

## OTIMIZAÇÃO DE REDES DE CIRCULAÇÃO DE PEDESTRES SUJEITAS A CONGESTIONAMENTO

P. C. Oliveira<sup>1</sup>, F. G. Amorim<sup>2</sup>, M. L. G. Toledo<sup>2</sup>, F. R. B. Cruz<sup>2</sup>

<sup>1</sup>Departamento de Ciência da Computação e <sup>2</sup>Departamento de Estatística

Instituto de Ciências Exatas – UFMG

31270-901 – Belo Horizonte – MG

E-mail: {pcampos,famorim,mlgt,fcruz}@ufmg.br

### Resumo

Apresentamos resultados de experimentos computacionais com um algoritmo para otimização em redes de filas do tipo  $M/G/c/c$  dependentes do estado, que são úteis para modelar o fenômeno de congestionamento. Diversas demandas, tamanhos de redes e configurações foram consideradas, tendo sido obtidos padrões surpreendentes de alocação. Uma configuração real foi examinada, evidenciando a aplicabilidade do algoritmo a casos práticos. Em geral, o algoritmo produziu soluções de boa qualidade, conforme demonstrado pelos nossos estudos de simulação. Embora a metodologia tenha sido aqui aplicada a redes de pedestres, ela é geral o bastante para ser utilizada em muito outros sistemas nos quais o fenômeno de congestionamento é significativo.

**Palavras-chave:** Teoria de filas; sistemas estocásticos; redes de filas dependentes do estado.

### Abstract

In this paper, we present computational experiments with an algorithm for optimization of  $M/G/c/c$  state-dependent queueing networks, which are very convenient in modeling congestions. Several demands, network sizes, and topologies were considered and unexpected allocation patterns were obtained. A network from a real world configuration was examined to illustrate the usefulness of the optimization algorithm. In general, the algorithm performed well, as demonstrated by our simulation studies. Although the methodology is applied only to pedestrian networks, it is general enough to be used in many other systems in which congestion is significant.

**Keywords:** Queue theory, stochastic systems, state-dependent queueing networks.

### 1. Introdução

Modelando por exemplo sistemas de computadores (Chaudhry et al., 2001), sistemas de distribuição (Kimura, 1996), sistemas de telecomunicações (Rubin & Zhang, 1991) e sistemas de produção (Harris & Powell, 1999; Enginarlar et al., 2002), redes de filas finitas têm se mostrado útil na prática. Uma rede de filas finitas, Figura 1, pode ser vista com um conjunto de filas organizadas de acordo com uma determinada topologia, com espaço para filas (*buffers*) de tamanho limitado em alguns ou todos os servidores. A consideração de *buffers* de tamanho finito leva ao fenômeno do bloqueio, que significa simplesmente que o usuário não pode passar para a próxima fila se o *buffer* correspondente estiver lotado (bloqueio após serviço). O usuário então permanece no servidor precedente à fila saturada e o serviço é suspenso até que ele possa seguir em frente.

Redes com bloqueio são difíceis de ser tratada e poucos são os resultados apresentados na literatura lidando com o problema (Gross & Harris, 1998). Resultados exatos são disponíveis apenas para redes simples, em série e com *buffers* de pequeno tamanho (Li et al., 2004). Além disso, a literatura considera normalmente chegadas exponenciais e serviços também exponenciais, havendo muito pouca teoria desenvolvida para redes mais gerais (Huang et al., 2002). Medidas de desempenho comumente consideradas são: a probabilidade de bloqueio, o número de usuários atendidos na unidade de tempo (o *throughput*), o tamanho médio da fila e o tempo de permanência do usuário no sistema (o *WIP*, *work-in-process*). Normalmente, essas medidas são estimadas por técnicas aproximadas (Kimura, 1996) e a

única forma de validação parece ser mesmo a simulação, uma vez que a análise exata é impossível para redes de grande porte.

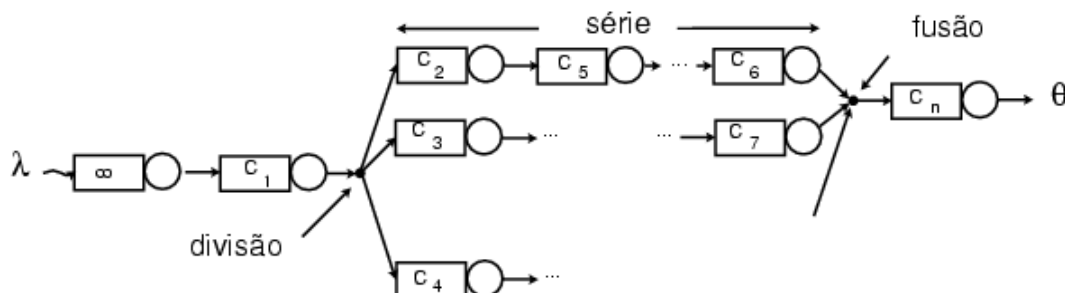


Figura 1: Rede de filas em topologia genérica.

