An Enhanced Algorithm to Solve Multiserver Retrial Queueing Systems with Impatient Customers

Tien Van Do [a,*] Nam H. Do [b] Jie Zhang [c]

[a] *MTA-BME Information Systems Research Group, Department of Networked Systems and Services, Budapest University of Technology and Economics, H-1117, Magyar tudósok körútja 2., Budapest, Hungary.*

[b] *Inter-University Centre for Telecommunications and Informatics, Budapest University of Technology and Economics, 4028 Debrecen, Kassai út 26., Hungary*

[c] *Communications Group, the Department of Electronic and Electrical Engineering, University of Sheffield, Mappin Street, Sheffield, S1 3JD UK*

Abstract

The homogenization of the state space for solving retrial queues refers to an approach where the performance of the M/M/c retrial queue with impatient customers and c servers is approximated with a retrial queue with a maximum retrial rate restricted beyond a given number of users in the orbit. As a consequence, the stationary distribution can be obtained by the matrix-geometric method, which requires the computation of the rate matrix. In this paper, we revisit an approach based on the homogenization of the state space. We provide the exact expression for the conditional mean number of customers based on the computation of the rate matrix R with the time complexity of $O(c)$. We develop simplified equations for the memory-efficient implementation of the computation of the performance measures. We construct an efficient algorithm for the stationary distribution with the determination of a threshold that allows the computation of performance measures with a specific accuracy.

Keywords: retrial queues, matrix-geometric method, spectral expansion, efficient algorithm

T. V. Do, N. H. Do, J. Zhang. An Enhanced Algorithm to Solve Multiserver Retrial Queueing Systems with Impatient Customers. *Computers & Industrial Engineering*, DOI:10.1016/j.cie.2013.04.008, 2013

---

* Corresponding author
Email addresses: do@hit.bme.hu (Tien Van Do), dohoai@hit.bme.hu (Nam H.

1 Introduction

Retrial queues have been used to take into account a phenomenon in modern information and telecommunication systems that blocked customers may re-request for service after a certain timeout [1–10]. In retrial queues a client who does not receive the allocation of a server joins the orbit and later initiates a request for service. The M/M/c retrial queue has been analyzed by many researchers because the stationary distribution when the number of servers is larger than two can be only obtained using approximate techniques [1,6–8,11,12].

Falin [13] presented necessary and sufficient conditions for ergodicity of the retrial queues $M/M/c$. A well-known approximation is based on the truncation of the state space at a sufficiently large level related to the number of customers in the orbit [13]. Another approximation based on the homogenization of the model was pioneered by Neuts and Rao [14], where the $M/M/c$ retrial queue is approximated by the multiserver retrial queue with the total retrial rate that does not depend on the number of clients in the orbit as long as the orbit contains the number of clients greater than the specified value N. Note that the discussion for the choice of N is presented in the recent book by Artalejo and Gómez-Corral on retrial queues [8]. With this assumption, the stationary probabilities of the M/M/c retrial queue can be estimated by any algorithm [15–19] based on the matrix-geometric method (MGM).

Recently, Domenech-Benlloch et al. [20] considered a multiserver retrial queue with the impatient phenomenon of customers waiting in the orbit. They proposed two different generalized truncated methods (called HM1 and HM2) based on the homogenization of the state space beyond a given number of users in the retrial orbit. The steady-state probabilities of the multiserver retrial queue with impatient customers are approximated with a modified retrial queue where the retrial rate beyond a certain level only depends on the conditional mean value of the number of customers in the orbit. Domenech-Benlloch et al. [20] also compared their methods with other well-known algorithms that belong to different categories [11] (approximations, finite truncated methods, generalized truncated methods). The authors [20] showed that the proposed HM2 method outperforms previous approaches from the aspect of accuracy at the price of increasing computation cost.

Based on the HM2 algorithm of Domenech-Benlloch et al. [20], our contributions allow an efficient computation for the stationary distribution and the performance measures. First, we revisit an approach based on the homogenization of the state space and provide an efficient method with the time complexity of only $O(c)$ to compute the rate matrix R. The method is based

Do).

on a property that the characteristic matrix polynomial has only a single non-zero eigenvalue and this single non-zero eigenvalue can be computed using the bisection method. Second, we derive an exact expression for the conditional mean number of customers. Third, we develop simplified equations that allow the memory-efficient implementation of the computation of the performance measures. Fourth, we construct an efficient computation for the stationary distribution with the determination of a threshold, which guarantees a specific accuracy for the computation of performance measures.

The rest of this paper is organized as follows. In Section 2 we summarize the considered queueing model with impatient customers. In Section 3 we present our new results that serve as the foundations of the computation. In Section 4 we provide some numerical results to illustrate the efficiency of our algorithm. Finally, Section 5 concludes our paper.

2 A Retrial Queueing Model with Impatient Customers

We consider a retrial queueing model with c homogenous servers and impatient customers. Inter-arrival times of customers are exponentially distributed with parameter λ. Holding times are exponentially distributed with parameter μ. Random variable $\gimel(t)$ represents the number of occupied servers at time t, hence $0 \leq \gimel(t) \leq c$ holds. A client joins the orbit in order to wait and retry upon when $\gimel(t) = c$. Let $\daleth(t)$ be the number of clients in the orbit waiting for retrial at time t. Each customer retries with rate μ_r. Hence, the total effective retrial rate, when $\daleth(t) = j$, is $j\mu_r$. A retrying customer either leaves the queue with probability P_{im} if all servers are busy upon the retrial or rejoins the orbit with probability $1 - P_{im}$. Note that a time between subsequent retrials of a specific user follows the exponential distribution with parameter μ_r.

This system can be represented by two-dimensional continuous-time Markov chain (CTMC) $Y = \{\gimel(t), \daleth(t)\}$ with state space $\{0, 1, \ldots, c\} \times \{0, 1, \ldots\}$. Let the steady-state probabilities of CTMC Y be denoted by $\pi_{i,j} = \lim_{t\to\infty} \Pr(\gimel(t) = i, \daleth(t) = j)$. Define the row vector $\mathbf{v}_j = [\pi_{0,j}, \ldots, \pi_{c,j}]$.

2.1 Notations

CTMC Y is driven by the following transitions.

(a) $A_j(i, k)$ denotes the transition rate from state (i, j) to state (k, j) $(0 \leq i, k \leq c\,;\; j = 0, 1, \ldots)$, which is caused by either the arrival of a customer (when $i < c$) or the leaving of a client after the expiry of a holding time.

Matrix A_j is of size $(c+1) \times (c+1)$ with elements $A_j(i,k)$. Since A_j is j-independent, it can be written as $A_j = A$. The nonzero elements of A_j are $A_j(i, i-1) = i\mu$ for $i = 1, \ldots, c+1$, and $A_j(i, i+1) = \lambda$ for $i = 0, \ldots, c$. Because A_j is j-independent, it can be written as

$$A_j = A = \begin{bmatrix} 0 & \lambda & 0 & \ldots & 0 & 0 & 0 \\ \mu & 0 & \lambda & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & & \ldots & (c-1)\mu & 0 & \lambda \\ 0 & 0 & & \ldots & 0 & c\mu & 0 \end{bmatrix}, \ \forall j \geq 0 \, .$$

(b) $B_j(i,k)$ represents the one-step upward transition rate from state (i,j) to state $(k, j+1)$ $(0 \leq i, k \leq c \, ; \ j = 0, 1, \ldots)$, which is caused by the arrival of a request when all servers are busy (i.e., when $i = c$), thus increasing $\daleth(t)$ by 1. Matrix B_j (B, since it is j-independent) is of size $(c+1) \times (c+1)$ with elements $B_j(i,k)$. The only nonzero element of B_j is $B_j(c,c) = \lambda$. Thus, we get

$$B_j = B = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & & \ldots & 0 & 0 & 0 \\ 0 & 0 & & \ldots & 0 & 0 & \lambda \end{bmatrix}, \ \forall j \geq 0 \, .$$

(c) $C_j(i,k)$ is the transition rate from state (i,j) to state $(k, j-1)$ $(0 \leq i, k \leq c \, ; \ j = 1, 2, \ldots)$, which is due to the successful retrial of a request from the orbit. Matrix C_j is of size $(c+1) \times (c+1)$ with its elements $C_j(i,k)$. The nonzero elements of C_j $(j \geq 1)$ are $C_j(i, i+1) = j\mu_r$ for $i = 0, \ldots, c$ and $C_j(c,c) = j\mu_r P_{im}$. Matrix C_j $(\forall j \geq 1)$ with elements $C_j(i,k)$ is written as

$$C_j = \begin{bmatrix} 0 & j\mu_r & 0 & & \ldots & 0 & 0 & 0 \\ 0 & 0 & j\mu_r & & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & & & \ldots & 0 & 0 & j\mu_r \\ 0 & 0 & & & \ldots & 0 & 0 & j\mu_r P_{im} \end{bmatrix}, \ \forall j \geq 1.$$

Note that $C_0 = 0$ by definition.

Let D^A, D^C and D^{C_j}, $j \geq 1$ denote diagonal matrices with the diagonal elements $D^A(i,i) = \sum_{k=0}^{c} A(i,k)$, $D^C(i,i) = \sum_{k=0}^{c} C(i,k)$ and $D^{C_j}(i,i) = \sum_{k=0}^{c} C_j(i,k)$ for $i = 0, \ldots, c$. The balance equations, which equate the probability fluxes from and to the states of CTMC Y, and the normalization equation pertaining to CTMC Y can be written as follows (see [3,8]):

$$\mathbf{v}_0 Q_1^{(0)} + \mathbf{v}_1 Q_2^{(1)} = \mathbf{0}, \tag{1}$$

$$\mathbf{v}_{j-1} Q_0^{(j-1)} + \mathbf{v}_j Q_1^{(j)} + \mathbf{v}_{j+1} Q_2^{(j+1)} = \mathbf{0} \ \ (j \geq 1), \tag{2}$$

$$\sum_{j=0}^{\infty} \mathbf{v}_j \mathbf{e}^T = 1.0 \ \ (normalization),$$

where $Q_0^{(j)} = B, j \geq 0$; $Q_1^{(j)} = A - D^A - B - D^{C_j}, j \geq 0$; $Q_2^{(j)} = C_j, \ j \geq 1$ and $\mathbf{e}$ is the row vector of size $c+1$ with each element equal to unity.

Using the similar argument as in [3–5,8], the infinitesimal generator matrix [15,16,21] of Y, that satisfies $[\mathbf{v}_0, \mathbf{v}_1, \ldots]Q_Y = \mathbf{0}$, can be constructed from equations (1) and (2) as follows:

$$Q_Y = \begin{bmatrix}
Q_1^{(0)} & Q_0^{(0)} & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
Q_2^{(1)} & Q_1^{(1)} & Q_0^{(1)} & 0 & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & Q_2^{(2)} & Q_1^{(2)} & Q_0^{(2)} & 0 & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & Q_2^{(3)} & Q_1^{(3)} & Q_0^{(3)} & 0 & \cdots & \cdots & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\cdots & \cdots & \cdots & Q_2^{(j)} & Q_1^{(j)} & Q_0^{(j)} & \cdots & \cdots & \\
\cdots & \cdots & \cdots & \cdots & Q_2^{(j+1)} & Q_1^{(j+1)} & Q_0^{(j+1)} & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & Q_2^{(j+2)} & Q_1^{(j+2)} & Q_0^{(j+2)} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{bmatrix}.$$

It is clear that Q_Y is a block tridiagonal matrix with

- $Q_Y(j, j+1) = Q_0^{(j)}, j \geq 0$, in the upper diagonal,
- $Q_Y(j, j) = Q_1^{(j)}, j \geq 0$ in the main diagonal,
- $Q_Y(j, j-1) = Q_2^{(j)}, \ j \geq 1$ in the lower diagonal.

2.2 An Approximation

Domenech-Benlloch et al. [20] suggested that the M/M/c retrial queue with impatient customers can be approximated by the solution of the modified multiserver retrial queue with the retrial rate

$$\mu_r(j) = \begin{cases} j\mu_r & \text{if } j < N \\ M(N)\mu_r & \text{if } j \geq N \end{cases},$$

where $M(N) = E[J|J \geq N]$ is the conditional mean number of customers.

As a consequence, the modified multiserver retrial queue is described by a CTMC $Z = \{\gimel_Z(t), \daleth_Z(t)\}$ with state space $\{0, 1, \ldots, c\} \times \{0, 1, \ldots\}$, where $\gimel_Z(t)$ represents the number of occupied servers at time t and $\daleth_Z(t)$ is the number of clients in the orbit waiting for retrial at time t. The steady-state probabilities of CTMC Z are denoted by $\widetilde{\pi}_{i,j} = \lim_{t\to\infty} \Pr(\gimel_Z(t) = i, \daleth_Z(t) = j)$, $j \geq 0, 0 \leq i \leq c$, and the row vectors $\widetilde{\mathbf{v}}_j = [\widetilde{\pi}_{0,j}, \ldots, \widetilde{\pi}_{c,j}]$, $j \geq 0$.

We define the transition rate matrices associated with CTMC Z as $\widetilde{A}_j$, $\widetilde{A}$, $\widetilde{B}_j$, $\widetilde{B}$, $\widetilde{C}_j$ and $\widetilde{C}$ for $j \geq 0$. Note that we have $\widetilde{A}_j = \widetilde{A} = A$ and $\widetilde{B}_j = \widetilde{B} = B$ for $j \geq 0$. Furthermore, $\widetilde{C}_j = C_j$ for $0 \leq j < N$ and

$$\widetilde{C}_j = \widetilde{C} = \begin{bmatrix} 0 & M(N)\mu_r & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & M(N)\mu_r & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & & \ldots & 0 & 0 & M(N)\mu_r \\ 0 & 0 & & \ldots & 0 & 0 & M(N)\mu_r P_{im} \end{bmatrix}, \quad \forall j \geq N.$$

For $j \geq N$, the balance equation of CTMC Z can be rewritten as

$$\widetilde{\mathbf{v}}_{j-1}\widetilde{Q}_0 + \widetilde{\mathbf{v}}_j\widetilde{Q}_1 + \widetilde{\mathbf{v}}_{j+1}\widetilde{Q}_2 = \mathbf{0} \ \ (j \geq N), \tag{3}$$

where $\widetilde{Q}_0 = \widetilde{B}$, $\widetilde{Q}_1 = \widetilde{A} - D^{\widetilde{A}} - \widetilde{B} - D^{\widetilde{C}}$, $\widetilde{Q}_2 = \widetilde{C}$.

