IX Encontro de Modelagem Computacional
Belo Horizonte,MG
15 a 17 de novembro de 2006
CEFET-MG - IPRJ/UERJ
ISBN: 978-85-99836-02-6

# OPTIMAL DESIGN OF NETWORKS OF GENERAL FINITE MULTI-SERVER QUEUES

**Frederico R. B. Cruz**- e-mail: fcruz@est.ufmg.br
Federal University of Minas Gerais, Department of Statistics
Av. Antônio Carlos, 6627 - 31270-901 - Belo Horizonte, MG, Brazil
**Milene S. Castro**- e-mail: mscastro@ufmg.br
Federal University of Minas Gerais, Department of Production Engineering
Av. Antônio Carlos, 6627 - 31270-901 - Belo Horizonte, MG, Brazil
**Helinton A. L. Barbosa**- e-mail: helinton@ufmg.br
Federal University of Minas Gerais, Department of Statistics
Av. Antônio Carlos, 6627 - 31270-901 - Belo Horizonte, MG, Brazil

***Abstract.*** *In this paper we address the topological network design of general service, finite waiting room, multi-server queueing networks. Several topologies are examined using an approximation method to estimate the performance of the queueing networks and an iterative search method to find the optimal buffer allocation within the network. Extensive computational results show that the buffer allocations are sound. The results were quite satisfactory and in most of the cases tested the approximate analytical results were within the 95% confidence intervals estimated by simulation. Additionally, quite different topologies may result in a similar performance, which may bring flexibility to the planner. Finally, it was confirmed that the coefficient of variation of the service times is significant in the buffer allocation.*

***Keywords:*** *Multi-server, finite networks, blocking probabilities, buffer allocation*

## 1. INTRODUCTION

The allocation of resources to process a flow of goods results in a finite queueing network wherever there is uncertainty about the flows and about the processing times of these goods at the nodes of the network. The allocation of resources we are concerned about here includes the buffers, the order of the servers, and their interaction. A relevant question is how we can effectively model, accurately predict their performance measures, and design these stochastic systems.

In this paper, the aim is to optimize the topology of finite queueing systems. Methods are sought to allow us to both model and construct algorithms to optimize these systems. This paper revisits previous works about single-server (Smith & Cruz, 2005) and multi-server (Smith et al., 2006) finite buffer systems. As such, with multi-server systems, we need to see how multi-servers affect the optimal buffer allocation and additionally how various topologies and systematic variations in the general service time coefficient of variation play out.

We are given a finite network $G(N, A)$ of a specified topology, with a set of nodes $N$, with general distributed service times, and a corresponding set of arc pairs $A$, with known routing probabilities. We seek to determine one of the most important performance measures of this network, the throughput. Because the network has finite capacity, there is blocking in the network that consequently gives rise to non-product form characteristics, which makes it very

difficult to derive the probability distribution of the number of customers within the network. Thus, one is forced to seek effective ways to decompose the problem to assess the performance measures of the system.

The paper is outlined as follows. In Section 2 of the paper, we describe the problem background and related works. In Section 3, we describe the mathematical programming formulation and, in Section 4, the algorithms we employ for its analysis. In Section 5, we describe our experimental results and, in Section 6, we conclude with open questions for future research.

## 2. PROBLEM BACKGROUND

The optimal design of finite queueing networks is a quite difficult problem for which there are limited published approaches in the literature. Exact approaches have been limited to the assumptions of exponential distributions, but these continuous time Markov Chain (CTMC) approaches may be limited to moderate sized networks since the state space explodes, although recent advances in solving huge Markov Chains my be found in the literature (Carrasco, 2006). Non-exponential service times within networks may be hard to analyze exactly, since the memoryless property of the exponential distribution no longer applies. Therefore, approximations are both reasonable and practical.

