frac{\left\|A^t\mathrm{e}^{k+1}\right\|^2}{\left\|A^t\mathrm{e}^k\right\|^2},$$

$$p^{k+1} = A^t\mathrm{e}^{k+1} + \phi^{k+1}p^k.$$

Endfor

This method will be convergent for any regular matrix A, since A^tA is symmetric positive definite. Nevertheless, this approach is avoided in pratice since the condition number of A^tA is much worse than the condition number of A; since the condition number $Z(A^tA) = Z^2(A)$, the rate of convergence is weaker and at each iteration it is necessary to compute two products of a matrix by a vector. However, the normal equation approach may be adequate in some situations. Indeed, there are even applications in which it is preferred to the usual Krylov techniques.

- The biconjugate gradient method (BiCG) is considered as a method for solving the following system

$$\begin{bmatrix} O & A \\ A^t & O \end{bmatrix} \begin{bmatrix} Q \\ X \end{bmatrix} = \begin{bmatrix} B \\ V \end{bmatrix},$$

where V is arbitrarily chosen. The classical version of BiCG (Fletcher 1975) is the following:

Starting from any X^0, let $\mathrm{e}^0 = B - AX^0$, $\widetilde{e}^0 = \mathrm{e}^0$, $p^0 = \mathrm{e}^0$, $\widetilde{p}^0 = \widetilde{e}^0$

For k=0,1,... **until** convergence **Do**

$$\alpha^k = \frac{\langle \widetilde{e}^k, \mathrm{e}^k \rangle}{\langle \widetilde{p}^k, Ap^k \rangle},$$

$$X^{k+1} = X^k + \alpha^k p^k,$$

$$\mathrm{e}^{k+1} = \mathrm{e}^k - \alpha^k Ap^k,$$

$$\widetilde{e}^{k+1} = \widetilde{e}^k - \alpha^k A^t\widetilde{p}^k,$$

$$\phi^{k+1} = \frac{\langle \widetilde{e}^{k+1}, \mathrm{e}^{k+1} \rangle}{\langle \widetilde{e}^k, \mathrm{e}^k \rangle},$$

$$p^{k+1} = \mathrm{e}^{k+1} + \phi^{k+1}p^k,$$

$$\widetilde{p}^{k+1} = \widetilde{e}^{k+1} + \phi^{k+1}\widetilde{p}^k.$$

Endfor The great advantage of the BiCG method is that it is simple and easy to encode and store information is not very high. However, this method has the disadvantage of being erratic, *i.e.*, unstable. Then the BiCGSTAB method inproves the BiCG method.

- The BiCG conjugate gradient squared method or BiCGS method corresponds to a faster iterative method than BiCG and is defined as follows:
Starting from any X^0, let $\mathrm{e}^0 = B - AX^0$, $\widetilde{p}^0 = p^0 = \mathrm{e}^0$
For k=0,1,... **until** convergence **Do**

$$\alpha^k = \frac{\langle \mathrm{e}^k, \mathrm{e}^0 \rangle}{\langle A\widehat{p}^k, \mathrm{e}^0 \rangle},$$

$$u^k = p^k - \alpha^k A\widetilde{p}^k,$$

$$X^{k+1} = X^k + \alpha^k (p^k + u^k),$$

$$\mathrm{e}^{k+1} = \mathrm{e}^k - \alpha^k A(p^k + u^k),$$

$$\phi^{k+1} = \frac{< \mathrm{e}^0, \mathrm{e}^{k+1} >}{< \mathrm{e}^0, \mathrm{e}^k},$$

$$p^{k+1} = \mathrm{e}^{k+1} + \phi^{k+1} u^k,$$

$$\widetilde{p}^{k+1} = p^{k+1} + \phi^{k+1}(u^k + \phi^{k+1}\widetilde{p}^k).$$

Endfor
Note that even though A^t is not needed anymore there are still two matrix vector products for each iteration.

- The BiCGSTAB method (Van der Vorst 1992) introduce a parameter ω^k chosen as to minimize the Euclidean norm of the residual. This leads to the BiCGSTAB algorithm described below:

Starting from any X^0, let $\mathrm{e}^0 = B - AX^0$, $p^0 = \mathrm{e}^0$, $\widetilde{e}^0$ arbitrary
For $k = 0, 1, \ldots$ **until** convergence **Do**

$$\alpha^k = \frac{\langle \widetilde{e}^0, \mathrm{e}^k \rangle}{\langle \widetilde{e}^0, Ap^k \rangle},$$

$$F^k = \mathrm{e}^k - \alpha^k Ap^k,$$

$$\omega^k = \frac{\langle AF^k, F^k \rangle}{\langle AF^k, \omega^k \rangle},$$

$$X^{k+1} = X^k + \alpha^k p^k + \omega^k F^k,$$

$$e^{k+1} = F^k - \omega^k A F^k,$$

$$\phi^{k+1} = \frac{\left(\left\langle \widetilde{e}^0, e^{k+1} \right\rangle .\alpha^k\right)}{\left(\left\langle \widetilde{e}^0, e^k \right\rangle .\omega^k\right)},$$

$$p^{k+1} = e^{k+1} + \phi^{k+1}(p^k - \omega^k A p^k).$$

Endfor

Note that, the BiCGSTAB method, is more stable and often gives better results.

