Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 186761, 18 pages
doi:10.1155/2012/186761

Research Article

Efficient Method to Approximately Solve Retrial Systems with Impatience

Jose Manuel Gimenez-Guzman,[1] M. Jose Domenech-Benlloch,[2] Vicent Pla,[2] Jorge Martinez-Bauset,[2] and Vicente Casares-Giner[2]

[1] *Departamento Automatica, Universidad de Alcalá, Alcalá de Henares, 28871 Madrid, Spain*
[2] *Departamento Comunicaciones, Universitat Politècnica de València, 46022 Valencia, Spain*

Correspondence should be addressed to Jose Manuel Gimenez-Guzman, josem.gimenez@uah.es

Received 30 June 2011; Accepted 18 October 2011

Academic Editor: Nicola Guglielmi

We present a novel technique to solve multiserver retrial systems with impatience. Unfortunately these systems do not present an exact analytic solution, so it is mandatory to resort to approximate techniques. This novel technique does not rely on the numerical solution of the steady-state Kolmogorov equations of the Continuous Time Markov Chain as it is common for this kind of systems but it considers the system in its Markov Decision Process setting. This technique, known as value extrapolation, truncates the infinite state space using a polynomial extrapolation method to approach the states outside the truncated state space. A numerical evaluation is carried out to evaluate this technique and to compare its performance with previous techniques. The obtained results show that value extrapolation greatly outperforms the previous approaches appeared in the literature not only in terms of accuracy but also in terms of computational cost.

1. Introduction

A common assumption when evaluating the performance of communication systems is that users that do not obtain an immediate service leave the system without retrying. However, due to the increasing number of customers and network complexity, the customer behavior in general, and the retrial phenomenon in particular, may have a nonnegligible impact on the system performance. For example, in mobile cellular networks the importance of the retrial phenomenon has been stressed in [1–3]. An extensive bibliography on retrial queues can be found in [4]. The modeling of repeated attempts has been a subject of numerous investigations, because these systems have a nonhomogeneous and infinite state space. However, it is known that the classical theory [5] is developed for random walks on the semistrip $\{0,\ldots,C\} \times \mathbb{Z}_+$ (being C the number of servers) with infinitesimal transitions subject to conditions of space homogeneity.

When the space-homogeneity condition does not hold, for example, in the case of retrial queues, the problem of calculating the equilibrium distribution has not been solved beyond approximate techniques when the number of servers is higher than two [6]. In particular, Marsan et al. [7] propose a well-known approximate technique for its analysis. In [8], a generalization of the approximate technique in [7] was proposed, showing a substantial improvement in the accuracy at the expense of a marginal increase of the computational cost. Those approximations are based on the reduction of an infinite state space to a finite one by aggregating states. Other solutions maintain the infinite state space but homogenize it beyond a given level in order to solve the system. These later models are known as generalized truncated models [6] and usually present the advantage of providing a much better accuracy than the finite methodologies [9]. In this category we find the models proposed by Falin [10], by Neuts and Rao [11], and by Artalejo and Pozo [6]. All these approaches rely on the numerical solution of the steady-state Kolmogorov equations of the Continuous Time Markov Chain (CTMC) that describes the system under consideration.

Very recently, however, an alternative approach for evaluating infinite state space Markov processes has been introduced by Leino et al. [12–14]. The new technique, named value extrapolation, does not rely on solving the global balance equations. This technique considers the system in its MDP (Markov Decision Process) setting and solves the expected value from the Howard equations written for a truncated state space. Instead of a simple truncation, the relative values of states just outside the truncated state space are estimated using a polynomial extrapolation based on the states inside, obtaining a closed system. Therefore, we can compute any performance parameter as far as we are capable to express it as the expected value of a random variable that is function of the system state.

So far the value extrapolation technique has been applied to multiclass single server queues showing very promising results. It must be noted that a key aspect on the application of value extrapolation lies on the election of the extrapolating function for the relative state values. Indeed, in [14] the authors show that by selecting an appropriate polynomial function the technique yields exact results for the moments of the queue length in a multiclass Discriminatory Processor-Sharing (DPS) system. Unfortunately, the appropriateness of the functional form of the extrapolation depends on the system and also on the revenue function, that is, the performance parameter we are interested in. Hence, there is no universal good choice for the extrapolating function. In this paper we address the application of the value extrapolation technique to an important class of queuing systems, for example, retrial queues, which are essentially different of the type of queues to which this technique has been applied. A potential drawback of value extrapolation compared to conventional state space truncation methods is that, since the stationary state probabilities are not obtained, if one want to compute several performance parameters, the technique has to be applied once per each of them. We apply well-known linear algebra algorithms to compute several performance parameters simultaneously, and through some series of numerical examples we show that, at least for the type of system that we are studying, the relative impact in terms of computational cost is marginal.

The application of the value extrapolation technique has only addressed problems in which relative state values are expected to follow a polynomial tendency. In this paper we develop the value extrapolation technique to solve a multiserver retrial system, addressing also the drawback of computing only a single performance parameter every time the technique is used.

In a first part of the paper, we develop the analytical part of the technique, defining the associated Howard equations of the model and the revenue functions. In a second part,

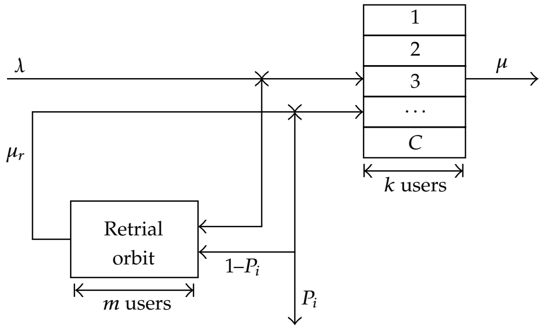


Figure 1: Retrial model under study.

we compare our technique with other previously proposed techniques in terms of accuracy and computational cost. Results show that the proposed technique clearly outperforms the rest of the studied techniques in terms of computational cost, and this improvement is even much higher in terms of accuracy.

The rest of the paper is structured as follows. Section 2 describes the system under study, while Section 3 introduces the solving technique used. In Section 4, the numerical analysis is carried out, evaluating the value extrapolation technique and comparing it with other previous solving techniques proposed in the literature. Final remarks and a summary of results are provided in Section 5.

2. System Model

The system under study is a generic retrial system including user impatience, that is, users leave the system with certain probability after a nonsuccessful retrial. A diagram of the system is shown in Figure 1. Users arrive following a Poisson process with rate λ to a system with C servers and request an exponentially distributed service time with rate μ. Without loss of generality, we consider that each user occupies one server. When a new request finds all servers occupied, it joins the retrial orbit with probability 1. After an exponentially distributed time of rate μ_r, this session retries. The reattempt is successful if it finds a free server. Otherwise, the user leaves the system with probability P_i or returns to the retrial orbit with probability $(1-P_i)$, starting the retrial procedure again. Note that we consider an infinite capacity for the retrial orbit.