O objeto de estudo deste artigo é um conjunto particular de redes de filas, do tipo  $M/G/c/c$  dependente do estado, apropriadas para modelar o congestionamento presente em sistemas de tráfego de pedestres (Cheah & Macgregor Smith, 1994), de veículos (Jain & MacGregor Smith, 1997), bem como outros sistemas de serviço e manufatura (Yuhaski & MacGregor Smith, 1989; MacGregor Smith & Li, 2001). Adotando-se a notação de Kendall,  $M$  indica que a chegada é um processo markoviano,  $G$  representa um serviço com distribuição geral e dependente do estado,  $c$  refere-se ao número de servidores e, por fim, assume-se que a capacidade total do sistema está restrita a  $c$  usuários. A principal característica do modelo de redes de filas aqui tratado é o decrescimento da taxa de serviço com o aumento do número de usuários no sistema, conforme ilustrado na Figura 2, para tráfego de pedestres.

Em áreas de circulação de pedestres, vários fatores podem ser apontados como responsáveis por diferentes velocidades de deslocamento. Por exemplo, adultos tendem a andar mais rápido do que crianças, e analogias semelhantes podem ser encontradas em diferentes grupos. Quando a densidade do tráfego aumenta, estes fatores têm um efeito grande sobre a velocidade de caminhada de pedestres individuais. Portanto, dentro de um corredor, o movimento mais rápido de pedestres tende a tornar-se mais lento à medida que o espaço do corredor é ocupado por mais e mais pedestres, resultando em um fluxo com características dependentes do estado.

Os objetivos deste artigo são apresentar uma metodologia para otimização de redes finitas dependentes do estado, comparar detalhadamente alternativas de modelagem de tráfego com a finalidade de verificação da robustez da abordagem, estudar empiricamente (McGeoch, 2001) o desempenho do algoritmo de otimização e exemplificar a utilização da metodologia em uma configuração real. Desta forma, na Seção 2, apresentamos a formalização do problema de alocação de serviços e capacidades, através de um modelo de programação matemática inteira. Apresentamos também os modelos estocásticos associados, incluindo a modelagem de uma única fila  $M/G/c/c$  (corredor), os modelos de congestionamento (linear e exponencial) e, finalmente, o algoritmo de alocação de capacidades. A Seção 3 apresenta resultados obtidos com algumas topologias básicas. Na Seção 4, apresentamos a modelagem e a análise de uma caixa de escadas submetida a surtos de demanda. A Seção 5 encerra o artigo com algumas observações finais.

## 2. Modelo de otimização

### 2.1. Formulação matemática

Assumimos que a topologia de rede seja conhecida de antemão, definida em um grafo  $G(V,A)$ , no qual  $V$  é o conjunto de nós (corredores) e  $A$  é o conjunto de arcos (conexões entre pares de corredores). O problema de alocação de serviços e capacidades (SCA) preocupa-se em determinar a mínima capacidade  $c_i$  que deve ser alocada à  $i$ -ésima fila da rede,  $\forall i \in V$ , para que as probabilidades de bloqueio nestas filas estejam abaixo de um limiar  $\varepsilon$  previamente especificado, isto é,  $p(c_i) \equiv \Pr[N_i=c_i] \leq \varepsilon$ ,

$\forall i \in V$ , com  $N_i$  representando o número (aleatório) de usuários na  $i$ -ésima fila. Uma possível formulação matemática para o problema SCA é a seguinte:

$$z = \min \left[ g(\mathbf{c}) = \sum_{\forall i \in N} f_i c_i \right], \quad (1)$$

s.a.:

$$p(c_i) \leq \varepsilon, \quad \forall i \in V, \quad (2)$$

$$c_i \in \{0, 1, \dots\}, \quad \forall i \in V, \quad (3)$$

que minimiza, em uma topologia qualquer, o custo total de alocação  $\sum_i f_i c_i$ , restrito ao provimento de uma probabilidade de bloqueio máxima  $\varepsilon$ , em todos os nós do grafo.