Note that, for the solution of very large scale algebraic linear systems, we can also consider vectorization or parallelization in the implementation of the previous methods.

Regarding the Gauss's method, it was cited for memory, and is not very well suited for solving large linear systems, even though the matrices are sparse. Note that, in order to decrease the influence of rounding error, in our numerical experiments, we have considered the Gauss's method with partial pivoting.

Remark 4.3. However it is possible to consider a Gauss's method with blocks elimination which have the advantage of reducing the complexity of the classical Gauss's method and could be parallelized in the case of very large systems.

5. Numerical experiments for the computation of P_K, $\overline{N}_o$ and $\overline{C}$

The first results of numerical experiments are summarized in Tables 1, 2, 3 and 4. For each implementation, the codes were written using MATLAB, allowing to work on 64-bits representation of the real numbers. We assume that the system parameters are given as follows: $K = 12$, $C = 8$, $\lambda = 8$, $\mu = 1$, $\nu = 1$, probability of deleting one customer from the service area $q_1 = 0.2$, $\theta = 1$, $\gamma = 2$, $n = 2$. So we have that the traffic intensity $\rho = \frac{\lambda}{C\mu+(K-C)\gamma} = \frac{1}{2} < 1$.

For the three performance characteristics, it is noted that the use of these 5 methods leads to consistent results for solving the corresponding algebraic systems. This suggests that the conditioning of the different matrices is relatively good. For large matrices, the results of the variants of the conjugate gradient method is preferable because the effect of the spread drop of errors is minimized.

For the variants of the conjugate gradient method the number of iterations is sometimes greater than the dimension of the matrix A; this is due to defects of orthogonality because of rounding errors. This could be seen in the extent it was possible to calculate the singular values of the matrices A and that the ratio of the extreme singular values is between a value of the order of 216 for a dimension of the matrix 52 and a value of the order of 711 for a dimension 156. This condition number increases with the dimension of the matrix. Furthermore the calculated residue is of the order of 10^{-16} in the case of using the method of Gauss and of the order of 10^{-8} when using one of the variants of the conjugate gradient methods. For the use of these methods, the accuracy difference is due to a less stringent stopping criterion due to the fact that the stop threshold is fixed to 10^{-8}. Regarding the convergence speed of the variants of the conjugate gradient methods we observe that the method with the best convergence rate is Bicgstab method, which is not in contradiction with the development of this method and is consistent with that we can find in the literature.

Another method that converges relatively well is the CGSsquared method where we find the characteristics of the conjugate gradient method when the matrix is to invert the matrix $A^t A$.This method is interesting insofar as the condition number is not too large.

The BiCG method converges more slowly whereas the method of the normal equation has an extremely slow convergence of the order of twice the dimension of the matrix.

The calculations are performed in a very short time. In terms of computing time it should be noted that Gauss leads to extremely short resolution times. However, in the case of very large matrices, this resolution times may worsen due to the fill-in phenomenon which appears in the matrix factorization.

As regards the time calculations corresponding to the use of the variants of the conjugate gradient method when A is not symmetric positive definites, the fastest way is the BiCG method, followed by Bicgstab method

TABLE 1. Blocking probability P_K with different methods.

