CBA 2004 - XV Congresso Brasileiro de Automática - 21 a 24 de setembro de 2004

GENERALIZED $M/G/c/c$ STATE DEPENDENT QUEUEING NETWORK ANALYSIS

H. F. Araújo*, M. L. G. Toledo†, F. R. B. Cruz†, J. MacGregor Smith‡

* *Departamento de Matemática e Estatística*
Pontifícia Universidade Católica de Minas Gerais
Belo Horizonte - MG, Brazil

† *Departamento de Estatística, Universidade Federal de Minas Gerais,*
31270-970 - Belo Horizonte - MG, Brazil

‡ *Department of Mechanical and Industrial Engineering, University of Massachusetts,*
Amherst MA 01003, USA

Emails: harley.fa@globo.com, mlgt@est.grad.ufmg.br, fcruz@ufmg.br,
jmsmith@ecs.umass.edu

Abstract— We describe a methodology for analysis of state-dependent $M/G/c/c$ queueing networks in which the service rate is subject to congestion, that is, is a function of the number of customers in the system. Important performance measurements are easily computed, such as the blocking probability, the throughput, the expected number of customers in the system (known as work-in-process), and expected waiting time. The methodology forms a basic building block useful in many practical applications and contexts. Preliminary computational results demonstrate that the methodology provides accurate results in many topological configurations.

Keywords— Queueing, Finite Capacity, State Dependent.

1 Introduction

A line (or queue) is formed every time there is more demand than service available. In order to answer crucial questions such as "what should the level of service be to ensure satisfactory service?", one needs to know for instance how long a customer may be willing to wait and needs to be able to compute performance measurements like how many customers are expected to be waiting. Our objective with this paper is to present a general methodology suitable for analysis of $M/G/c/c$ state dependent queueing networks (**M**arkovian interarrival-time distribution, **G**eneral service-time distribution, c parallel servers, and total capacity c). We assume that service times are dependent on the number of customers in the system (see Figure 1).

1.1 Motivation

One problem that could be treated by the methodology presented here is related to pedestrian flows (Fruin, 1971; Tregenza, 1976; Yuhaski and MacGregor Smith, 1989). Other possible applications include vehicular traffic modeling (Alfa and Neuts, 1995; Jain and MacGregor Smith, 1997), industrial engineering problems (MacGregor Smith and Daskalaki, 1988), and others with decreasing service rates with increasing customer traffic (MacGregor Smith and Chikhale, 1995).

Many studies of pedestrian walking speeds have been conducted (Fruin, 1971; Tregenza, 1976) and there are several common factors that can be associated with the different pedestrian walking speeds. For instance, men tend to walk faster than women. Mean walking speeds tend to vary among people of different age groups, as adolescents generally walk faster than do adults. A general decline in walking speed is observed with an increase in age. Finally, trip purpose can also affect walking speed of a pedestrian. As the traffic density increases, however, these factors have a much smaller effect on the walking speed of individual pedestrians since the progress of faster moving pedestrians tends to become impeded by slower moving pedestrians as the limited floor space of the corridor becomes occupied by more pedestrians. Thus, when the crowd density becomes even slightly moderate, crowd density tends to become the most significant factor in determining pedestrian walking speeds, as can be seen in Figure 1, which presents experimental curves (a through f) that relate the walking speed of a pedestrian to the crowd density, based on various empirical studies (Tregenza, 1976).

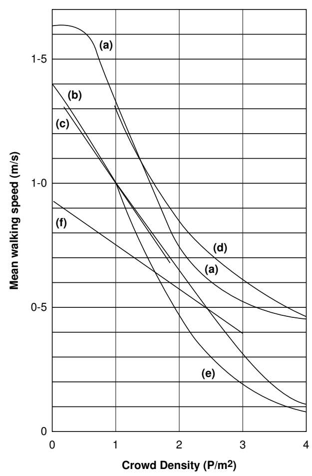

Figure 1: Average walking speed for pedestrians.