The coefficient matrices in the difference equations (3) are j-independent. This leads to the following solution based on the MGM (see [16])

$$\widetilde{\mathbf{v}}_j = \widetilde{\mathbf{v}}_{N-1}R^{j-N+1} \ \ (j \geq N-1), \tag{4}$$

where R is the unique minimal nonnegative solution of the quadratic matrix equation $\widetilde{Q}_0 + R\widetilde{Q}_1 + R^2\widetilde{Q}_2 = 0$ (see [15,16]). After the computation of R, the rate matrix, the steady-state probabilities for states $0 \leq j \leq N-1$ can be determined by solving the balance equations pertaining to the levels $0 \leq j < N$ and the normalization equation.

Algorithm 1 The HM2 algorithm

$M_0(N) = N$
$k = 0$
repeat
  $k = k + 1$
  Compute R matrix based on the logarithmic reduction algorithm [16]
  Compute $M_k(N)$ using equation (5)
until $|M_k(N) - M_{k-1}(N)|/M_{k-1}(N) < \epsilon_M$
Solve for $\mathbf{v}_j$ for $j = 0, \ldots, N$

Because R and $\widetilde{\mathbf{v}}_N$ depend on $M(N)$, we can get the fixed-point iteration

$$M(N) = \frac{\sum_{j=N}^{\infty} j\widetilde{\mathbf{v}}_j\mathbf{e}}{\sum_{j=N}^{\infty} \widetilde{\mathbf{v}}_j\mathbf{e}} = \frac{\widetilde{\mathbf{v}}_N[R(\mathbf{I} - R)^{-1} + N\mathbf{I}](\mathbf{I} - R)^{-1}\mathbf{e}}{\widetilde{\mathbf{v}}_N(\mathbf{I} - R)^{-1}\mathbf{e}}, \tag{5}$$

where $\mathbf{I}$ is the identity matrix of size $(c+1)\times(c+1)$. Hence, Domenech-Benlloch et al. proposed Algorithm 1 (called HM2) in [20].

3 An Enhanced Algorithm

The stationary distribution of CTMC Y is approximated by the steady-state probabilities of CTMC Z. Therefore, we need to compute the following quantities associated with CTMC Z:

- the rate matrix R,
- the conditional mean number of customers $M(N)$,
- the steady-state probabilities for states $0 \le j \le N - 1$,
- the estimation of N.

Note that R can be computed by the original algorithm of the MGM [15] and further improved algorithms of MGM [16,18,19]. However, the time complexity of these algorithms is $O(c^3)$.

In what follows, we provide a method to compute the rate matrix (Theorem 1) in Section 3.1. We derive the exact and simplified formula for the computation of the conditional mean number of customers in the orbit (Corollary 1). As a consequence, we can compute the rate matrix R and the conditional mean number $M(N)$ of customers in a very efficient way. We provide a method to determine the steady-state probabilities for states $0 \leq j \leq N-1$ in Section 3.2. We provide the new formulae of performance measures and the relation between performance measures in Section 3.3. Next, we present our new result and our algorithm for the computation of the initial value of N in Section 3.4.

3.1 The computation of matrix R and $M(N)$

In Theorem 1 we prove that the characteristic matrix polynomial has only a single non-zero eigenvalue, and it can be computed using the bisection method. As a consequence, the rate matrix R has a special form and a method can be constructed to compute the rate matrix R with the computational complexity of $O(c)$. The property that the characteristic matrix polynomial has only a single non-zero eigenvalue allows the derivation of an exact equation for the conditional mean number of customers $M(N)$.

Theorem 1 *The rate matrix R has all rows of elements equal to zero except the last row $\mathbf{r} = [r_0, r_1, \ldots, r_c]$, where $r_c = x_c$ is the single eigenvalue of characteristic matrix polynomial $Q(x, M(N)) = \widetilde{Q}_0 + \widetilde{Q}_1 x + \widetilde{Q}_2 x^2$ in the interval $(0, 1)$ (the corresponding left-eigenvector is $\boldsymbol{\psi}_c = [\psi_{c,0}, \psi_{c,2}, \ldots, \psi_{c,c}]$ with $\psi_{c,c} = 1$) and $r_i = x_c \psi_{c,i}$ for $0 \leq i < c$. The computational complexity for r_c and $\boldsymbol{\psi}_c$ is $O(c)$.*

Proof. The steady-state probabilities of the CTMC Z are expressed as

$$\widetilde{\mathbf{v}}_j = \sum_{k=0}^{c} b_k \boldsymbol{\psi}_k x_k^{j-N+1} \quad (j \geq N-1), \tag{6}$$

where b_k are suitable coefficients to be determined using the balance equations pertaining to rows 0 to $N-1$ and the normalization equation, $(x_k, \boldsymbol{\psi}_k)$, $k = 0, \ldots, c$ are the left eigenvalue-eigenvector pairs of $Q(x, M(N)) = \widetilde{Q}_0 + \widetilde{Q}_1 x + \widetilde{Q}_2 x^2$ inside the unit circle. They satisfy, $\boldsymbol{\psi}_k Q(x_k, M(N)) = \mathbf{0}$; $det[Q(x_k, M(N))] = 0, \quad k = 0, \ldots, c$.

Since the $(c+1) \times (c+1)$ tri-diagonal matrix $Q(x, M(N))$ can be expressed

$$Q(x, M(N)) = \begin{bmatrix} q_{1,1}(x) & q_{1,2}(x) & 0 & \ldots & 0 & 0 \\ q_{2,1}(x) & q_{2,2}(x) & q_{2,3}(x) & \ldots & 0 & 0 \\ 0 & q_{3,2}(x) & q_{3,3}(x) & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & q_{c,c-1}(x) & q_{c,c}(x) & q_{c,c+1}(x) \\ 0 & 0 & \ldots & 0 & q_{c+1,c}(x) & q_{c+1,c+1}(x) \end{bmatrix},$$

where

$$\begin{aligned}
q_{1,1}(x) &= -(\lambda + M(N)\mu_r)x, \\
q_{i,i}(x) &= -(\lambda + M(N)\mu_r + (i-1)\mu)x \\
&\quad (i = 2, \ldots, c), \\
q_{c+1,c+1}(x) &= \lambda - (\lambda + c\mu + M(N)\mu_r P_{im})x \\
&\quad + M(N)\mu_r P_{im} x^2, \\
q_{i,i+1}(x) &= \lambda x + M(N)\mu_r x^2 \qquad (i = 1, \ldots, c), \\
q_{i+1,i}(x) &= \mu i x \qquad (i = 1, \ldots, c).
\end{aligned}$$

It is easy to verify that $Q(x, M(N))$ has c zero-eigenvalues. Let the null-eigenvalues be $x_0, \ldots, x_{c-1}$ with corresponding independent left-eigenvectors $\boldsymbol{\psi}_0 = [1, 0, \ldots, 0]$, $\boldsymbol{\psi}_2 = [0, 1, 0, \ldots, 0]$,$\ldots$,$\boldsymbol{\psi}_{c-1} = [0, 0, \ldots, 1, 0]$, respectively. As a consequence, $Q(x, M(N))$ should have a single non-zero eigenvalue x_c strictly inside the unit disk to ensure that the stationary distribution of CTMC $\widetilde{Y}$ exists.

Let $L(x, M(N))$ and $U(x, M(N))$ denote the component matrices in the LU decomposition of $Q(x, M(N)) = L(x, M(N))U(x, M(N))$ for any specific value x. Due to the tri-diagonal structure, the component matrices of the LU decomposition of $Q(x, M(N))$ can be written as follows

$$L(x, M(N)) =$$
$$\begin{bmatrix}
l_1(x, M(N)) & 0 & 0 & \ldots & 0 & 0 \\
\mu x & l_2(x, M(N)) & 0 & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \ldots & \mu(c-1)x & l_c(x, M(N)) & 0 \\
0 & 0 & \ldots & 0 & \mu c x & l_{c+1}(x, M(N))
\end{bmatrix},$$

$$U(x, M(N)) =$$
$$\begin{bmatrix}
1 & u_1(x, M(N)) & \ldots & 0 & 0 & 0 & 0 \\
0 & 1 & u_2(x, M(N)) & \ldots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & & \ldots & 0 & 1 & u_c(x, M(N)) \\
0 & 0 & & \ldots & 0 & 0 & 1
\end{bmatrix}.$$

By equating the corresponding elements of $Q(x, M(N))$ and $L(x, M(N)) \cdot U(x, M(N))$, and using some algebraic simplifications, we get

$$
\begin{aligned}
l_1(x, M(N)) &= q_{1,1}(x) = -(\lambda + M(N)\mu_r)x, && (7)\\
l_i(x, M(N)) + \mu(i-1)xu_{i-1}(x, M(N)) &= q_{i,i}(x),\\
&(i = 2, \ldots, c+1), && (8)\\
l_i(x, M(N))u_i(x, M(N)) &= \lambda x + M(N)\mu_r x^2,\\
&(i = 1, \ldots, c). && (9)
\end{aligned}
$$

Based on the elementary rules of matrix algebra, we obtain

$$
\begin{aligned}
Det[Q(x, M(N))] &= Det[L(x, M(N))]Det[U(x, M(N))]\\
&= \prod_{i=1}^{c+1} l_i(x, M(N)). && (10)
\end{aligned}
$$

From equations (7),(8) and (9), it can be verified that $l_i(x_c, M(N)) \neq 0$ $(1 \leq i \leq c)$. Hence, $Det[Q(x_c, M(N))] = 0$ (from equation (10)) gives rise to $l_{c+1}(x_c, M(N)) = 0$. This means x_c is the root of $l_{c+1}(x, M(N))$ in the interval $(0, 1)$. The bisection method [22] can be applied to find the root of $l_{c+1}(x, M(N))$ in the interval $(0, 1)$.

Since $(x_c, \boldsymbol{\psi}_c)$ are left eigenvalue-eigenvector pair, we can write

$$
\begin{aligned}
\boldsymbol{\psi}_c Q(x_c, M(N)) &= \mathbf{0},\\
\boldsymbol{\psi}_c L(x_c, M(N))U(x_c, M(N)) &= \mathbf{0},\\
\boldsymbol{\psi}_c L(x_c, M(N))U(x_c, M(N))U(x_c, M(N))^{-1} &=\\
\mathbf{0}U(x_c, M(N))^{-1}&,\\
\text{because } U(x_c, M(N)) \text{ is non-singular}&,\\
\boldsymbol{\psi}_c L(x_c, M(N)) &= \mathbf{0}.
\end{aligned} \tag{11}
$$

Expanding equation (11) we obtain the recursive relations $\psi_{c,i} = \frac{-(i+1)\mu x_c \psi_{c,i+1}}{l_{i+1}(x_c)}$ between $\psi_{c,i}$ and $\psi_{c,i+1}$, for $i = c-1, \ldots, 0$.

There are a number of eigenvectors corresponding to the same eigenvalue, but the ratio of the elements in these eigenvectors does not change. Applying this property, we can determine $\boldsymbol{\psi}_c = [\psi_{c,0}, \psi_{c,1}, \ldots, \psi_{c,c}]$ by setting $\psi_{c,c} = 1$ and using the above recursive relations and equations (7),(8), (9), to compute $\psi_{c,i}$ for $i = c-1, \ldots, 0$.

From (4) and (6) we get $R = \Psi^{-1} \cdot diag(0, 0, \ldots, 0, x_c) \cdot \Psi$, where $\Psi = [\psi_0, \psi_1, ..., \psi_c]^T$. Therefore, the rate matrix R has all rows of elements equal to zero except the last row $\mathbf{r} = [r_0, r_1, \ldots, r_c]$. Furthermore, we get $r_i = x_c\psi_{c,i}$ for $0 \leq i < c$ and $r_c = x_c$ after a simple algebraic step.

The complexity for computing r_c is $O(c)$ as the consequence of the following facts: (i) the last element $l_{c+1}(x, M(N))$ in the main diagonal of $L(x, M(N))$ can be determined after $c+1$ steps due to the tri-diagonal structure of $Q(x, M(N))$; (ii) the number of iterations in the bisection method for finding a root in the interval $(0, 1)$ to achieve the solution tolerance ϵ_r is $log_2(1/\epsilon_r)$ (see [22]). □

Corollary 1 *The conditional mean value $M(N) = E[J|J \geq N]$ of the number J of customers in the orbit under the condition $J \geq N$ can be expressed in the following closed-form:*

$$M(N) = N - 1 + \frac{1}{1 - r_c} = N + \frac{r_c}{1 - r_c}. \tag{12}$$

Proof. The consequence of Theorem 1 and (4) is

$$\widetilde{\mathbf{v}}_N = \widetilde{\mathbf{v}}_{N-1}R = \widetilde{\pi}_{c,N-1}\mathbf{r}, \tag{13}$$

$$\mathbf{r}R = r_c\mathbf{r}, \quad R^2 = r_cR. \tag{14}$$

Substituting (13) to (5) we obtain

$$\begin{aligned} M &= \frac{\widetilde{\pi}_{c,N-1}\mathbf{r}[R(\mathbf{I} - R)^{-1} + N\mathbf{I}](\mathbf{I} - R)^{-1}\mathbf{e}}{\widetilde{\pi}_{c,N-1}\mathbf{r}(\mathbf{I} - R)^{-1}\mathbf{e}} \\ &= \frac{\mathbf{r}[R(\mathbf{I} - R)^{-1} + N\mathbf{I}](\mathbf{I} - R)^{-1}\mathbf{e}}{\mathbf{r}(\mathbf{I} - R)^{-1}\mathbf{e}}. \end{aligned} \tag{15}$$

Since R has all rows with zero-elements except the last row, the rank of R is 1. Applying the result of [23], we can write

$$(\mathbf{I} - R)^{-1} = \mathbf{I} - \frac{\mathbf{I}(-R)\mathbf{I}}{1 - tr(R\mathbf{I})} = \mathbf{I} + \frac{R}{1 - r_c}, \tag{16}$$

where the trace $tr(R\mathbf{I})$ of matrix $R\mathbf{I}$ is the sum of all the elements on the main diagonal of matrix $R\mathbf{I}$. Substituting (16) into (15) and utilizing (14) yields (12) after some algebraic steps. □

Corollary 1 expresses that the conditional mean value $M(N) = E[J|J \geq N]$ of the number J of customers in the orbit under the condition $J \geq N$ is the simple function of the single eigenvalue and N. Note that this result is the direct consequence of Theorem 1.

