Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

ON-LINE ERROR BOUNDS FOR STEADY-STATE APPROXIMATIONS: A POTENTIAL SOLUTION TO THE INITIALIZATION BIAS PROBLEM

Enver Yücesan
Luk N. Van Wassenhove
Klenthis Papanikas

Technology Management Area
INSEAD
Boulevard de Constance
77305 Fontainebleau Cedex, FRANCE

Nico M. van Dijk

Faculty of Economics and Econometrics
University of Amsterdam
Roettersstraat 11
1018 WB Amsterdam, THE NETHERLANDS

ABSTRACT

By studying performance measures via reward structures, on-line error bounds are obtained by successive approximation. These bounds indicate when to terminate computation with guaranteed accuracy; hence, they provide insight into steady-state convergence. The method therefore presents a viable alternative to steady-state computer simulation where the output series is typically contaminated with initialization bias whose impact on the output cannot be easily quantified. The method is illustrated on capacitated queueing networks. The results indicate that the method offers a practical tool for numerically approximating performance measures of queueing networks. Results on steady-state convergence further quantify the error involved in analyzing an inherently transient system using a steady-state model.

1 INTRODUCTION

Man-made systems such as manufacturing processes, telecommunication systems or transportation networks are typically referred to as *Discrete Event Dynamic Systems* (DEDS). These are systems with piecewise constant trajectories, where state changes are induced by the occurrence of asynchronous events. A fundamental obstacle to the study of DEDS is the lack of a comprehensive framework for modeling and analysis of such systems.

Discrete-event computer simulation is the most flexible approach in assessing the performance of DEDS. Performance measures of interest such as average throughput, average response time or system utilization are typically steady-state measures. Roughly speaking, these are measures that represent averages over a relatively long period of time, after the impact of start-up conditions has dissipated. Initialization bias can be a major source of error in estimating the steady-state value of a performance measure in a simulation experiment. This is due to arbitrarily specified start-up conditions of a simulation run that fail to reflect typical long-term behavior of the system.

Transient effects have received considerable attention in the simulation literature (Yücesan 1993). Very broadly, two general approaches have been proposed: investigation of initialization policies to promote rapid convergence to steady-state conditions (for example, Kelton 1985, 1989) and construction of heuristic procedures to determine a truncation point, the observation after which data are retained for analysis (for example, Schruben 1981, 1982). Proposed heuristics range from the visual inspection of the output series to statistical tests, from asymptotic theory to the maximum-entropy principle.

None of these heuristic techniques, however, offer any insights on the impact of the transient effects on the resulting estimates or on the rate of convergence to steady state. Such insights can only be obtained for special cases. For instance, it is well known that the transition matrix of a Markov chain converges at a geometric rate under a constraining constant depending on its second eigenvalue (Çinlar 1975). However, eigenvalues can only be obtained in simplistic situations; their numerical calculation rapidly becomes prohibitively expensive, if not impossible, for complex systems.

Abate and Whitt (1986, 1987) investigate the transient behavior of the single-server queue to determine whether the steady-state descriptions are reasonable. Odoni and Roth (1983) conduct an empirical investigation of the transient behavior of stationary, single-server Markovian queues. They observe that, for many queueing systems, the rate at which a system converges to its steady-state characteristics, independently of the system's initial state, eventually becomes dominated by an exponential term of the form $\exp\{-t/\tau\}$, where t is time and τ is a characteristic of the queueing system.

Kelton and Law (1985) and Kelton (1985) analyze the transient probabilistic structure of Markovian and Erlangian queueing systems, proposing computational algorithms to obtain the required probabilities. These results are further used to investigate the question of initializing simulations so as to promote rapid convergence to steady state. Kelton (1989) incorporates the maximum entropy principle in a heuristic to determine initial conditions that promote rapid convergence.

No other results, even experimental ones, seem to be reported. Therefore, very little is known in general about how fast steady state is reached. In addition, analytic closed-form expressions, particularly product-form expressions, are highly restricted when such practical considerations as finite buffer constraints, breakdowns, and job interferences are taken into account (Van Dijk 1993).

The objective of the paper is to introduce a method to compute steady-state values for performance measures with an *explicit consideration of the transient effects*. The method can be used for direct numerical computation. It can therefore provide an indicator for the actual steady-state convergence, offering a potential solution for the imitialization bias problem in discrete event simulation.