1.2 *Outline*

This paper is organized as follows. In Section 2, we present some background material with an analytical congestion model for single queues. Then, in Section 3, we present the Generalized Expansion Method used to model more complex topologies, along with a new implementation of it. In Section 4, we present computational results of our network analysis methodology and compare it with simulations. Finally, Section 5 completes the paper with conclusions and final remarks.

2 Congestion Models

2.1 *Analysis of Single Queues*

In order to develop the model, we shall concentrate on the pedestrian traffic flow problem, but the methodology can be extended to other contexts (Jain and MacGregor Smith, 1997). We are assuming that the time of arrival to a corridor for each pedestrian is independent of that of other pedestrians arriving at the entrance of the corridor. Additionally, we assume that pedestrians arrive to the corridor according to a Poisson process with rate λ. The M in the model represents these exponential arrival times to the corridor. Bulk arrivals can also occur, however, it is reasonable to suggest that as pedestrians enter the doorways, they do so individually, so Poisson arrivals are very reasonable.

In the $M/G/c/c$ state dependent queueing model of a corridor, the corridor behaves as the server to its occupants. The number of servers is equal to the capacity of the corridor c. Pedestrians that arrive to the corridor when it is at capacity may not enter the corridor. The queue consists entirely of the corridor without any additional buffer floor space. Therefore, the total number of pedestrians that are allowed in the entire queueing system is also equal to the capacity of the corridor.

The service time of the queueing model of a corridor is equal to the time for a pedestrian to traverse the entire length of the corridor. The rate at which this traversal occurs, the service rate $f(n)$, is dependent on the number of occupants n within the corridor and follows a general distribution G. We assume that the pedestrian crowd density within the corridor is approximately uniformly distributed throughout the corridor, and that it is this corridor density that determines the average walking speed of a pedestrian within the corridor. The queueing model is state dependent because a change in the number of pedestrians within the corridor will change the service rate of every pedestrian within the corridor. Thus, if there are n pedestrians within a corridor, all of them will have service rate $f(n)$. If there is an arrival to the corridor, and that pedestrian enters the corridor, the service rate will change to $f(n+1)$. Likewise, if a pedestrian traverses the length of the corridor and leaves the system, the service rate will change to $f(n-1)$.

The limiting probabilities for the number of pedestrians in a $M/M/c/c$ queueing model have been developed before (Yuhaski and MacGregor Smith, 1989) and it has been shown that $M/M/c/c$ and $M/G/c/c$ state dependent queues are stochastically equivalent (Cheah and MacGregor Smith, 1994). Thus, the limiting probabilities for the number of pedestrians in an $M/G/c/c$ state dependent queueing model $p_n = \Pr\{N = n\}$ are as follows:

$$p_n = \left\{ \frac{\Big[\lambda E[T_1]\Big]^n}{n! f(n) f(n-1) \cdots f(2) f(1)} \right\} p_0, \qquad (1)$$

for $n = 1, 2, \ldots, c$, in which

$$p_0^{-1} = 1 + \sum_{i=1}^{c} \left\{ \frac{\Big[\lambda E[T_1]\Big]^i}{i! f(i) f(i-1) \cdots f(2) f(1)} \right\}, \qquad (2)$$

is the empty system probability, λ is the arrival rate, $E[T_1] = l/V_1$ is the expected service time of a lone occupant in a corridor of length l, considering that $V_1 \approx 1.5$ m/s is the speed of a lone pedestrian, and $f(n) = V_n/V_1$ is the service rate, considered to be the ratio of the average walking speed of n people in the corridor to that of a lone occupant V_1.