In the past two-moment approximations have been very successful (Smith, 2003; Smith & Cruz, 2005; Smith et al., 2006) and we shall also follow this approach here. Methodologies for approximating the blocking probability in $M/G/1/K$ and $M/G/c/K$ systems have a long and detailed history, which following Kendall's notation stand for systems with Markovian (exponential) inter-arrival time distribution, General service time distribution, 1 (or $c$) servers in parallel, and a total capacity $K$ including the servers. Exact methods are not feasible for large $c$ and $K$ since the memoryless property of the exponential distribution no longer applies. Approximations essentially begin with Gelenbe's approach which is based on a diffusion approximation (Gelenbe, 1975). Also, formulas based on the steady-state probabilities of infinite systems by Schweitzer & Konheim (1978), Tijms (1987), and Sakasegawa et al. (1993) have been developed. Finally, two-moment approximations emerged from Tijms (1992, 1994), Kimura (1996b,a), and Smith (2003)

Because the buffer allocation problem is a solution to an integer stochastic problem with a nonlinear objective function and constraints (not found in closed form), heuristic approaches have dominated optimal ones. The buffer allocation problem has been treated by many authors. Approaches include those based on dynamic programming (Yamashita & Onvural, 1994), search methods (Smith & Cruz, 2005), metaheuristics (Spinellis et al., 2000), and simulation-based methods (Harris & Powell, 1999).

## 3. MATHEMATICAL MODELS

### 3.1 Notation

This section presents the notation needed for the paper:

$\lambda_j$ := Poisson arrival rate to node $j$;

$\mu_j$ := mean service rate at node $j$;

$c$ := number of servers;

$\rho = \lambda/(\mu c)$ := the traffic intensity;

$B_j$ := buffer capacity at node $j$ *excluding* those in service;

$K_j$ := Buffer capacity at node $j$ *including* those in service;

$p_K$ := blocking probability of finite queue of size $K$;

$s^2 = \mathrm{V}ar(T_s)/\mathrm{E}(T_s)^2$ := squared coefficient of variation of the service time, $T_s$;

$\Theta$ := throughput rate.

### 3.2 Mathematical programming formulation

In this paper, we will consider the following type of optimization problem, which also was the central objective used by Smith & Cruz (2005) and Smith et al. (2006)

$$Z = \min\left(f(\mathbf{x}) = \sum_{\forall i} x_i\right), \tag{1}$$

subject to:

$$\begin{aligned} \Theta(\mathbf{x}) &\geq \Theta^\tau, & (2)\\ x_i &\in \{1, 2, \ldots\},\ \forall i, & (3) \end{aligned}$$

that minimizes the total buffer allocation $\sum_{\forall i} x_i$, constrained to provide the minimum throughput $\Theta^\tau$. In the above formulation $\Theta_j^\tau$ is a threshold throughput value and $x_i \equiv K_i$ is the decision variable, which is the total buffer capacity at the $i$-th queue.

In this paper only Markovian arrival processes will be considered because exact results can be derived for these systems. Besides, results for general arrivals are scarce and limited to single servers (see, for instance, the paper by Kim & Chae, 2003).

### 3.3 Blocking probabilities in single queues

The blocking probability for an $M/M/1/K$ system with $\rho < 1$ is well-known from any textbook (Gross & Harris, 1985)

$$p_K = \frac{(1-\rho)\rho^K}{1-\rho^{K+1}}.$$

If the integrality of $K$ is relaxed, one can express $K$ in terms of $\rho$ and $p_K$ and arrive at a closed-form expression for the buffer size which is the smallest integer $K$ not inferior to

$$\frac{\ln\left(\frac{p_K}{1-\rho+p_K\rho}\right)}{\ln(\rho)}.$$

In two previous papers (Smith, 2003; Smith & Cruz, 2005), it was showed that once one has the closed form expression for the pure buffer $B^* = K^* - 1$ in an $M/M/1/K$ system, one can use a two-moment approximation scheme based on Kimura's and Tijms' work (Kimura, 1996b,a; Tijms, 1992, 1994) to develop the buffer size $B^*$ for general service. For $c = 1$ and $s^2$, we have an approximation to the optimal buffer size $B^*$ for $M/G/1/K$ systems

$$B^* = \frac{\left[\ln\left(\frac{p_K}{1-\rho+p_K\rho}\right) + \ln(\rho)\right]\left(2 + \sqrt{\rho}s^2 - \sqrt{\rho}\right)}{2\ln(\rho)}.$$