## 2.2. Modelo analítico de um único corredor

No contexto aqui considerado, um nó da rede (vide Figura 1) representa um corredor (área de circulação). Para um único corredor, o tempo de serviço é igual ao tempo gasto para que um pedestre o atravesse. A taxa de serviço  $f(n)$  depende do número de usuários  $n$  e segue uma distribuição geral  $G$  dependente do estado. Considera-se, ainda, que a densidade de aglomeração de pedestres dentro do sistema está aproximadamente distribuída de maneira uniforme e que essa densidade determina a velocidade média de um indivíduo. Assim, se existirem  $n$  pedestres dentro de um corredor, esses terão uma taxa de serviço  $f(n)$ , enquanto que numa determinada chegada ou saída, essa taxa de serviço será alterada para  $f(n+1)$  ou  $f(n-1)$ , respectivamente.

Sob esse ponto de vista, assumindo-se o modelo de filas  $M/G/c/c$  dependentes do estado, pode-se expressar a distribuição de probabilidades do número de pedestres por (Cheah & MacGregor Smith, 1994):

$$p(n) \equiv P[N = n] = \left[ \frac{[\lambda \times E[T_1]]^n}{n! f(n) \dots f(2) f(1)} \right] \times p(0), n = 1, 2, \dots, c, \quad (4)$$

sendo

$$p(0) \equiv P[N = 0] = 1 / \left\{ 1 + \sum_{i=1}^c \left[ \frac{[\lambda \times E[T_1]]^i}{i! f(i) \dots f(2) f(1)} \right] \right\}, \quad (5)$$

em que  $p(0)$  é a probabilidade de não se ter nenhum usuário no sistema,  $\lambda$  é a taxa de chegada,  $E[T_1]$  é o tempo médio de serviço para um único usuários no sistema, e  $f(n)$  é a taxa de serviço para  $n$  usuários simultâneos no sistema.

Utilizando-se a Equação (4), é possível calcularem-se medidas de desempenho. Essas medidas incluem, entre outras, a probabilidade de bloqueio, a taxa de atendimento, o número médio de usuários e o tempo médio no sistema. O bloqueio ocorre quando o número de usuários  $N$  for igual à capacidade máxima  $c$  do sistema:

$$p_{\text{bloc}} \equiv p(c) \equiv \Pr[N = c]. \quad (6)$$

A taxa de ingresso efetiva no sistema (ou taxa de atendimento) pode então ser denotada por:

$$\theta = \lambda [1 - p(c)]. \quad (7)$$

O número médio (esperado) de usuários no sistema é:

$$L \equiv E[N] = \sum_{n=1}^c np(n). \quad (8)$$

O tempo médio no sistema (tempo de serviço esperado) pode ser dado pela lei de Little:

$$W \equiv E[T] = L / \theta. \quad (9)$$

### 2.3. Modelos de congestionamento para corredores

Precisamos agora definir as taxas de serviço  $f(n)$ , utilizadas nas Eq.(4) e (5). Estaremos utilizando a seguinte notação:

- $V_n$  → velocidade média para uma ocupação de  $n$  pedestres;
- $V_1$  → velocidade média para um único pedestre no sistema, que corresponde a 1,5 m/s;
- $V_a$  → velocidade média para uma densidade de aglomeração de 2 ped/m<sup>2</sup>;
- $V_b$  → velocidade média para uma densidade de aglomeração de 4 ped/m<sup>2</sup>;
- $\gamma$  e  $\beta$  → parâmetros de forma e escala do modelo exponencial;

Utilizando-se o modelo de filas  $M/G/c/c$  dependentes do estado, um corredor (área de circulação) pode ser considerado como servidor para seus usuários. O número de servidores em paralelo é igual à capacidade  $c$  do corredor, que também representa o número total de usuários permitidos simultaneamente no sistema, e é dado pela expressão:

$$c = \lfloor 5 \times l \times w \rfloor, \quad (9)$$

na qual  $\lfloor x \rfloor$  representa o maior inteiro não superior a  $x$ ,  $l$  é o comprimento do corredor e  $w$  é a sua largura. Note-se que, de acordo com estudos realizados por Tregenza (1976), 5 ped/m<sup>2</sup> representa a máxima densidade de aglomeração admissível em uma área de circulação. Finalmente, o tempo médio de serviço para um único usuário no sistema,  $E[T_1]$ , utilizado nas Eq. (4) e (5), pode ser definido como sendo dado por  $l/V_1$ .