		Gauss	BICG	BICGSTAB	BICGS	CGNR
dim $A = 52$	P_K	0.02911268	0.02911268	0.02911269	0.02911269	0.02911268
	max error	$8.88.10^{-16}$	$2.37.10^{-8}$	$2.14.10^{-8}$	$4.42.10^{-8}$	$1.60.10^{-8}$
$L = 3$	Iteration	–	50	32	36	79
	time	0.092	0.149	0.180	0.384	0.251
cond$(A) = 2.16762645.10^2$	Norm final	–	$3.396.10^{-7}$	$3.515.10^{-7}$	$1.069.10^{-6}$	$1.631.10^{-7}$
dim $A = 78$	P_K	0.02926945	0.02926945	0.02926946	0.02926945	0.0292694
	max error	$1.66.10^{-15}$	$5.82.10^{-8}$	$1.57.10^{-8}$	$6.29.10^{-9}$	$1.29.10^{-8}$
$L = 5$	Iteration	–	61	37	44	123
	time	0.143	0.276	0.214	0.504	0.482
cond$(A) = 3.23163797.10^2$	Norm final	–	$1.214.10^{-7}$	$5.865.10^{-7}$	$1.604.10^{-7}$	$1.937.10^{-7}$
dim $A = 104$	P_K	0.02931432	0.02931432	0.02931433	0.02931432	0.02931432
	max error	$2.22.10^{-15}$	$5.02.10^{-8}$	$2.43.10^{-8}$	$2.77.10^{-8}$	$2.33.10^{-8}$
$L = 7$	Iteration	–	65	39	48	164
	time	0.184	0.302	0.196	0.417	0.943
cond$(A) = 4.40963672.10^2$	Norm final	–	$2.465.10^{-7}$	$2.320.10^{-7}$	$7.528.10^{-6}$	$2.691.10^{-7}$
dim $A = 130$	P_K	0.02932648	0.02932648	0.02932648	0.02932648	0.02932648
	max error	$3.55.10^{-15}$	$4.98.10^{-8}$	$1.37.10^{-8}$	$2.88.10^{-8}$	$1.31.10^{-8}$
$L = 9$	Iteration	–	72	42	53	206
	time	0.305	0.417	0.311	0.532	1.226
cond$(A) = 5.699423.10^2$	Norm final	–	$2.051.10^{-7}$	$3.250.10^{-7}$	$2.495.10^{-7}$	$1.651.10^{-7}$
dim$A = 143$	P_K	0.02932859	0.02932859	0.02932859	0.02932859	0.02932859
	max error	$3.55.10^{-15}$	$2.37.10^{-8}$	$2.09.10^{-8}$	$2.73.10^{-8}$	$2.01.10^{-8}$
$L = 10$	Iteration	–	73	44	56	225
	time	0.333	0.285	0.227	0.638	2.099
cond$(A) = 6.38853813.10^2$	Norm final	–	$7.229.10^{-7}$	$1.058.10^{-7}$	$2.823.10^{-7}$	$1.009.10^{-6}$
dim $A = 156$	P_K	0.02932966	0.02932966	0.02932965	0.02932966	0.02932966
	max error	$2.66.10^{-15}$	$1.98.10^{-8}$	$3.17.10^{-8}$	$9.96.10^{-9}$	$9.68.10^{-9}$
$L = 11$	Iteration	–	80	44	57	247
	time	0.403	0.314	0.193	0.600	1.808
cond$(A) = 7.10838072.10^2$	Norm final	–	$1.776.10^{-7}$	$1.297.10^{-7}$	$0.003.10^{-7}$	$1.636.10^{-7}$

TABLE 2. Mean number of customers in the orbit $\overline{N}_o$ with different methods.

