Efficient and accurate solution of multiserver retrial systems with user impatience through the value extrapolation technique

Jose Manuel Gimenez-Guzman, M^{a} Jose Domenech-Benlloch,
Vicent Pla, Vicente Casares-Giner and Jorge Martinez-Bauset
Dept. Comunicaciones, Universidad Politecnica de Valencia, UPV
Camino de Vera s/n, 46022, Valencia, Spain. Phone: +34 963877767
Email: jogiguz@upvnet.upv.es, mdoben@doctor.upv.es,{vpla,vcasares,jmartinez}@dcom.upv.es

Keywords: Queueing theory, Markov processes, retrial systems, value extrapolation, performance evaluation.

Abstract

The retrial phenomenon plays an important role in many types of communication networks, and therefore, it should not be ignored in their analysis. Unfortunately, these systems do not present an exact analytic solution, so it is mandatory to resort to approximate techniques in order to compute their performance parameters. To the best of our knowledge, although there is a wide literature related to retrial systems, all the approximate techniques appeared up to now are based in computing the steady states probabilities and, later, computing the desired performance parameters. In this paper we use another approach to solve the retrial system based on a different metric: the relative state values which appear in the Howard equations. A numerical evaluation is carried out to evaluate this new technique and we compare its performance with previous methods. The obtained results show that the proposed solution 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 [1]. The modeling of repeated attempts has been a subject of numerous investigations, because these systems have a non homogeneous and infinite state space. However, it is known that the classical theory [2] is developed for random walks on the semi-strip $\{0,\ldots,C\} \times \mathbb{Z}_{+}$ with infinitesimal transitions subject to conditions of space-homogeneity.

When the space-homogeneity condition does not hold, e.g. in the case of retrial queues, the problem of calculating the equilibrium distribution has not been solved beyond approximate methods when the number of servers in higher than 2 [3]. In particular Marsan et al. [4] propose a well-known approximate technique for its analysis. In [5] a generalization of the approximate method in [4] 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 a infinite state space to a finite one by aggregating states. Other solutions mantain the infinite state space but homogeneize it beyond a given level in order to solve the system. These later models are known as generalized truncated models [3], and usually present the advantage of providing a much better accuracy than the finite methodologies. In this category we find the models proposed by Falin [6], by Neuts and Rao [7] and by Artalejo and Pozo [3]. 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. [8–10]. The new method, named value extrapolation, does not rely on solving the global balance equations. This method 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 [8–10]. 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 [10] the authors shown that by selecting an appropriate polynomial function the method 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, i.e., the performance

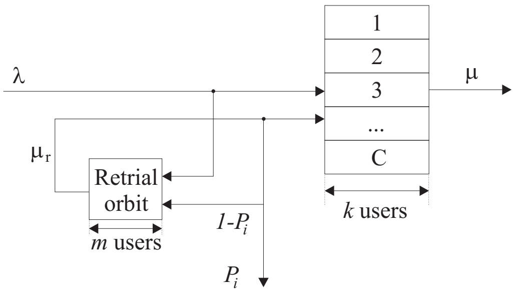

Figure 1. Retrial model under study.

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, e.g. retrial queues, which are essentially different of the type of queues to which this method 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 method has to be applied once per each of them. We apply well-known linear algebra algorithms to compute several performance parameters simultaneously and through a numerical a 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 [8, 9]. 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 method, defining the associated Howard equations of the model and the revenue functions. In a second part, we compare our method with other previously proposed methods in terms of accuracy and computational cost. Results show that the proposed method 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 method used. In Section 4 the numerical analysis is carried out, evaluating the value extrapolation technique and comparing it with other previous solving methods proposed in the literature. Final remarks and a summary of results are provided in Section 5.

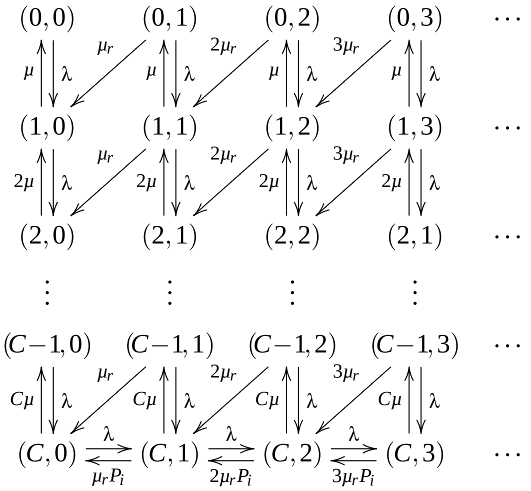

Figure 2. Transition diagram.