The task is to compute the steady-state probabilities for states $0 \le j \le N-1$ by solving the balance equations pertaining to the levels $0 \le j < N$ and the normalization equation.

Let us introduce auxiliary variables $u_{i,j} = \widetilde{\pi}_{i,j}/\widetilde{\pi}_{c,N-1}$ and $\mathbf{u}_j = [u_{0,j}, \ldots, u_{c,j}]$, $j \ge 0$. Hence $u_{c,N-1} = 1$. Note that $\widetilde{\pi}_{-1,j} = u_{-1,j} = 0$ by definition.

From (4), we have $\widetilde{\mathbf{v}}_N = \widetilde{\mathbf{v}}_{N-1}R$. Thus, $\widetilde{\pi}_{i,N} = r_i\widetilde{\pi}_{c,N-1}$ for $i = 0, \ldots, c$. Using the relation between $\widetilde{\pi}_{c,N-1}$ and $\widetilde{\pi}_{i,N}$, we can write the balance equation for level $N-1$ as

$$\begin{aligned}
&(\lambda + i\mu + (N-1)\mu_r)\widetilde{\pi}_{i,N-1} \\
&= \lambda\widetilde{\pi}_{i-1,N-1} + M(N)\mu_r\widetilde{\pi}_{i-1,N} + (i+1)\mu\widetilde{\pi}_{i+1,N-1} \\
&= \lambda\widetilde{\pi}_{i-1,N-1} + M(N)\mu_r r_{i-1}\widetilde{\pi}_{c,N-1} + (i+1)\mu\widetilde{\pi}_{i+1,N-1}, \\
&\qquad 0 \le i < c.
\end{aligned}$$

$$\begin{aligned}
&(\lambda + c\mu + (N-1)\mu_r P_{im})\widetilde{\pi}_{c,N-1} \\
&= \lambda\widetilde{\pi}_{c-1,N-1} + M(N)\mu_r\widetilde{\pi}_{c-1,N} \\
&\quad + \lambda\widetilde{\pi}_{c,N-2} + M(N)\mu_r P_{im}\widetilde{\pi}_{c,N} \\
&= \lambda\widetilde{\pi}_{c-1,N-1} + M(N)\mu_r r_{c-1}\widetilde{\pi}_{c,N-1} \\
&\quad + \lambda\widetilde{\pi}_{c,N-2} + M(N)\mu_r r_c P_{im}\widetilde{\pi}_{c,N-1}.
\end{aligned}$$

Therefore, we obtain

$$\begin{aligned}
&(\lambda + i\mu + (N-1)\mu_r)u_{i,N-1} \\
&= \lambda u_{i-1,N-1} + M(N)\mu_r r_{i-1} + (i+1)\mu u_{i+1,N-1}, \\
&\qquad 0 \le i < c.
\end{aligned} \tag{17}$$

$$\begin{aligned}
&(\lambda + c\mu + (N-1)\mu_r P_{im}) \\
&= \lambda u_{c-1,N-1} + M(N)\mu_r u_{c-1,N} \\
&\quad + \lambda u_{c,N-2} + M(N)\mu_r P_{im} u_{c,N} \\
&= \lambda u_{c-1,N-1} + M(N)\mu_r r_{c-1} \\
&\quad + \lambda u_{c,N-2} + M(N)\mu_r r_c P_{im}.
\end{aligned}$$

Expanding the balance equation pertaining to level j, $0 \le j < N-1$, we get

$$\begin{aligned}
(\lambda + i\mu + j\mu_r)\widetilde{\pi}_{i,j} &= \lambda\widetilde{\pi}_{i-1,j} \\
&\quad + (j+1)\mu_r\widetilde{\pi}_{i-1,j+1} + (i+1)\mu\widetilde{\pi}_{i+1,j}, \\
&\qquad 0 \le i < c, 0 \le j < N-1,
\end{aligned} \tag{18}$$

$$
\begin{aligned}
(\lambda + c\mu + j\mu_r P_{im})\widetilde{\pi}_{c,j} = \lambda \widetilde{\pi}_{c-1,j} \\
+ (j+1)\mu_r \widetilde{\pi}_{c-1,j+1} + \lambda \widetilde{\pi}_{c,j-1} + (j+1)\mu_r P_{im} \widetilde{\pi}_{c,j+1}, \\
0 \le j < N-1, \quad (19)
\end{aligned}
$$

which follows

$$
\begin{aligned}
(\lambda + i\mu + j\mu_r) u_{i,j} = \lambda u_{i-1,j} \\
+ (j+1)\mu_r u_{i-1,j+1} + (i+1)\mu u_{i+1,j}, \\
0 \le i < c, 0 \le j < N-1, \quad (20)
\end{aligned}
$$

$$
\begin{aligned}
(\lambda + c\mu + j\mu_r P_{im}) u_{c,j} = \lambda u_{c-1,j} \\
+ (j+1)\mu_r u_{c-1,j+1} + \lambda u_{c,j-1} + (j+1)\mu_r P_{im} u_{c,j+1}, \\
0 \le j < N-1. \quad (21)
\end{aligned}
$$

Note that both equations (17) and (20) have the tridiagonal form if $u_{i-i,j+1}$ is determined in the previous step:

$$
\alpha_{ij} u_{i-1,j} + \beta_{i,j} u_{i,j} + \gamma_{ij} u_{i+1,j} = \omega_{i,j}
$$

for $0 \le i \le c-1$ and $0 \le j \le N-1$. Therefore, the Thomas algorithm can be used to solve the steady-state probabilities in an efficient way [24]. The adaptation of the Thomas algorithm for the present problem is presented in Algorithm 2, where line 6 is the result of equating the flow rate into and out of level j of the orbit.

Algorithm 2 Computation of the stationary probabilities $\tilde{\mathbf{v}}_j,\ j = 0, \ldots, N-1$

$b_{0,j} = 0$; $D_{0,j} = 0$
for $j = N-1$ to 0 **do**
  if $j == N-1$ **then**
    $u_{c,N-1} = 1$
  else
    $u_{c,j} = \dfrac{(j+1)\mu_r}{\lambda}\left(\sum_{i=0}^{c-1} u_{i,j+1} + P_{im} u_{c,j+1}\right)$
  end if
  $b_{0,j} = 0$; $D_{0,j} = 0$
  for $i = 0$ to $c-1$ **do**
    $\beta_{i,j} = \lambda + i\mu + j\mu_r$
    $\gamma_{i,j} = -(i+1)\mu$, $\alpha_{i,j} = -\lambda$
    if $i > 0$ **then**
      if $j == N-1$ **then**
        $\omega_{i,j} = M\mu_r r_{i-1}$
      else
        $\omega_{i,j} = (j+1)\mu_r u_{i-1,j+1}$
      end if
      $b_{i,j} = \dfrac{i\mu(b_{i-1,j} + j\mu_r)}{b_{i-1,j} + \beta_{0,j}}$
      $D_{i,j} = \omega_{i,j} - \dfrac{\alpha_{ij} D_{i-1,j}}{b_{i-1,j} + \beta_{0,j}}$
    end if
  end for
  for $i = c-1$ to 0 **do**
    $u_{i,j} = \dfrac{D_{i,j} - \gamma_{i,j} u_{i+1,j}}{b_{i,j} + \beta_{0,j}}$
  end for
end for

From the normalization equation $\sum_{j=0}^{N-2} \tilde{\mathbf{v}}_j \mathbf{e} + \tilde{\mathbf{v}}_{N-1}(\mathbf{I} - R)^{-1}\mathbf{e} = 1$, we get

$$\tilde{\pi}_{c,N-1} = \frac{1}{\sum_{j=0}^{N-2} \mathbf{u}_j \mathbf{e} + \mathbf{u}_{N-1}(\mathbf{I} - R)^{-1}\mathbf{e}}.$$

Then, we obtain

$$\tilde{\pi}_{i,j} = u_{i,j}\tilde{\pi}_{c,N-1} = \frac{u_{i,j}}{\sum_{j=0}^{N-2} \mathbf{u}_j \mathbf{e} + \mathbf{u}_{N-1}(\mathbf{I} - R)^{-1}\mathbf{e}}.$$

The blocking probability P_b, the immediate service probability P_{is}, the delayed service probability P_{ds}, the nonservice probability P_{ns} and the mean number of users in the retrial orbit N_{ret} can be determined [20] as follows:

$$\begin{aligned}
P_b &= \sum_{m=0}^{N-1} \widetilde{\mathbf{v}}_m \, \mathbf{z} + \widetilde{\mathbf{v}}_N (\mathbf{I} - R)^{-1}\mathbf{z}, \\
P_{is} &= 1 - P_b, \\
P_{ds} &= \lambda^{-1}\mu_r \left[\sum_{m=0}^{N-1} m\widetilde{\mathbf{v}}_m \, \mathbf{o} + M(N)\widetilde{\mathbf{v}}_N (\mathbf{I} - R)^{-1}\mathbf{o} \right], \\
P_{ns} &= \lambda^{-1} P_{im}\mu_r \left[\sum_{m=0}^{N-1} m\widetilde{\mathbf{v}}_m \, \mathbf{z} + M(N)\widetilde{\mathbf{v}}_N (\mathbf{I} - R)^{-1}\mathbf{z} \right], \\
N_{ret} &= \sum_{m=0}^{N-1} m\widetilde{\mathbf{v}}_m \, \mathbf{e} \\
&\quad + \widetilde{\mathbf{v}}_N \left(R(\mathbf{I} - R)^{-1} + N\mathbf{I} \right) (\mathbf{I} - R)^{-1}\mathbf{e}, \qquad (22)
\end{aligned}$$

where $\mathbf{z} = [0, 0, \ldots, 0, 1]$, $\mathbf{o} = [1, 1, \ldots, 1, 0]$, $\mathbf{e} = [1, 1, \ldots, 1, 1]$. Note that $\mathbf{e} = \mathbf{o} + \mathbf{z}$. Because $P_{is} + P_{ds} + P_{ns} = 1$ (see [20]) and $P_{is} = 1 - P_b$, we get $P_b = P_{ds} + P_{ns}$.

However, the direct application of equations (22) defined in [20] is not efficient when one implements a computer program. To compute the performance measures in an efficient way, we derive simpler equations than (22) after some algebraic steps that are presented in Proposition 1, 2, 3 and 4. Note that the equations for the performance measures are utilized in Algorithm 4 (see Section 3.4).

Let us define the following quantities

$$\mathbf{a} = \widetilde{\mathbf{v}}_N \left(R(\mathbf{I} - R)^{-1} + N\mathbf{I} \right) (\mathbf{I} - R)^{-1} \qquad (23)$$

$$a_1 = \sum_{m=0}^{N-1} m\widetilde{\mathbf{v}}_m\mathbf{o} + M(N)\widetilde{\mathbf{v}}_N(\mathbf{I} - R)^{-1}\mathbf{o} \qquad (24)$$

$$a_2 = \sum_{m=0}^{N-1} m\widetilde{\mathbf{v}}_m\mathbf{z} + M(N)\widetilde{\mathbf{v}}_N(\mathbf{I} - R)^{-1}\mathbf{z}. \qquad (25)$$

Therefore, $P_{ds} = \lambda^{-1}\mu_r a_1$.

Proposition 1 *The nonservice probability is expressed as*

$$P_{ns} = \lambda^{-1} P_{im} \mu_r a_2 = \lambda^{-1} P_{im} \mu_r \left(\sum_{m=1}^{N-2} m \widetilde{\pi}_{c,m} + \frac{(M(N)-1)\widetilde{\pi}_{c,N-1}}{1-r_c} \right). \tag{26}$$

Proof.