From Eq. (1), important performance measures can be derived:

$$\left.\begin{array}{rcl} p_c &=& \Pr\{N = c\}, \\ \theta &=& \lambda(1 - p_c), \\ L &=& E[N] = \displaystyle\sum_{n=1}^{c} n p_n, \\ W &=& E[T] = L/\theta, \end{array}\right\} \qquad (3)$$

in which p_c is the blocking probability, θ is the throughput in ped/s, L is the expected number of customers in the corridor (also known as work-in-process, WIP), and W, here derived from Little's law, is the expected service time in seconds. The average walking speed $f(n)$ for n pedestrians in the corridor could be calculated using several different models. In the following section, we shall see two of them (Yuhaski and MacGregor Smith, 1989).

2.2 *Service Rate Functions*

Basically, what one wants is that the congestion model represents the effect depicted in Figure 1, in which the service rate depends on the number of customers in the system. One possibility is to assume a linear model in which the service rate decays following the expression:

$$f(n) = \frac{V_n}{V_1} = \frac{c+1-n}{c}, \tag{4}$$

in which V_n is the average walking speed for n pedestrians in a corridor, V_1 is the typical walking speed of a lone occupant, and c is the capacity of the corridor which is assumed to be:

$$c = \lfloor klw \rfloor, \tag{5}$$

in which $\lfloor x \rfloor$ is the largest integer not superior to x, l is the length, w is the width, and k is the capacity of the corridor per square-unit. Considering pedestrian related applications and realizing that k represents the crowd density, a reasonable value for k would be 5, since at this density, movement ceases, in accordance to experimental studies (Tregenza, 1976).

On the other hand, it is also possible to assume an exponential model in which the service rate decays following the expression:

$$f(n) = \exp\left[-\left(\frac{n-1}{\beta}\right)^{\gamma}\right], \tag{6}$$

in which

$$\gamma = \log\left[\frac{\log(V_a/V_1)}{\log(V_b/V_1)}\right] / \log\left(\frac{a-1}{b-1}\right) \tag{7}$$

and

$$\beta = \frac{a-1}{[\log(V_1/V_a)]^{1/\gamma}} = \frac{b-1}{[\log(V_1/V_b)]^{1/\gamma}}, \tag{8}$$

in which V_n and V_1 are as defined before. The values a and b are arbitrary points used to adjust the exponential curve. In pedestrian related applications, commonly used values are $a = 2lw$ and $b = 4lw$ corresponding to crowd densities of 2 and 4 ped/m^2 respectively. Looking at the curves presented in Figure 1, reasonable values for such points are $V_a = 0.64$ and $V_b = 0.25$.

3 The Generalized Expansion Method

The Generalized Expansion Method (GEM) was proposed by Kerbache and MacGregor Smith (1987) back in the eighties and has a long tradition in the subject. The GEM is a combination of repeated trials and node-by-node decomposition approximation methods, with a key characteristic that an artificial holding node is added preceding each finite queue in the network in order to register blocked customer that attempt to enter the finite node when it is at capacity, see Figure 2. By the addition of holding nodes, the queueing network is "expanded" into an equivalent Jackson network, in which each node can then be decomposed and analyzed separately. Now, we shall describe briefly the GEM for series queues.

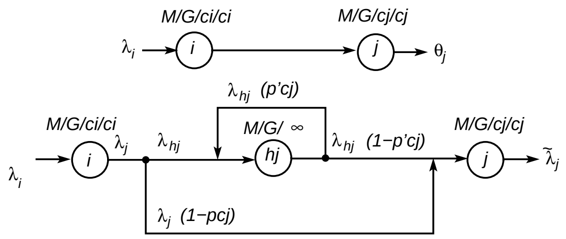

Figure 2: Generalized expansion method.

3.1 Series Queues

Figure 2 shows an example of corridors configured in series and the corresponding queueing network. The GEM consists of three stages that are performed for each finite node in the original queueing network.