The key idea is to evaluate average performance measures by means of Markovian cumulative reward structures. A rather simple result, that has been developed within Markov decision theory, can then be adopted directly. While this result was originally introduced by Odoni (1969) and Popyack (1985) to determine the accuracy of successive approximation schemes in dynamic programming, it has not been applied for the purpose of approximating steady-state performance measures. To this end, a somewhat simplified result, adopted from the literature, is presented for direct application to Markov reward chains. Its deployment is illustrated through some generic queueing networks. An illustrative set of numerical results is included. A more extensive study is undertaken in Papanikas (1992). These results show that the error bounding technique can be used in a straightforward fashion to determine truncation points for computations and to guarantee a level of accuracy with respect to a steady-state parameter value.

The paper is organized as follows. Section 2 introduces the technique and discusses its application to discrete event simulation. A numerical illustration is provided in Section 3 together with a sensitivity analysis investigating the performance of the technique. Concluding comments are included in Section 4.

2 METHODOLOGY

Consider a continuous-time Markov chain (CTMC), $\{X(t): t \geq 0, (\Omega, \mathbf{Q}, \mathbf{r})\}$, with a state space $\mathbf{\Omega} = \{0,1,...,N\}$, an infinitesimal generator matrix with transition rate *q(i,j)* from state *i* to state *j*, $\mathbf{Q} = [q(i,j)]$, and a one-step reward rate, *r(i),* when the system is in state *i*. We further assume that:

- The Markov process is irreducible with respect to Ω with a unique steady-state distribution $\{\pi(i)\}_{i\in\Omega}$.
- Without loss of generality, a value $D < \infty$ exists for which

$$\sum_{j\neq i} q(i,j) \leq D \quad (i \in \Omega) \tag{2.1}$$

- For a given reward rate, *r(i),* the measure **g** is well defined by

$$\mathbf{g} = \sum_i \pi(i) \cdot r(i) \tag{2.2}$$

It is easier to obtain the average expected reward, **g**, after applying the standard uniformization technique (Tijms 1986, p.110). The purpose of uniformization is the prospect of studying our continuous-time model as a discrete-time Markov chain with a probabilistically equivalent generator matrix.

The uniformized one-step transition probability matrix $\mathbf{P} = [p(i,j)]$ is defined as :

$$p(i,j) = \begin{cases} q(i,j)/D & (j \neq i) \\ 1 - \sum_{i\neq j} q(i,j)/D & (j = i) \end{cases} \tag{2.3}$$

Note that the generator $\mathbf{Q}^d$ of the discrete-time Markov chain over epochs {0, 1/D, 2/D, ...} with one-step transition matrix **P** coincides with the infinitesimal generator matrix $\mathbf{Q} = [q(i,j)]$ of the continuous-time model, as $\mathbf{Q}^d = (\mathbf{P} - \mathbf{I}) \cdot D = \mathbf{Q}$ where **I** is the identity matrix.

The fact that the two models have equivalent generator matrices implies that the steady-state distributions are also equal, i.e., $\pi = \pi^d$. The latter are determined by $\pi\mathbf{Q} = 0$ and $\pi^d\mathbf{Q}^d = 0$. As a consequence, related performance measures such as average expected reward **g** for a given reward rate **r** are also equal. We can thus restrict our attention to the discrete-time model.

In order to study the average reward for our discrete-time model and with $\mathbf{P}^k$, the k^{th} power of the transition matrix **P**, for each $n = 0,1,2,\ldots$, we define the expected total reward functions V_n by:

$$V_n(i) = \sum_{k=0}^{n-1} \mathbf{P}^k r = \sum_{k=0}^{n-1} \left\{ \sum_j p^k(i,j) \cdot \mathbf{r}(j) \right\} \tag{2.4}$$

In words, this expression represents the expected cumulative reward over n steps, starting in state i at time 0 and receiving a reward $r(j)$ upon visiting state j.

Then, by virtue of our assumptions, we conclude that, for an arbitrary initial state i, the following expression is well defined and represents the average expected reward per unit time:

$$\mathbf{g} = \lim_{n\to\infty} \frac{1}{n} V_n(i). \qquad (2.5)$$

This average reward value accurately represents different performance measures of interest through the appropriate specification of the reward rate function **r**.