If $s^2 = 1$ and $c = 1$, then the formula yields the same expression as for the $M/M/1/K$ formula, when we subtract the space for the server. As one might expect, we can continue this process of developing $p_K$ since one can obtain $B^*$ and $p_K$ for different values of $c$ and thus develop closed form expressions of the buffer size and blocking probabilities for $M/G/c/K$ systems (Smith et al., 2006).

### 3.4 Blocking probabilities in networks of queues

The Generalized Expansion Method (GEM) is a robust and effective approximation technique developed by Kerbache & Smith (1987) to derive performance measures of finite queueing networks. As described in previous papers, this method is characterized as a combination of repeated trials and node-by-node decomposition solution procedures. Methodologies for computing performance measures for a finite queueing network use primarily the following two kinds of blocking:

**Type I:** The upstream node $i$ gets blocked if the service on a customer is completed but it cannot move downstream due to the queue at the downstream node j being full. This is sometimes referred to as Blocking After Service (BAS) (Onvural, 1990).

**Type II:** The upstream node is blocked when the downstream node becomes saturated and service must be suspended on the upstream customer regardless of whether service is completed or not. This is sometimes referred to as Blocking Before Service (BBS) (Onvural, 1990).

The GEM uses Type I blocking, which is common in production and manufacturing, transportation and other similar systems. Consider a single node with finite capacity $K$ (including service). This node essentially oscillates between two states — the saturated phase and the unsaturated phase. In the unsaturated phase, node $j$ has at most $K - 1$ customers (in service or in the queue). On the other hand, when the node is saturated no more customers can join the queue. Refer to Fig. 1 for a graphical representation of the two scenarios.

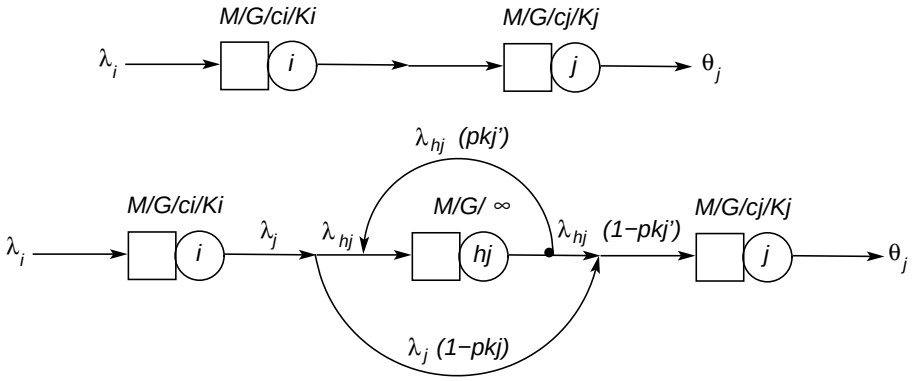


**Figure 1:** Generalized expansion method.

The GEM has the following three stages:

**Stage I:** Network Reconfiguration;

**Stage II:** Parameter Estimation;

**Stage III:** Feedback Elimination.

Details on the GEM will not be given here and can be found in the paper by Kerbache & Smith (1987). The GEM 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

$$\tilde{\mu}_i^{-1} = \mu_i^{-1} + p_K \mu_h^{-1}.$$

To recapitulate, we first expand the network; followed by approximation of the routing probabilities, due to blocking, and the service delay in the holding node $h_j$ and finally the feedback arc at the holding node is eliminated. Once these three stages are complete, we have an expanded network which can then be used to compute the performance measures for the original network. As a decomposition technique this approach allows successive addition of a holding node for every finite node, estimation of the parameters and subsequent elimination of the holding node. An important point about this process is that we do not physically modify the networks, only represent the expansion process through the software.