Substituting (16) to the definition of **a**, we obtain

$$\begin{aligned}
\mathbf{a} &= \widetilde{\mathbf{v}}_N \left(R(\mathbf{I}-R)^{-1} + N\mathbf{I} \right) (\mathbf{I}-R)^{-1} \\
&= \widetilde{\mathbf{v}}_N \left(R(\mathbf{I} + \frac{R}{1-r_c}) + N\mathbf{I} \right) (\mathbf{I}-R)^{-1} \\
&= \widetilde{\mathbf{v}}_N \left(\frac{R}{1-r_c} + N\mathbf{I} \right) (\mathbf{I}-R)^{-1} \ \text{(using (13))} \\
&= \widetilde{\pi}_{c,N-1}\mathbf{r} \left(\frac{R}{1-r_c} + N\mathbf{I} \right) (\mathbf{I}-R)^{-1} \ \text{(using (14))} \\
&= \widetilde{\pi}_{c,N-1} \left(\frac{r_c\mathbf{r}}{1-r_c} + N\mathbf{r} \right) (\mathbf{I}-R)^{-1} \ \text{(using (13))} \\
&= \widetilde{\pi}_{c,N-1} M(N)\mathbf{r}(\mathbf{I}-R)^{-1} \ \text{(using (12))} \\
&= M(N)\widetilde{\mathbf{v}}_N(\mathbf{I}-R)^{-1} \ \text{(using (13))} \qquad (27) \\
&= \widetilde{\pi}_{c,N-1} M(N)\mathbf{r} \left(\mathbf{I} + \frac{R}{1-r_c} \right) \ \text{(using (16))} \\
&= \widetilde{\pi}_{c,N-1} M(N) \left(\mathbf{r} + \frac{r_c\mathbf{r}}{1-r_c} \right) \ \text{(using (14))} \\
&= \frac{\widetilde{\pi}_{c,N-1} M(N)}{1-r_c}\mathbf{r}. \qquad (28)
\end{aligned}$$

Substituting (28) into (25), we obtain

$$
\begin{aligned}
a_2 &= \sum_{m=0}^{N-1} m\widetilde{\mathbf{v}}_m\, \mathbf{z} + \mathbf{a}\, \mathbf{z} \\
&= \sum_{m=0}^{N-1} m\widetilde{\mathbf{v}}_m\, \mathbf{z} + \frac{\widetilde{\pi}_{c,N-1} M(N)}{1-r_c} \mathbf{r}\, \mathbf{z} \\
&= \sum_{m=1}^{N-1} m\widetilde{\pi}_{c,m} + \frac{\widetilde{\pi}_{c,N-1} M(N) r_c}{1-r_c} \\
&= \sum_{m=1}^{N-2} m\widetilde{\pi}_{c,m} + \widetilde{\pi}_{c,N-1} \left[N-1 + \frac{r_c M(N)}{1-r_c} \right] \\
&= \sum_{m=1}^{N-2} m\widetilde{\pi}_{c,m} + \frac{(M(N)-1)\widetilde{\pi}_{c,N-1}}{1-r_c} \;\; \text{(using (12))}.
\end{aligned} \tag{29}
$$

Equation (29) yields (26).□

Proposition 2 *The mean number of users in the retrial orbit is*

$$
N_{ret} = a_1 + a_2. \tag{30}
$$

Proof. From the definition of a_1 and a_2, we get

$$
\begin{aligned}
a_1 + a_2 &= \sum_{m=0}^{N-1} m\widetilde{\mathbf{v}}_m(\mathbf{o}+\mathbf{z}) + M(N)\widetilde{\mathbf{v}}_N(\mathbf{I}-R)^{-1}(\mathbf{o}+\mathbf{z}) \\
&= \sum_{m=0}^{N-1} m\widetilde{\mathbf{v}}_m\mathbf{e} + M(N)\widetilde{\mathbf{v}}_N(\mathbf{I}-R)^{-1}\mathbf{e}.
\end{aligned}
$$

Utilizing (27) and the definition of N_{ret}, we obtain equation (30). □.

Proposition 3 *We can obtain the blocking probability P_b as follows:*

$$
P_b = \sum_{m=0}^{N-2} \widetilde{\pi}_{c,m} + \frac{\widetilde{\pi}_{c,N-1}}{1-r_c}. \tag{31}
$$

Proof. Utilizing (16), we obtain

$$
\begin{aligned}
P_b &= \sum_{m=0}^{N-1} \widetilde{\mathbf{v}}_m \, \mathbf{z} + \widetilde{\mathbf{v}}_N (\mathbf{I} - R)^{-1}\mathbf{z} \\
&= \sum_{m=0}^{N-1} \widetilde{\mathbf{v}}_m \, \mathbf{z} + \widetilde{\mathbf{v}}_{N-1} R(\mathbf{I} + \frac{R}{1-r_c})\mathbf{z} \\
&= \sum_{m=0}^{N-1} \widetilde{\mathbf{v}}_m \, \mathbf{z} + \widetilde{\mathbf{v}}_{N-1} \frac{R}{1-r_c}\mathbf{z} \\
&= \sum_{m=0}^{N-1} \widetilde{\mathbf{v}}_m \, \mathbf{z} + \widetilde{\pi}_{c,N-1} \frac{\mathbf{r}}{1-r_c}\mathbf{z} \\
&= \sum_{m=0}^{N-2} \widetilde{\mathbf{v}}_m \, \mathbf{z} + \widetilde{\mathbf{v}}_{N-1}\mathbf{z} + \widetilde{\pi}_{c,N-1} \frac{\mathbf{r}}{1-r_c}\mathbf{z} \\
&= \sum_{m=0}^{N-2} \widetilde{\pi}_{c,m} + \widetilde{\pi}_{c,N-1} + \widetilde{\pi}_{c,N-1} \frac{r_c}{1-r_c} \\
&= \sum_{m=0}^{N-2} \widetilde{\pi}_{c,m} + \frac{\widetilde{\pi}_{c,N-1}}{1-r_c} . \Box .
\end{aligned}
$$

Proposition 4 *The following relation exists between the performance measures*

$$
N_{ret} = \frac{\lambda}{\mu_r} \left(\frac{P_{ns}(1-P_{im})}{P_{im}} + P_b \right) . \tag{32}
$$

Proof. From $P_{ds} = \lambda^{-1}\mu_r a_1$, $P_{ns} = \lambda^{-1} P_{im}\mu_r a_2$, $P_b = P_{ds} + P_{ns}$ and $N_{ret} = a_1 + a_2$, we get

$$
\begin{aligned}
N_{ret} &= \frac{\lambda}{\mu_r}(\frac{P_{ns}}{P_{im}} + P_{ds}) \\
&= \frac{\lambda}{\mu_r}(\frac{P_{ns}}{P_{im}} + P_b - P_{ns}) \\
&= \frac{\lambda}{\mu_r} \left(\frac{P_{ns}(1-P_{im})}{P_{im}} + P_b \right) . \Box
\end{aligned}
$$

3.4 An Estimation of N

Domenech-Benlloch et al. [20] presented some numerical results concerning choosing the appropriate value of threshold N to achieve the required accuracy of the approximation of performance measures. However, the authors [20] did not present a systematic way to find the appropriate value of threshold N. In this section, we will show an efficient method to estimate threshold N.

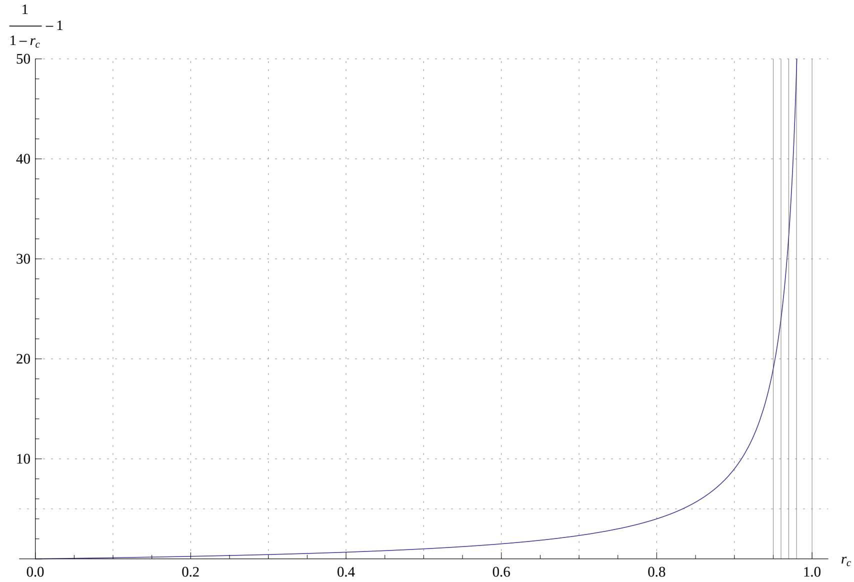

Fig. 1. $1/(1-r_c)-1$ vs r_c

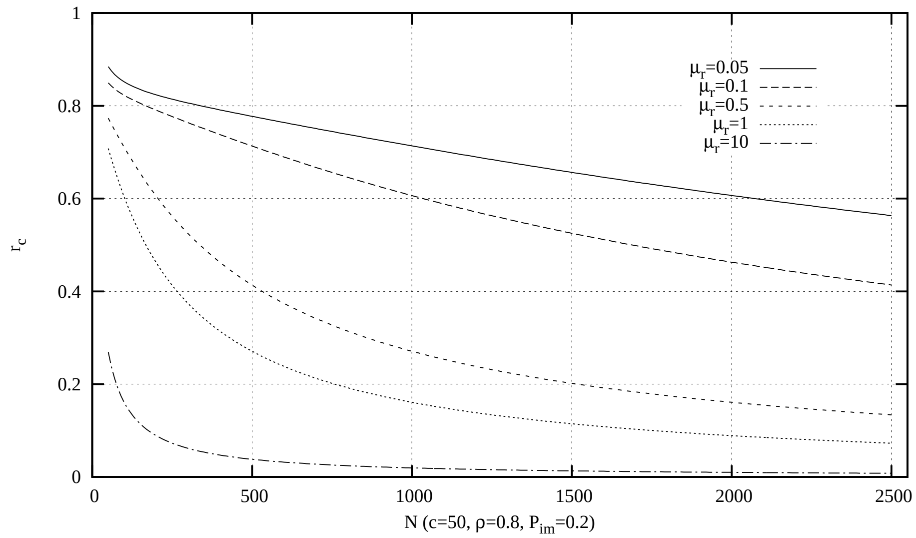

Fig. 2. r_c vs N for $c = 50$, $\rho = 0.8$, $P_{im} = 0.2$ and $\mu = 1$

From equation (4), the tail of the distribution $\tilde{\mathbf{v}}_j$ is geometrically distributed with parameter r_c. In the stable state of the system described by CTMC $\widetilde{Y}$, the higher the value N we choose, the smaller the value of $\mathbf{v}_N$ is. It is anticipated that the higher the chosen value of N is the smaller the value of r_c is if the system is in the stable state (see Figure 2).

As one observes from Figure 1, the slope of the tangent line to the curve $1/(1-r_c)-1$ increases as r_c approaches 1, and $M(N)-N$ too. Let r_{th} be the selected upper limit of r_c (the purpose of the selected upper limit is to search for the initial value of N to save the computational time), i.e., $0 < r_c \leq r_{th}$.

Algorithm 3 To choose N

$N \leftarrow N_{ini}$
repeat
    $N \leftarrow N + 1$
until $l_{c+1}(r_{th}, N - 1 + 1/(1 - r_{th}) \leq 0$

Algorithm 4 Proposed algorithm

Call algorithm 3 to choose N
$step \leftarrow 1$
repeat
    $N \leftarrow N + step$
    $M_0 \leftarrow N$
    $k \leftarrow 0$
    repeat
        $k \leftarrow k + 1$
        Compute the root x_c of $l_{c+1}(x, M_{k-1})$ in the interval $(0, r_{th}]$
        $M_k = N + \frac{r_c}{1-r_c}$ (using equation (12))
        $iteration\_error = |M_k - M_{k-1}|/M_{k-1}$
        if $l_{c+1}(x_c, M_{k-1}) > \epsilon_r$ **then**
            $k \leftarrow 0$
            $N \leftarrow N + 1$
            $M_0 \leftarrow N$
            $iteration\_error = 1$
        end if
    until $iteration\_error < \epsilon_M$
    Compute $\boldsymbol{\psi}_c$ and R
    Call Algorithm 2
    Compute performance measures
    Compute $Converge$
until $Converge$

As stated before we also need to find the initial value of N. To save the computational time of the search for the initial value of N, we only check whether $l_{c+1}(x, M(N))$ has a root in the interval $(0, r_{th}]$ instead of determining r_c. Because $l_{c+1}(0, M(N)) = \lambda$ holds, we have to examine whether $l_{c+1}(r_{th}, M(N)) \leq 0$ in the first stage of our proposed solution. However, $M(N)$ is not known in advance. To resolve this problem, Theorem 2 is applied to choose the initial value of N (see Algorithm 3), where $l_{c+1}(r_{th}, N-1+1/(1-r_{th})) \leq 0$ is verified.

Theorem 2 *If r_c is the eigenvalue of $Q(x, M(N))$ in the interval $(0, 1)$, and $l_{c+1}(r_{th}, N - 1 + 1/(1 - r_{th})) \leq 0$ for $r_{th} > 0$, then r_c is bounded by r_{th} (i.e., $0 < r_c \leq r_{th}$).*

Proof.

Assume that $r_{th} < r_c$, which follows $l_{c+1}(r_{th}, M(N)) > 0$ because r_c is the eigenvalue of $Q(x, M(N))$ in the interval $(0, 1)$ (i.e., $l_{c+1}(r_c, M(N)) = 0$) and $l_{c+1}(0, M(N)) = \lambda$.

Note that $l_{c+1}(x, y)$ is a monotonically decreasing function with respect to y for any constant x, $0 < x < 1$, because

- $l_{c+1}(0, y) = \lambda$ holds,
- $l_{c+1}(x, y)$ has a single root with respect to x in the interval (0,1) for any constant y, and
- the higher the value of y is the smaller the root with respect to x is.

We have $N-1+1/(1-r_{th}) < M(N) = N-1+1/(1-r_c)$ from the assumption. Therefore, $l_{c+1}(r_{th}, N-1+1/(1-r_{th})) > l_{c+1}(r_{th}, M(N))$.

Using $l_{c+1}(r_{th}, M(N)) > 0$, we obtain $l_{c+1}(r_{th}, N-1+1/(1-r_{th})) > 0$, which contradicts the given condition $l_{c+1}(r_{th}, N-1+1/(1-r_{th})) \leq 0$. Therefore, $r_{th} < r_c$ does not hold, which yields that r_c is bounded by r_{th}. $\square$

Theorem 2 is used in our proposed Algorithm 4 to find the initial value of N.

3.5 The Convergence Criterion of a Proposed Algorithm

We present the details of a proposed computational procedure in Algorithm 4 that integrates key results in Section 3.1-3.4. In Algorithm 4 two loops are applied after the initial choice of N. The inner loop is needed to find $M(N)$ of a certain value N, while the outer loop is to tune N to obtain the required accuracy of the estimation of performance measures. A notable feature of the proposed procedure compared to the HM2 algorithm is that the computation of $M(N)$ does not require the computation of the steady-state probabilities. Furthermore, the algorithm is enhanced with the computation of threshold N.