Example Consider an M/M/1/N loss system (that is, a single-server system with finite waiting room of size N-1, and exponential interarrival and service times) with arrival rate λ and service rate μ. The measures in Table 1 can then be obtained, where **r** is the reward rate function and $1\{A\}$ represents the indicator of an event A; i.e., $1\{A\} = 1$ if event A occurs and $1\{A\} = 0$ otherwise. □

Table 1: Performance Measures

g	**r**
Throughput	$\mu \cdot 1\{z > 0\}$
Loss probability	$1\{z = N\}$
Tail probability	$1\{z > t\}$
Mean number	z

The following simple recursion relation, directly obtained from (2.4), sometimes referred to as a one-step Markov reward or dynamic programming relation, provides a simple computational scheme:

$$Vn(i) = r(i) + \sum_j p(i,j) Vn-1(j) \qquad n \geq 1. \qquad (2.6)$$

This successive approximation is used to calculate or better approximate the value **g**, starting with $V_0(i) = 0$ for all i. Furthermore, this recursion can be used to approximate **g**, without averaging, by simply providing monotonically converging lower and upper bounds for **g**. These bounds provide a *guarantee of accuracy* for the reported result in situations where the numerical computation is terminated after exhausting a computing budget. To this end, define values $\boldsymbol{M}_n$ and $\boldsymbol{m}_n$ by

$$\begin{aligned} \boldsymbol{m}_n &= \min_j \left[V_n(j) - V_{n-1}(j)\right] \\ \boldsymbol{M}_n &= \max_j \left[V_n(j) - V_{n-1}(j)\right] \end{aligned} \qquad (2.7)$$

The following result, stated without proof, then applies. It is adopted from results in Markov decision theory (e.g., see Tijms 1986, Chapter 3). In that setting, it has already been introduced by Odoni (1969), and further extended, most notably, by Popyack (1985). In a non-decision setting, however, it remains largely unexploited.

Result: $\boldsymbol{m}_n \leq \boldsymbol{m}_{n+1} \leq \mathbf{g} \leq \boldsymbol{M}_{n+1} \leq \boldsymbol{M}_n \quad (n \geq 0) \qquad (2.8)$

2.1 Application to Discrete Event Computer Simulation

Generalized semi-Markov processes (GSMPs) are viewed not only as a precise model for describing DEDS, but also as a mathematical setting for analyzing discrete event processes (Glynn 1989). As such, they represent a widely recognized modeling and analysis framework for discrete event simulations (Schruben and Yücesan 1993).

A Generalized Semi-Markov Scheme (GSMS) is a four-tuple (**S**,**A**,E,p), where **S** is the (finite or countably infinite) state space and **A** is the (finite) event set. E is the event list, E: $\mathbf{S} \to 2^{\mathbf{A}}$; that is, for each state s$\in$**S**, E(s) denotes the set of active (feasible) events in state s. The evolution of the system, through state changes upon the occurrence of events, is governed by the probability measure p. That is, if e$\in E$(s), then upon the occurrence of event e the system moves to state s'$\in$**S** with probability p(s';s,e). We obtain a GSMP from a GSMS by introducing a sequence of clock samples, $\omega = \{\omega_e(n) : e \in \mathbf{A}, n = 0,1,2,...\}$, representing the time from the n^{th} activation of event e to the n^{th} scheduled occurrence of event e. The clock samples furnish the temporal dynamics lacking in GSMS.

Informally, the evolution of a GSMP can be summarized as follows: compare clock values of all active events in the current state s, determine the triggering event e^* (the one with the smallest clock value), update the process state s' in accordance with p(s';s, e^*), and update the clock values (update the values for all events that remain active in the new state -sometimes referred to as the 'residual lifetimes,' set new values for events that become active in the new state, and turn off the clocks for all events that become inactive (infeasible) in the new state). For a formal discussion, see Cassandras (1993, p.201).

Example (continued) The GSMP for the capacitated single-server queue is a birth-and-death process. The state space is **S** = {0,1,2,...,N} and the event set is **A** = {arrival, departure}. For $1 \leq s < N$, E(s)={arrival, departure}, while $E(0)$={arrival} and $E(N)$={departure}. Arrivals increment and departures decrement the state; hence, p(s+1;s,arrival)=1 for $0 \leq s < N$, while p(s-1;s,departure)=1 for $1 \leq s \leq N$. All other transitions have zero probability. The interarrival times, $\omega_{arrival}(n)$, and the service times, $\omega_{departure}(n)$, are all independent exponentially distributed random variables. □

Note that the process is called *Markovian* since the state transitions are governed by the probability measure p(s';s,e), which, in turn, depends only on the current state of the system. If all clock sequences are driven by the exponential distribution, we have a *continuous-time Markov process.* On the other hand, allowing any general probabilistic structure to drive the clock sequences produces a *semi-Markov process.* The term *generalized* is used to emphasize that residual lifetime distributions and/or state transition probabilities can depend on the entire history of the process. For example, these probability distributions may explicitly depend on the time of the n^{th} transition.