		Gauss	BICG	BICGSTAB	BICGS	CGNR
dim $A = 52$	$\overline{N}_o$	0.27201425	0.27201425	0.27201424	0.27201425	0.27201425
	max error	$1.42108547.10^{-14}$	$1.31042519.10^{-8}$	$3.03612995.10^{-8}$	$3.05643510.10^{-10}$	$1.73245946.10^{-8}$
$L = 3$	Iteration	–	50	35	41	79
	time	0.081	0.686	0.135	0.385	0.224
cond$(A) = 2.16762645.10^2$	Norm final	–	$1.610.10^{-7}$	$1.688.10^{-7}$	$4.507.10^{-7}$	$5.816.10^{-7}$
dim $A = 78$	$\overline{N}_o$	0.27597878	0.27597878	0.27597877	0.27597878	0.27597878
	max error	$5.68.10^{-14}$	$3.96.10^{-8}$	$1.75.10^{-8}$	$2.37.10^{-8}$	$2.45.10^{-8}$
$L = 5$	Iteration	–	60	38	45	124
	time	0.116	2.882	0.302	0.453	0.473
cond$(A) = 3.23163797.10^2$	Norm final	–	$1.008.10^{-7}$	$2.096.10^{-7}$	$3.125.10^{-5}$	$2.807.10^{-7}$
dim $A = 104$	$\overline{N}_o$	0.27730483	0.27730484	0.27730483	0.27730483	0.27730483
	max error	$9.59.10^{-14}$	$3.46.10^{-8}$	$2.61.10^{-8}$	$6.97.10^{-9}$	$2.82.10^{-8}$
$L = 7$	Iteration	–	64	41	50	165
	time	0.171	0.218	0.233	0.428	0.668
cond$(A) = 4.40963672.10^2$	Norm final	–	$7.060.10^{-7}$	$2.229.10^{-7}$	$1.281.10^{-7}$	$3.415.10^{-7}$
dim $A = 130$	$\overline{N}_o$	0.27772422	0.27772422	0.27772422	0.27772422	0.27772422
	max error	$1.13.10^{-13}$	$3.52.10^{-8}$	$7.17.10^{-9}$	$3.17.10^{-8}$	$1.29.10^{-8}$
$L = 9$	Iteration	–	72	45	52	206
	time	0.256	0.445	0.202	0.535	0.810
cond$(A) = 5.699423.10^2$	Norm final	–	$4.193.10^{-7}$	$7.570.10^{-7}$	$1.008.10^{-6}$	$2.978.10^{-7}$
dim $A = 143$	$\overline{N}_o$	0.27780651	0.27780651	0.27780652	0.27780651	0.27780651
	max error	$1.13.10^{-13}$	$5.76.10^{-8}$	$3.40.10^{-8}$	$1.25.10^{-8}$	$1.04.10^{-8}$
$L = 10$	Iteration	–	76	48	56	227
	time	0.341	1.430	0.304	0.767	1.142
cond$(A) = 6.38853813.10^2$	Norm final	–	$3.780.10^{-7}$	$2.716.10^{-7}$	$4.407.10^{-6}$	$1.810.10^{-7}$
dim $A = 156$	$\overline{N}_o$	0.27785138	0.27785138	0.27785139	0.27785138	0.27785138
	max error	$1.98.10^{-13}$	$1.54.10^{-8}$	$2.66.10^{-8}$	$3.37.10^{-9}$	$1.23.10^{-8}$
$L = 11$	Iteration	–	81	46	59	248
	time	0.409	1.099	1.776	0.712	1.328
cond$(A) = 7.10838072.10^2$	Norm final	–	$6.289.10^{-7}$	$2.185.10^{-7}$	$1.727.10^{-7}$	$1.625.10^{-7}$

which leads to a time resolution almost similar. The method of the normal equation requires 1.5 times longer and CGS squared method requires about two times longer.

Note that in the previous results, matrices were represented as a solid shape for iterative methods. However, MATLAB provides a sparse representation of these matrices which leads to reduce by a factor of about 10 times the calculation of these methods.

TABLE 3. Mean number of customers in the service and waiting areas $\overline{C}$ with different methods.

		Gauss	BICG	BICGSTAB	BICGS	CGNR
dim $A = 52$	$\overline{C}$	7.22378737	7.22378737	7.22378738	7.22378737	7.22378737
	max error	$7.10.10^{-14}$	$1.79.10^{-8}$	$2.32.10^{-8}$	$6.18.10^{-8}$	$3.35.10^{-8}$
$L = 3$	Iteration	–	53	36	41	81
	time	0.090	0.164	0.165	0.338	0.239
cond$(A) = 2.16762645.10^2$	Norm final	–	$2.132.10^{-7}$	$3.325.10^{-7}$	$1.928.10^{-7}$	$1.182.10^{-7}$
dim $A = 78$	$\overline{C}$	7.22332619	7.22332619	7.22332620	7.22332619	7.22332619
	max error	$7.10.10^{-14}$	$2.57.10^{-8}$	$9.02.10^{-9}$	$2.61.10^{-8}$	$2.85.10^{-8}$
$L = 5$	Iteration	–	60	39	53	125
	time	0.522	0.357	0.195	0.579	0.597
cond$(A) = 3.23163797.10^2$	Norm final	–	$1.123.10^{-7}$	$1.298.10^{-7}$	$2.975.10^{-5}$	$1.261.10^{-7}$
dim $A = 104$	$\overline{C}$	7.22322004	7.22322004	7.22322004	7.22322004	7.22322004
	max error	$5.68.10^{-14}$	$3.73.10^{-8}$	$2.97.10^{-8}$	$2.03.10^{-8}$	$1.46.10^{-8}$
$L = 7$	Iteration	–	73	40	56	166
	time	0.621	0.296	0.169	0.859	0.624
cond$(A) = 4.40963672.10^2$	Norm final	–	$2.535.10^{-6}$	$1.511.10^{-6}$	$6.189.10^{-7}$	$1.403.10^{-7}$
dim $A = 130$	$\overline{C}$	7.22319659	7.22319659	7.22319659	7.22319659	7.22319659
	max error	$8.52.10^{-14}$	$2.94.10^{-8}$	$1.14.10^{-8}$	$1.98.10^{-8}$	$1.10.10^{-8}$
$L = 9$	Iteration	–	72	49	61	207
	time	0.984	1.406	1.213	0.522	1.295
cond$(A) = 5.699423.10^2$	Norm final	–	$2.056.10^{-7}$	$7.875.10^{-6}$	$8.615.10^{-7}$	$1.303.10^{-7}$
dim $A = 143$	$\overline{C}$	7.22319313	7.22319313	7.22319314	7.22319313	7.22319313
	max error	$1.13.10^{-13}$	$3.16.10^{-8}$	$3.63.10^{-8}$	$5.37.10^{-8}$	$1.23.10^{-8}$
$L = 10$	Iteration	–	76	51	56	227
	time	1.235	0.311	0.240	0.638	1.071
cond$(A) = 6.38853813.10^2$	Norm final	–	$2.278.10^{-6}$	$1.056.10^{-7}$	$2.984.10^{-4}$	$2.179.10^{-7}$
dim $A = 156$	$\overline{C}$	7.22319153	7.22319153	7.22319153	7.22319153	7.22319153
	max error	$7.10.10^{-14}$	$2.21.10^{-8}$	$2.76.10^{-8}$	$1.30.10^{-7}$	$2.38.10^{-8}$
$L = 11$	Iteration	–	80	48	70	247
	time	1.387	0.644	0.561	0.657	1.160
cond$(A) = 7.10838072.10^2$	Norm final	–	$4.497.10^{-5}$	$1.382.10^{-7}$	$1.172.10^{-7}$	$3.493.10^{-7}$