To define the convergence criterion *Converge* we made the following investigation: we compute the performance measures by running the core (between lines 4 and 19 of Algorithm 3) of our algorithm with fixing N for parameters $\mu = 1/180$, $\mu_r = 0.01$, $P_{im} = 0.2$, $\epsilon_M = 10^{-3}$ and $\epsilon_r = 10^{-10}$ (Figures 3 and 4). From Figures 3 and 4 the performance measures converges as N grows. Furthermore, a high oscillation is observed in Figure 4 as well (see the e-companion for more numerical results with other settings of parameters). Therefore, the convergence criterion (to determine when the performance measures reach the "stable state") is defined as the relative error between two moving averages concerning a specific performance measure

$$Converge := \frac{|\sum_{i=L}^{K} \chi_i/(K-L+1) - \sum_{i=1}^{K} \chi_i/K|}{\sum_{i=L}^{K} \chi_i/(K-L+1)} < \epsilon_p,$$

where χ_i's, $i = 1, \ldots, K$, are the latest values of a chosen performance measure (e.g., N_{ret}) determined so far by the algorithm, ϵ_p is the specified accuracy, K and L are parameters. It is obvious that the minimum choice of K and L is $K = 3$ and $L = 2$. Furthermore, the higher the values of K and L are, the more time is needed, but the better the guarantee of convergence is ensured.

In Table 1 we summarize the computational time of the algorithm for $\rho = \lambda/(\mu c)$, $\mu = 1/180$, $\mu_r = 0.01$, $P_{im} = 0.2$. As observed the algorithm successfully stops in the stable region of performance measures (see Figures 3 and 4). From numerical results (see the e-companion for results with other settings of K and L), the minimum choice of $K = 3$ and $L = 2$ can guarantee that the proposed algorithm steps over the "oscillation period". Results in Table 1 and the e-companion empirically show that P_{ns} can be set as the main performance measure in the convergence criterion *Converge* for determining all performance measures.

4 Computational Times of the Proposed Algorithm

We plot the computational time versus c and N in Figures 5 and 6, on a machine with Intel® Core™ 2 Duo T9400 2.53 GHz processor (note that the algorithm is implemented in Mathematica) for parameters $\rho = \lambda/(\mu c) = 0.8$, $\mu = 1$, $P_{im} = 0.2$, $\epsilon_r = 10^{-10}$, $\epsilon_M = 10^{-3}$. In the curves the computational time of the HM2 algorithm [20] and the core (between lines 4 and 19 of Algorithm 3) of our algorithm. As observed the computational time of the original algorithm HM2 is a rapidly increasing function of c, while the computational complexity of the core of our algorithm is only $O(c)$.

In Figure 7, we plot the computational time of our algorithm versus ρ and c. It is observed from Table 1 and Figure 7 that ρ impacts the convergence of the algorithms. The explanation behind this phenomenon is that the higher ρ is the more likely the oscillation is (see Figure 4 and illustrations in the e-companion). Therefore, more computational time is needed to step over the oscillation. In other words, the algorithm needs more time to reach the convergence due to the oscillation phenomenon at large values of ρ. However, the

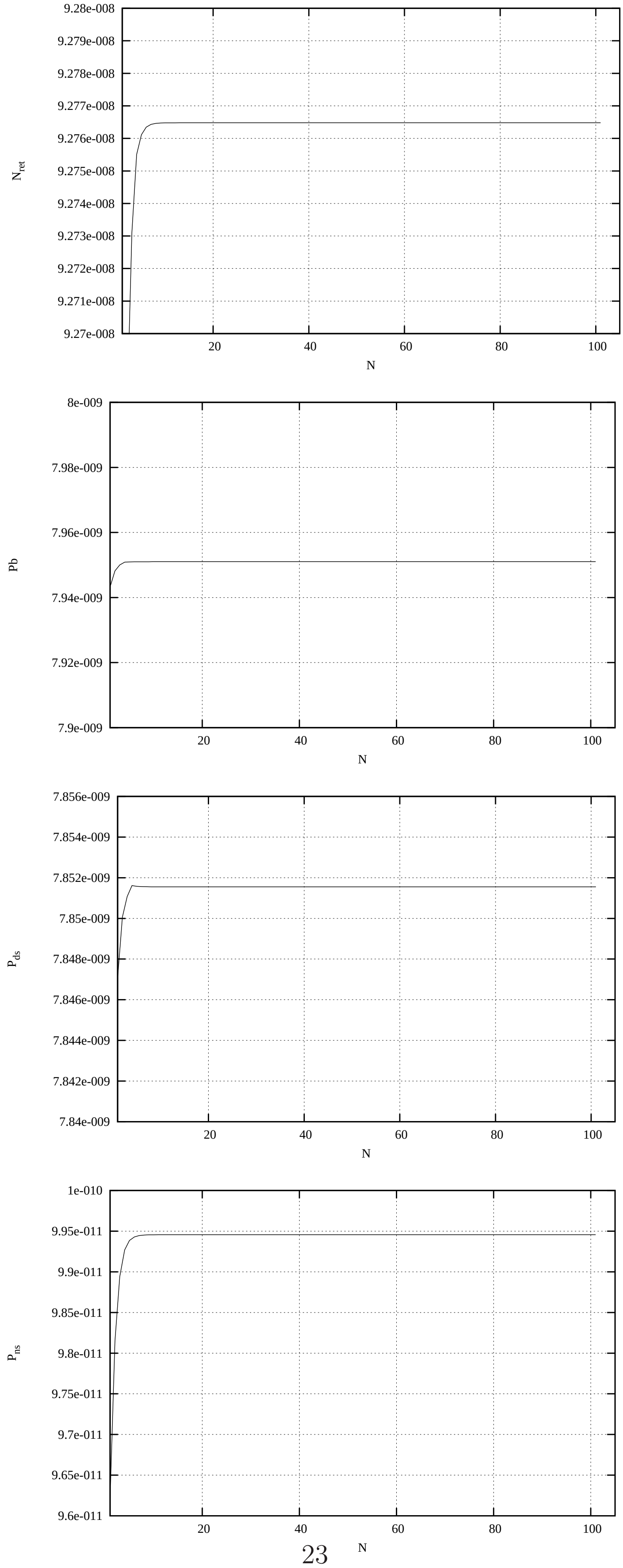

Fig. 3. Performance measures vs N for $c = 50$, $\rho = 0.4$, $\mu = 1/180$, $\mu_r = 0.01$, $P_{im} = 0.2$

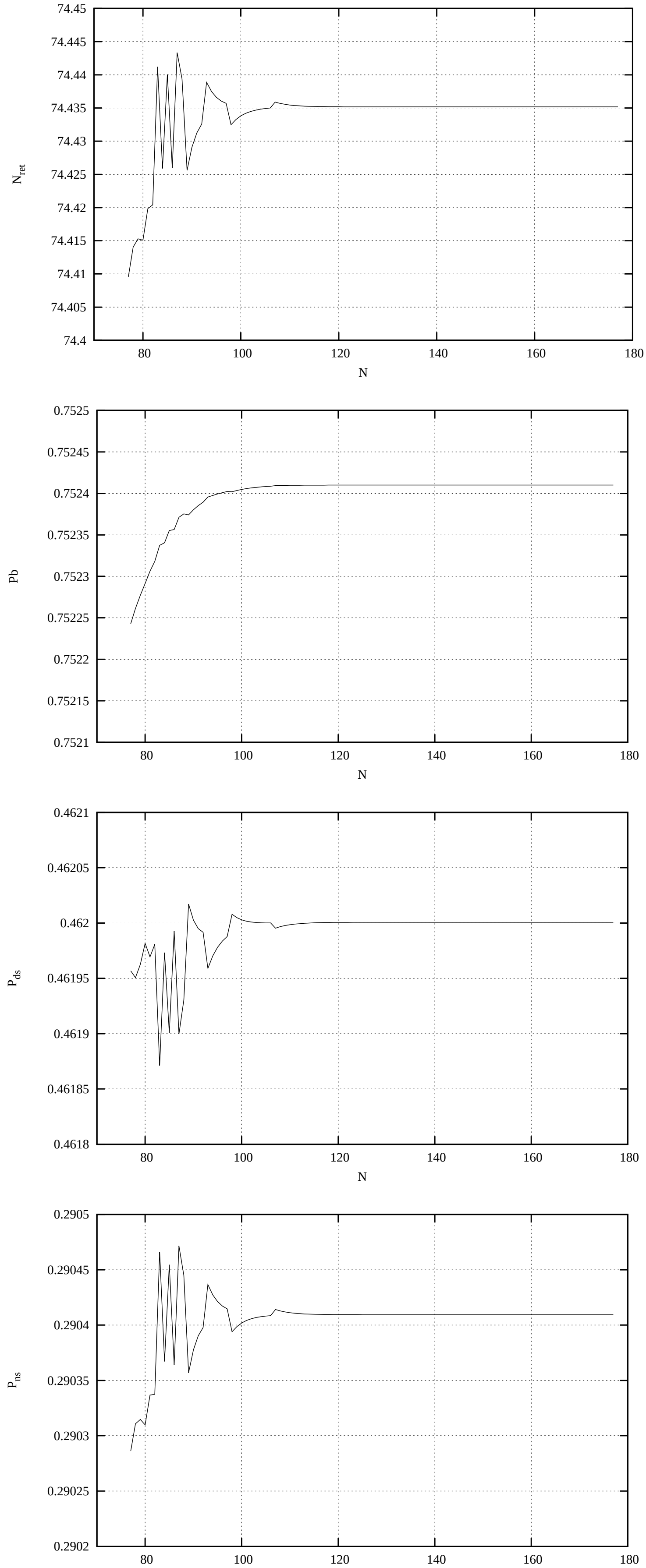

Fig. 4. Performance measures vs N for $c = 50$, $\rho = 1.4$, $\mu = 1/180$, $\mu_r = 0.01$, $P_{im} = 0.2$

Table 1
N and computational time of the algorithm for $K = 3, L = 2, \epsilon_p = 10^{-5}, \epsilon_M = 10^{-3}, \epsilon_r = 10^{-10}, \rho = \lambda/(\mu c), \mu = 1/180, \mu_r = 0.01, P_{im} = 0.2, r_{th} = 0.95$

		$c = 50$		$c = 100$		$c = 200$		$c = 500$		$c = 1000$	
ρ		N	Time (s)	N	Time (s)	N	Time (s)	N	Time (s)	N	Time (s)
0.4	N_{ret}	17	0.889	6	0.765	8	1.435	39	16.318	13	13.121
	P_b	16	0.858	4	0.468	8	1.42	39	16.255	13	13.275
	P_{ds}	16	0.873	5	0.609	8	1.42	39	16.38	13	13.166
	P_{ns}	23	1.233	23	2.574	12	2.449	39	16.396	13	13.12
0.8	N_{ret}	38	2.511	29	3.697	46	9.703	35	15.646	13	15.287
	P_b	38	2.574	32	4.57	41	8.408	35	15.647	12	13.744
	P_{ds}	38	2.511	32	4.446	41	8.377	35	15.709	11	11.762
	P_{ns}	47	3.182	32	4.617	46	9.751	36	17.035	40	52.355
1.0	N_{ret}	44	2.855	76	7.504	117	14.133	206	35.491	363	82.619
	P_b	44	2.792	76	7.769	117	14.227	206	35.756	363	82.509
	P_{ds}	44	2.698	76	7.737	124	16.114	206	35.506	363	82.914
	P_{ns}	44	2.777	79	8.129	124	16.255	206	36.099	363	82.477
1.4	N_{ret}	102	3.042	198	7.957	397	26.411	858	81.417	1679	264.001
	P_b	100	2.621	198	8.064	397	26.427	858	81.589	1679	266.122
	P_{ds}	106	3.463	198	8.003	397	26.318	858	81.339	1679	264.781
	P_{ns}	106	3.464	202	9.375	397	26.458	858	81.354	1679	266.013

computational time complexity is still of $O(c)$ for a specific ρ.

Remark. It is worth emphasizing that approaches belonging to the category "Approximations" (Domenech-Benlloch et al. [20]) produced unacceptable errors in most cases. Therefore, we do not focus on the comparison with these algorithms in this paper. The HM2 algorithm overcomes other approaches in the term of the accuracy. Our algorithm has the same accuracy as the HM2. However, it is much faster than the HM2 algorithm. We have shown that the computational time complexity of our algorithm is of $O(c)$. To our best knowledge, we do not know that there is any other algorithm which has the computational time complexity of $O(c)$ for the M/M/c retrial queue with impatient customers and has the same accuracy as of the HM2 algorithm (see Domenech-Benlloch et al. [20] and Do [3] for the overview of the latest algorithms).