The model considered can be represented as a bidimensional CTMC, $S(t) = \{K(t), M(t)\}$, where $K(t)$ is the number of sessions being served and $M(t)$ the number of users in the retrial orbit at time t. The state space of the process is defined by

$$\mathcal{S} := \{s = (k, m) : k \leq C; m \in \mathbb{Z}_+\}. \tag{2.1}$$

Figure 2 shows the transition diagram of such system, showing two important properties in the dimension corresponding to the number of users in the retrial orbit: on the one hand its infinite cardinality and on the other hand its space-heterogeneity produced by the fact that retrial rate depends on the number of customers in the retrial orbit.

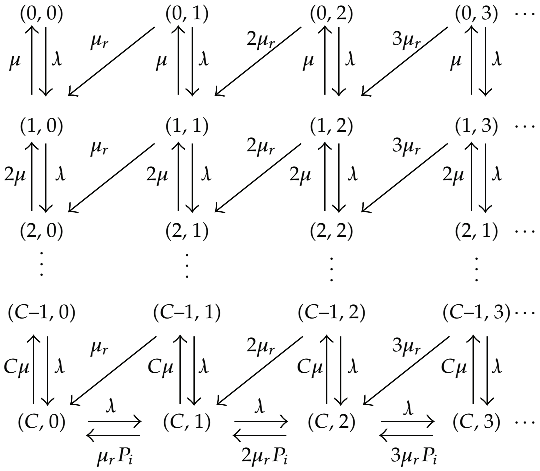


Figure 2: Transition diagram.

3. Solving Technique

In this section we develop the value extrapolation technique for the model presented in Section 2. Additionally, we present some particularities that should be taken into account when using this technique.

3.1. MDP Settings

As it has been aforementioned, the problem under interest has not a closed form solution when $C > 2$ [6], so approximation techniques are mandatory. To the best of our knowledge, all the approximate techniques that have appeared in literature compute the steady state probabilities ($\pi(s)$) using the balance equations in order to compute the desired performance parameters, that is, solving the linear system of equations:

$$\pi(s) \sum_{s' \neq s} q_{ss'} = \sum_{s' \neq s} \pi(s') q_{ss'} \quad \forall s \in \mathcal{S}, \tag{3.1}$$

along with the normalization condition $\sum_s \pi(s) = 1$, where $q_{ss'}$ represents the transition rate from state s to s'.

Notwithstanding, value extrapolation is not based on the probability of being in a certain state, but on a new metric called relative state values. Relative state values appear when we consider the system in the setting of an MDP. Formally, an MDP can be defined as a tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}\}$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $\mathcal{P}$ is a state transition function, and $\mathcal{R}$ is a revenue function. The state of the system can be controlled by choosing actions a from $\mathcal{A}$, influencing in this way the state transitions. The transition function $\mathcal{P} : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}_+$ specifies the transition rate between a pair of states when a certain action is taken at the original state. The first characteristic of the value extrapolation technique is the necessity of the definition of a revenue function that must be a function of the system state, that is, $r(s)$. Following the definition of the revenue function for every state, in the steady state, a mean revenue rate of the entire process can be introduced as $r = \sum_{s \in \mathcal{S}} \pi(s) r(s)$. In the value extrapolation technique, the revenue function $\mathcal{R}$ has to be defined so that the resulting average revenue r coincides with the desired performance metric.

Once we have defined the MDP framework as well as the revenue function, we are in a position to define the relative state values. It is obvious that after performing an action in state s the system will collect a revenue for that action $(r(s))$, but as the number of transitions increases, the average revenue collected converges to r. The relative state value $(v(\mathrm{s}))$ is equal to the difference between the total revenue incurred when the system starts at state s and the total revenue incurred in a system for which the revenue rate at all states is r:

$$v(s) = E\left[\int_0^\infty (r(S(t)) - r)dt \mid S(0) = s\right]. \tag{3.2}$$

The Howard equations relate revenues, relative state values, and transition rates:

$$r(s) - r + \sum_{s'} q_{ss'}\left(v(s') - v(s)\right) = 0 \quad \forall s \in \mathcal{S}. \tag{3.3}$$

The Howard equations represent the *policy evaluation* phase of the well-known *policy iteration* algorithm, the most widespread dynamic programming technique, proposed in [15]. The Howard equations that correspond to the system under study are

$$\begin{aligned}
&r(k,m) - r + \lambda(v(k+1,m) - v(k,m)) + k\mu(v(k-1,m) - v(k,m))\\
&\quad + m\mu_r(v(k+1,m-1) - v(k,m)) = 0 \quad \text{if } k < C,\\
&r(C,m) - r + \lambda(v(C,m+1) - v(C,m)) + C\mu(v(C-1,m) - v(C,m))\\
&\quad + m\mu_r P_i(v(C,m-1) - v(C,m)) = 0 \quad \text{if } k = C.
\end{aligned} \tag{3.4}$$

As we can observe the number of states is infinite because m can take any value in $\mathbb{Z}_+$, thus we need to truncate the state space to $\widehat{\mathcal{S}}$. In our case, the truncated state space is defined by

$$\widehat{\mathcal{S}} := \{s = (k,m) : k \leq C; m \leq Q\}. \tag{3.5}$$

In general, Q is known as the truncation level. As we choose a higher value of Q, we can expect a higher accuracy as the system is more similar to the original one, but we will have a higher computation cost too. Therefore, the objective will be to achieve a certain accuracy with the minimum value of Q.

There will be as many Howard equations as number of states, $|\widehat{\mathcal{S}}|$. The number of unknowns will be the $|\widehat{\mathcal{S}}|$ relative state values plus the expected revenue r, that is, $|\widehat{\mathcal{S}}| + 1$ unknowns. However, as only the differences in the relative values appear in the Howard equations, we can set $v(0) = 0$, so we will have a solvable linear system of equations with the same number of equations as unknowns.

3.2. Polynomial Fitting

The traditional truncation sets $q_{ss'} = 0$ for all $s' \notin \widehat{\mathcal{S}}$, but value extrapolation performs a more efficient truncation. Basically, value extrapolation considers the relative state values outside $\widehat{\mathcal{S}}$ that appear in the Howard equations as an extrapolation of some relative state values inside $\widehat{\mathcal{S}}$. As we truncate the retrial orbit dimension beyond a value Q, the value extrapolation technique uses the state value of some states in $\widehat{\mathcal{S}}$ to approximate $v(C, Q+1)$, which is expected to improve the accuracy significantly, as it is better than ignoring these relative state values. Note that if extrapolation yielded the exact value for those states outside $\widehat{\mathcal{S}}$, the results obtained by solving the truncated model would be exact. Also note that including value extrapolation neither increases the computational cost nor increases the number of Howard equations, which remains equal to $|\widehat{\mathcal{S}}| = (C+1) \times (Q+1)$.