Stage 1: Network Reconfiguration

For each node with finite capacity, an artificial node is added directly preceding it, as shown in Figure 2. Customers that attempt to move to the new node but are unable since it is at capacity are re-routed to the artificial node. The probability that an arriving customer is blocked by node j equals p_{c_j}. Thus, with probability $(1-p_{c_j})$, it will enter node j, and with probability p_{c_j} it will enter holding node h_j. The holding node is modeled as an $M/G/\infty$ queue, so that there will be no waiting to enter this node.

After service at the holding node, the customer will be blocked with a new probability, p'_{c_j}. With probability $(1 - p'_{c_j})$, it will proceed to the following node. Otherwise, it must retrace its path through the feedback loop into artificial node h_j again.

Stage 2: Parameter Estimation

The value of p_{c_j} can be determined from known analytical results. For $M/G/c/c$ state dependent queues, such a value is given directly by Eq. (1), i.e., $p_{c_j} = \Pr\{N = c_j\}$. The value of p'_{c_j} is determined from approximation results. After a customer completes its service at holding node h_j, it is forced to return with probability p'_{c_j}, for another immediate service delay. An approximation that uses diffusion techniques states that (Labetoulle and Pujolle, 1980):

$$p'_{c_j} = \left\{ \frac{\mu_j + \mu_{h_j}}{\mu_{h_j}} - \frac{\lambda\left[(r_2^{c_j} - r_1^{c_j}) - (r_2^{c_j-1} - r_1^{c_j-1})\right]}{\mu_{h_j}\left[(r_2^{c_j+1} - r_1^{c_j+1}) - (r_2^{c_j} - r_1^{c_j})\right]} \right\}^{-1}, \tag{9}$$

in which r_1 and r_2 are the roots to:

$$\lambda_{\text{ext}} - (\lambda_{\text{ext}} + \mu_{h_j} + \mu_j)x + \mu_{h_j}x^2 = 0. \tag{10}$$

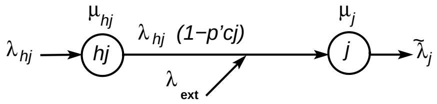

Figure 3: External arrival rate λ_{ext}.

Defined with help of Figure 3, the external arrival rate λ, used in Eq. (10), is:

$$\begin{cases} \lambda_{\text{ext}} & = & \tilde{\lambda}_j - \lambda_{h_j}(1 - p'_{c_j}), \\ \tilde{\lambda}_j & = & \lambda_j(1 - p_{c_j}), \\ \lambda_{h_j} & = & \lambda_j(p_{c_j}), \\ \lambda_j & = & \lambda_i(1 - p_{c_i}) = \tilde{\lambda}_i. \end{cases} \quad (11)$$

Using renewal theory, it can be shown that the service rate of the holding node is, in the exponential case, as follows (Kleinrock, 1975):

$$\mu_{h_j} = \frac{2\mu_j}{1 + \sigma_j^2 \mu_j^2}, \quad (12)$$

in which σ_j^2 is the service time variance. However, since the service rate is state dependent for the queue system we deal with here, a reasonable assumption is to consider the worst case:

$$\mu_{h_j} = \mu_j \approx \frac{c_j}{E[T_1]/f(c_j)}, \quad (13)$$

in which c_j is the maximum number of servers in parallel and $E[T_1]/f(c_j)$ is the service time for c_j occupants.

Stage 3: Feedback Elimination and Service Update

A reconfiguration of the holding node is performed, so that the strong dependencies in arrival processes caused by the repeated visits (feedback) to the artificial node are removed. The feedback arc is removed from the holding node by recomputing the service rate at this node as follows:

$$\mu'_h = (1 - p'_{c_j})\mu_{h_j}. \quad (14)$$

Finally, the average service time that a customer spends at node i preceding node j is given by:

$$\tilde{\mu}_i^{-1} = \mu_i^{-1} + p_{c_j}(\mu'_h)^{-1}. \quad (15)$$

Eq. (15) represents the last step of the GEM, which ultimate goal is to provide an approximation scheme to update the service rates of upstream nodes that takes into account all blocking after service in there, caused by downstream nodes.