No modelo de congestionamento, o fluxo uni-direcional de pedestres através dos corredores tem velocidade média  $V_n$  como uma função do número de indivíduos  $n$  que ocupam o corredor e a sua respectiva capacidade  $c$ . Baseado nessa e em outras observações empíricas, modelos analíticos linear e exponencial foram desenvolvidos para essa velocidade (uni-direcional) em função do número de pedestres em um corredor (Yuhaski & MacGregor Smith, 1989).

De acordo com Tregenza (1976), a velocidade de um único ocupante no corredor,  $V_1$ , é cerca de 1,5 m/s. Assumindo que uma população de  $n=c+1$  não é possível,  $V_n$  é zero para todo  $n \geq c+1$ . Uma relação linear que satisfaz a tais condições é como segue:

$$V_n = \frac{V_1}{c}(c+1-n). \quad (10)$$

Conforme o formato das curvas apresentadas por Tregenza (1976), Figura 2, um modelo exponencial também parece fornecer uma boa aproximação para a média da velocidade de um pedestre *versus* a densidade de aglomeração. De fato, um modelo exponencial de fluxo uni-direcional foi proposto com sucesso por Yuhaski e Smith (1989):

$$V_n = V_1 \exp \left[ - \left( \frac{n-1}{\beta} \right)^\gamma \right], \quad (11)$$

no qual:

$$\gamma = \ln \left[ \frac{\ln(V_a / V_1)}{\ln(V_b / V_1)} \right] / \ln \left( \frac{a-1}{b-1} \right), \quad \beta = \frac{a-1}{[\ln(V_1 / V_a)]^{\frac{1}{\gamma}}} = \frac{b-1}{[\ln(V_1 / V_b)]^{\frac{1}{\gamma}}}, \quad (12)$$

$V_1=1,5$  m/s,  $V_a=0,64$  m/s,  $V_b=0,25$  m/s,  $a=2 \times l \times w$  e  $b=4 \times l \times w$ . A Figura 3 mostra o comportamento da taxa de serviço para pedestres,  $f(n)=V_n/V_1$ , com o aumento da densidade, para fluxos unidirecionais, considerando-se os modelos linear e exponencial.

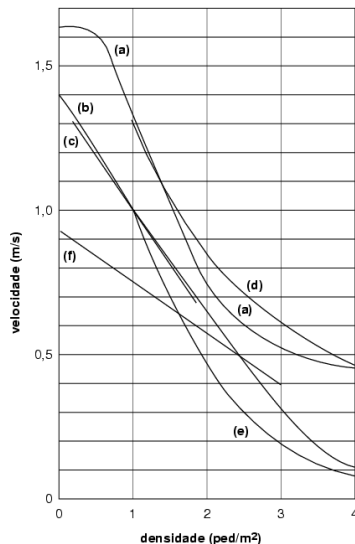


Figura 2: Curvas empíricas de Tregenza (1976).