GSMP provides a precise modeling framework for discrete event simulations. There have been recent applications where it is also used for simulation analysis. For example, Glynn and Iglehart (1988) develop likelihood ratios for importance sampling leading to efficient queueing simulations. Glynn (1989) generalizes this result to DEDS and points out applications to parameter optimization. He also develops an external control variate scheme within the GSMP framework for variance reduction in simulation experiments. Glasserman (1991) uses the GSMP framework to introduce conditions validating the applicability of infinitesimal perturbation analysis to DEDS. Glasserman and Yao (1992) identify cases where variance reduction can be guaranteed through the application of common random numbers in a simulation experiment.

Our method provides a numerical alternative for performance assessment of DEDS that can be modeled as a Markov or a semi-Markov process. Note that although the approach was introduced within the context of continuous-time Markov processes, its application to semi-Markov processes is immediate through a data transformation (Tijms 1986, p. 200). While some generality of the GSMP framework is lost, the domain of applicability encompasses a large class of queueing models that are widely used to analyze manufacturing systems, telecommunication, service, and computer networks (Van Dijk 1993). The key advantage of our approach, however, is the guarantee of accuracy it offers for the steady-state performance metric. Such a guarantee of accuracy would be equally valuable in situations where computation is terminated prematurely due to a limited computing budget. With such a measure of accuracy, our method can be viewed as a viable numerical alternative to discrete event computer simulation.

3 AN ILLUSTRATION

In this section, we illustrate our approach through a simple but nevertheless unsolvable illustrative example: a finite-buffer assembly line. A sensitivity analysis is also conducted to study the behavior of the method with respect to system characteristics as well as the algorithm parameters.

3.1 Model

We consider a simple tandem queue that consists of two single-server queues in series as depicted in Figure 1. Jobs arrive at the first queue according to a Poisson process with parameter λ. The service requirements are assumed to be exponentially distributed with parameters μ_1 and μ_2 at queues 1 and 2, respectively. Jobs are served in a first-come-first-served order. In addition, queue 1 has a capacity for at most N_1 jobs in total and queue 2 for at most N_2.

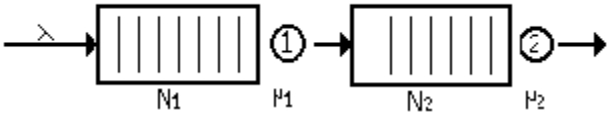

Figure 1: Capacitated Tandem Queue

The state of the system can be described by the vector $\bar{n} = (n_1,n_2)$ where n_i denotes the number of customers at station i, $i = 1, 2$. By $\bar{n}+e_i$ ($\bar{n}-e_i$), we denote the state of the system equal to $\bar{n}$ except for one customer more (less) at station i where $\bar{n}-e_i = \bar{n}$ for $n_i=0$. Consequently, by $\bar{n}-e_i+e_j$ we denote the state equal to $\bar{n}$ with one customer moved from station i to station j, where i=0 corresponds to an external arrival at station j (j can only be the first station of the configuration) and j=0 to a departure from the system at station i (again i is restricted to be the second station in the configuration).

A customer is denied access upon arrival when the first queue of the network is saturated ($n_1 = N_1$). A customer is recirculated to the first station if, upon its service completion at station 1, the second station is saturated ($n_2 = N_2$). Here we note that, because of the memoryless property of the exponential distribution, one can also state that the first station stops servicing when the second one is saturated. This, in turn, is known as *communication blocking* (Altiok and Perros 1986).

The transition rates can be summarized in the following way:

$$q(\bar{n},\bar{n}') = \begin{cases} \lambda \cdot 1\{n_1 < N_1\} & \bar{n}' = \bar{n} + e_1 \\ \mu_2 \cdot 1\{n_2 > 0\} & \bar{n}' = \bar{n} - e_2 \\ \mu_1 \cdot 1\{n_1 > 0\} 1\{n_2 < N_2\} & \bar{n}' = \bar{n} - e_1 + e_2 \end{cases}$$

Hence, with $D = \lambda + \mu_1 + \mu_2$, the uniformization constant, in accordance with (2.3), yields the probability of staying in the same state:

$$q(\bar{n},\bar{n}) = (1 - [\lambda \cdot 1\{n_1 < N_1\} - \mu_2 \cdot 1\{n_2 > 0\} - \mu_1 \cdot 1\{n_1 > 0\} 1\{n_2 < N_2\}] / D)$$

We consider two performance measures:

1. The throughput, T, of the system is defined as the number of customers departing from the second

queue per unit of time. The reward rate, $r(\bar{n})$, in this case is given by:

$$r(n_1,n_2) = \begin{cases} \mu_2, n_2 > 0, \\ 0, n_2 = 0. \end{cases}$$

2. The expected sojourn time, *W*, is defined as the time a customer spends in the system, including both the service time and the waiting time in each queue. This sojourn time can be calculated via Little's law ($L = T \cdot W$), where *L* is the average number of customers in the system.