## 4. ALGORITHMS

The primal optimization problem with $M/M/c/K$ and $M/G/c/K$ systems that will be examined here is given by Eq. (1)–(3). One way to incorporate the throughput constraint, Eq. (2), is through a penalty function approach, such as the Lagrangean relaxation (for a recently published tutorial, see the paper by Lemaréchal, 2003).

Thus, defining a dual variable $\alpha$ and relaxing constraint (2), the following penalized problem is given

$$Z_\alpha = \min \left[ \sum_{\forall i} x_i + \underbrace{\alpha\Big(\Theta^\tau - \Theta(\mathbf{x})\Big)}_{\leq 0} \right], \tag{4}$$

subject to:

$$x_i \in \{1, 2, \ldots\}, \ \forall i, \tag{5}$$

$$\alpha \geq 0. \tag{6}$$

Notice that for any vector $\mathbf{x}$ feasible — that is, Eq. (2) and (3) must hold — the term $\alpha\Big(\Theta^\tau - \Theta(\mathbf{x})\Big)$ must be non-positive and is a penalty of the objective function related to the difference between the threshold throughput, $\Theta^\tau$, and the effective throughput, $\Theta(\mathbf{x})$. Thus, it follows that $Z_\alpha \leq Z$, that is, $Z_\alpha$ is an inferior limit for $Z$, the optimal solution for the primal problem, given by Eq. (1)–Eq. (3).

The Lagrangean relaxation of the primal problem, $Z_\alpha$, plus an additional relaxation of the integrality constraints for $x_i$, is a classical unconstrained optimization problem. In the particular formulation of the problem, the $x_i$ variables become the decision variables under optimization control. While these are essentially integer variables, they can be reasonably approximated by round off from the nonlinear programming solver.

While the GEM will be used to compute the throughput, Powell's algorithm will be used to search for the optimal buffer vector. Powell's method (for details, see the book by Himmelblau, 1972), locates a minimum of a non-linear function $f(\mathbf{x})$ by successive one-dimensional

searches from an initial starting point $\mathbf{x}^{(0)}$ along a set of conjugate directions. These conjugate directions are generated within the procedure itself. Powell's method is based on the idea that if a minimum of a non-linear function $f(\mathbf{x})$ is found along $p$ conjugate directions in a stage of the search, and an appropriate step is made in each direction, the overall step from the beginning to the $p$-th step is conjugate to all of the $p$ sub-directions of the search. We have seen reports (Smith & Cruz, 2005; Smith et al., 2006) of a remarkable success with coupling Powell's algorithm and the GEM.

## 5. EXPERIMENTAL RESULTS

In this section of the paper, we will provide experimental results of the network design methodology above described. We will present results for two-node and three-node queueing networks, which extend and corroborate in some aspects the experimental results presented by Smith et al. (2006).

### 5.1 Two-node/three-server Networks

The simplest network is a two-node/three-server topology involving single and two-servers arranged in a simple series connection, as seen in Fig. 2. We would like to test what are the buffers needed for this type of topology and whether one topology (i.e. server order) is better than another.

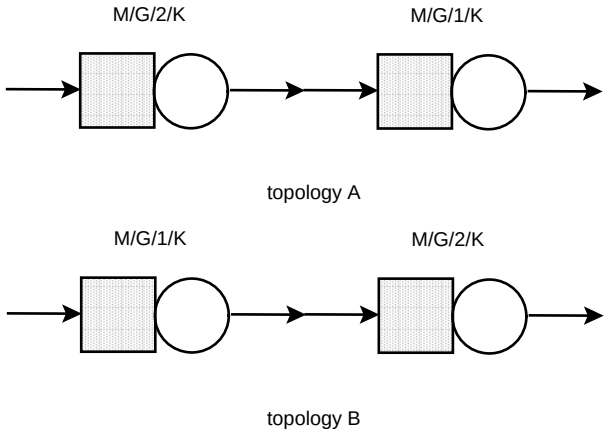


**Figure 2:** Two-node/three-server network topology.

In the first experiment, presented in Table 1, we fix the arrival rate to the network with $\lambda = 1$ and service rates of the different servers to $\mu = 4$. We would like to examine what buffers are needed for these two alternative network topologies. We will also vary the coefficient of variation of the service time $s^2$ to see how the buffer is affected by the service time variability.