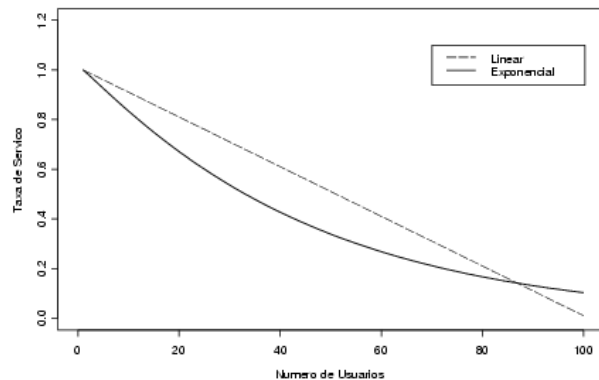


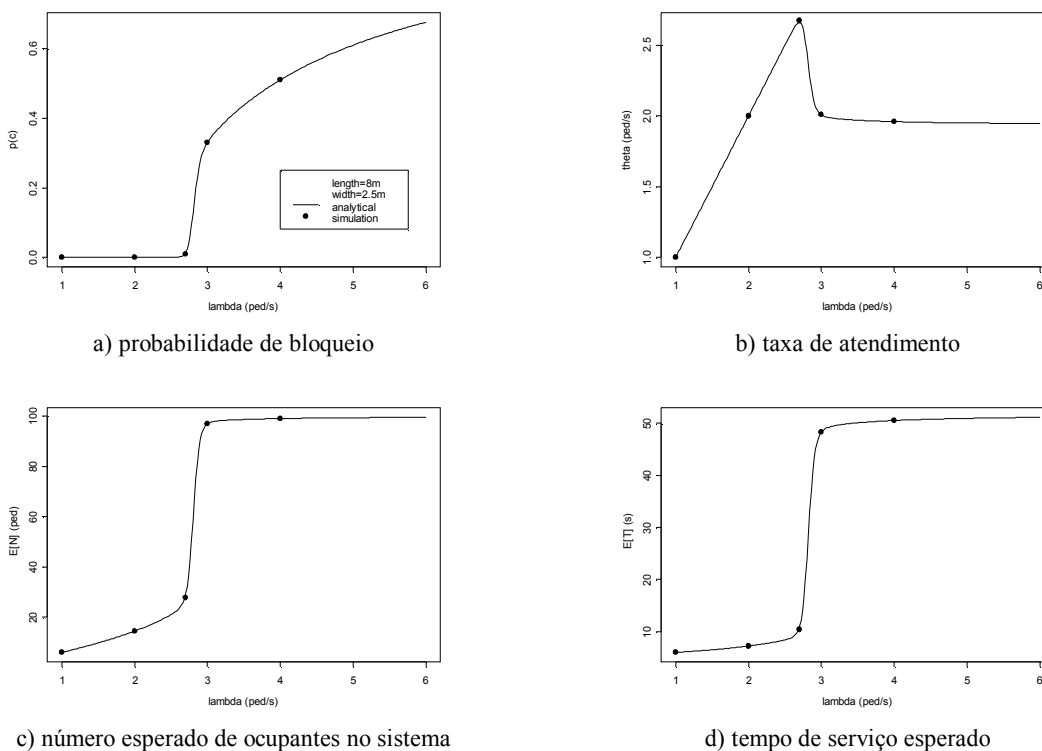
Figura 3: Taxa de serviço  $f(n)$  em função do número de usuários  $n$  para um corredor de  $8 \times 2,5$  m.

## 2.4. Comparação entre modelos de congestionamento

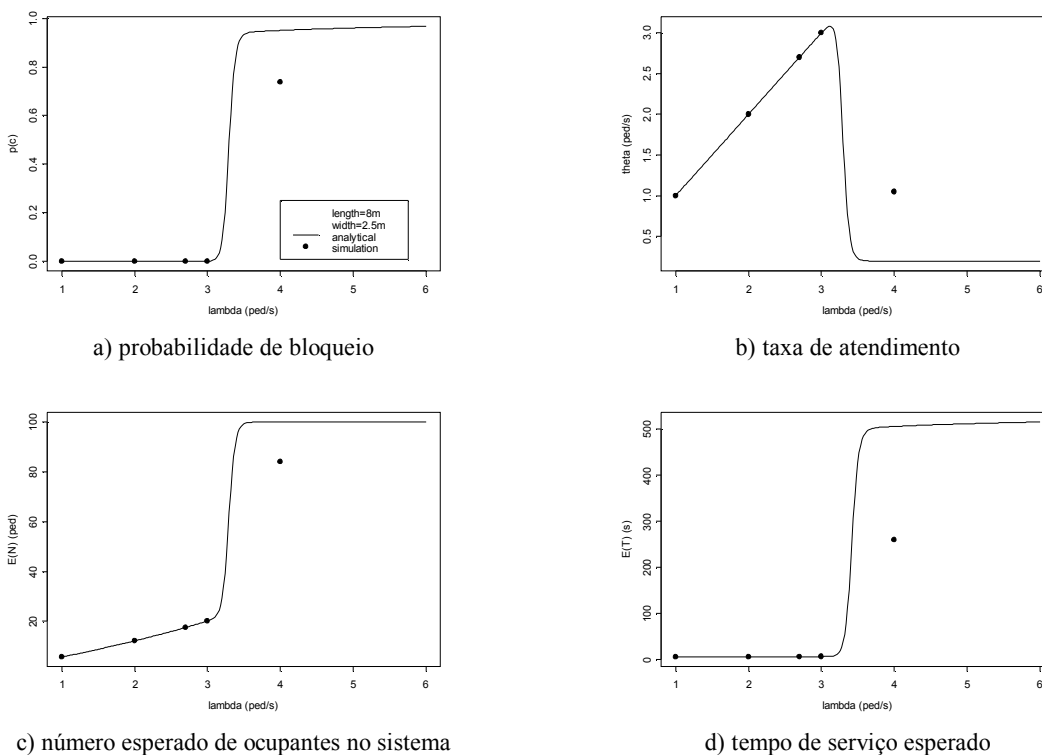
Realizamos experimentos com os modelos linear e exponencial. Não temos conhecimento de nenhum estudo assim detalhado. Para confirmar a precisão das estimativas, utilizamos um modelo de simulação a eventos discretos (Cruz et al., 2004), realizando simulações por 20.000 segundos e 30 replicações, para construção do I.C. de 95%. Primeiramente foram realizadas simulações para os modelos exponencial e linear em um corredor de  $8,0 \times 2,5$  m, e diferentes taxas de chegada  $\lambda$  (Figura 4 e 5). Posteriormente foram feitas simulações para um corredor de 8,0 m de comprimento e diferentes larguras, com  $\lambda=2,5$  e 5,0 (figuras não apresentadas aqui). Finalmente foram realizadas simulações variando-se o comprimento de um corredor de 2,5 m de largura, com  $\lambda=2,5$  e 5,0 (figuras não apresentadas aqui).