3.2 Numerical Results

Characteristics of the tandem queue used in our experiments are summarized in Table 2. The parameters of the algorithm are given below.

The uniformization variable D and the corresponding step-size 1/D per transition was chosen to be

$$D = \lambda + \mu_1 + \mu \qquad (3.1)$$

As stopping criterion, we used:

$$\mathbf{M}_n - \mathbf{m}_n \leq \varepsilon = 0.001. \qquad (3.2)$$

Table 2: System Characteristics for the Queueing Example

Characteristics of the system			
No. of queues	**Servers per queue**	**Capacity**	**Utilization**
2	1	1 to 30	20% to 120%

3.3 Sensitivity Analysis and Further Discussion of Numerical Results

In this section we discuss the operating characteristics of the proposed method in detail. In particular, we will elaborate upon the following aspects:

- convergence rate,
- sensitivity with respect to characteristics of the system under study (capacity, utilization), and
- sensitivity with respect to its operating parameters (uniformization variable, stopping criterion).

Though these discussions are not directly necessary for the actual application of the method, we believe them to be of interest to practitioners for gaining further insight into some of the underlying features of the algorithm. Though such results are usually not reported (in detail), we find it useful to include them.

3.3.1 Performance of the Method: Convergence Rate

A representative illustration of the numerical experiments is provided in Figures 2 (throughput) and 3 (sojourn time) for different utilization levels. The illustration pertains to a system consisting of two single-server queues in tandem, each with a waiting room for four customers. These figures indicate that the lower and upper bounds $\mathbf{m}_n$ and $\mathbf{M}_n$ can be used as simple practical values to quickly determine the value of **g** within a prescribed accuracy. As such they support the Markov reward method as an easy-to-use tool in practice.

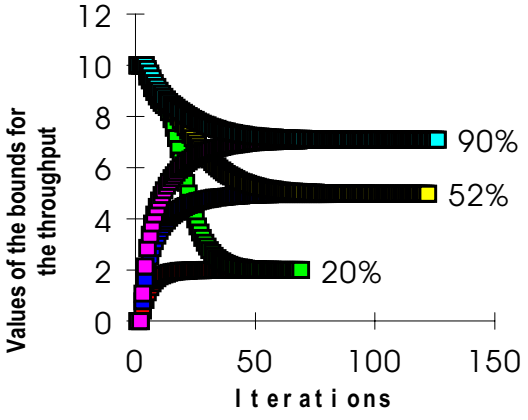

Figure 2: Convergence Rate for System Throughput

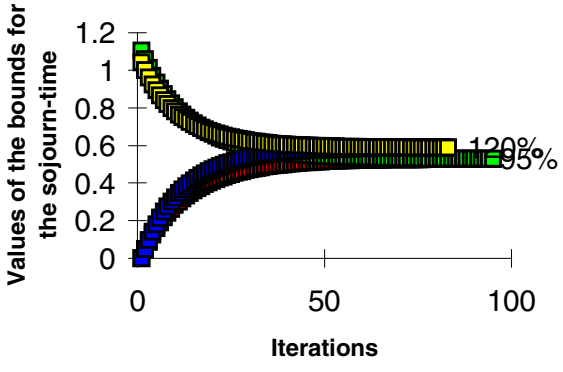

Figure 3: Convergence Rate for Sojourn Time

In addition to computational accuracy, the results provide an indication of the actual convergence to steady-state in terms of transitions (or events). For the original (non-uniformized) continuous-time Markov chain $\{X(t) : t \geq 0\}$, we can consider the transitions from state *i* to occur at a uniform rate of D rather than at a state-dependent rate $\Sigma\, q(i,j)$. However, only a fraction $\Sigma q(i,j)/D$ are real transitions out of state *i* corresponding to an event in the original queueing process such as an arrival or a service completion. The remainder are just fictitious transitions that leave the process unchanged in state *i*. As a consequence, the number of iterations required can be regarded as an upper

bound on the number of events (arrivals or service completions) that must be executed in a discrete-event simulation before the prescribed accuracy for a steady-state performance value is achieved.