3.2 Split and Merge Topologies

Merging and splitting topologies, can also be handled by the GEM. In splitting topologies, the

```
algorithm
    read G(V, A) and arrival rates, λi, ∀ i ∈ V
    read routing probabilities, pij, (i, j) ∈ A
        /* pre-evaluate every node */
    P ← ∅
    while P ≠ V
        choose k ∈ (V \ P)
        if {i|(i, k) ∈ A} ⊆ P then
                /* compute performance measures */
            E[T1]k ← lk/V1
            compute Pr{N = ck}, θk, Lk, Wk
            P ← P ∪ {k}
                /* forward information */
            λj ← λj + pkj θk, ∀ (k, j) ∈ A
        end if
    end while
        /* evaluate every node */
    P ← ∅
    θi^max ← ∞, ∀ i ∈ V
    while P ≠ V
        choose k ∈ (V \ P)
        if {j|(k, j) ∈ A} ⊆ P then
                /* update performance measures */
            E[T1]k ← min E[T1]k
                    s.t.: θk ≤ θk^max,
                          E[T1]k ≥ lk/V1
            compute Pr{N = ck}, θk, Lk, Wk
            P ← P ∪ {k}
                /* back-propagate information */
            update θi^max, ∀ (i, k) ∈ A
        end if
    end while
    write Pr{N = ci}, θi, Li, Wi, ∀ i ∈ V
end algorithm
```

Figure 4: Performance evaluation algorithm.

routing probabilities need to be known, then, the throughput has only to be decomposed according to the branching probabilities. Similar changes must to be made in the merging topology cases. In such cases, the arrival rate to the merging node is the sum of the throughput of the previous nodes. In conclusion, any general network topology can be analyzed once these building blocks are complete. Now, we shall describe a new algorithm to compute Eq. (15), especially developed for $M/G/c/c$ state dependent queueing networks.

3.3 Data Structures and Algorithm

First, let us define the network under study as a graph $G(V, A)$, in which V is a finite set of nodes (corridors) and A, a finite set of arcs (connections between pair of corridors).

The proposed algorithm, Figure 4, is based on the fact that blocking in the downstream nodes cause blocking after service at the upstream nodes. As a result, the effective service rates at the upstream nodes are reduced in a similar fashion as in Markovian systems, in accordance to Eq. (15). In $M/G/c/c$ state-dependent queueing networks, a similar effect is present, regardless of the dynamic changes in the service rates as users joint or leave the system. Thus, the algorithm progressively adjusts the state-dependent service rate curve, as blocking increases in the downstream nodes. The adjustments are made simply considering that the flow conservation throughout the networks is a constraint that must be satisfied.

The performance evaluation algorithm, see Figure 4, is comprised of two parts. First, a pre-

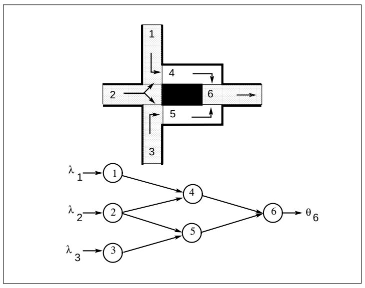

Figure 5: Corridors in a mixed topology.

evaluation is performed to every node in the network disregarding any possible inter-node blocking effect. In other words, the pre-evaluation procedure approximates the actual network to a Jackson network, in which each node can be analyzed separately. Notice that the pre-evaluation procedure is a variant of a labeling algorithm for determination of shortest paths in graphs (Dijkstra, 1959). Under very light traffic, with negligible inter-node blocking effects, the network is actually a Jackson network and the pre-evaluation procedure alone will do just fine for an accurate approximation of all performance measures.

For example, in the network illustrated in Figure 5, a possible valid sequence to perform pre-evaluations is $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 6$. The reason that a node could be pre-evaluated only after all of its predecessors is that one needs to know the overall arrival rate at this node before proceeding with any calculation and, clearly, the total arrival rate is a function of the outputs of all predecessor nodes.