In order to evaluate the analytical results, simulation runs of 20 replications, with a warm up period of 2,000 time units, and 200,000 time units for each run were carried out in Arena (Kelton et al., 2001). These run length and number of replications reduced the standard deviation of the statistics of the simulation output to a reasonably accurate level. The general service times for the $s^2 = \{0.5, 2.0\}$ were simulated by a Gamma distribution (Kelton et al., 2001). The experiments took place on a Pentium 4 3.0 GHz 2 MB CPU, 1.0 GB RAM, under Windows XP operating system.

The results seen in Table 1 are impressive. The $\delta$ in the 9th column of the result tables refers to the half-width of the 95% confidence intervals (CI). In most of the cases, the analytical throughput value was within the 95% CI. The buffer allocations are symmetric for all cases,

**Table 1:** Two-node/three-server results.

| | | | | | | | Simulation | | |
|---|---|---|---|---|---|---|---|---|---|
| $\lambda$ | $\mu$ | $s^2$ | **c** | **x** | $\theta(\mathbf{x})$ | $Z_\alpha$ | $\theta(\mathbf{x})^s$ | $\delta$ | $Z_\alpha^s$ |
| 1.0 | (4,4) | 0.5 | (2,1) | (3,4) | 0.999 | 8.000 | 0.997* | 0.001 | 9.71 |
| | | | (1,2) | (4,3) | 0.999 | 8.000 | 0.998 | 0.001 | 8.78 |
| | | 1.0 | (2,1) | (3,4) | 0.998 | 9.000 | 0.997 | 0.001 | 9.67 |
| | | | (1,2) | (4,3) | 0.998 | 9.000 | 0.997 | 0.001 | 10.26 |
| | | 2.0 | (2,1) | (4,5) | 0.999 | 10.000 | 0.999 | 0.001 | 10.38 |
| | | | (1,2) | (5,4) | 0.999 | 10.000 | 0.997* | 0.001 | 12.01 |

* The 95% CI does not cover the analytical result.

and there is not any difference in the optimal solution values for either topology. Thus, it is difficult to say whether one topology is better than another, simply because the optimization methodology made sure that the resulting buffer allocations were appropriate for each of the topologies. If one did not optimize the buffer allocations, then perhaps one topology might dominate the other. However, it is difficult to derive heuristic rules (e.g. always place the multi-servers first in the topology) prior to an optimization procedure to say which topology is better.

In another experiment with two-node networks, let us assume that the service time of the two-server node is smaller than the service time of the single server node (see Fig. 2). This represents a bottleneck situation. Let us assume that the service time of the two-server queue has $\mu = 4$ while the service time at the single-server queue has $\mu = 8$. We get the experimental results presented in Table 2.

**Table 2:** Two-node/three-server bottleneck results.

| | | | | | | | Simulation | | |
|---|---|---|---|---|---|---|---|---|---|
| $\lambda$ | $\mu$ | $s^2$ | **c** | **x** | $\theta(\mathbf{x})$ | $Z_\alpha$ | $\theta(\mathbf{x})^s$ | $\delta$ | $Z_\alpha^s$ |
| 1.0 | (4,8) | 0.5 | (2,1) | (3,3) | 0.999 | 7.000 | 0.997* | 0.001 | 8.670 |
| | (8,4) | | (1,2) | (3,3) | 0.999 | 7.000 | 0.999 | 0.001 | 6.720 |
| | (4,8) | 1.0 | (2,1) | (3,3) | 0.999 | 7.000 | 0.997* | 0.001 | 8.870 |
| | (8,4) | | (1,2) | (3,3) | 0.999 | 7.000 | 0.998 | 0.001 | 8.130 |
| | (4,8) | 2.0 | (2,1) | (4,3) | 0.999 | 8.000 | 0.999 | 0.001 | 8.320 |
| | (8,4) | | (1,2) | (3,4) | 0.999 | 8.000 | 0.996* | 0.001 | 10.930 |

* The 95% CI does not cover the analytical result.