Analisando a Figura 4 e a Figura 5 podemos observar que não há diferença substancial entre os modelos linear e exponencial e a simulação. Além disso, nota-se que para todas as medidas de desempenho o modelo linear parece ser mais pessimista que o modelo exponencial. Notamos que a probabilidade de bloqueio para o modelo exponencial começa a crescer a partir de uma taxa de chegada menor do que aquela para o modelo linear. Porém, neste caso a probabilidade de bloqueio cresce bruscamente a partir de  $\lambda=3,0$  até atingir um valor constante, mais alto que para o caso anterior. Podemos notar efeito semelhante quando variamos a largura e fixamos as outras medidas (não apresentado). Além disto, é notável que as curvas para os dois modelos, exponencial e linear, apresentam uma forma muito semelhante. Neste caso, notamos que todas as medidas de desempenho melhoram com o aumento da largura do corredor, indicando que está é uma boa variável para o nosso algoritmo de otimização. Finalmente, ao analisarmos a variação do comprimento, também verificamos que a máxima probabilidade de bloqueio é menor para o modelo exponencial (não apresentado). A taxa de atendimento tem um comportamento semelhante para os dois modelos. Um fato muito curioso é que quando o modelo de congestionamento é exponencial o tempo de serviço esperado  $E(T)$  cresce

linearmente com o comprimento, mas quando o modelo é linear, o mesmo cresce exponencialmente. Notamos pouca influência do comprimento nas medidas de desempenho.



**Figura 4: Medidas de desempenho para um corredor de  $8,0 \times 2,5$  m e diferentes taxas de chegada para o modelo exponencial.**



**Figura 5: Medidas de desempenho para um corredor de  $8,0 \times 2,5$  m e diferentes taxas de chegada para o modelo linear.**

## 2.4. Análise de desempenho em redes de corredores

O problema de determinação das medidas de desempenho fica significativamente mais complexo, se estamos interessados em filas configuradas em rede. Note por exemplo que na topologia apresentada na Figura 1, a probabilidade de bloqueio na  $i$ -ésima fila depende de todo o fluxo anterior e também das probabilidades de bloqueio nas filas seguintes. O Método da Expansão Generalizado (GEM) é um algoritmo que permite lidar com tais redes. O GEM é uma combinação de métodos de aproximações sucessivas e decomposição nó-a-nó. Este método tem como característica chave a colocação de um nó artificial antes de cada fila finita no sistema, a fim de registrar os pedestres bloqueados que tentam entrar num corredor finito quando este está em seu limite. Desta forma, a rede de filas é “expandida” em uma rede de Jackson equivalente, na qual cada nó pode ser decomposto e analisado separadamente. Adaptações do GEM a topologias genéricas de redes M/G/c/c podem ser encontradas no trabalho de Araújo et al. (2004).

## 2.5. Algoritmo de resolução

O algoritmo para resolução do problema SCA é apresentado na Figura 6. O algoritmo principal implementa uma variação do método de busca por coordenadas. Assim o algoritmo lê as configurações iniciais e então é alocada uma capacidade inicial que satisfaça às restrições, ou seja, que dê uma probabilidade de bloqueio bem inferior ao limiar dado. Por conveniência, a capacidade inicial é dada na forma  $2^M$ , na qual M é um número "grande".