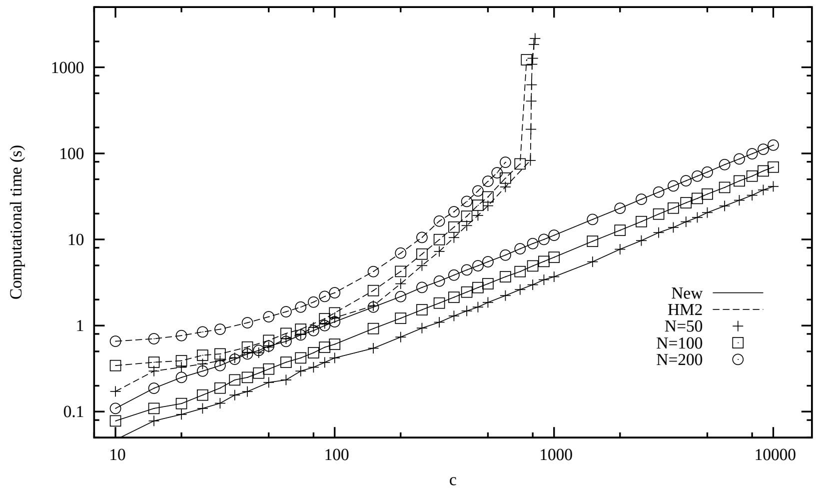

Fig. 5. Computational time vs c for $\mu = 1$, $\mu_r = 0.5$, $\rho = \lambda/(\mu c) = 0.8$, $P_{im} = 0.2$, $\epsilon_M = 10^{-3}$ and $\epsilon_r = 10^{-10}$

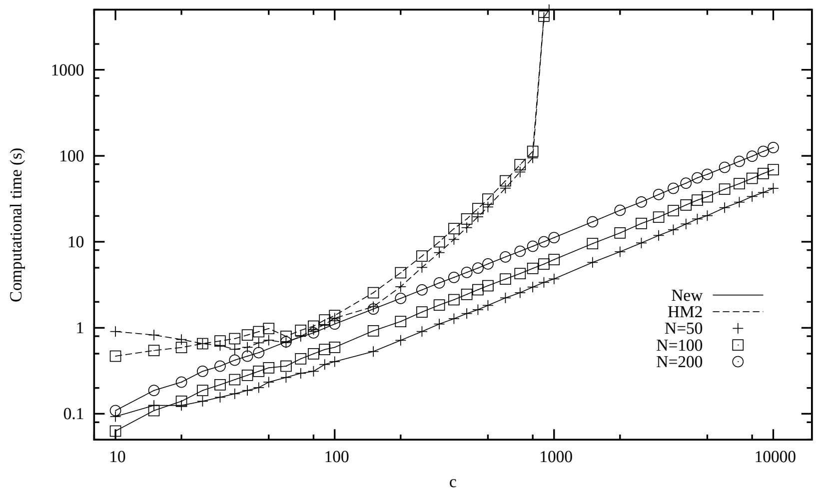

Fig. 6. Computational time vs c for $\mu = 1$, $\mu_r = 0.05$, $\rho = \lambda/(\mu c) = 0.8$, $P_{im} = 0.2$, $\epsilon_M = 10^{-3}$ and $\epsilon_r = 10^{-10}$

5 Conclusions

We have presented our contributions that enable the approximation of the performance of a multiserver retrial queue with impatient customers. We derived exact expressions for the computation of the rate matrix and the conditional mean number of customers. We explore the behavior of performance measures versus N, then the estimation of threshold N is derived. We have presented some important properties and provided a proof of concerning the determination of important quantities. Based on the derivations, we have constructed an algorithm to solve the multiserver retrial queue with impatient customers.

In the future, we will investigate the application of the proposed approach for

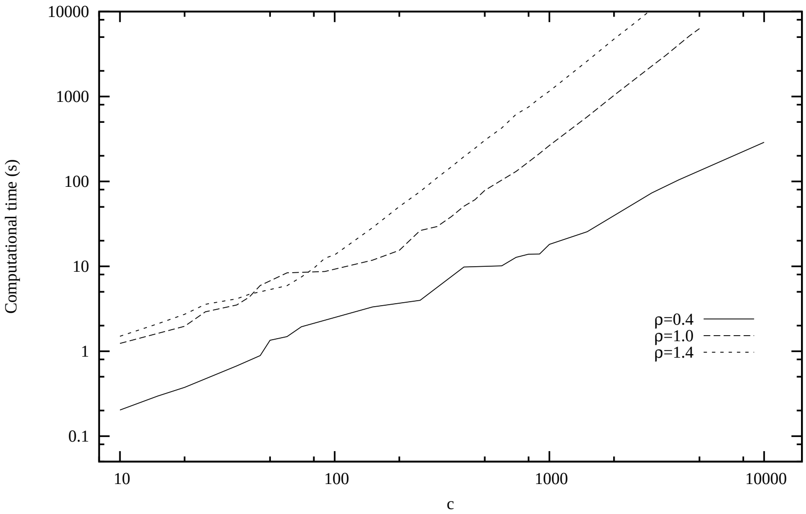

Fig. 7. Computation time vs c and ρ for $K = 3$, $L = 2$, $\mu = 1.0$, $\mu_r = 0.5$, $P_{im} = 0.2$, $\epsilon_M = 10^{-3}$, $\epsilon_p = 10^{-5}$ and $\epsilon_r = 10^{-10}$, $r_{th} = 0.95$

queueing problems [3,4] in cellular mobile networks.

Acknowledgement

The publication was supported by the TÁMOP-4.2.2.C-11/1/KONV-2012-0001 project. The project has been supported by the European Union, co-financed by the European Social Fund.

The authors are grateful for anonymous reviewers for comments that help to improve the presentation of this paper.

References

[1] Artalejo, J. R. (1999) Accessible Bibliography on Retrial Queues. *Math. Compt. Model*, **30**, 1–6.

[2] Artalejo, J. R. (2010) Accessible bibliography on retrial queues: Progress in 2000-2009. *Mathematical and Computer Modelling*, **51**, 1071–1081.

[3] Do, T. V. (2010) A new computational algorithm for retrial queues to cellular mobile systems with guard channels. *Computers & Industrial Engineering*, **59**, 865 – 872.

[4] Do, T. V. (2011) Solution for a retrial queueing problem in cellular networks with the fractional guard channel policy. *Mathematical and Computer Modelling*, 53(11–12):2058–2065, 2011.

[5] Do, T. V. and Chakka, R. (2010) An Efficient Method to Compute the Rate Matrix for Retrial Queues with Large Number of Servers, *Applied Mathematics Letters*, vol. 23, pp. 638–643, 2010.

[6] Falin, G. I. (1990) A Survey of Retrial Queues. *Queueing Systems*, **7**, 127–168.

[7] Falin, G. I. and Templeton, J. G. C. (1997) *Retrial Queues*. Chapman & Hall, London.

[8] Artalejo, J. R. and Gómez-Corral, A. (2008) *Retrial Queueing Systems*. Springer-Verlag, Berlin Heidelberg.

[9] Wüchner, P., Sztrik, J., and de Meer, H. (2009) Finite-source M/M/S retrial queue with search for balking and impatient customers from the orbit. *Computer Networks*, **53**, 1264–1273.

[10] Wang, J., Cao, J., and Li, Q. (2001) Reliability analysis of the retrial queue with server breakdowns and repairs. *Queueing Syst.*, **38**, 363–380.

[11] Artalejo, J. and Pozo, M. (2002) Numerical calculation of the stationary distribution of the main multiserver retrial queue. *Annals of Operations Research*, **116**, 41–56. 10.1023/A:1021359709489.

[12] Yang, T. and Templeton, J. G. C. (1982) A Survey on Retrial Queues. *Queueing Systems*, **2**, 201–233.

[13] Falin, G. I. (1984) On Sufficient Conditions for Ergodicity of Multichannel Queueing Systems with Repeated Calls. *Adv. Appl. Prob.*, **16**, 447–448.

[14] Neuts, M. F. and Rao, B. M. (1990) Numerical investigation of a multiserver retrial model. *Queueing Syst.*, **7**, 169–189.

[15] Neuts, M. F. (1981) *Matrix Geometric Soluctions in Stochastic Model*. Johns Hopkins University Press, Baltimore.

[16] Latouche, G. and Ramaswami, V. (1999) *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM Series on Statistics and Applied Probability.

[17] Mitrani, I. and Chakka, R. (1995) Spectral expansion solution for a class of Markov models: Application and comparison with the matrix-geometric method. *Performance Evaluation*, **23**, 241–260.

[18] Bini, D. A., Latouche, G., and Meini, B. (2005) *Numerical Methods for Structured Markov Chains (Numerical Mathematics and Scientific Computation)*. Oxford University Press, Inc., New York, NY, USA.

[19] Naoumov, V., Krieger, U. R., and Warner, D. (1997) Analysis of a Multi-Server Delay-Loss System With a General Markovian Arrival Process. In Chakravarthy, S. R. and Alfa, A. S. (eds.), *Matrix-analytic methods in stochastic models*, September, Lecture Notes in Pure and Applied Mathematics, **183**, pp. 43–66. Marcel Dekker.

[20] Domenech-Benlloch, M. J., Gimenez-Guzman, J. M., Pla, V., Martinez-Bauset, J., and Casares-Giner, V. (2008) Generalized truncated methods for an efficient solution of retrial systems. *Mathematical Problems in Engineering, (2008), doi:10.1155/2008/183089*, **2008**.

[21] Kleinrock, L. (1975) *Queueing Systems: Vol I.* Wiley.

[22] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007) *Numerical Recipes 3rd Edition: The Art of Scientific Computing.* Cambridge University Press, New York, NY, USA.

[23] Miller, K. S. (1981) On the inverse of the sum of matrices. *Mathematics Magazine*, **54**, 67–72.

[24] Conte, S. and deBoor, C. (1972) *Elementary Numerical Analysis.* McGraw-Hill, New York.

An Enhanced Algorithm to Solve Multiserver Retrial Queueing Systems with Impatient Customers
e-companion

Preprint submitted to

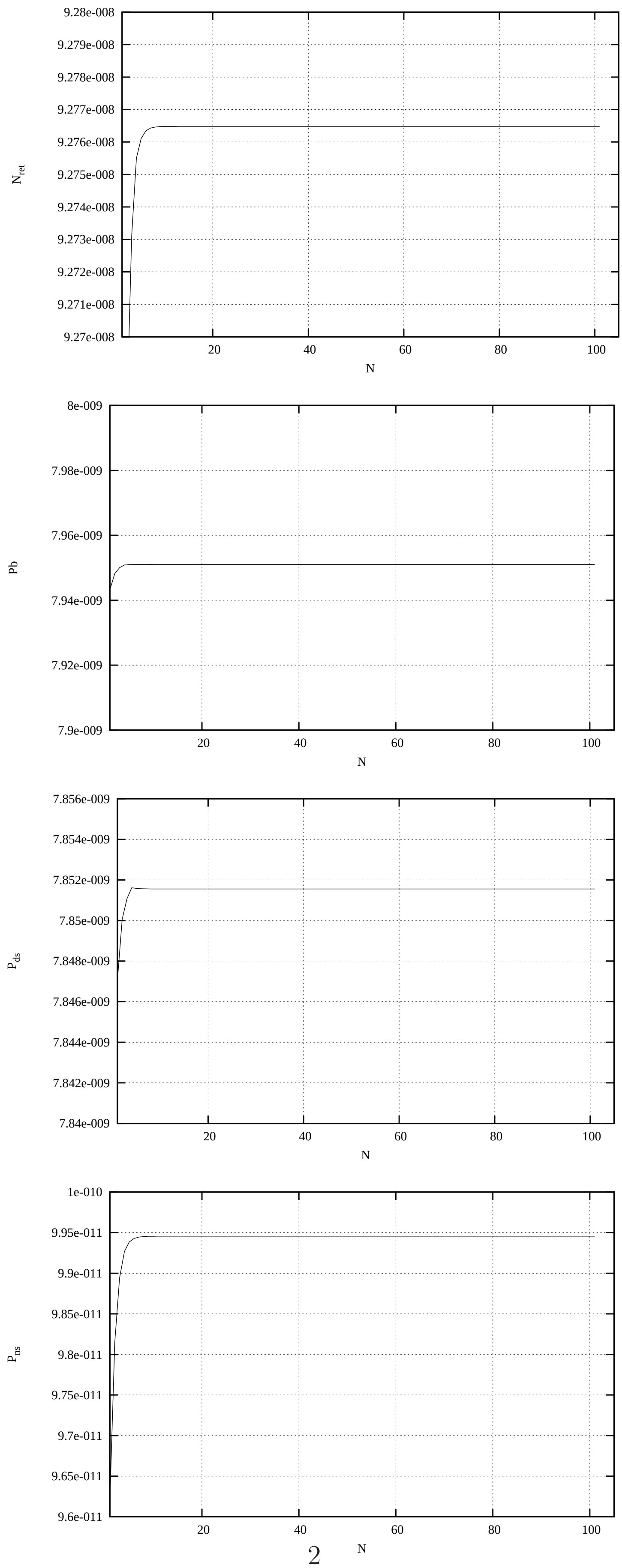

Fig. 1. Performance measures vs N for $c = 50$, $\rho = 0.4$, $\mu = 1/180$, $\mu_r = 0.01$, $P_{im} = 0.2$

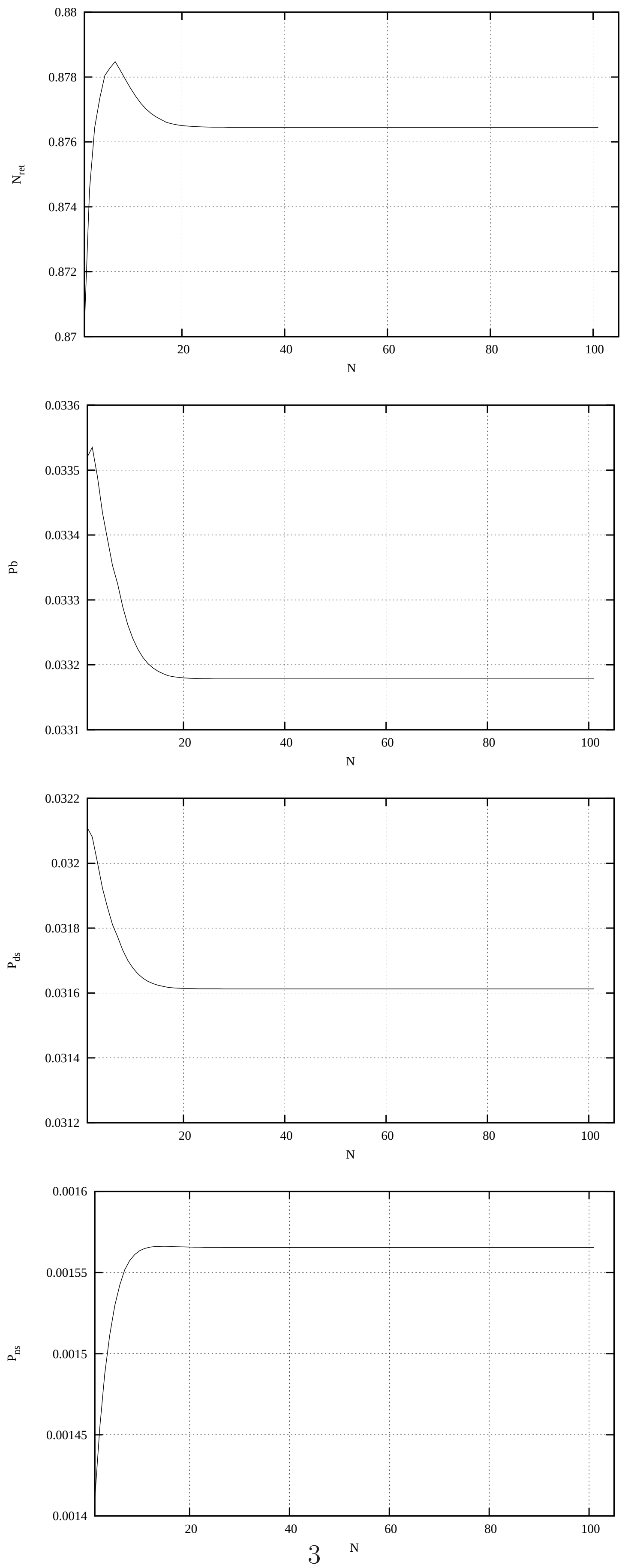

Fig. 2. Performance measures vs N for $c = 50$, $\rho = 0.8$, $\mu = 1/180$, $\mu_r = 0.01$, $P_{im} = 0.2$