We have also calculated the blocking probability for $\lambda = 14$, $K = 12$, $C = 8$, $\mu = 1$, $\nu = 1$, probability of deleting one customer from the service area $q_1 = 0.2$, $\theta = 1$, $\gamma = 2$, $n = 2$ (so we have that the traffic intensity $\rho = 0.875 < 1$). From Table 4, it is possible to deduce the same conclusions as previously. We note that for numerical experimentations, the truncation level L was chosen to be greater than the degree n of the polynomial used as extrapolating function. Moreover, in the case of $\rho = 0.5$, the numerical results have a little change from $L = 3$ (see Tabs. 1, 2, 3); whereas when the traffic intensity is high ($\rho = 0.875$), the same tensency is observed from $L = 20$ (see Tab. 4).

In conclusion, it is very difficult to choose an efficient method for the solution of the algebraic system to solve. When the size of the matrix is large, the Gauss's method should be avoided. The method which works well on average is the BiCG method. The nice thing with BiCG is that it is simple and easy to code; moreover the corresponding storage is low. The disadvantage of BiCG is that sometimes the convergence is erratic. In this respect BiCGSTAB often gives better results.

To be sure, it is best, as we have done, using several methods and to compare the obtained results in order to verify the numerical stability of the computed results. As previously said the BiCGSTAB method seems to give very good numerical results.

Now we study the effects of some pertinent parameters characterizing the considered system, such as retrial rate θ and deleting rate η_1. The numerical results are obtained by applying the Value Extrapolation method, where the algebraic linear system $AX = B$ was resolved by BICGSTAB numerical method, and summarized in Table 5 and Figure 1. We have considered the cases where $\rho = 0.5$ and $\rho = 0.875$.

From Table 5, we can see, as is expected, that increasing the deleting rate η_1 results in a sensitive improvement of the numerical values of $\overline{N}_o$ and $\overline{C}$ and P_K. In Figure 1, we present the influence of the retrial rate θ on the system performance. We can observe that increasing θ ameliorates the mean number of customers in the orbit, $\overline{N}_o$, as well as that in the service and waiting areas $\overline{C}$, but it deteriorates the numerical values of the blocking probability, P_K.

Finally, we conclude that Value Extrapolation approach can be successfully used to find the solution for the considered multiserver queueing model, in particular when the system is under heavy traffic.

Table 4. Blocking probability P_K for $\rho = 0.875$ with different methods.