Summarizing, the objective of value extrapolation is to find an extrapolation function that fits with some points in $\widehat{\mathcal{S}}$ so that it approximates also points outside $\widehat{\mathcal{S}}$. It is important to choose a fitting function that makes the Howard equations remain a closed system of linear equations. The most common fitting functions that fulfill that condition are the polynomials. We can use all the states in $\widehat{\mathcal{S}}$ into the fitting procedure (global fitting) or, what is most commonly used, only a subset ($\mathcal{S}_f$) of them (local fitting).

For the sake of simplicity, in the following description we will assume there exists a mapping $\mathcal{W}$ from the two-dimensional set of states into a single-dimensional set, for example, the real numbers: $\mathcal{W} : \widehat{\mathcal{S}}_f \to \mathbb{R}$. Hence, below we deal with states as if they were real values given as $w = W(s)$. The specific mapping used for the model under study is specified later on.

The choice of $\mathcal{W}$ will highly depend on the states we want to extrapolate its relative state value. Note also that the function $f(w)$ and the set $\widehat{\mathcal{S}}_f$ need to be chosen so that the parameters in $f(w)$ have unambiguous values, that is, in the case of choosing a polynomial as the fitting function, the number of different points in $\widehat{\mathcal{S}}_f$ has to be equal or greater than the number of coefficients in the polynomial. In general, the procedure to compute the coefficients of the fitting polynomial a_i consists in minimizing the least mean squared error

$$E = \sum_{w \in \mathcal{W}} \left(f(w) - v(w)\right)^2. \tag{3.6}$$

Then the optimal values for the a_i's can be computed by solving the equations

$$\frac{\partial E}{\partial a_i} = 0 \quad \forall i. \tag{3.7}$$

In our case, we are using as many points as the number of parameters of the fitting polynomial, so the fitting procedure is an ordinary polynomial interpolation and $E = 0$, that is, all the considered points will lie in the curve of the polynomial. In this case, the problem can be formulated as follows. Given a set of $n = |\mathcal{W}(\widehat{\mathcal{S}}_f)| = |\widehat{\mathcal{S}}_f|$ points $(w_0, v(w_0)), \dots, (w_{n-1}, v(w_{n-1}))$, where there are not two identical w_i, we can determine an $(n-1)$-th degree polynomial so that $f(w_i) = v(w_i)$ for $i = 0, \dots, n-1$, where

$$f(w) = a_0 + a_1 w + a_2 w^2 + \cdots + a_{n-1} w^{n-1}. \tag{3.8}$$

The interpolating polynomial satisfies the following n linear equations:

$$f(w_i) = a_0 + a_1 w_i + a_2 w_i^2 + \cdots + a_{n-1} w_i^{n-1} = v(w_i) \quad i = 0, \ldots, n-1, \tag{3.9}$$

which in a matrix form are

$$Aa = \begin{bmatrix} 1 & w_0 & \ldots & w_0^{n-1} \\ 1 & w_1 & \ldots & w_1^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & w_{n-1} & \ldots & w_{n-1}^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} v(w_0) \\ v(w_1) \\ \vdots \\ v(w_{n-1}) \end{bmatrix} = b. \tag{3.10}$$

The matrix of coefficients of this system (A) is a Vandermonde matrix, whose determinant is nonvanishing and therefore A is invertible. Thus, there always exists a unique solution to the considered linear system of equations or, equivalently, there exists a unique polynomial that goes through all the n points. However, Vandermonde matrices are often badly conditioned, specially if some w_i are very close, so the procedure to compute the fitting polynomial is also badly conditioned. It is important to note that the unicity of the fitting polynomial does not mean that it cannot be written in a basis different from the standard basis. More concretely in this work we have used the Lagrange basis.

For the considered interpolation problem, the polynomial in its Lagrange setting is a linear combination

$$L(w) = \sum_{j=0}^{n-1} v(w_j) \ell_j(w) \tag{3.11}$$

of Lagrange basis polynomials

$$\ell_j(w) = \prod_{\substack{i=0 \\ i \neq j}}^{n-1} \frac{w - w_i}{w_j - w_i} = \frac{w - w_0}{w_j - w_0} \cdots \frac{w - w_{j-1}}{w_j - w_{j-1}} \frac{w - w_{j+1}}{w_j - w_{j+1}} \cdots \frac{w - w_{n-1}}{w_j - w_{n-1}}. \tag{3.12}$$

For the truncated problem of interest and as shown in Figure 3, we will have a Howard equation in which appears $v(C, Q+1)$, that is a state value of a state that does not belong to $\widehat{\mathcal{S}}$. Therefore, we must approximate the value $v(C, Q+1)$ by using some relative state values of states belonging to $\widehat{\mathcal{S}}$. It is important to emphasize that for the extrapolation of $v(C, Q+1)$ we only use states from the last row of the model shown in Figure 3, that is, states of the form $s = (C, m)$, with varying m. With this choice, we define the mapping $\mathcal{W}$ as $\mathcal{W}((C, m)) = m$.

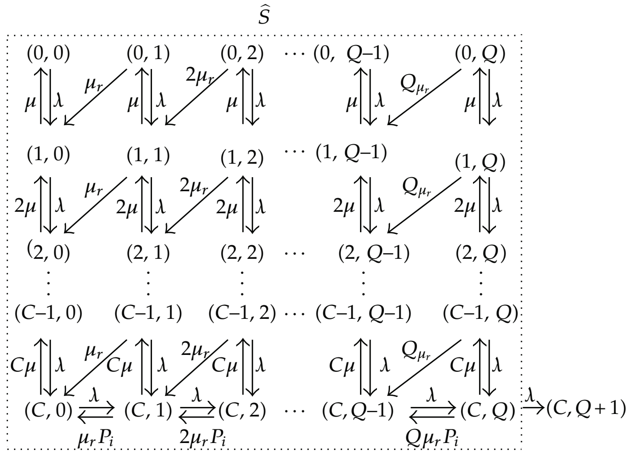


Figure 3: Truncated model and states that appear in Howard equations outside the truncated model.

Moreover, we use an $(n-1)$-th degree polynomial that interpolates the n points in $\mathcal{S}_f := \{s_i = (C, Q-i) \mid i = 0, \ldots, n-1\}$ and then $\mathcal{W}(\mathcal{S}_f) = \{w_i = Q-i \mid i = 0, \ldots, n-1\}$:

$$\begin{aligned}
w_0 &= Q && \longrightarrow && v(w_0) = v(C, Q), \\
w_1 &= Q-1 && \longrightarrow && v(w_1) = v(C, Q-1), \\
& && \vdots && \\
w_j &= Q-j && \longrightarrow && v(w_j) = v(C, Q-j), \\
& && \vdots && \\
w_{n-1} &= Q-(n-1) && \longrightarrow && v(w_{n-1}) = v(C, Q-(n-1)).
\end{aligned} \tag{3.13}$$

This way, the general form of the extrapolation state when using an $(n-1)$-th degree polynomial is

$$v^{(n)}(C, Q+1) = L^{(n)}(Q+1) = \sum_{j=0}^{n-1} v(C, Q-j)\ell_j(Q+1). \tag{3.14}$$

For example, in the case of linear extrapolation $(n = 2)$, we use $(Q, v(C, Q))$ and $(Q-1, v(C, Q-1))$, having

$$\begin{aligned}
v^{(2)}(C, Q+1) = L^{(2)}(Q+1) &= v(C, Q)\ell_0(Q+1) + v(C, Q-1)\ell_1(Q+1) \\
&= v(C, Q)\frac{(Q+1)-(Q-1)}{Q-(Q-1)} + v(C, Q-1)\frac{(Q+1)-Q}{(Q-1)-Q} \\
&= 2v(C, Q) - v(C, Q-1).
\end{aligned} \tag{3.15}$$

Table 1: Revenue function definition.