2. SYSTEM MODEL

The system under study is a generic retrial system including user impatience, i.e., users leave the system with certain probability after a non successful retrial. As shown in Fig. 1, an infinite number of users arriving following a Poisson process with rate λ contend for access to a system with C servers, requesting an exponentially distributed service time with rate μ. Without loss of generality, we consider that each user occupies one resource unit. 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, being a retrial 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, $S(t)$. Being k the number of sessions being served and m the number of users in the retrial orbit the state space is defined by

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

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 on the retrial orbit.

3. SOLVING TECHNIQUE

3.1. Theoretical framework

As it has been aforementioned, the problem under interest has not a closed form solution when $C > 2$ [3], so approximation techniques are mandatory. To the best of our knowledge, all the approximate methods appeared in literature compute the steady state probabilities using the balance equations in order to compute the desired performance parameters, i.e. solving the linear system of equations:

$$\pi(s)\sum_{s'} q_{ss'} = \sum_{s'} \pi(s')q_{ss'} \qquad \forall s$$

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

Notwithstanding, the method we use [8, 9] 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 to other states when a certain action is taken at a given 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, i.e., $r(s)$. Following the definition of the revenue function for every state, we will also have a mean revenue rate of the entire process (r), which will be the performance metric we want to compute.

Once 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 mean revenue collected converges to r. The relative state value ($v(s)$) tells how much greater the expected cumulative revenue over an infinite time horizon is when the system starts from the initial state s in comparison with r.

$$v(s) = E\left[\int_{t=0}^{\infty}(r(S(t))-r)dt\middle|S(0)=s\right] \qquad (1)$$

The equations that relate revenues, relative state values and transition probabilities are the Howard equations defined by:

$$r(s)-r+\sum_{s'} q_{ss'}(v(s')-v(s)) = 0 \quad \forall s \qquad (2)$$

The Howard equations represent the *policy evaluation* phase of the well-known *policy iteration* algorithm, the most widespread dynamic programming technique, proposed in [11]. There will be as much Howard equations as number of states, $|\mathcal{S}|$. The number of unknowns will be the $|\mathcal{S}|$ relative state values plus the expected revenue r, i.e, $|\mathcal{S}|+1$ 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.

The Howard equations that correspond to the system under study are:

For $k<C$:

$$\begin{aligned} r(k,m)-r+\lambda[v(k+1,m)-v(k,m)]+ \\ +k\mu[v(k-1,m)-v(k,m)]+ \\ +m\mu_r[v(k+1,m-1)-v(k,m)]=0 \end{aligned} \qquad (3)$$

For $k=C$:

$$\begin{aligned} r(C,m)-r+\lambda[v(C,m+1)-v(C,m)]+ \\ +C\mu[v(C-1,m)-v(C,m)]+ \\ +m\mu_r P_i[v(C,m-1)-v(C,m)]=0 \end{aligned} \qquad (4)$$

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 $\hat{\mathcal{S}}$. In our case, the truncated state space is defined by:

$$\hat{\mathcal{S}} := \{s=(k,m) : k\le C; m\le Q\}$$

3.2. Value extrapolation: polynomial fitting

The traditional truncation consists of doing $q_{ss'}=0 \quad \forall s'\notin\hat{\mathcal{S}}$ but the value extrapolation method performs a more efficient truncation. Basically, value extrapolation considers the relative state values outside $\hat{\mathcal{S}}$ that appear in the Howard equations as an extrapolation of some relative state values inside $\hat{\mathcal{S}}$. As we truncate the retrial orbit dimension beyond a value Q, the value extrapolation technique uses the state value of some states in $\hat{\mathcal{S}}$ 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 the relative values outside $\hat{\mathcal{S}}$ were correctly extrapolated, the results obtained by solving the truncated model would be exact. Also note that including value extrapolation neither increase the computational cost nor increase the number of Howard equations, remaining in $|\hat{\mathcal{S}}|=(C+1)\times(Q+1)$.