The method is observed to converge fairly quickly; however, its rate of convergence, to the best of our knowledge, has not yet been formally determined. We argue that it actually follows a geometric pattern. We used the method of least - squares estimation for the purpose of curve fitting because the data failed to satisfy the underlying assumptions of regression analysis. The method has been applied to different cases with different capacities and utilization levels. Figure 4 shows, for the configuration with capacity 1 and 90 % utilization, the differences of the upper bound for the value of the system throughput, as calculated through the program at every step, and the steady-state value of the throughput, computed by solving a set of differential-difference equations. In both cases the values are plotted against the number of iterations. Our results are consistent with the derivations in Çinlar (1975, p.378), where geometric convergence rates are established for irreducible, aperiodic Markov chains. The results support the utility of the method as a practical tool for the steady-state approximations for queueing network performance measures.

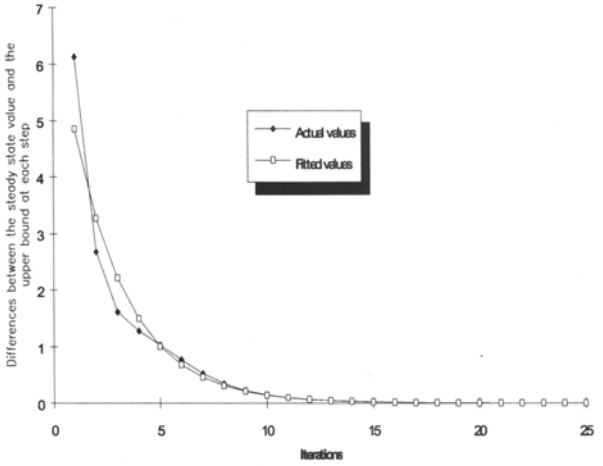

Figure 4: Convergence Rate of the Algorithm

3.4 Characteristics of the Configuration

An examination of the convergence rate with respect to changes in the system characteristics is certainly of great interest. Since the work in this paper has been restricted to relatively simple systems, we have only examined the impact of two system characteristics, utilization and capacity.

As anticipated, an increase in the value of these two parameters necessitates a larger number of steps for convergence. An increase in utilization modifies the distribution of the probability mass throughout the state space. For instance, at high utilization levels, most of the mass is concentrated around the congested states, that are attained very quickly. For low utilization rates, however, the opposite is true. As depicted in Figure 5, convergence is slower when the probability mass is more evenly distributed over the state space; those corresponds to the range of 0.6 to 0.8.

Increasing capacity results in an exponential growth of the number of iterations. Such a growth can quickly lead to prohibitively large run-time requirements (Figure 6). This is also an intuitive result as an increase in capacity corresponds to an increase in the number of states. In general, a system with a larger state space will take longer to attain steady state.

The steep increase in the number of iterations for systems with larger state space appears to limit the applicability of the proposed method to complex systems. There are, however, two approaches to mitigate this problem: truncation of the state space and acceleration of the convergence rate. Tijms (1986, p.195) proposes a relaxation method that modifies the computation of the reward function at each step through a dynamically selected relaxation factor to accelerate convergence. Our experimentation with the relaxation method did not yield satisfactory results; the number of iterations required for convergence was reduced only by one or two. Nevertheless, the use of a relaxation factor could prove to be fruitful when the algorithm is applied to systems with a larger state space. Next, we discuss the truncation of the state space.

3.5 Experimental Truncation Error Bound

Examination of a system with a very large state space is a difficult task. Thus, such a state space is normally truncated to be able to obtain a solution, but with no guarantee for the accuracy of the results. A question of great interest is then whether this truncation error could possibly be quantified.

Truncation errors have been experimentally obtained (for the configuration considered in section 3.1). The results indicate that a significant reduction in run times can be obtained by truncating the state space while incurring only a small truncation error. For instance, a truncation from 900 states (capacity of 30 at each station) down to 625 yields a truncation error in the values of the average throughput of 0.038 (0.43%). Table 3 shows the error in throughput at a 90% utilization level due to the truncation of capacity from 30 customers at each station to smaller values.

Further analysis reveals that the capacity restriction imposed on our configuration hardly makes any difference, compared to the uncapacitated system, when the maximum allowed number of customers increases above 60. Thus, the existing *steady-state* equations of the equivalent *product-form* configuration could be used to compute the value of the system throughput with a negligible error. Papanikas (1992) discusses analytic methods to provide *a priori* error bounds in Markov chains.