Blocking probability	P_{b}	$r(k,m) = 1$ for $k = C$, for all m $r(k,m) = 0$ otherwise
Nonservice probability	P_{ns}	$r(k,m) = m\mu_r P_i/\lambda$ for $k = C$, for all m $r(k,m) = 0$ otherwise
Mean number of users retrying	N_{ret}	$r(k,m) = m$ for all k, for all m
Probability of being in state (K,M)	$\pi(K,M)$	$r(k,m) = 1$ for $k = K$, $m = M$ $r(k,m) = 0$ otherwise
Probability of having K busy servers	$B(K)$	$r(k,m) = 1$ for $k = K$, for all m $r(k,m) = 0$ otherwise

Following a similar procedure, we can obtain the next relationship for $n = 3$ and $n = 4$:

$$\begin{aligned} v^{(3)}(C,Q+1) &= 3v(C,Q) - 3v(C,Q-1) + v(C,Q-2), \\ v^{(4)}(C,Q+1) &= 4v(C,Q) - 6v(C,Q-1) + 4v(C,Q-2) - v(C,Q-3). \end{aligned} \tag{3.16}$$

In general, for $(n-1)$-th degree polynomials and using the Lagrange basis to reduce the complexity of the procedure, a simple closed-form expression for the extrapolated value can be obtained by

$$v^{(n)}(C,Q+1) = \sum_{k=0}^{n-1} (-1)^k \binom{n}{k+1} v(C,Q-k), \tag{3.17}$$

where n is the number of coefficients taken for Lagrange polynomials.

3.3. *Revenue Function*

As performance parameters are not computed from the steady state probabilities as usual, it is important to explain more carefully how they are computed. By definition, $r(s)$ is the revenue rate obtained when the system is in state s. Therefore, we must define the revenue as the performance parameter we want to compute. The effect of that action is that the computed r will be the performance parameter we are looking for. Additionally, the inputs $r(s)$ in the Howard equations must be properly set. Table 1 gives several examples on how $r(s)$ can be set in order to obtain certain performance parameters such as: blocking probability $P_{\mathrm{b}} = \mathrm{Prob}\{K = C\}$, mean number of users in the retrial orbit $N_{\mathrm{ret}} = E[M]$, nonservice probability P_{ns} (probability of a user leaving the system due to impatience without obtaining service), probability of being in a certain state $\pi(K,M)$, and probability of having K busy servers $B(K)$.

As an example, we focus on the blocking probability and we define the revenue function to be 1 in those states in which an attempt is blocked, that is, when $r(C,m) = 1$, for all m, and 0 in the rest of states, $r(k,m) = 0$, $k \neq C$, for all m.

3.4. Effect of the Value Extrapolation into the Howard Equations

In our problem, and as mentioned above, we will only have to replace $v(C,Q+1)$ by its approximate value in the Howard equation that corresponds to the state $v(C,Q)$. As an example, if we use linear extrapolation ($n=2$), that equation becomes

$$\begin{aligned} r(C,Q) - r + v(C,Q)\left(-\lambda - C\mu - QP_i\mu_r\right) + \lambda v(C,Q+1) + C\mu v(C-1,Q) \\ + QP_i\mu_r v(C,Q-1) = r(C,Q) - r + v(C,Q)\left(\lambda - C\mu - QP_i\mu_r\right) + C\mu v(C-1,Q) \\ + \left(QP_i\mu_r - \lambda\right) v(C,Q-1) = 0. \end{aligned} \tag{3.18}$$

As $v(C,Q+1)$ no longer appears in the Howard equations, the linear system of equations we have consists of $(C+1)\times(Q+1)$ equations with the same number of unknowns. This system can be expressed in matrix form for simplicity reasons. Therefore, the system can be seen as $\mathbf{xT}=\mathbf{b}$, where $\mathbf{x}$ is a vector with the $(C+1)\times(Q+1)$ unknowns (r and the relative state values $v(s)$) and $\mathbf{b}$ are the negative revenue rates for the different states:

$$\begin{aligned} \mathbf{x} &= \left[r \quad v(0,1) \quad \cdots \quad v(0,Q) \quad v(1,0) \quad \cdots \quad v(C,Q)\right], \\ \mathbf{b} &= \left[-r(0,0) \quad -r(0,1) \quad \cdots \quad -r(C,Q)\right]. \end{aligned} \tag{3.19}$$

Matrix $\mathbf{T}$ represents the matrix of coefficients and can be constructed making all the elements in the first row of matrix $\mathbf{T_0}$ equal to −1.

Matrix $\mathbf{T_0}$ is given by

$$\mathbf{T_0} = \begin{bmatrix} \mathbf{A}_1^0 & \mathbf{A}_0^0 & \cdots & 0 & 0 \\ \mathbf{A}_2^1 & \mathbf{A}_1^1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \mathbf{A}_1^{C-1} & \mathbf{A}_0^{C-1} \\ 0 & 0 & \cdots & \mathbf{A}_2^{C} & \mathbf{A}_1^{C} \end{bmatrix}, \tag{3.20}$$

where the submatrices are defined as

$$\begin{aligned} \mathbf{A}_0^k &= (k+1)\mu\mathbf{I}, \quad \text{for } 0 \le k \le (C-1), \\ \mathbf{A}_2^k &= \begin{bmatrix} \lambda & \mu_r & 0 & \cdots & 0 & 0 \\ 0 & \lambda & 2\mu_r & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda & Q\mu_r \\ 0 & 0 & 0 & \cdots & 0 & \lambda \end{bmatrix}, \quad \text{for } 1 \le k \le C, \end{aligned}$$

$$\mathbf{A}_1^k = \begin{bmatrix} \alpha & 0 & 0 & \cdots & 0 \\ 0 & \alpha-\mu_r & 0 & \cdots & 0 \\ 0 & 0 & \alpha-2\mu_r & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha-Q\mu_r \end{bmatrix}, \quad \text{for } 0 \le k \le (C-1),\ \alpha = -\lambda - k\mu. \tag{3.21}$$

When $k = C$, using linear ($n = 2$) and quadratic ($n = 3$) extrapolation, we obtain, respectively,

$$\begin{gathered} \mathbf{A}_1^C = \begin{bmatrix} \beta & P_i\mu_r & \cdots & 0 & 0 \\ \lambda & \beta-P_i\mu_r & \cdots & 0 & 0 \\ 0 & \lambda & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \beta-(Q-1)P_i\mu_r & QP_i\mu_r-\lambda \\ 0 & 0 & \cdots & \lambda & \lambda-C\mu-QP_i\mu_r \end{bmatrix}, \\ \mathbf{A}_1^C = \begin{bmatrix} \beta & P_i\mu_r & \cdots & 0 & 0 \\ \lambda & \beta-P_i\mu_r & \cdots & 0 & 0 \\ 0 & \lambda & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & (Q-1)P_i\mu_r & \lambda \\ 0 & 0 & \cdots & \beta-(Q-1)P_i\mu_r & QP_i\mu_r-3\lambda \\ 0 & 0 & \cdots & \lambda & 2\lambda-C\mu-QP_i\mu_r \end{bmatrix}, \end{gathered} \tag{3.22}$$

where $\beta = -\lambda - C\mu$.