Summarizing, the objective of value extrapolation is to find a function $f(s)$ that fits with some points $(s,v(s))$ for $s\in\hat{\mathcal{S}}$ so that it approximates also $(s,v(s))$ for $s\notin\hat{\mathcal{S}}$. 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 acomplish that fact are the polynomials.

We can use all $(s,v(s))$-pairs of the state space into the fitting procedure (global fitting) or, what is most commonly used, only a subset ($\mathcal{S}_f$) of them (local fitting). The choice of $\mathcal{S}_f$ will highly depend on the state we want to extrapolate its relative state value. Note also that function $f(s)$ and set $\mathcal{S}_f$ need to be chosen so that parameters have unambiguous values, i.e. in the case of choosing a polynomial as the fitting

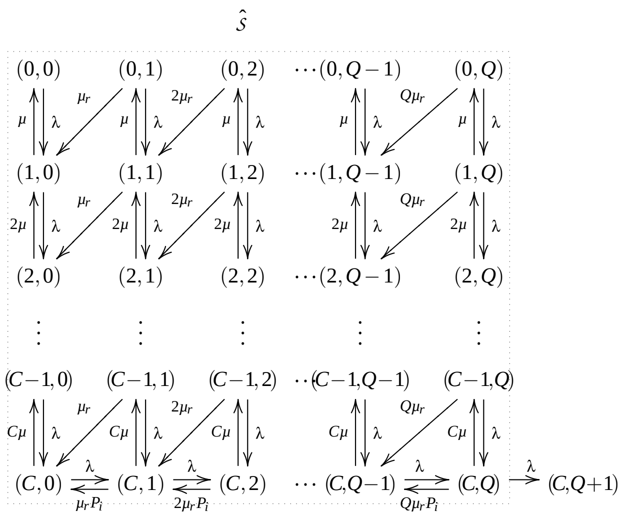

Figure 3. Truncated model and states that appear in Howard equations outside the truncated model.

function, the number of different points in $\mathcal{S}_f$ has to be equal or greater than the number of coefficients in the polynomial.

As shown in Fig. 3, for the truncated problem of interest 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 $\hat{\mathcal{S}}$. Therefore, we must approximate the value $v(C,Q+1)$ by using some relative state values of states belonging to $\hat{\mathcal{S}}$. For this purpose we have used a $(n-1)$-th degree polynomial that interpolates the n points in $\{(m, v_m)|v_m = v(C,m), Q-n < m \leq Q$. After some algebra, and using the Lagrange basis to reduce the complexity of the procedure, we obtain a simple closed-form expression for the extrapolated value

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

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 are they computed. By definition, $r(s)$ is the expected immediate revenue 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 summarizes the different $r(s)$ to be set in order to obtain several performance parameters. Additionally to the well-known blocking probability (P_b) and the mean number of users in the retrial orbit (N_{ret}), we must define the non-service probability (P_{ns}). This later probability describes the behaviour of retrial systems more accurately, as defines the probability of a user leaving the system without obtaining service due to impatience.

Table 1. Revenue function definition.

Performance parameter	Acronym	Value
Blocking probability	P_b	$r(k,m)=1$ for $k=C, \forall m$
		$r(k,m)=0$ otherwise
Non-service probability	P_{ns}	$r(k,m)=\frac{m\mu_r P_i}{\lambda}$ for $k=C, \forall m$
		$r(k,m)=0$ otherwise
Mean number of users retrying	N_{ret}	$r(k,m)=m$ $\forall k, \forall m$

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

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 that equation will be:

$$\begin{aligned} r(C,Q)-r+v(C,Q)[-\lambda-C\mu-QP_i\mu_r]+ \\ +\lambda v(C,Q+1)+C\mu v(C-1,Q)+QP_i\mu_r v(C,Q-1)= \\ =r(C,Q)-r+v(C,Q)[\lambda-C\mu-QP_i\mu_r]+ \\ +C\mu v(C-1,Q)+[QP_i\mu_r-\lambda]v(C,Q-1)=0 \end{aligned}$$

As $v(C,Q+1)$ no longer appears into 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 relative state values for the different states. Matrix $\mathbf{T}$ represents the matrix of coefficients. Note that the size of matrix $\mathbf{T}$ does not depend on the order of the polynomial used to perform the extrapolation. This characteristic has the advantage that there will not be any difference into the computation cost when using higher order extrapolation.