		Gauss	BICG	BICGSTAB	BICGS	CGNR
dim $A = 52$	P_K	0.38290061	0.38290061	0.38290061	0.38290061	0.38290061
	max error	$3.55.10^{-15}$	$5.53.10^{-8}$	$7.36.10^{-9}$	$3.13.10^{-8}$	$2.73.10^{-8}$
$L = 3$	Iteration	–	52	34	38	78
	time 0.328	0.662	0.131	0.091	0.079	
cond(A) = $1.17227318.10^3$	Norm final	–	$1.915.10^{-7}$	$3.036.10^{-7}$	$2.075.10^{-7}$	$1.220.10^{-7}$
dim $A = 78$	P_K	0.40008909	0.40008909	0.40008909	0.40008909	0.40008909
	max error	$3.55.10^{-15}$	$3.96.10^{-8}$	$1.96.10^{-8}$	$1.47.10^{-8}$	$3.45.10^{-8}$
$L = 5$	Iteration	–	61	36	51	118
	time	0.511	0.472	0.155	0.145	0.239
cond(A) = $1.44079784.10^3$	Norm final	–	$1.059.10^{-7}$	$1.597.10^{-6}$	$3.556.10^{-7}$	$2.853.10^{-7}$
dim $A = 104$	P_K	0.41359627	0.41359627	0.41359627	0.41359627	0.41359627
	max error	$7.10.10^{-15}$	$2.70.10^{-8}$	$2.75.10^{-8}$	$3.86.10^{-8}$	$1.55.10^{-8}$
$L = 7$	Iteration	–	73	38	54	157
	time	0.921	1.320	0.199	0.185	0.264
cond(A) = $1.75185380.10^3$	Norm final	–	$3.353.10^{-6}$	$8.283.10^{-7}$	$2.338.10^{-7}$	$2.402.10^{-7}$
dim A= 130	P_K	0.42413493	0.42413493	0.42413493	0.42413493	0.42413493
	max error	$1.06.10^{-14}$	$3.35.10^{-8}$	$1.79.10^{-8}$	$3.15.10^{-8}$	$1.47.10^{-8}$
$L = 9$	Iteration	–	77	44	60	195
	time	0.957	0.654	0.315	0.207	0.509
cond(A) = $2.07888660.10^3$	Norm final	–	$3.431.10^{-7}$	$5.758.10^{-7}$	$3.711.10^{-5}$	$2.603.10^{-7}$
dim $A = 143$	P_K	0.42848557	0.42848557	0.42848557	0.42848557	0.42848557
	max error	$1.06.10^{-14}$	$6.14.10^{-9}$	$1.01.10^{-8}$	$3.25.10^{-8}$	$2.15.10^{-8}$
$L = 10$	Iteration	–	82	50	62	216
	time	1.204	0.731	0.278	0.315	0.497
cond(A) = $2.24511964.10^3$	Norm final	–	$4.851.10^{-7}$	$6.578.10^{-7}$	$7.334.10^{-6}$	$1.500.10^{-7}$
dim $A = 208$	P_K	0.44351337	0.44351337	0.44351337	0.44351337	
	max error	$2.84.10^{-14}$	$4.47.10^{-8}$	$1.49.10^{-8}$	$1.45.10^{-8}$	$1.68.10^{-8}$
$L = 15$	Iteration	–	94	53	73	313
	time	3.353	1.430	1.308	0.671	1.822
cond(A) = $3.07463557.10^3$	Norm final	–	$2.151.10^{-6}$	$4.063.10^{-7}$	$2.712.10^{-7}$	$1.296.10^{-7}$
dim $A = 273$	P_K	0.45132744	0.45132744	0.45132744	0.45132744	0.45132744
	max error	$4.26.10^{-14}$	$2.68.10^{-8}$	$1.72.10^{-8}$	$1.49.10^{-8}$	$1.30.10^{-8}$
$L = 20$	Iteration	–	122	63	84	407
	time	6.543	1.828	0.883	0.894	3.334
cond(A) = $3.86199503.10^3$	Norm final	–	$1.785.10^{-7}$	$3.803.10^{-7}$	$7.895.10^{-7}$	$1.071.10^{-7}$
dim $A = 338$	P_K	0.45535911	0.45535911	0.45535911	0.45535911	0.45535911
	max error	$2.84.10^{-14}$	$4.03.10^{-8}$	$1.68.10^{-8}$	$1.38.10^{-7}$	$1.88.10^{-8}$
$L = 25$	Iteration	–	151	74	111	502
	time	12.624	2.731	1.056	2.017	5.702
cond(A) = $4.58704316.10^3$	Norm final	–	$1.180.10^{-7}$	$1.333.10^{-6}$	$3.072.10^{-6}$	$1.480.10^{-7}$
dim $A = 403$	P_K	0.45743145	0.45743145	0.45743145	0.45743145	0.45743145
	max error	$5.68.10^{-14}$	$3.51.10^{-8}$	$2.34.10^{-8}$	$2.04.10^{-7}$	$1.34.10^{-8}$
$L = 30$	Iteration	–	187	85	151	606
	time	19.155	6.685	1.575	2.712	8.223
cond(A) = $5.26662753.10^3$	Norm final	–	$4.254.10^{-6}$	$6.253.10^{-7}$	$1.711.10^{-6}$	$1.189.10^{-7}$

Table 5. Influence of the deleting rate η_1 ($\theta = 10$).