In general, if the extrapolation is done with $n \le Q+1$ points, the matrix $\mathbf{A}_1^C$ is given as

$$\mathbf{A}_1^C = \begin{bmatrix} \beta & P_i\mu_r & \cdots & 0 & \lambda c_Q^{(n)} \\ \lambda & \beta-P_i\mu_r & \cdots & 0 & \lambda c_{Q-1}^{(n)} \\ 0 & \lambda & \cdots & 0 & \lambda c_{Q-2}^{(n)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & (Q-1)P_i\mu_r & \lambda c_2^{(n)} \\ 0 & 0 & \cdots & \beta-(Q-1)P_i\mu_r & QP_i\mu_r+\lambda c_1^{(n)} \\ 0 & 0 & \cdots & \lambda & -\lambda-C\mu-QP_i\mu_r+\lambda c_0^{(n)} \end{bmatrix}, \tag{3.23}$$

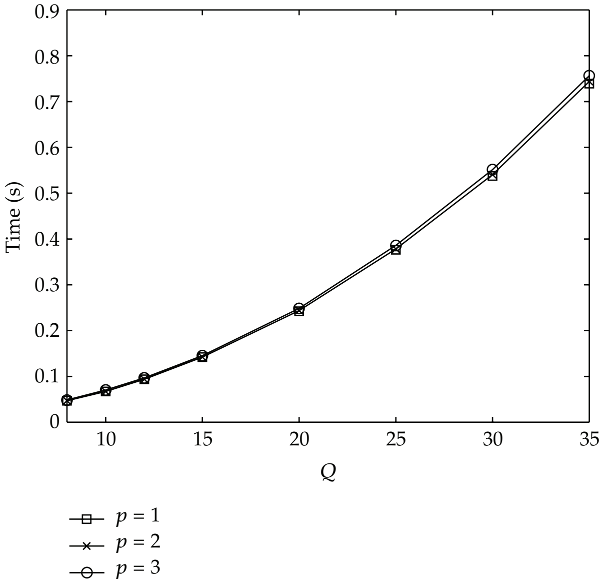


Figure 4: Computation cost when solving p performance parameters simultaneously.

where

$$c_l^{(n)} = \begin{cases} (-1)^l \begin{pmatrix} n \\ l+1 \end{pmatrix} & \text{if } \mathrm{l} < n, \\ 0 & \text{if } l \geq n. \end{cases} \tag{3.24}$$

Note that the size of matrix $\mathbf{T}$ does not depend on the order of the polynomial used to perform the extrapolation; only the last column in the matrix A_1^C depends on the polynomial adjustment. This characteristic has the advantage that there will not be any difference in the computation cost when using higher order of extrapolation.

The main drawback of the value extrapolation technique is that this technique is only able to compute one performance parameter each time we solve the system. Notwithstanding, we can overcome this drawback in the following way. In a general manner, the solution of the system $\mathbf{xT} = \mathbf{b}$ can be obtained using the inverse matrix of $\mathbf{T}$ by doing $\mathbf{x} = \mathbf{bT^{-1}}$. Note also that choosing a different performance parameter to solve will only affect to the values in $\mathbf{b}$. Therefore, computing a second performance parameter will only increase the computation expenses by the cost of the product $\mathbf{bT^{-1}}$, as the rest of the process (specifically the computation of the inverse matrix $\mathbf{T^{-1}}$) is solved only once. Similarly, we can compute several performance parameters with a marginal increase in the computation cost using LU factorization, as the first part of the procedure (the factorization, which represents the most computationally expensive part) is done only once for the $\mathbf{T}$ matrix. This characteristic of the value extrapolation technique can be observed in Figure 4, where we show that the computation time (results have been obtained using Matlab running on an Intel Pentium IV 3 GHz) is only marginally increased when we compute additional performance parameters.

4. Results

In order to evaluate and compare the proposed technique, we have studied its performance in several scenarios. Letting $\rho = \lambda/(C\mu)$, we have studied different system loads by modifying λ and keeping $C = 50$ resource units and $\mu^{-1} = 180\,\text{s}$. The retrial phenomenon has been configured with $\mu_r^{-1} = 100\,\text{s}$ and $P_i = 0.2$. Although only one configuration of the retrial orbit has been chosen, there will be fairly different working points, as the system load is widely modified.

For obtaining the results, we have used the relative error of different performance parameters, defined for a generic performance parameter Ψ by $\epsilon_\Psi = |\Psi^{\text{approx}} - \Psi^{\text{exact}}|/\Psi^{\text{exact}}$. In order to obtain an accurate enough estimate of Ψ which can be used as Ψ^{exact}, we ran all techniques with increasing and sufficiently high values of Q so that the value of Ψ had stabilized up to the 14th decimal digit. As expected all techniques converged to the same value in the performance parameters under study, $\Psi \in \{P_\text{b}, P_\text{ns}, N_\text{ret}\}$.

4.1. Value Extrapolation Evaluation

Table 2 shows the minimum value of Q needed to obtain a relative error lower than 10^{-8} for different performance parameters and loads (columns) and for different orders of the extrapolation polynomials (rows). Note that VEx denotes the use of an extrapolation polynomial of order $x = (n - 1)$. The number in bold indicates the lowest truncation level of all the polynomials studied. Finally, the last row of Table 2 shows the exact value of the studied performance parameter for that scenario.

From Table 2 we conclude that there is not a clear choice in the order of the best polynomial. In general, neither the lowest nor the highest order polynomials are recommendable, so we recommend to use the intermediate cases. Furthermore, the fact that using VEx enforces us to use a model with $Q \geq x$ (see Section 3.2) must be considered in the choice of the polynomial. For that reason we can conclude that, for the problem and scenario of interest and for the relative accuracy we want to achieve, VE8 represents a good tradeoff between accuracy and value of Q needed. Therefore, hereafter we will use the polynomial of order 8 (VE8) and we will simply denote it as VE.

4.2. Comparison with Other Techniques

In this section we compare the performance of value extrapolation with other techniques based on the traditional approach of solving the steady state probabilities using the balance equations for later computing the performance parameters of interest. Although other approaches exist, we have chosen the technique proposed in [8], referred to hereafter as FM, and the one proposed by Neuts and Rao in [11], referred to as NR. Note that we have not compared the results with the technique proposed by Artalejo and Pozo [6] as this last technique does not include the impatience phenomenon, so it is not directly applicable. A similar reasoning can be done for the technique proposed by Falin [10].