Unfortunately, in order to solve system $\mathbf{xT} = \mathbf{b}$, we can not make use of methodologies that make use of the block tridiagonal structure [12, 13] to effectively solve such systems. So it is mandatory to use a general procedure to solve the linear system of equations, such as Gauss-Seidel, Gauss-Jordan or LU factorization methods.

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 (specially 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 supposes the most computation consumption part) is done only once for the $\mathbf{T}$ matrix.

4. RESULTS

In order to evaluate and compare the proposed method 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 $\mu^{-1} = 180$ s. The retrial phenomenon has been configured with $\mu_r^{-1} = 100$ s and $P_i = 0.2$. Although only a 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 $\varepsilon_\Psi = |\Psi^{approx} - \Psi^{exact}|/\Psi^{exact}$. In order to obtain an accurate enough estimate of Ψ which can be used as Ψ^{exact}, we ran all methods 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 methods converged to the same value in the performance parameters under study, $\Psi \in \{P_b, P_{ns}, N_{ret}\}$.

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. 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.

Although from Table 2 there is not a clear choice in the order of the best polynomial, the general trend shows that increasing the order generally increases the performance, but this increasement is lower as we use higher orders. Although the computation cost is almost the same independently of the order of the extrapolation polynomial used, in this case we do not recommend to use higher order polynomials, due to the fact that using VEx enforces us to use a model with $Q \geq x$ with the choice of $\mathcal{S}_f$ explained in Section 3.2. 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 minimum 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 the value extrapolation method with other methods 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 method proposed in [5], refered hereafter as FM method, and the one proposed by Neuts and Rao in [7], refered as NR method. Note that we have not compared the results with the method proposed by Artalejo and Pozo [3] as this last method does not include the impatience phenomenon, so it is not directly applicable. A similar reasoning can be done for the method proposed by Falin [6].

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 methods. Results show that the value extrapolation method clearly outperforms classical methods 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 Figs. 4-6 we plot the relative error for P_b, P_{ns} and N_{ret} respectively when $\rho = 0.7$ and for the different methods deployed. Results show that, for a same value of Q, VE greatly improves the performance of NR and FM (being FM slightly worse than NR). The difference in the relative errors is around 4 to 5 orders of magnitude, which supposes a very clear improvement.

4.3. Computational cost

Initially, one of the main drawbacks of the value extrapolation method is that it is only able to compute a performance parameter each time the system is solved. However, in Section 3.4 we have glimpsed that solving several performance parameters at the same time is not expected to severely increase the computation time. In Fig. 7 we show the computation time[1] needed to obtain a different number of parame-

[1] Results have been obtained using Matlab running on an Intel Pentium IV 3GHz.

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

	$\varepsilon_{P_b} < 10^{-8}$			$\varepsilon_{P_{ns}} < 10^{-8}$			$\varepsilon_{N_{ret}} < 10^{-8}$		
	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$
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**
Value	$3.89 \cdot 10^{-6}$	0.0045	0.1353	$6.05 \cdot 10^{-8}$	$1.34 \cdot 10^{-4}$	0.01096	$5.74 \cdot 10^{-5}$	0.09806	4.4789

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

	$\varepsilon_{P_b} < 10^{-8}$			$\varepsilon_{P_{ns}} < 10^{-8}$			$\varepsilon_{N_{ret}} < 10^{-8}$		
	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$
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**

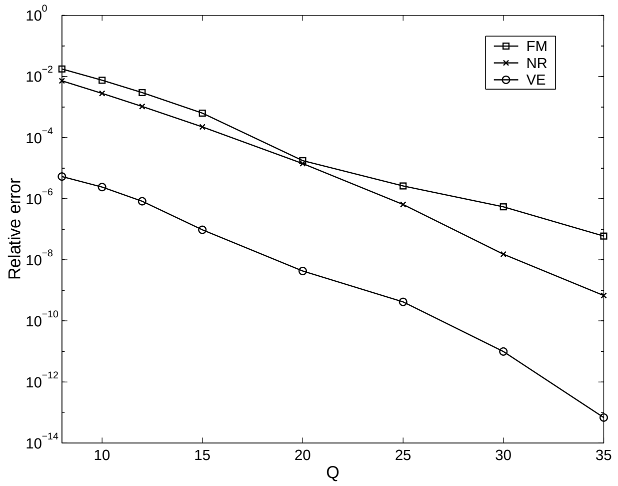

Figure 4. Relative error in P_b for different methods.