The second part of the algorithm, see Figure 4, evaluates the performance measures, but now seeking flow conservation. By flow conservation we mean that $\theta_{k_1} \leq \theta_{k_2} \leq \cdots \leq \theta_{k_k}$ for every disjoint path $(k_1, k_2, \cdots, k_k)$ from an origin k_1 to a destination k_k in the queueing network. The estimates are improved by means of adjustments in the expected service time for lone occupant of each node k, $E[T_1]_k$. In order to describe the procedure, we define $\theta_k^{\max}$ as the maximum throughput of node k, i.e. the maximum flow that can be forwarded to the set of successors of k. Thus, if k is an end node (no successors), obviously $\theta_k^{\max} = \infty$. Otherwise, the update of $\theta_k^{\max}$ will depend on the knowledge of the throughput of all successors.

Initially, all nodes are assumed to have no limits and $\theta_k^{\max}$ is set to infinity. Starting from the end nodes, $E[T_1]_k$ is minimized, subject to providing a throughput less or equal to $\theta_k^{\max}$ and not being inferior to l_k/V_1, in which l_k is the length of the kth corridor and V_1, the walking speed for lone occupant ($V_1 \approx 1.5$ m/s, for the pedestrian application). Then, all performance measures of k are updated, taking into account this new $E[T_1]_k$. Finally, $\theta_i^{\max}$ is updated for the set of predecessors of k. Updating is done by assigning to predecessors an equal part of θ_k. If some predecessor is not able to provide its share, the flow is rerouted among the other predecessor, to ensure flow conservation.

The evaluation procedure was designed to correct estimates in cases such as those with bottlenecks in the downstream nodes. The effect of the procedure in networks submitted to extremely light traffic and without any bottlenecks should be marginal to none. The evaluation procedure is also a labeling algorithm working in reverse. For the network presented in Figure 5, a possible valid sequence to perform the evaluations is $6 \rightarrow 4 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3$, since a node can be evaluated only if all of its predecessors are already evaluated.

4 Computation Experiments

The performance evaluation algorithm was coded in C++ and all computational experiments were carried out on a PC, CPU Pentium II 400 MHz, 256 MB RAM, running Windows NT 4.0 operating system. For the sake of the argument, only the exponential congestion model, Eq. (6), was considered. In order to confirm the accuracy of the estimates, a discrete-event digital simulation model (Cruz et al., 2003) was used. After much experimentation (not shown) for reaching steady-state, simulations were run for 24,000 seconds, assuming the first 4,000 seconds as a *burn-in* period. Confidence intervals of 95% were computed based on 30 replications. The cpu times reported are only for the simulations. The performance evaluation algorithm usually runs within a single minute.

We show results for a three-node version of the system presented in Figure 5 in which corridor #2 does not receive flows, corridors #1 and #4 are collapsed into one single corridor #1, and corridors #3 and #5 are collapsed into one single corridor #3. All nodes were assumed 8.5 m long. Collapsed nodes #1 and #3 are 2.4 m wide, and node #6 is only 1.2 m wide. Notice that besides receiving flows from two nodes, node #6 is a bottleneck.

Analytical and simulation results for all performance measures are available in Table 1. Here, we can confirm the accuracy of the analytical model in almost all situations tested. In extreme cases of heavy unbalance between λ_1 and λ_3, the error can be as large as 92% (see Table 1) in the expected number of users and in the expected service times. While this imbalance is not likely to happen in practice, it is intended to illustrate a worst case scenario. Notice, however, that the throughput, the most regarded performance measure, is always extremely accurate. Also large is the error for the blocking probability in node #6. Here, we can see that the analytical method tends to overestimate this measure.