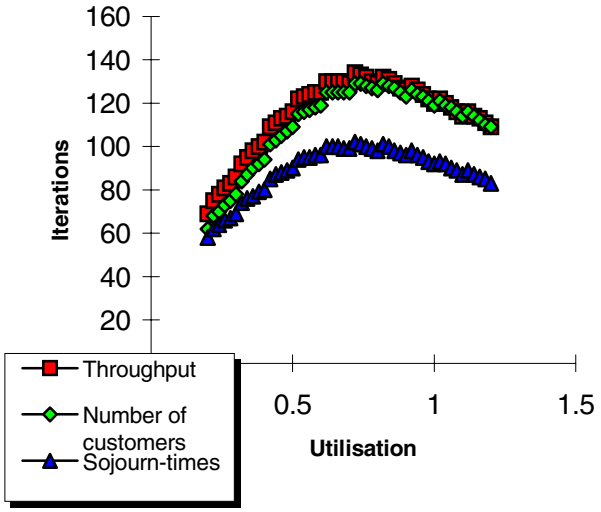

Figure 5: The Number of Iterations versus Utilization

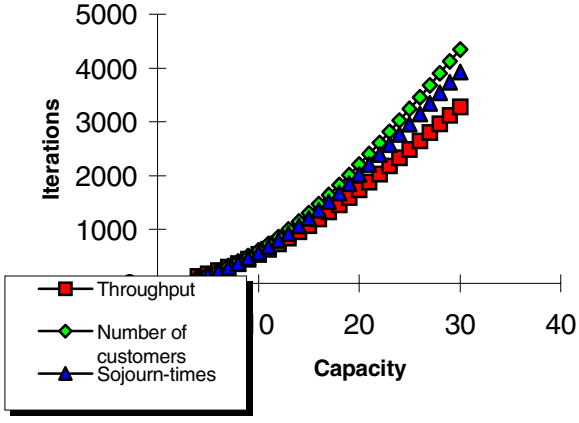

Figure 6: The Number of Iterations versus System Capacity

3.5.1 Characteristics of the Algorithm

The convergence rate of the algorithm is also influenced by its parameters: the value of the uniformization variable, D, and the value of the stopping criterion, ε. The increase in the number of iterations resulting from an increase in the value of D can be estimated through the following equation : (D / Iterations) = (D' / Iterations'). As the best possible value for D, we recommend the one that satisfies equation (2.1) with equality.

The value of the stopping criterion has a big impact on convergence rate. A smaller value, that will provide a more accurate approximation, will result in a larger number of iterations. An appreciation of the changes in convergence rate, expressed as the number of iterations, due to changes in the values of D and ε can be obtained from Table 4.

Table 3: Error in State Space Truncation

Capacity	Avg T-put	Difference	% Δ
10	8.385727	0.567223	6.33
15	8.699578	0.253372	2.83
20	8.842422	0.110528	1.23
25	8.914626	0.038324	0.43

Table 4: Sensitivity to Parameters

	Utilization (%)			
D	**20**	**50**	**90**	**120**
32	103	149	139	109
40	130	188	174	137
ε				
0.001	69	116	126	109
0.0001	82	143	157	137

4 CONCLUDING COMMENTS

A computational method is introduced to calculate the values of performance measures by successive Markov reward approximations. This method enables one to obtain on-line error bounds for the accuracy of the approximation. Extensive numerical analysis conducted in Papanikas (1992) and summarized in this paper illustrates that these bounds can be quite practical to provide a guaranteed accuracy within a reasonably small number of steps: as such, the results support the approach as a practical alternative to digital simulation, where no guarantees can be furnished on steady-state estimations. The bounds could also be used to show how the convergence rates depend on system input parameters, such as capacity and load. This is particularly useful for the design and operation of modern manufacturing systems where steady-state conditions can never be attained due to small lot sizes and a great variety of products. Chance (1993), for example, describes a queueing network model of such a semiconductor manufacturing process with re-entrant routing, rework, and machine breakdowns.

It should be noted that there exists a rich literature on numerical techniques applied to Markovian modeling (Freiberger and Grenander 1971). A comparison of such techniques can be found in (Stewart 1978). We have not considered them since the method of successive Markov reward approximations is, in our view, a modern and proven technique that is intuitive, easy to implement, and efficient. An added benefit is its ready generalizability to semi-Markovian processes, that significantly increases its domain of applicability and makes it a viable alternative to discrete event computer simulation.