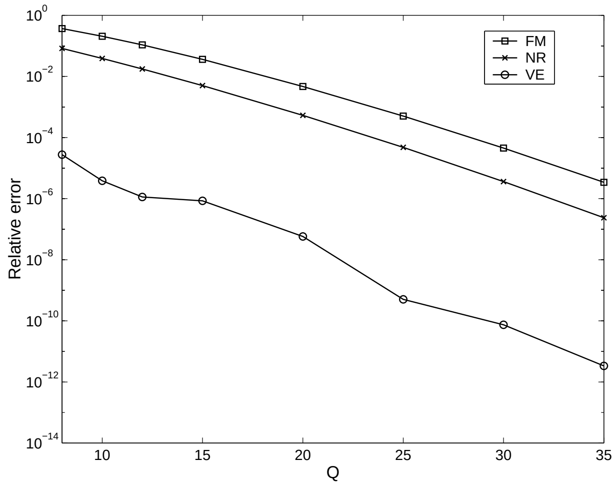

Figure 5. Relative error in P_{ns} for different methods.

ters (p). Observing Fig. 7 it follows that the computation cost is only marginally increased when we compute another performance parameter.

Although it is shown that VE method clearly outperforms NR and FM methods, it is interesting to study their associated computational cost. In Fig. 8 we plot the time needed to solve a model for the different methods studied and for different values of Q. Note that the computation cost chosen for the value extrapolation method has been obtained when solving the three aforementioned perfomance parameters. Note also that, although it has been obtained using VE8, choosing a diferent order for the extrapolation polynomial would not

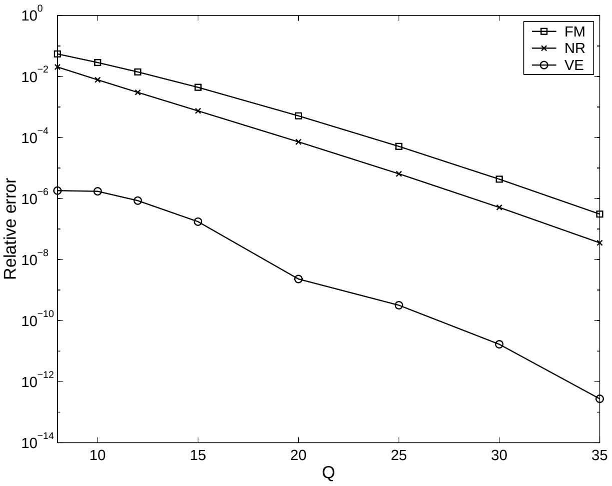

Figure 6. Relative error in N_{ret} for different methods.

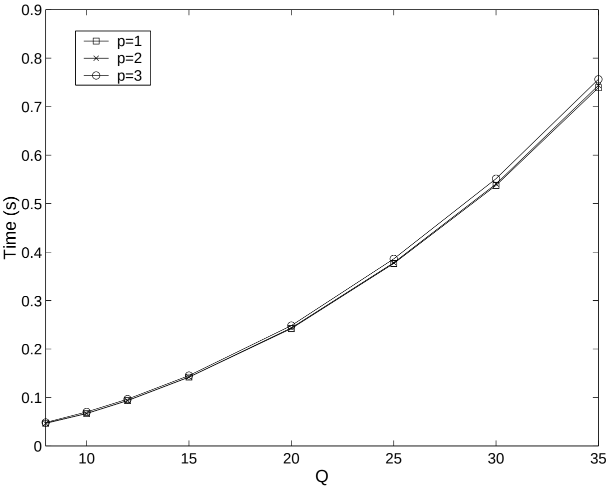

Figure 7. Computation cost when solving p performance parameters simultaneously.

change the computation cost, as the linear system of equations to solve remains of the same size. Results should be interpreted carefully, as the implementation of each method can vary. For solving the systems obtained in the FM and NR methods we have made use of the efficient algorithm described in [12] that can take advantage of the tridiagonal structure that presents the system under study. Unfortunately, the linear system of equations obtained in the VE method has no longer such a block tridiagonal structure, so we must use a more general method. In this case we have used LU factorization. Therefore, the results shown could vary by using different methodologies. Figure 8 shows that the cost of solving FM is much higher than NR and VE, as it involves an iterative procedure that makes necessary to solve the system more than once. Comparing NR with VE we show that their computational cost is similar, being lower for VE for low values of Q. However, the scalability of the method described in [12] is better, so the slope of cost of NR is lower than the obtained in LU factorization (VE method). As VE method needs a much lower value of Q to achieve a certain accuracy and the computation cost of solving the system for the same value of Q is similar, we conclude that using VE method is highly recomendable for solving retrial systems.