In Table 3 we show the minimum values of Q needed to obtain a relative error lower than 10^{-8} for different performance parameters and for the aforementioned techniques. Results show that value extrapolation clearly outperforms classical techniques as it needs a much lower value of Q to achieve a certain accuracy in all the scenarios under study and for all the parameters studied. Similarly, in Figures 5, 6 and 7, we plot the relative error for P_b, P_ns

Table 2: Minimum value of Q to obtain relative errors (ϵ) lower than 10^{-8}.

	$\epsilon_{P\mathrm{b}} < 10^{-8}$			$\epsilon_{P_{\mathrm{ns}}} < 10^{-8}$			$\epsilon_{N_{\mathrm{ret}}} < 10^{-8}$		
ρ	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9
VE1	20	32	61	25	41	64	22	37	57
VE2	14	31	53	21	35	58	17	32	54
VE3	15	18	48	19	31	53	16	26	50
VE4	12	25	47	17	30	48	14	26	47
VE5	12	24	44	12	24	44	9	18	43
VE6	10	20	41	14	26	44	11	22	39
VE7	**7**	21	39	11	24	42	**8**	21	40
VE8	8	17	39	11	23	**36**	**8**	19	39
VE9	9	19	38	**10**	22	39	9	**13**	**34**
VE10	10	16	35	**10**	21	39	10	17	35
VE11	11	18	**31**	11	**16**	37	11	18	37
VE12	12	15	40	12	20	42	12	17	42
VE13	13	**14**	43	14	19	43	13	18	43
VE14	14	23	48	26	25	48	14	24	48
VE15	15	25	56	15	29	56	15	25	56
VE16	16	27	56	18	29	57	28	27	57
Exact Value	$3.89 \cdot 10^{-6}$	0.0045	0.1353	$6.05 \cdot 10^{-8}$	$1.34 \cdot 10^{-4}$	0.0110	$5.74 \cdot 10^{-5}$	0.0981	4.4789

Table 3: Minimum Q value to obtain relative errors (ϵ) lower than 10^{-8}.

	$\epsilon_{P_{\mathrm{b}}} < 10^{-8}$			$\epsilon_{P_{\mathrm{ns}}} < 10^{-8}$			$\epsilon_{N_{\mathrm{ret}}} < 10^{-8}$		
	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$
FM	23	39	68	29	46	70	25	42	53
NR	20	31	61	25	41	64	22	38	65
VE8	**8**	**17**	**39**	**11**	**23**	**36**	**8**	**19**	**39**

and N_{ret}, respectively, when $\rho = 0.7$ and for the different techniques deployed. Results show that, for a same value of Q, VE is able to obtain lower relative errors than NR and FM. The difference in the relative errors is around 4 to 5 orders of magnitude, which supposes a very clear improvement.

4.3. Computation Cost

Although it is shown that VE clearly outperforms NR and FM techniques, it is interesting to study their associated computation cost. In Figures 8, 9, and 10, we plot the time needed to achieve a certain relative error for P_{b}, P_{ns}, and N_{ret} using the different techniques under study. Note also that, although it has been obtained using VE8, choosing a diferrent order for the extrapolation polynomial would not change the computation cost, as the linear system of equations to be solved remains of the same size. However, results should be interpreted carefully, because computation costs highly depend on the algorithm used to solve the resulting system of equations. More concretely, for solving the systems obtained in the FM and NR techniques, we have made use of the efficient algorithm described in [16] that

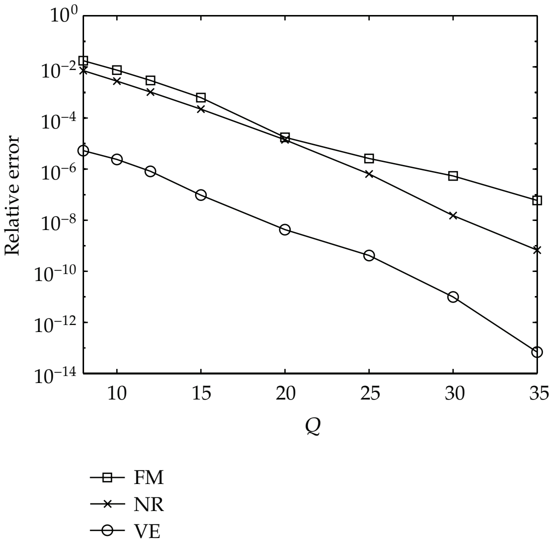


Figure 5: Relative error in P_b for different techniques.

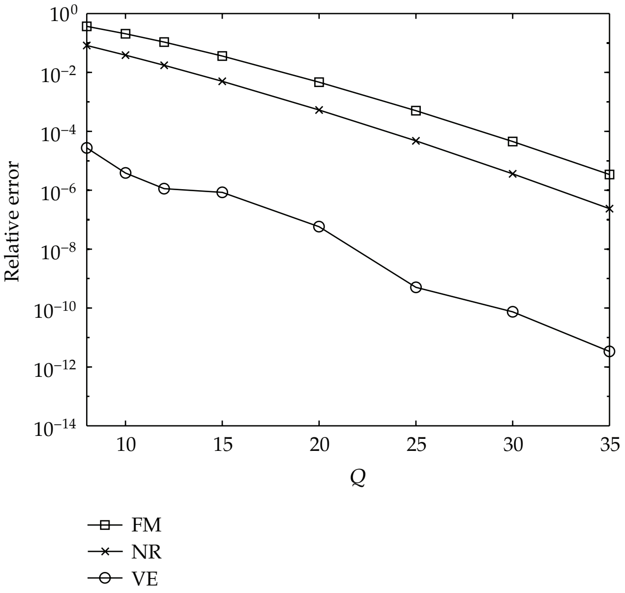


Figure 6: Relative error in P_{ns} for different techniques.

takes advantage of the block-tridiagonal structure that presents the infinitesimal generator. Unfortunately, the linear system of equations obtained in VE has no longer such a block-tridiagonal structure, and therefore we must use a more general and less efficient algorithm. More concretely, we have used LU factorization. Figures 8–10 show that VE achieves a certain accuracy faster than the other techniques under study.