As in previous experimental results, Table 2 indicates that more buffer space may be allocated to the two-server node rather than less since they represent the bottlenecks. Symmetric buffer allocations occur and no difference occurs in the objective function values of the topologies. Thus, it is difficult to say which topology is better. Additionally, the throughput is within the 95% CI in almost all cases.

### 5.2 Three-node/five-server Networks

Extending the experiments to more complex series networks, we examine a three-node/five-server queueing network. Figure 3 represents the possible topologies with one single server and two two-server queues in a series topology.

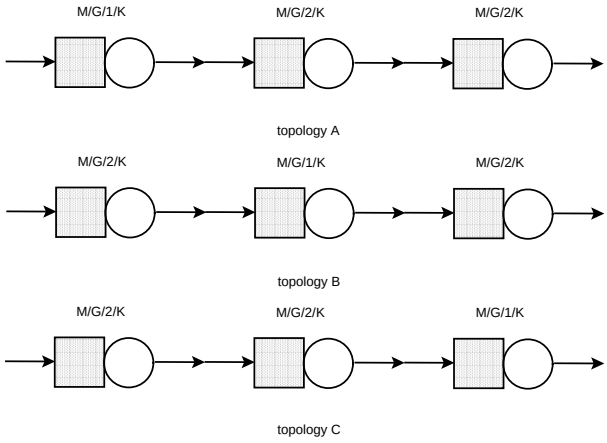


**Figure 3:** Three-node/five-server network topology.

The results may be seen in Table 3. It is interesting that no matter $s^2$, the buffer at the single server, which is the botleneck, is increased in relation to the two-server nodes. Additionally, in the high $s^2 = \{2.0\}$, the buffers are increased in comparison with low $s^2$. Thus, the effect of variability is important in the buffer allocation. Concerning which topology is best, once again, it is difficult to say since all give the same $\theta(\mathbf{x})$.

**Table 3:** Three-node/five-server results.

| | | | | | | | Simulation | | |
|---|---|---|---|---|---|---|---|---|---|
| $\lambda$ | $\mu$ | $s^2$ | **c** | **x** | $\theta(\mathbf{x})$ | $Z_\alpha$ | $\theta(\mathbf{x})^s$ | $\delta$ | $Z_\alpha^s$ |
| 1.0 | (4,4,4) | 0.5 | (1,2,2) | (4,3,3) | 0.998 | 12.000 | 0.999 | 0.001 | 10.980 |
| | | | (2,1,2) | (3,4,3) | 0.998 | 12.000 | 0.997 | 0.001 | 12.840 |
| | | | (2,2,1) | (3,3,4) | 0.998 | 12.000 | 0.997 | 0.001 | 12.520 |
| | | 1.0 | (1,2,2) | (4,3,3) | 0.997 | 13.000 | 0.997 | 0.001 | 12.680 |
| | | | (2,1,2) | (3,4,3) | 0.997 | 13.000 | 0.997 | 0.001 | 12.980 |
| | | | (2,2,1) | (3,3,4) | 0.997 | 13.000 | 0.996 | 0.001 | 13.520 |
| | | 2.0 | (1,2,2) | (5,4,4) | 0.998 | 15.000 | 0.998 | 0.001 | 15.390 |
| | | | (2,1,2) | (4,5,4) | 0.998 | 15.000 | 0.999 | 0.001 | 13.840 |
| | | | (2,2,1) | (4,4,5) | 0.998 | 15.000 | 1.000* | 0.001 | 13.430 |

* The 95% CI does not cover the analytical result.