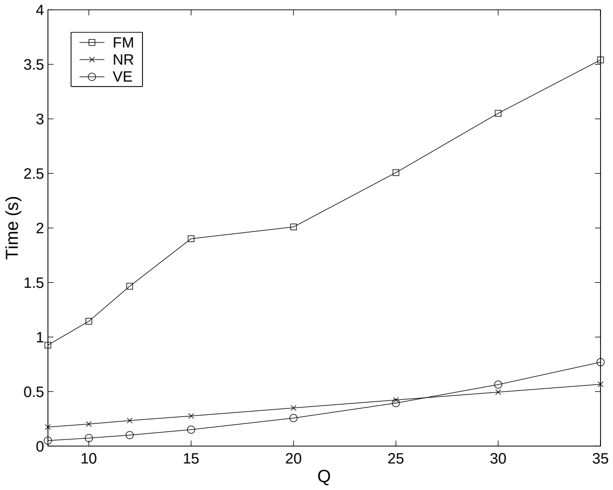

Figure 8. Computation cost for different methodologies.

5. CONCLUSIONS

The retrial phenomenon has a nonnegligible impact into the communication networks. However, multiserver retrial systems have not an exact solution when the number of servers is higher than two, as their state space present space heterogeneity along an infinite dimension. For that reason, it is mandatory to develop approximate techniques in order to solve these systems. To the best of our knowdledge, all the methods 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 method, truncation of the state space can be done in a more efficient way, as the state values outside the truncated state space are 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, concluding that higher order polynomials use to give better solutions. On the other hand, they need a higher truncation level so we must choose a tradeoff between both facts. Later, we have compared the new method with two well-known approaches appeared in the literature [5, 7] 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.

ACKNOWLEDGEMENTS

This work was supported by the Spanish Government (30% PGE) and the European Commission (70% FEDER) through projects TSI2005-07520-C03-03 and TEC2004-06437-C05-01 and by *Cátedra Telefónica de Internet y Banda Ancha* (e-BA) from the Universidad Politécnica de Valencia. Besides M. Jose Domenech-Benlloch was supported by the Spanish Ministry of Education and Science under contract AP-2004-3332.

REFERENCES

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

[2] M. Neuts, *Matrix-geometric Solutions in Stochastic Models: An Algorithmic Approach*. The Johns Hopkins University Press, 1981.

[3] 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.

[4] M. A. Marsan, G. Marco 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, Feb. 2001.

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

[6] G. Falin, “Calculation of probability characteristics of a multiline system with repeat calls,” *Moscow University Computational Mathematics and Cybernetics*, vol. 1, pp. 43–49, 1983.

[7] M. Neuts and B. Rao, “Numerical investigation of a multiserver retrial model,” *Queueing systems*, vol. 7, pp. 169–190, 1990.

[8] J. Leino, A. Penttinen, and J. Virtamo, “Flow level performance analysis of wireless data networks: A case study,” in *Proceeding of IEEE ICC*, vol. 3, 2006, pp. 961–966.

[9] J. Leino and J. Virtamo, “An approximative method for calculating performance measures of markov processes,” in *Proceeding of the 1st international conference on Performance evaluation methodolgies and tools*, Pisa, Italy, Oct. 2006.

[10] ——, “Determining the moments of queue-length distribution of discriminatory processor-sharing systems with phase-type service requirements,” in *Proceedings of 3rd EuroNGI Conference on Next Generation Internet Networks*, Trondheim, Norway, May 2007, pp. 205–208.

[11] R. Howard, *Dynamic Programming and Markov Processes*. MIT Press, 1960.

[12] D. Gaver, P. Jacobs, and G. Latouche, “Finite birth-and-death models in randomly changing environments,” *Advances in Applied Probability*, vol. 16, pp. 715–731, 1984.

[13] L. D. Servi, “Algorithmic solutions to two-dimensional birth-death processes with application to capacity planning,” *Telecommunication Systems*, vol. 21, no. 2–4, pp. 205–212, 2002.