5. Conclusions

Multiserver retrial systems have not an exact solution when the number of servers is higher than two, as their state space presents space heterogeneity along an infinite dimension. For

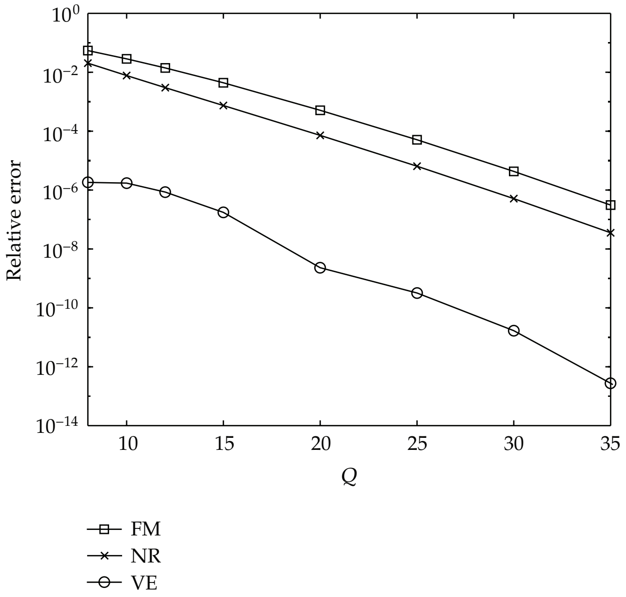


Figure 7: Relative error in N_{ret} for different techniques.

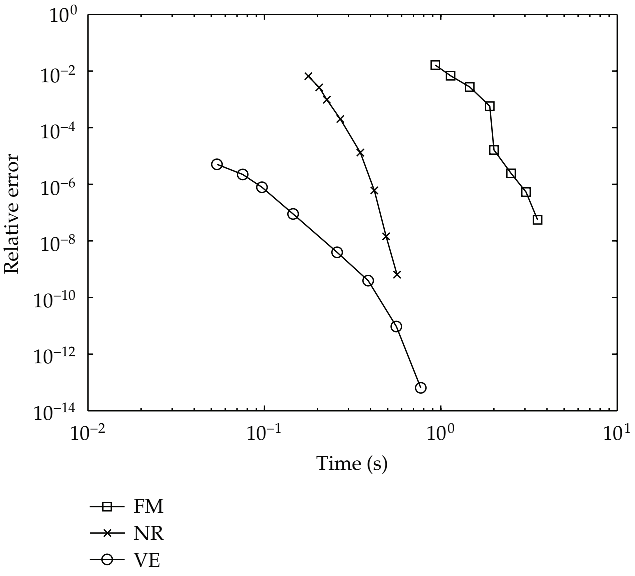


Figure 8: Relative error in P_{b} versus computation cost.

that reason, it is mandatory to develop approximate techniques in order to solve these systems. To the best of our knowledge, all the techniques studied in the literature to solve these systems are based on their steady state probabilities. In this paper we propose an alternative technique based on a different metric: the relative state values and the Howard equations that relate them, instead of the balance equations. With this technique, truncation of the state space can be done in a more efficient way, as the state values outside the truncated state space are

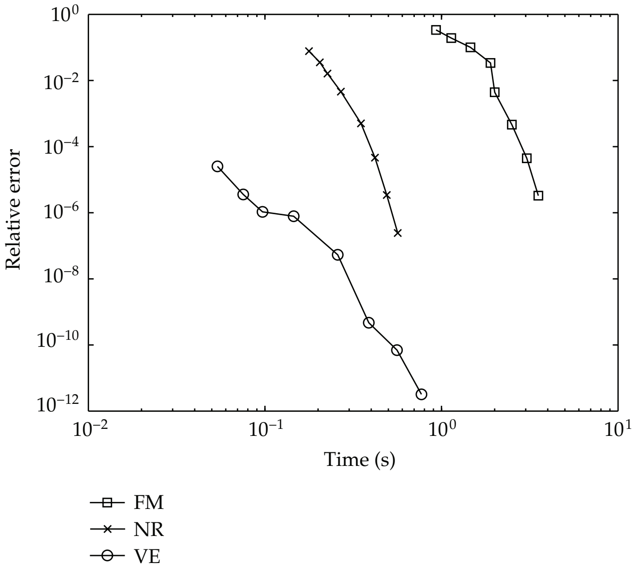


Figure 9: Relative error in P_{ns} versus computation cost.

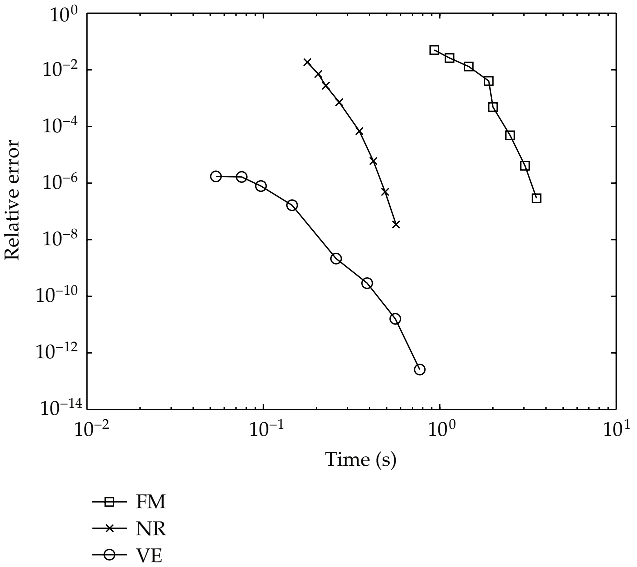


Figure 10: Relative error in N_{ret} versus computation cost.

extrapolated from some known state values. In order to preserve the linearity of the resulting system of equations, we have only used polynomials as extrapolation functions.

In a first part, we have studied the use of different orders for the extrapolation polynomials. Later, we have compared the new technique with two well-known approaches appeared in the literature [8, 11] in terms of accuracy and computational cost. Results show that the proposed technique highly improves the previous approaches in terms of computational cost and, specially, in terms of accuracy, so its use is highly recommended.

Acknowledgments

This work has been supported by the Spanish government under Projects TIN2010-21378-C02-02 and TIN2008-06739-C04-02/TSI and by Comunidad de Madrid through Project S-2009/TIC-1468.

References

[1] P. Tran-Gia and M. Mandjes, "Modeling of customer retrial phenomenon in cellular mobile networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 8, pp. 1406–1414, 1997.

[2] J. M. Gimenez-Guzman, M. J. Domenech-Benlloch, J. Martinez-Bauset, V. Pla, and V. Casares-Giner, "Analysis of a handover procedure with queueing, retrials and impatient customers," in *Proceedings of the 3rd International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (HET-NETs '05)*, 2005.

[3] J. M. Gimenez-Guzman, M. J. Domenech-Benlloch, V. Pla, V. Casares-Giner, and J. Martinez-Bauset, "Guaranteeing seamless mobility with user redials and automatic handover retrials," *Journal of Universal Computer Science*, vol. 14, no. 10, pp. 1597–1624, 2008.

[4] J. R. Artalejo, "Accessible bibliography on retrial queues: progress in 2000–2009," *Mathematical and Computer Modelling*, vol. 51, no. 9-10, pp. 1071–1081, 2010.

[5] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: an Algorithmic Approach*, vol. 2 of *Johns Hopkins Series in the Mathematical Sciences*, Johns Hopkins University Press, Baltimore, Md, USA, 1981.

[6] J. R. Artalejo and M. Pozo, "Numerical calculation of the stationary distribution of the main multiserver retrial queue," *Annals of Operations Research*, vol. 116, no. 1–4, pp. 41–56, 2002.