<pre> <b>Algorithm</b> leia <math>G(V,A)</math> leia as probabilidades de roteamento <math>p_{ij}, \forall (i,j) \in A</math> leia as taxas de chegada <math>\lambda_i, \forall i \in V</math> /* encontre um vetor inicial viável */ <b>for</b> <math>\forall i \in V</math> <b>do</b> <math>c_i^{opt} \leftarrow c_i^{sup} \leftarrow 2^M</math> <b>end for</b> /* encontre uma solução ótima */ iter <math>\leftarrow 0</math> <b>repeat</b> iter <math>\leftarrow</math> iter+1 /* otimize a <math>i</math>-ésima fila */ <b>for</b> <math>\forall i \in V</math> <b>do</b> OptQueue(<math>i, c^{sup}</math>) <b>end for</b> /* atualize para a melhor solução */ <b>if</b> <math>g(c^{sup}) &lt; g(c^{opt})</math> <b>then</b> <math>c^{opt} \leftarrow c^{sup}</math> tire rótulo de todos os nós <b>else</b> <b>exit</b> <b>end if</b> <b>end repeat</b> write <math>c^{opt}</math> <b>end algorithm</b> </pre>	<pre> <b>Algorithm</b> OptQueue(<math>i, c^{sup}</math>) /* passo recursivo de rotulação */ <b>for</b> <math>\forall (j,i) \in A</math> <b>do</b> <b>if</b> nó <math>j</math> está sem rótulo <b>then</b> OptQueue(<math>j, c^{sup}</math>) rotule nó <math>j</math> <b>end if</b> <b>end for</b> /* isole o ótimo */ j <math>\leftarrow 1</math> <math>c_i^{inf} \leftarrow c_i^{sup} \leftarrow 2^j</math> <b>while</b> <math>p(c_i^{sup}) &gt; \epsilon, \forall i \in V</math> j <math>\leftarrow j+1</math> <math>c_i^{sup} \leftarrow 2^j</math> <b>end while</b> /* reduzir o intervalo */ <math>c_i^{can} \leftarrow c_i^{sup}</math> <b>while</b> <math>(c_i^{sup} - c_i^{inf}) &gt; 1</math> <math>c_i^{can} \leftarrow (c_i^{inf} + c_i^{sup})/2</math> <b>if</b> <math>p(c_i^{can}) \leq \epsilon, \forall i \in V</math> <b>then</b> <math>c_i^{sup} \leftarrow c_i^{can}</math> <b>else</b> <math>c_i^{inf} \leftarrow c_i^{can}</math> <b>end if</b> <b>end while</b> <b>end algorithm</b> </pre>
a) otimização da rede	b) otimização de uma única fila

Figura 6: Algoritmo de resolução do problema SCA.

Na busca local, o algoritmo escolhe um nó  $i$  (corredor) e para ele define dois limites para a sua capacidade, que, para facilitar a busca, é sempre potência de 2. O primeiro valor, denominado limite inferior  $c_i^{inf}$ , é uma capacidade que com certeza *não* atende às restrições de as probabilidades de bloqueio serem menores que o limiar dado. Já  $c_i^{sup}$ , denominado limite superior, é um valor que garante a viabilidade da solução. A partir daí, o algoritmo começa a redução do intervalo tirando a média dos dois limites, determinando  $c_i^{can} = (c_i^{sup} + c_i^{inf})/2$ . Se a probabilidade de bloqueio considerando-se o valor  $c_i^{can}$  for menor que o limiar  $\epsilon$ , então isto significa que o limite superior  $c_i^{sup}$  está superestimado e pode ser atualizado para o valor  $c_i^{can}$ . Caso contrário, o limite inferior  $c_i^{inf}$  está subestimado, podendo portanto ser atualizado para o valor  $c_i^{can}$ . E assim é feito até que a diferença entre estes dois limites seja menor ou igual à unidade. Então, o valor dado para o limite superior  $c_i^{sup}$  é

a menor capacidade para a qual a probabilidade de bloqueio é inferior ou igual ao limiar  $\varepsilon$ . Note que para a convergência, a convexidade da função objetivo é assumida.

### 3. Experimentos computacionais

Os experimentos foram realizados em um computador com processador Intel® Celeron Processor, 28 MB RAM com sistema operacional Microsoft® Windows 98. Os algoritmos utilizados foram programados em C++ e, no modelo SCA, foi considerado um  $\varepsilon=0,001$  (0,1%). A qualidade das soluções foi atestada por meio de simulações (Cruz et al., 2004), durante 20.000 segundos e com 30 replicações, para construção dos IC's.

#### 3.1. Análise empírica de desempenho do algoritmo de resolução

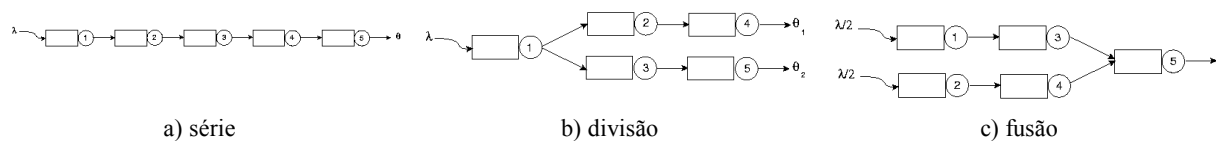


Figura 7: Topologias básicas consideradas, para cinco nós.