In order to determine the effect of the $s^2$ on the buffer allocation, let us isolate one configuration $\mathbf{c} = (1, 2, 2)$ and vary $s^2$ to see how the buffer allocation changes. Table 4 presents the results. When $s^2 = 0$, the buffer allocation is not different at the single server node in relation to the two-server nodes, and then changes above $s^2 = 0.3$, when the buffer at the single node becomes larger than at the two-server nodes. This is very interesting and somewhat unpredictable showing that the buffer allocation may be susceptible to slight changes in the service time variability, $s^2$.

## 6. SUMMARY AND CONCLUSIONS

We have shown a recently developed approach to the buffer allocation problem of finite open queueing networks with general service and multiple-servers. We have described both the derivation of the blocking probability formulas used in the experiments and the optimization

**Table 4:** More of three-node/five-server results.

| | | | | | | | Simulation | | |
|---|---|---|---|---|---|---|---|---|---|
| $\lambda$ | $\mu$ | $s^2$ | **c** | **x** | $\theta(\mathbf{x})$ | $Z_\alpha$ | $\theta(\mathbf{x})^s$ | $\delta$ | $Z_\alpha^s$ |
| 1.0 | (4,4,4) | 0.0 | (1,2,2) | (3,3,3) | 0.998 | 11.00 | 0.997 | 0.001 | 12.20 |
| | | 0.1 | (1,2,2) | (3,3,3) | 0.998 | 11.00 | 0.996* | 0.001 | 13.10 |
| | | 0.2 | (1,2,2) | (3,3,3) | 0.998 | 11.00 | 0.996* | 0.001 | 13.14 |
| | | 0.3 | (1,2,2) | (3,3,3) | 0.997 | 12.00 | 0.995* | 0.001 | 14.50 |
| | | 0.4 | (1,2,2) | (4,3,3) | 0.998 | 12.00 | 0.999 | 0.001 | 10.89 |
| | | 0.5 | (1,2,2) | (4,3,3) | 0.998 | 12.00 | 0.999 | 0.001 | 10.98 |
| | | 0.6 | (1,2,2) | (4,3,3) | 0.998 | 12.00 | 0.999 | 0.001 | 11.26 |
| | | 0.7 | (1,2,2) | (4,3,3) | 0.998 | 12.00 | 0.998 | 0.001 | 12.00 |
| | | 0.8 | (1,2,2) | (4,3,3) | 0.997 | 13.00 | 0.998 | 0.001 | 12.06 |
| | | 0.9 | (1,2,2) | (4,3,3) | 0.997 | 13.00 | 0.998 | 0.001 | 12.08 |

* The 95% CI does not cover the analytical result.

methodology. Numerous experiments illustrating the scope and limitations of the approach have been shown.

In general, the buffer allocations derived by the algorithms, symmetric for the cases tested, made sense. The results were quite satisfactory as in most of the cases tested the approximate analytical results were within the 95% confidence intervals estimated by simulation. Another interesting result is that quite different topologies (e.g., topologies A and B in the two-node/three-server networks) may result in a similar performance, of course if the buffer allocation is optimal. Thus it is difficult to derive heuristic rules, such as 'always place the multi-servers first in the topology', prior to an optimization procedure to say which topology is best. Finally, it was shown that the coefficient of variation of the service times is significant in the buffer allocation for both uniform and bottlenecked server networks. We hope that the reader had sensed the power of this approach and the ability we now have to tackle these complex network planning and design problems.

### 6.1 Open Questions

This research could evolve in many directions. It includes various applications of the algorithm to practical networks, such as in manufacturing and assembly problems, facility planning and layout design, telecommunication, and computer system network design problems. Also we have not examined in any detail the situation in which the number of servers $c$ is treated as a decision variable.

### Acknowledgements

The research of Frederico Cruz has been partially funded by the CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) of the Ministry for Science and Technology of Brazil, grants 201046/1994-6, 301809/1996-8, 307702/2004-9, and 472066/2004-8, the FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais), grants CEX-289/98 and CEX-855/98, and PRPq-UFMG, grant 4081-UFMG/RTR/FUNDO/PRPq/99.