[7] M. A. Marsan, G. De Carolis, E. Leonardi, R. Lo Cigno, and M. Meo, "Efficient estimation of call blocking probabilities in cellular mobile telephony networks with customer retrials," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 2, pp. 332–346, 2001.

[8] M. J. Doménech-Benlloch, J. M. Giménez-Guzmán, J. Martínez-Bauset, and V. Casares-Giner, "Efficient and accurate methodology for solving multiserver retrial systems," *Electronics Letters*, vol. 41, no. 17, pp. 967–969, 2005.

[9] M. J. Domenech-Benlloch, J. M. Gimenez-Guzman, V. Pla, J. Martinez-Bauset, and V. Casares-Giner, "Generalized truncated methods for an efficient solution of retrial systems," *Mathematical Problems in Engineering*, vol. 2008, Article ID 183089, 15 pages, 2008.

[10] G. I. Falin, "Calculation of probability characteristics of a multiline system with repeat calls," *Moscow University Computational Mathematics and Cybernetics*, no. 1, pp. 43–49, 1983.

[11] M. F. Neuts and B. M. Rao, "Numerical investigation of a multiserver retrial model," *Queueing Systems*, vol. 7, no. 2, pp. 169–190, 1990.

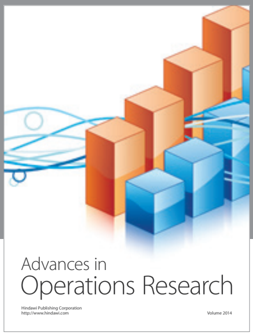
[12] J. Leino, A. Penttinen, and J. Virtamo, "Flow level performance analysis of wireless data networks: a case study," in *Proceedings of the IEEE International Conference on Communications, (ICC '06)*, vol. 3, pp. 961–966, July 2006.

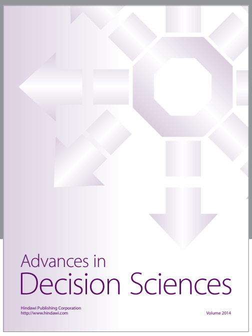
[13] J. Leino and J. Virtamo, "An approximative method for calculating performance measures of markov processes," in *Proceedings of the 1st International Conference on Performance Evaluation methodolgies and tools*, Pisa, Italy, 2006.

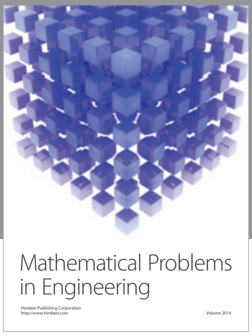
[14] J. Leino and J. Virtamo, "Determining the moments of queue-length distribution of discriminatory processor-sharing systems with phase-type service requirements," in *Proceedings of the 3rd EuroNGI Conference on Next Generation Internet Networks*, pp. 205–208, Trondheim, Norway, 2007.

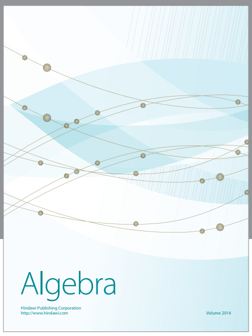
[15] R. A. Howard, *Dynamic Programming and Markov Processes*, The Technology Press of MIT, Cambridge, Mass, USA, 1960.

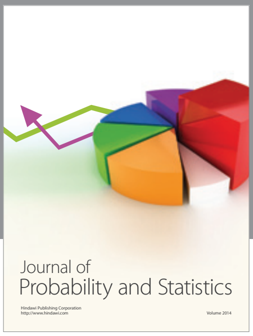
[16] D. P. Gaver, P. A. Jacobs, and G. Latouche, "Finite birth-and-death models in randomly changing environments," *Advances in Applied Probability*, vol. 16, no. 4, pp. 715–731, 1984.


Advances in
Operations Research

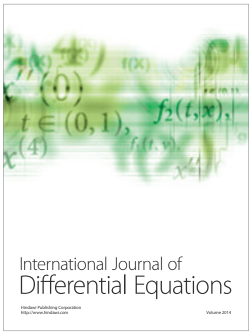

Advances in
Decision Sciences

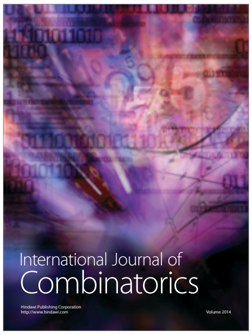

Mathematical Problems
in Engineering

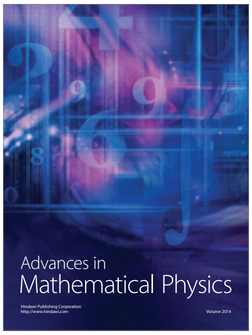

Algebra

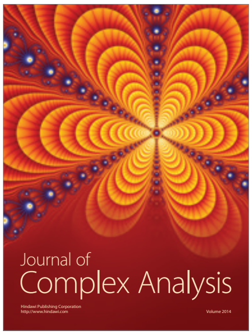

Journal of
Probability and Statistics

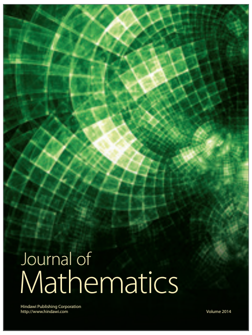
The Scientific
World Journal

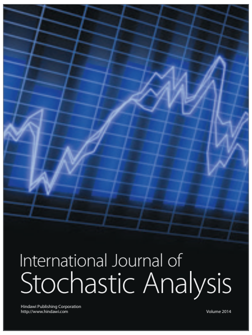

International Journal of
Differential Equations

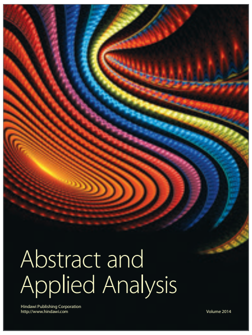

International Journal of
Combinatorics

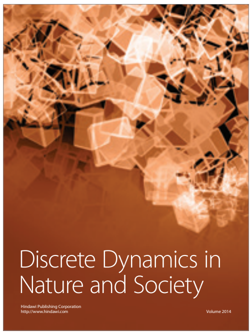

Advances in
Mathematical Physics

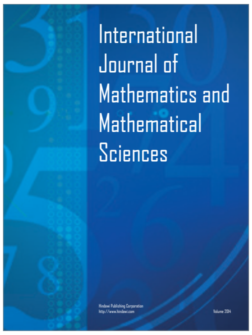

Journal of
Complex Analysis


Journal of
Mathematics


International Journal of
Stochastic Analysis

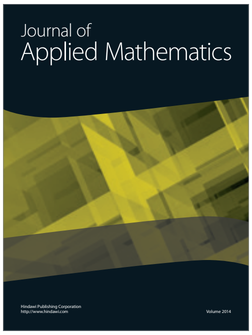

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


Journal of
Applied Mathematics

Journal of
Optimization