ρ	P.M $\ddots$ η_1	0.2	0.4	0.6	0.8
	P_K	0.03720550	0.03407397	0.03113462	0.02838345
$\rho = 0.5$	$\overline{N}_o$	0.07300968	0.06666848	0.06074052	0.05521398
	$\overline{C}$	7.20986589	7.05917957	6.90568391	6.74955662
	P_K	0.54650889	0.53587380	0.52501298	0.51393124
$\rho = 0.875$	$\overline{N}_o$	5.59698010	5.44014686	5.28335946	5.12670179
	$\overline{C}$	10.91069305	10.85756761	10.80163107	10.74276398

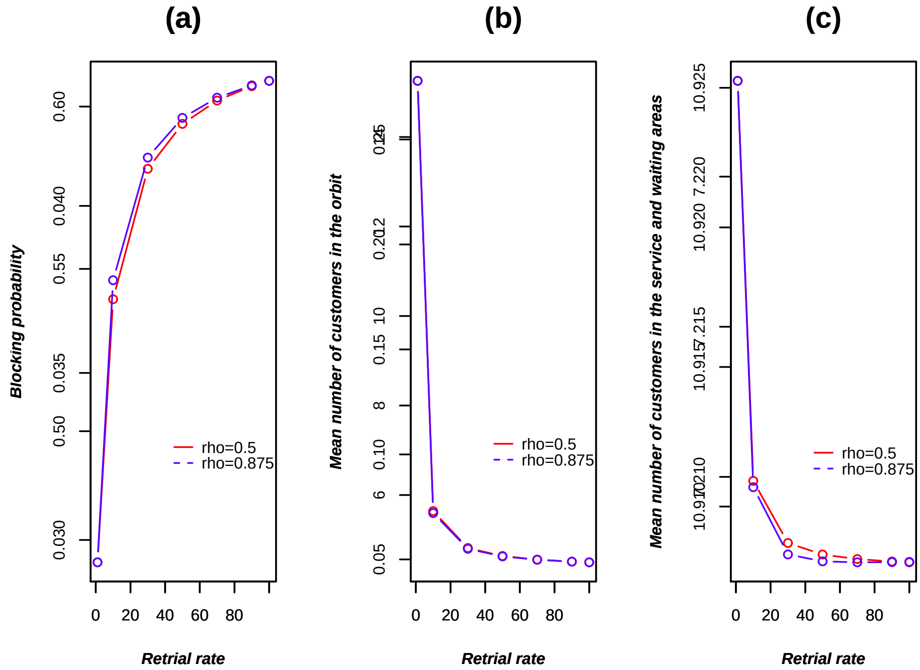

FIGURE 1. Influence of the retrial rate θ ($\eta_1 = 0.2$).

Acknowledgements. The authors thank very much Professor Philippe Marthon for his interest to this work and for his constructive criticism and formulated suggestions.

References

[1] V.M. Abramov, Analysis of multiserver retrial queueing system: a martingale approach and an algorithm of solution. *Ann. Oper.* **141** (2006) 19–52.

[2] B. Almási, T. Bérczes, A. Kuki, J. Sztrik and J. Wang, Performance modeling of finite-source cognitive radio networks. *Acta Cybernetica* **22** (2016) 617–631.

[3] J.R. Artalejo, G-networks: a versatile approach for work removal in queueing systems. *EJOR* **126** (2000) 233–249.

[4] J.R. Artalejo, Retrial queues: an algorithmic approach. *J. Egyptian Math. Soc.* **17** (2009) 83–101.

[5] J.R. Artalejo and A. Gomez−Corral, *Retrial Queueing Systems: A Comput. Approach.* Springer (2008).

[6] J.R. Artalejo, A. Gomez-Corral and M.F. Neuts, Analysis of multiserver queues with constant retrial rate. *Eur. J. Oper. Res.* **135** (2001) 569–581.

[7] J.R. Artalejo and V. Pla, On the impact of customer balking, impatience and retrials in telecommunication systems. *Comput. Math. Appl.* **57** (2009) 217–229.

[8] J.R. Artalejo and M. Pozo, Numerical calculation of the stationary distribution of the main multiserver retrial queue. *Ann. Oper. Res.* **116** (2002) 41–56.

[9] K. Avrachenkov and U. Yechiali, Retrial networks with finite buffers and their application to internet data traffic. *Probab. Eng. Inform. Sci.* **22** (2008) 519–536.

[10] T.V. Do, An efficient computation algorithm for a multiserver feedback retrial queue with a large queueing capacity. *Appl. Math. Model.* **34** (2010) 2272–2278.

[11] T.V. Do, Solution for a retrial queueing problem in cellular networks with the Fractional Guard Channel Policy. *Math. Comput. Model.* **53** (2011) 2059–2066.

[12] T.V. Do, N.H. Do and J. Zhang, An enhanced algorithm to solve multiserver retrial queueing systems with impatient customers. *Comput. Industrial Eng.* **65** (2013) 719–728.