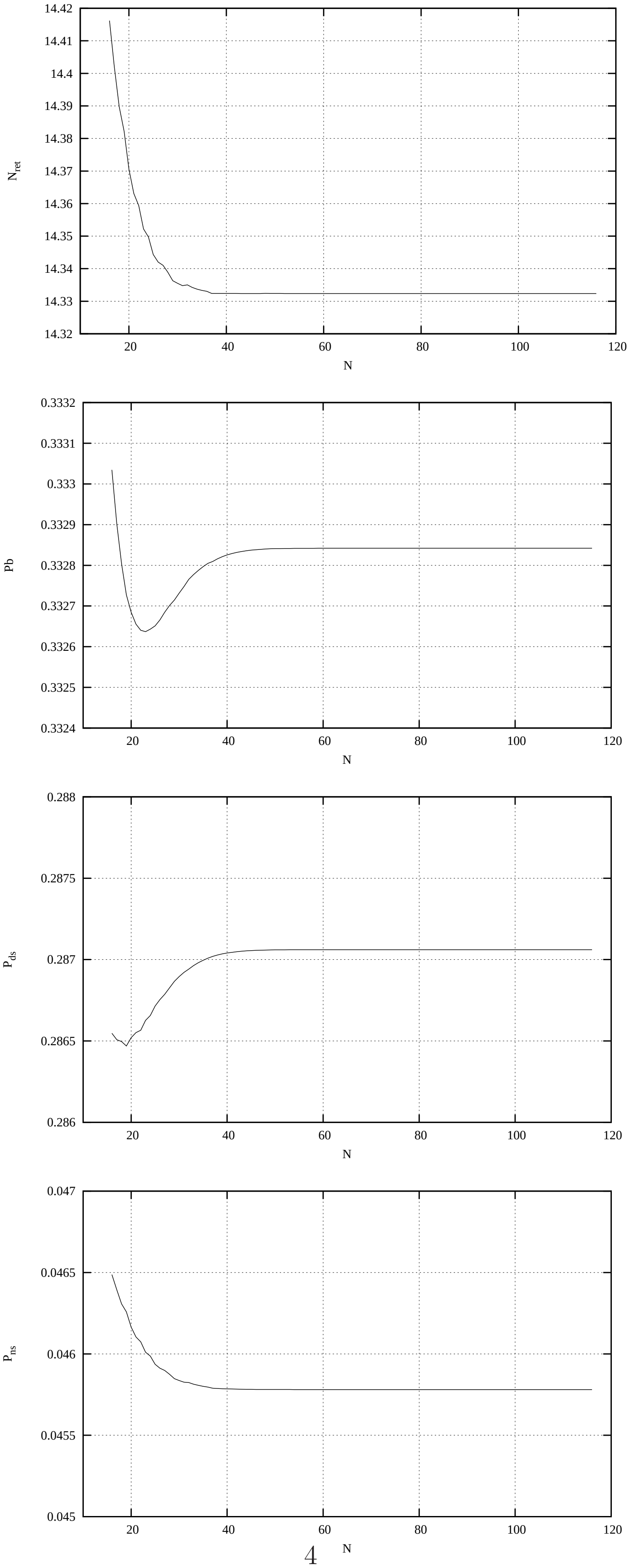

Fig. 3. Performance measures vs N for $c = 50$, $\rho = 1.0$, $\mu = 1/180$, $\mu_r = 0.01$, $P_{im} = 0.2$

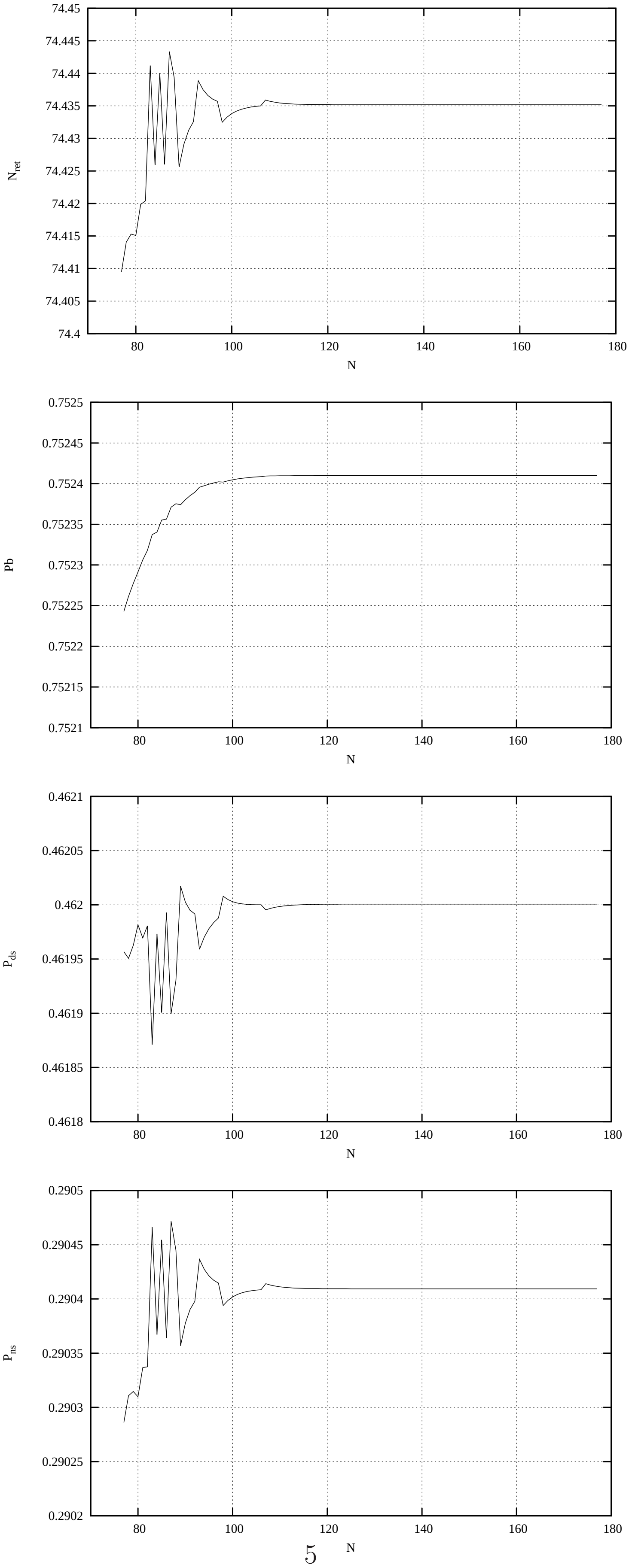

Fig. 4. Performance measures vs N for $c = 50$, $\rho = 1.4$, $\mu = 1/180$, $\mu_r = 0.01$, $P_{im} = 0.2$

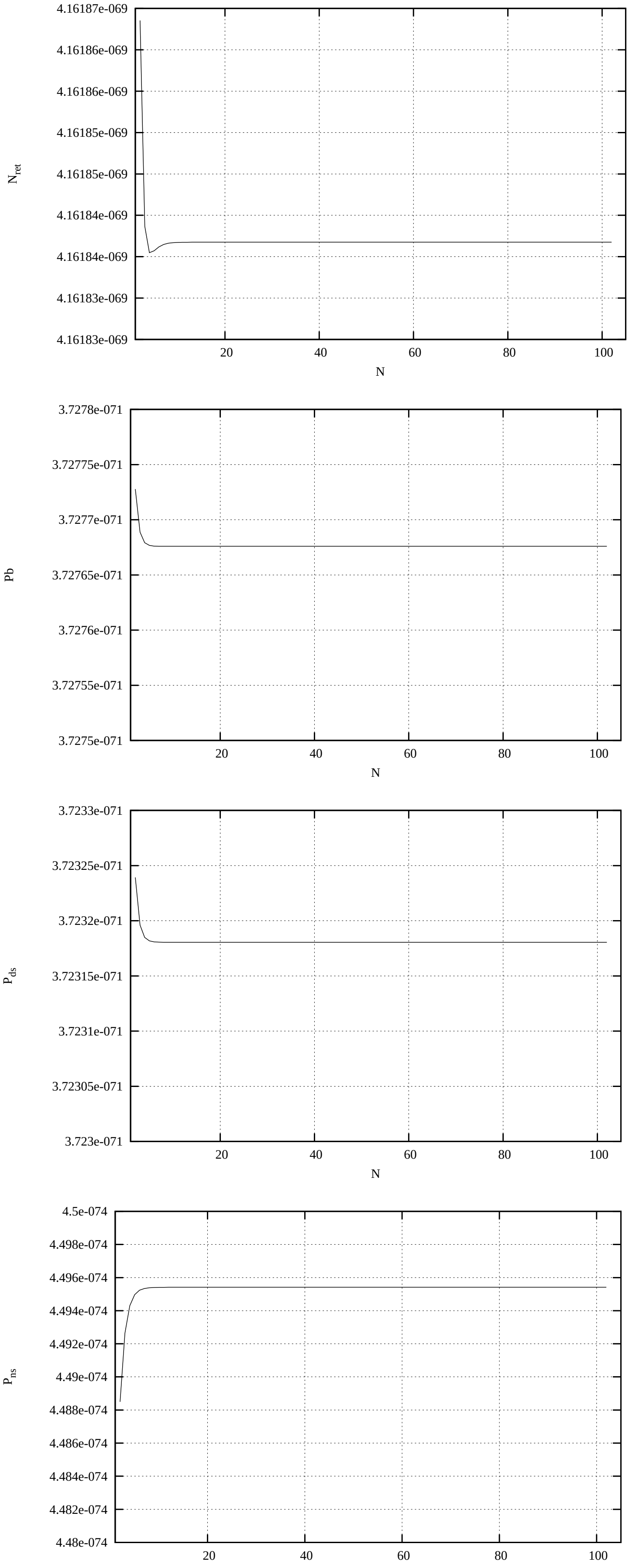

Fig. 5. Performance measures vs N for $c = 500$, $\rho = 0.4$, $\mu = 1/180$, $\mu_r = 0.01$, $P_{im} = 0.2$

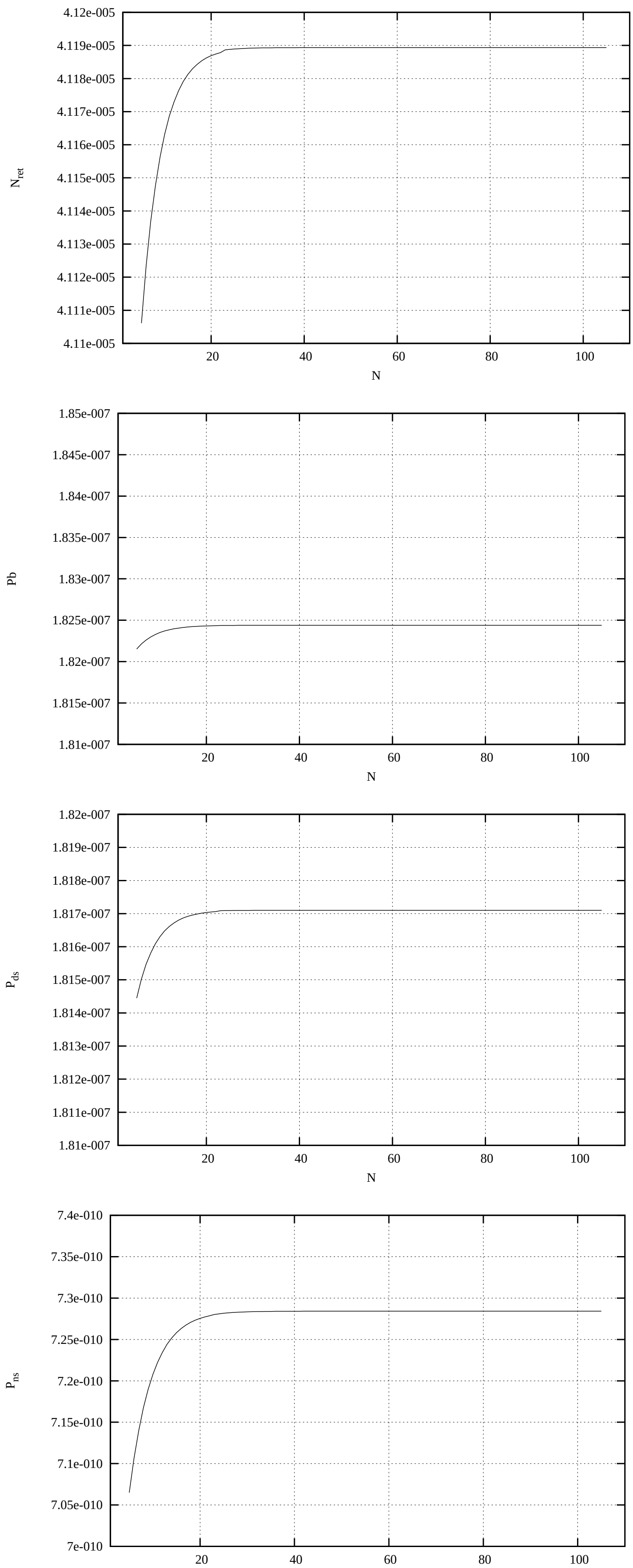

Fig. 6. Performance measures vs N for $c = 500$, $\rho = 0.8$, $\mu = 1/180$, $\mu_r = 0.01$, $P_{im} = 0.2$