REFERENCES

Abate, J., and W. Whitt. 1986. Transient behavior of the M/M/1 queue: starting at the origin. *Queueing Systems* 2(1): 41-65.

Abate, J., and W. Whitt. 1987. Transient behavior of regulated Brownian motion, II: non-zero initial conditions. *Advances in Applied Probability* 19: 599-631.

Altiok, T., and H.G. Perros. 1986. Open networks of queues with blocking: split and merge configurations," *IIE Transactions* 18: 251-261.

Cassandras, C.G. 1993. *Discrete event systems: modeling and performance analysis.* Boston, MA: Irwin..

Chance, F.D. 1993. Conjectured upper bounds on transient mean total waiting times in queueing networks. In *Proceedings of the 1993 Winter Simulation Conference,* ed. G.E. Evans, M. Mollaghashemi, E.C. Russell, and W.E. Biles, 414-421. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Çinlar, E. 1975. *Introduction to stochastic processes*. Englewood Cliffs, NJ: Prentice Hall.

Freiberger, W.F., and U. Grenander. 1971. *A short course in computational probability and statistics*, New York, NY: Springer-Verlag.

Glasserman, P. 1991. *Gradient estimation via perturbation analysis*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Glasserman, P., and D.D. Yao. 1992. Some guidelines and guarantees for common random numbers. *Management Science* 38 (6): 884-908.

Glynn, P.W. 1989. A GSMP formalism for discrete event systems. *Proceedings of the IEEE.* 77: 14-23.

Glynn, P.W., and D.L. Iglehart. 1988. Simulation methods for queues: an overview," *Queueing Systems.* 3: 221-256.

Kelton, W. D. 1985. Transient exponential-Erlang queues and steady-state simulation. *Communications of the ACM* 28: 741-749.

Kelton, W.D. 1989. Random initialization methods in simulation. *IIE Transactions* 21: 355-367.

Kelton, W.D., and A.L. Law. 1985. The transient behavior of the M/M/s queue with implications for steady-state simulations. *Operations Research*. 33: 378-396.

Odoni, A. R. 1969. On finding the maximal gain for Markov decision processes," *Operations Research.* 17: 857 - 860.

Odoni, A.R., and E. Roth. 1983. An empirical investigation of the transient behavior of stationary queueing systems. *Operations Research.* 31: 432-455.

Papanikas, K. 1992. On-line error bounds for steady state approximations, Unpublished MSc. Thesis, Department of Operational Research, University of Southampton, Southampton, U.K.

Popyack, J. L. 1985. Accelerated convergence for successive approximations in Markov decision processes, Research report, Drexel University, Philadelphia.

Schruben, L. 1981. Control of initialization bias in multivariate simulation response. *Communications of the ACM* 24: 246-252.

Schruben, L. 1982. Detecting initialization bias in simulation output. *Operations Research,* 30: 569-590.

Schruben, L., and E. Yücesan. 1993. Modeling paradigms for discrete event simulation. *Operations Research Letters.* 13: 265-275.

Stewart, W.J. 1978. A comparison of numerical techniques in Markov modeling. *Communications of the ACM* 21: 144-152.

Tijms, H. C. (1986), *Stochastic modeling and analysis; a computational approach*. New York, NY: Wiley.

Van Dijk, N. M. 1993 *Queueing networks and product forms*. New York, NY: Wiley.

Yücesan, E. 1993. Randomization tests for initialization bias in simulation output. *Naval Research Logistics* 40: 643-663.

AUTHOR BIOGRAPHIES

ENVER YÜCESAN is a Professor of Operations Management at the European Institute of Business Administration, INSEAD. He received his Ph.D. from Cornell University in 1989. His research interests include simulation optimization and supply chain management. He is a member of IIE and INFORMS. His email and web addresses are <enver.yucesan@insead.edu> and <www.insead.edu/facultyresearch/tm/yucesan>.

LUK N. VAN WASSENHOVE is the Henri Ford Chaired Professor of Manufacturing at INSEAD. His research and teaching are concerned with integrated operations management, continual improvement, and learning. His email and web addresses are <luk.van.wassenhove@ insead.edu> and <www.insead.edu/ facultyresearch/tm/wassenhove>.

KLENTHIS PAPANIKAS has obtained his Master's Degree in Operations Research at the University of Southampton. The work presented in this paper is based on his dissertation.

NICO M. VAN DIJK is a Professor of Operations Research at the University of Amsterdam/Incontrol Business Engineers. His email address is <nico.van.dijk@ incontrol.nl>.