Milene S. Castro and H. A. L. Barbosa have been funded by the FAPEMIG.

## REFERENCES


Carrasco, J. A., 2006. Two methods for computing bounds for the distribution of cumulative reward for large Markov models. *Performance Evaluation* (in press).

Gelenbe, E., 1975. On approximate computer system models. *Journal of the ACM*, vol. 22, n. 2, pp. 261–269.

Gross, D. & Harris, C., 1985. *Fundamentals of Queueing Theory*. John Wiley & Sons, New York.

Harris, J. H. & Powell, S. G., 1999. An algorithm for optimal buffer placement in reliable serial lines. *IIE Transactions*, vol. 31, pp. 287–302.

Himmelblau, D. M., 1972. *Applied Nonlinear Programming*. McGraw-Hill Book Company, New York.

Kelton, D., Sadowski, R. P., & Sadowski, D. A., 2001. *Simulation with Arena*. MacGraw Hill College Div., New York, NY, USA.

Kerbache, L. & Smith, J. MacGregor, 1987. The generalized expansion method for open finite queueing networks. *European Journal of Operational Research*, vol. 32, pp. 448–461.

Kim, N. K. & Chae, K. C., 2003. Transform-free analysis of the $GI/G/1/K$ queue through the decomposed Little's formula. *Computers & Operations Research*, vol. 30, n. 3, pp. 353–365.

Kimura, T., 1996a. Optimal buffer design of an $M/G/s$ queue with finite capacity. *Communications in Statistics - Stochastic Models*, vol. 12, n. 1, pp. 165–180.

Kimura, T., 1996b. A transform-free approximation for the finite capacity $M/G/s$ queue. *Operations Research*, vol. 44, n. 6, pp. 984–988.

Lemaréchal, C., 2003. The omnipresence of Lagrange. *4OR*, vol. 1, pp. 7–25.

Onvural, R. O., 1990. Survey of closed queueing networks with blocking. *ACM Computing Surveys*, vol. 22, n. 2, pp. 83–121.

Sakasegawa, H., Miyazawa, M., & Yamazaki, G., 1993. Evaluating the overflow probability using the infinite queue. *Management Science*, vol. 39, n. 10, pp. 1238–1245.

Schweitzer, P. J. & Konheim, A. G., 1978. Buffer overflow calculations using an infinite-capacity model. *Stochastic Processes and their Applications*, vol. 6, n. 3, pp. 267–276.

Smith, J. MacGregor, 2003. $M/G/c/k$ blocking probability models and system performance. *Performance Evaluation*, vol. 52, n. 4, pp. 237–267.

Smith, J. MacGregor & Cruz, F. R. B., 2005. The buffer allocation problem for general finite buffer queueing networks. *IIE Transactions on Design & Manufacturing*, vol. 37, n. 4, pp. 343–365.

Smith, J. MacGregor, Cruz, F. R. B., & van Woensel, T., 2006. Topological network design of general, finite, multi-server queueing networks. *Performance Evaluation* (under review).

Spinellis, D., Papadopoulos, C. T., & Smith, J. MacGregor, 2000. Large production line optimization using simulated annealing. *International Journal of Production Research*, vol. 38, n. 3, pp. 509–541.

Tijms, H. C., 1987. *Stochastic Modelling and Analysis: A Computational Approach*. John Wiley & Sons, New York.

Tijms, H. C., 1992. Heuristics for finite-buffer queues. *Probability in the Engineering and Informational Sciences*, vol. 6, pp. 267–276.

Tijms, H. C., 1994. *Stochastic Models: An Algorithmic Approach*. John Wiley & Sons, New York.

Yamashita, H. & Onvural, R., 1994. Allocation of buffer capacities in queueing networks with arbitrary topologies. *Annals of Operations Research*, vol. 48, pp. 313–332.