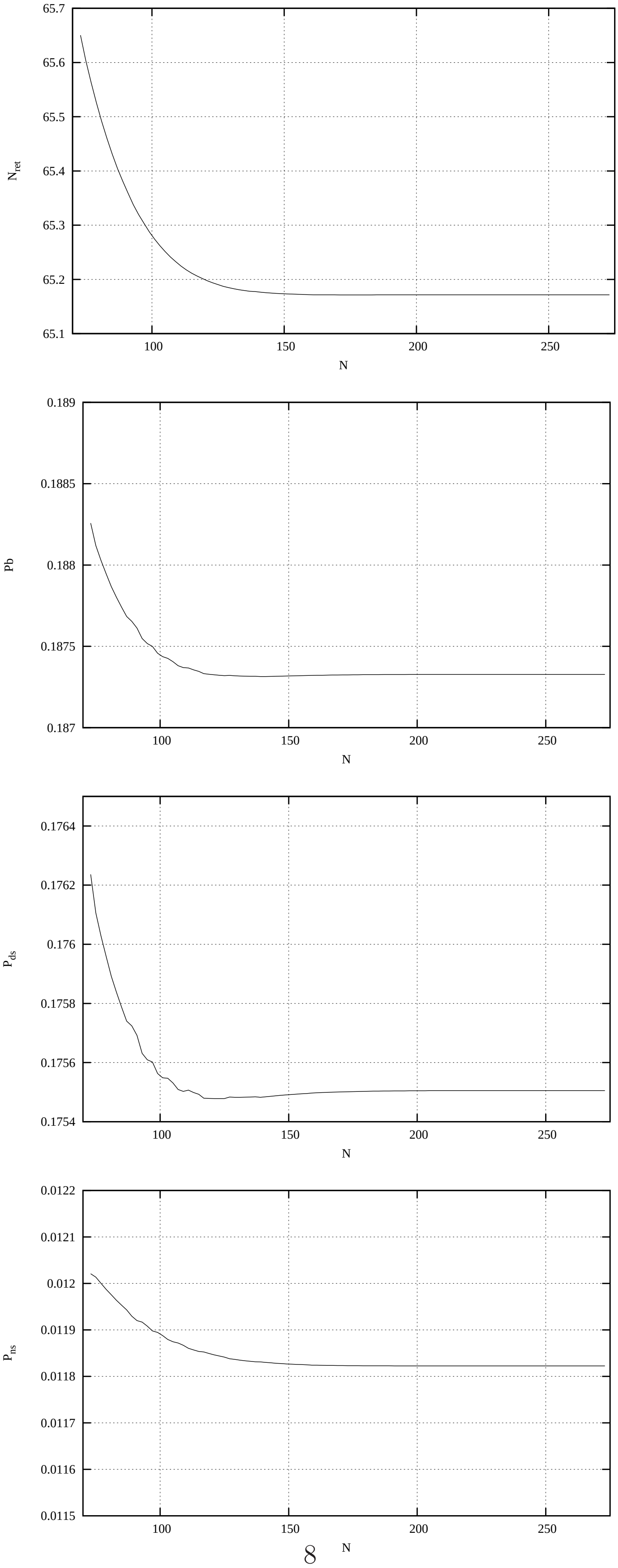

Fig. 7. Performance measures vs N for $c = 500$, $\rho = 1.0$, $\mu = 1/180$, $\mu_r = 0.01$, $P_{im} = 0.2$

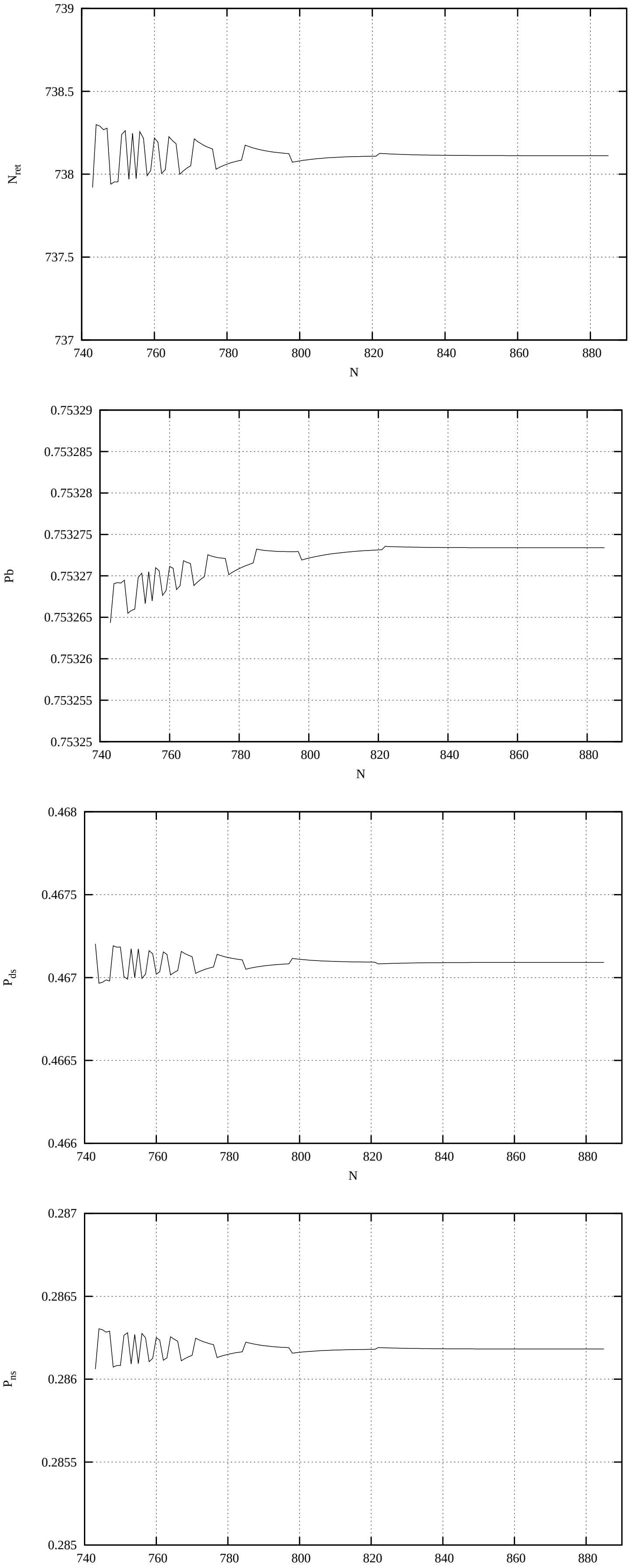

Fig. 8. Performance measures vs N for $c = 500$, $\rho = 1.4$, $\mu = 1/180$, $\mu_r = 0.01$, $P_{im} = 0.2$

Table 1
N and the computational time of the proposed algorithm for $K = 3$, $L = 2$ $\epsilon_p = 10^{-5}$, $\epsilon_M = 10^{-3}$, $\epsilon_r = 10^{-10}$, $\rho = \lambda/(\mu c)$, $\mu = 1/180$, $\mu_r = 0.01$, $P_{im} = 0.2$, $r_{th} = 0.95$

		$c = 50$		$c = 100$		$c = 200$		$c = 500$		$c = 1000$	
ρ		N	Time (s)	N	Time (s)	N	Time (s)	N	Time (s)	N	Time (s)
0.4	N_{ret}	17	0.889	6	0.765	8	1.435	39	16.318	13	13.121
	P_b	16	0.858	4	0.468	8	1.42	39	16.255	13	13.275
	P_{ds}	16	0.873	5	0.609	8	1.42	39	16.38	13	13.166
	P_{ns}	23	1.233	23	2.574	12	2.449	39	16.396	13	13.12
0.8	N_{ret}	38	2.511	29	3.697	46	9.703	35	15.646	13	15.287
	P_b	38	2.574	32	4.57	41	8.408	35	15.647	12	13.744
	P_{ds}	38	2.511	32	4.446	41	8.377	35	15.709	11	11.762
	P_{ns}	47	3.182	32	4.617	46	9.751	36	17.035	40	52.355
1.0	N_{ret}	44	2.855	76	7.504	117	14.133	206	35.491	363	82.619
	P_b	44	2.792	76	7.769	117	14.227	206	35.756	363	82.509
	P_{ds}	44	2.698	76	7.737	124	16.114	206	35.506	363	82.914
	P_{ns}	44	2.777	79	8.129	124	16.255	206	36.099	363	82.477
1.4	N_{ret}	102	3.042	198	7.957	397	26.411	858	81.417	1679	264.001
	P_b	100	2.621	198	8.064	397	26.427	858	81.589	1679	266.122
	P_{ds}	106	3.463	198	8.003	397	26.318	858	81.339	1679	264.781
	P_{ns}	106	3.464	202	9.375	397	26.458	858	81.354	1679	266.013

Table 2
N and the computational time of the proposed algorithm for $K = 4$, $L = 2$, $\epsilon_p = 10^{-5}$, $\epsilon_M = 10^{-3}$, $\epsilon_r = 10^{-10}$, $\rho = \lambda/(\mu c)$, $\mu = 1/180$, $\mu_r = 0.01$, $P_{im} = 0.2$, $r_{th} = 0.95$

		$c = 50$		$c = 100$		$c = 200$		$c = 500$		$c = 1000$	
ρ		N	Time (s)	N	Time (s)	N	Time (s)	N	Time (s)	N	Time (s)
0.4	N_{ret}	23	1.061	7	0.765	11	1.841	43	17.504	14	13.541
	P_b	17	0.78	5	0.484	11	1.809	43	17.487	14	13.588
	P_{ds}	17	0.749	5	0.499	11	1.779	43	17.254	14	13.462
	P_{ns}	23	1.06	36	3.213	14	2.559	43	17.316	14	13.447
0.8	N_{ret}	47	2.808	30	3.713	53	10.639	36	15.694	15	16.614
	P_b	47	2.761	34	4.352	46	8.705	36	15.631	12	12.465
	P_{ds}	47	2.761	34	4.337	46	8.814	36	15.616	12	12.387
	P_{ns}	47	2.793	34	4.305	53	10.608	37	16.926	43	52.057
1.0	N_{ret}	52	2.933	79	7.223	124	14.445	225	44.959	364	95.52
	P_b	52	2.932	79	7.332	124	14.617	225	44.85	364	95.02
	P_{ds}	52	2.933	79	7.363	124	14.633	225	44.975	364	94.584
	P_{ns}	61	3.698	83	7.971	138	18.362	225	44.788	364	94.833
1.4	N_{ret}	106	3.042	202	8.409	404	28.735	863	98.421	1684	332.251
	P_b	102	2.62	202	8.362	404	28.657	863	98.515	1684	331.315
	P_{ds}	106	3.011	202	8.377	404	29.343	863	98.405	1684	330.16
	P_{ns}	106	3.042	202	8.362	404	28.735	863	98.078	1684	332.641

Table 3
N and the computational time of the proposed algorithm for $K = 5$, $L = 2$, $\epsilon_p = 10^{-5}$, $\epsilon_M = 10^{-3}$, $\epsilon_r = 10^{-10}$, $\rho = \lambda/(\mu c)$, $\mu = 1/180$, $\mu_r = 0.01$, $P_{im} = 0.2$, $r_{th} = 0.95$

		$c = 50$		$c = 100$		$c = 200$		$c = 500$		$c = 1000$	
ρ		N	Time (s)	N	Time (s)	N	Time (s)	N	Time (s)	N	Time (s)
0.4	N_{ret}	29	1.389	14	1.295	12	2.122	55	23.447	15	15.351
	P_b	23	1.045	6	0.655	12	2.122	55	23.447	15	15.21
	P_{ds}	23	1.061	6	0.624	12	2.138	55	23.416	15	15.288
	P_{ns}	29	1.341	37	3.447	18	3.245	55	23.588	15	15.288
0.8	N_{ret}	49	2.932	32	3.993	56	11.715	37	17.316	16	18.642
	P_b	49	2.98	45	5.553	53	10.671	37	16.973	13	14.055
	P_{ds}	49	2.979	45	5.491	53	10.686	37	17.051	13	14.04
	P_{ns}	49	2.949	45	5.569	56	12.137	40	19.126	49	59.717
1.0	N_{ret}	54	3.198	83	7.94	138	18.346	229	52.573	386	134.051
	P_b	54	3.198	83	8.003	138	18.58	229	52.65	386	132.102
	P_{ds}	54	3.104	83	8.018	138	18.626	229	52.276	386	133.583
	P_{ns}	54	3.182	83	8.05	140	20.03	229	52.261	386	133.162
1.4	N_{ret}	111	3.573	204	9.547	407	32.947	872	123.319	1688	406.523
	P_b	106	3.01	204	9.454	407	32.76	872	123.287	1688	409.581
	P_{ds}	111	3.51	204	9.531	407	33.166	872	121.665	1688	411.032
	P_{ns}	111	3.573	204	9.563	407	32.838	872	122.321	1688	408.005