Table 1: Performance measures for a three-node merging network.

		node 1 (8.5 × 2.4)				node 3 (8.5 × 2.4)				node 6 (8.5 × 1.2)				
			simulation				simulation				simulation			
λ_1 λ_3		anal.	aver.	95% CI	Error	anal.	aver.	95% CI	Error	anal.	aver.	95% CI	Error	CPU
2.9 0.1	p_c	0.7050	0.7135	[0.7130;0.7139]	-1.2%	0.0000	0.0000	[0.0000;0.0000]	0.00%	0.5267	0.4994	[0.4994;0.4995]	5.46%	
	θ	0.8554	0.8302	[0.8294;0.8310]	3.0%	0.1000	0.1006	[0.0998;0.1014]	-0.58%	0.9554	0.9308	[0.9308;0.9308]	2.64%	
	L	101.6	101.5	[101.5 ; 101.5]	0.0%	0.5726	7.066	[7.001 ; 7.131]	**-91.9%**	50.05	51.00	[51.00 ; 51.00]	-1.86%	
	W	118.8	122.3	[122.2 ; 122.4]	-2.9%	5.7256	70.24	[70.00 ; 70.48]	**-91.8%**	52.39	54.79	[54.79 ; 54.79]	-4.39%	(0h 26m 24s)
2.5 0.5	p_c	0.8114	0.8127	[0.8121;0.8133]	-0.2%	0.0572	0.0716	[0.0684;0.0748]	-20.2%	0.6855	0.4995	[0.4995;0.4996]	37.2%	
	θ	0.4714	0.4683	[0.4671;0.4694]	0.7%	0.4714	0.4625	[0.4613;0.4637]	1.92%	0.9428	0.9308	[0.9308;0.9308]	1.29%	
	L	101.8	101.7	[101.7 ; 101.8]	0.1%	42.32	88.28	[87.77 ; 88.79]	-52.1%	50.53	51.00	[51.00 ; 51.00]	-0.92%	
	W	215.9	217.2	[216.7 ; 217.7]	-0.6%	89.76	190.9	[189.6 ; 192.1]	-53.0%	53.60	54.79	[54.79 ; 54.79]	-2.18%	(1h 46m 31s)
2.1 0.9	p_c	0.7755	0.7839	[0.7829;0.7848]	-1.1%	0.4762	0.4690	[0.4666;0.4713]	1.55%	0.6857	0.4996	[0.4996;0.4996]	37.3%	
	θ	0.4714	0.4537	[0.4519;0.4556]	3.9%	0.4714	0.4771	[0.4752;0.4789]	-1.18%	0.9428	0.9308	[0.9308;0.9308]	1.29%	
	L	101.7	101.6	[101.5 ; 101.7]	0.1%	100.9	98.79	[98.68 ; 98.91]	2.09%	50.53	51.00	[51.00 ; 51.00]	-0.92%	
	W	215.8	223.9	[222.9 ; 224.8]	-3.6%	214.0	207.1	[206.3 ; 207.9]	3.30%	53.60	54.79	[54.79 ; 54.79]	-2.18%	(1h 57m 32s)
1.7 1.3	p_c	0.7227	0.7279	[0.7266;0.7292]	-0.7%	0.6374	0.6394	[0.6378;0.6409]	-0.31%	0.6857	0.4997	[0.4996;0.4997]	37.2%	
	θ	0.4714	0.4619	[0.4601;0.4636]	2.1%	0.4714	0.4689	[0.4672;0.4707]	0.53%	0.9428	0.9308	[0.9308;0.9308]	1.29%	
	L	101.6	101.2	[101.2 ; 101.3]	0.4%	101.4	100.8	[100.7 ; 100.9]	0.62%	50.53	51.00	[51.00 ; 51.00]	-0.92%	
	W	215.6	219.2	[218.4 ; 220.1]	-1.7%	215.1	215.0	[214.2 ; 215.8]	0.08%	53.60	54.79	[54.79 ; 54.79]	-2.18%	(1h 58m 52s)
1.5 1.5	p_c	0.6857	0.6892	[0.6877;0.6907]	-0.5%	0.6857	0.6898	[0.6886;0.6910]	-0.59%	0.6857	0.4997	[0.4996;0.4997]	37.2%	
	θ	0.4714	0.4656	[0.4636;0.4676]	1.2%	0.4714	0.4652	[0.4632;0.4672]	1.34%	0.9428	0.9308	[0.9308;0.9308]	1.29%	
	L	101.5	101.1	[101.0 ; 101.2]	0.4%	101.5	101.1	[101.0 ; 101.2]	0.46%	50.53	51.00	[51.00 ; 51.00]	-0.92%	
	W	215.4	217.2	[216.2 ; 218.2]	-0.8%	215.4	217.3	[216.4 ; 218.2]	-0.88%	53.60	54.79	[54.79 ; 54.79]	-2.18%	(1h 58m 35s)