[13] M.J. Domenech−Benloch, J.M. Gimenez−Guzman, V. Pla, J. Martinez-Bauset and V. Casares-Giner, Generalized truncated methods for an efficient solution of retrial systems. *Math. Probl. Eng.* **2008** (2008) 183089.

[14] M.J. Domenech−Benloch, J.M. Gimenez−Guzman, V. Pla, J. Martinez−Bauset and V. Casares-Giner, On the convergence of truncated processes of multiserver retrial queues. *Math. Probl. Eng.* **2010** (2010) 580349.

[15] G.I. Falin, A survey of retrial queues. *Queueing Syst.* **7** (1990) 127–168.

[16] G.I. Falin and J.G.C. Templeton, Retrial Queues. Chapman and Hall (1997).

[17] E. Gelenbe, Random neural networks with negative and positive signals and product form solution. *Neural Comput.* **1** (1989) 502–510.

[18] E. Gelenbe, P. Glynn and K. Singman, Queues with negative arrivals. *J. Appl. Probab.* **28** (1991) 245–250.

[19] J.M. Gimenez−Guzman, M.J. Domenech−Belloch, V. Pla, V. Casares−Giner and J. Martinez-Bauset, Value extrapolation technique to solve retrial queues: a comparative perspective. *ETRI J.* **30** (2008) 492–494.

[20] T. Hanschke, Explicit formulas for the characteristics of the M/M/2/2 queue with repeated attempts. *J. App. Probab.* **24** (1987) 486–494.

[21] O. Ibe, Markov Processess for Stochastic Modeling. Elsevier, Elsevier Academic Press (2009).

[22] D.L. Isaacson and R.W. Madsen, *Markov Chains, Theory and Applications*. John Wiley Sons (1976).

[23] B. Krishna Kumar and J. Raja, On multiserver feedback retrial queues with balking and control retrial rate. *Ann. Oper. Res.* **141** (2006) 211–232.

[24] B. Krishna Kumar, R. Rukmani and V. Thangaraj, On multiserver feedback retrial queue with finite buffer. *Appl. Math. Model.* **33** (2009) 2062–2083.

[25] P. Lascaux, R. Théodor, Analyse numérique matricielle appliquée *à* l'art de l'ingénieur. Masson Tomes 1 and 2 (1986–1987).

[26] J. Leino and J. Virtamo, An approximate method for calculating performance measures of Markov processes. *Proc. Valuetools* (2006).

[27] V. A. Malyshev and M.V. Menshikov, Ergodicity, continuity and analyticity of countable Markov chains. *Proc. Moscow Math. Soc.* **39** (1979) 3–48.

[28] G. Meurant, Computer solution of large linear systems. North Holland (1999).

[29] M.F. Neuts and B.M. Rao, Numerical investigation of a multiserver retrial model. *Queueing Syst.* **7** (1990) 169-190.

[30] A.G. Pakes, Some conditions for ergodicity and recurrence of Markov chains. *Operat. Res.* **17** (1969) 1058–1061.

[31] T. Phung−Duc, Multiserver retrial queues with two types of non persistent customers. *Asia-Pacific J. Oper. Res.* **31** (2014) 1440009.

[32] T. Phung−Duc, Asymptotic analysis for markovian queues with two types of non persistent retrial customers. *Appl. Math. Comput.* **265** (2015) 768–784.

[33] T. Phung−Duc and K. Kawanishi, Multiserver retrial queues with after-call work. *Numer. Algebra, Control Optimiz.* **1** (2011) 639–656.

[34] T. Phung−Duc and K. Kawanishi, Performance analysis of call centers with abandonment, retrial and after-call work. *Performance Evaluation* **80** (2014) 43–62.

[35] M.L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming. *Wiley Series in Probability and Statistics*. Wiley (2008).

[36] Y. Saad, Iterative methods for sparse linear systems. PWS Publishing Company (1996).

[37] S.N. Stepanov, Markov models with retrials: the calculation of stationary performance measures based on the concept of truncation. *Math. Comput. Model.* **30** (1999) 207–228.

[38] R.L. Tweedie, Sufficient conditions for regularity, recurrence and ergodicity of Markov processes. *Math. Proc. Cambridge Philosoph. Soc.* **78** (1975) 125–136.

[39] J. Wu and Z. Liang, A single server retrial G-queue with priority and unreliable server under Bernoulli vacation schedule. *Comput. Industrial Eng.* **64** (2013) 84–93.

[40] D.Y. Yang, J.C. Ke and C.H. Wu, The multi-server retrial system with Bernoulli feedback and starting failures. *Inter. J. Comput. Math.* **92** (2015) 954–969.