5 Concluding Remarks

We have presented a general methodology suitable for the analysis of $M/G/c/c$ systems with state dependent service rates. The importance of this model was stressed. We have also discussed in detail the application of the model to pedestrian traffic flow problems. Computational results demonstrated that the methodology is accurate and may be considered as a basic building block for more complex approaches, such as those including optimization problems that embed congestion effects.

Acknowledgements

Maria Luíza G. Toledo is funded by the CNPq of the Ministry for Science and Technology of Brazil, grant 103106/2003-0. The research of Frederico R. B. Cruz has been partially funded by the CNPq, grants 301809/96-8 and 201046/94-6, the FAPEMIG, grants CEX-289/98 and CEX-855/98, and PRPq-UFMG, grant 4081-UFMG/RTR/FUNDO/PRPq/99.

References

Alfa, A. S. and Neuts, M. F. (1995). Modeling vehicular traffic using the discrete time Markovian arrival process, *Transportation Science* **29**(2): 109–117.

Cheah, J. and MacGregor Smith, J. (1994). Generalized $M/G/C/C$ state dependent queueing models and pedestrian traffic flows, *Queueing Systems* **15**: 365–386.

Cruz, F. R. B., MacGregor Smith, J. and Medeiros, R. O. (2003). An $M/G/C/C$ state dependent network simulation model, *Computers & Operations Research* (accepted).

Dijkstra, E. W. (1959). A note on two problems in connection with graphs, *Numerical Mathematics* **1**: 269–271.

Fruin, J. J. (1971). *Pedestrian Planning and Design*, Metropolitan Association of Urban Designers and Environmental Planners, Inc., New York, NY, USA.

Jain, R. and MacGregor Smith, J. (1997). Modeling vehicular traffic flow using $M/G/C/C$ state dependent queueing models, *Transportation Science* **31**(4): 324–336.

Kerbache, L. and MacGregor Smith, J. (1987). The generalized expansion method for open finite queueing networks, *European Journal of Operational Research* **32**: 448–461.

Kleinrock, L. (1975). *Queueing Systems*, Vol. I: Theory, John Wiley & Sons, New York, NY, USA.

Labetoulle, J. and Pujolle, G. (1980). Isolation method in a network of queues, *IEEE Transactions on Software Engineering* **SE-6**(4): 373–381.

MacGregor Smith, J. and Chikhale, N. (1995). Buffer allocation for a class of nonlinear stochastic knapsack problem, *Annals of Operations Research* **58**: 323–360.

MacGregor Smith, J. and Daskalaki, S. (1988). Buffer space allocation in automated assembly lines, *Operations Research* **36**(2): 343–358.

Tregenza, P. R. (1976). *The Design of Interior Circulation*, Van Nostrand Reinhold Company, New York, NY, USA.

Yuhaski, S. J. and MacGregor Smith, J. (1989). Modeling circulation systems in buildings using state dependent models, *Queueing Systems* **4**: 319–338.