A Figura 7 mostra as três topologias consideradas. Embora sejam apresentadas apenas redes com cinco nós, também foram utilizadas redes com três e sete nós, nestas topologias.

Tabela 1: Resultados para a topologia série.

$\lambda$	c (ped)	w (m)	max $p(c_i)$	IC (95%)	tempo (s)
1	(42;45;46)	(1,06;1,14;1,16)	0,0028	(-0,0006;0,0062)	0,43
	(42;45;46;47;48)	(1,06;1,14;1,16;1,19;1,21)	0,0028	(-0,0006;0,0062)	1,4
	(42;45;46;47;48;48;48)	(1,06;1,14;1,16;1,19;1,21;1,21;1,21)	0,0028	(-0,0006;0,0062)	2,9
2	(79;79;82)	(1,99;1,99;2,06)	0,0080	(-0,0004;0,0165)	0,61
	(79;79;82;83;83)	(1,99;1,99;2,06;2,09;2,09)	0,0080	(-0,0004;0,0165)	2,0
	(79;79;82;83;83;84;87)	(1,99;1,99;2,06;2,09;2,09;2,11;2,19)	0,0080	(-0,0004;0,0165)	4,1
4	(151;152;155)	(3,79;3,81;3,89)	0,0000	(0,0000;0,0000)	1,0
	(151;152;155;157;158)	(3,79;3,81;3,89;3,94;3,96)	0,0000	(0,0000;0,0000)	3,2
	(151;152;155;157;158;160;161)	(3,79;3,81;3,89;3,94;3,96;4,01;4,04)	0,0000	(0,0000;0,0000)	6,3
8	(296;297;302)	(7,41;7,44;7,56)	0,0000	(0,0000;0,0000)	1,7
	(296;297;302;302;305)	(7,41;7,44;7,56;7,56;7,64)	0,0000	(0,0000;0,0000)	5,4
	(296;297;302;302;305;306;307)	(7,41;7,44;7,56;7,56;7,64;7,66;7,69)	0,0000	(0,0000;0,0000)	11

Sobre a topologia série apresentada na Tabela 1, o fato curioso a observar-se é o padrão de alocação crescente de capacidades ao longo da rede. Isto pode ser explicado pelo fato de que a probabilidade de bloqueio depende do fluxo dos corredores anteriores. Assim, para que o fluxo não tenha bloqueio, há a necessidade de aumentarem-se as capacidades dos corredores do final da topologia.

Tabela 2: Resultados para a topologia divisão.

$\lambda$	c (ped)	w (m)	max $p(c_i)$	IC (95%)	tempo (s)
1	(42;26;26)	(1,06;0,66;0,66)	0,0011	(0,0005;0,0017)	0,34
	(42;26;26;28;28)	(1,06;0,66;0,66;0,71;0,71)	0,0011	(0,0005;0,0017)	1,1
	(42;26;26;28;28;29;29)	(1,06;0,66;0,66;0,71;0,71;0,74;0,74)	0,0011	(0,0005;0,0017)	2,2
2	(79;42;44)	(1,99;1,06;1,11)	0,0028	(0,0014;0,0042)	0,47
	(79;42;44;47;48)	(1,99;1,06;1,11;1,19;1,21)	0,0013	(0,0005;0,0021)	1,5
	(79;42;44;47;48;48;49)	(1,99;1,06;1,11;1,19;1,21;1,21;1,24)	0,0024	(0,0009;0,0038)	3,0
4	(151;79;81)	(3,79;1,99;2,04)	0,0069	(-0,0017;0,0156)	0,70
	(151;79;81;81;83)	(3,79;1,99;2,04;2,04;2,09)	0,0188	(0,0018;0,0357)	2,1
	(151;79;81;81;83;87;87)	(3,79;1,99;2,04;2,04;2,09;2,19;2,19)	0,0050	(0,0007;0,0093)	4,1
8	(296;152;152)	(7,41;3,81;3,81)	0,0000	(0,0000;0,0000)	1,2
	(296;152;152;156;160)	(7,41;3,81;3,81;3,91;4,01)	0,0000	(0,0000;0,0000)	3,5
	(296;152;152;156;160;160;161)	(7,41;3,81;3,81;3,91;4,01;4,01;4,04)	0,0000	(0,0000;0,0000)	6,7

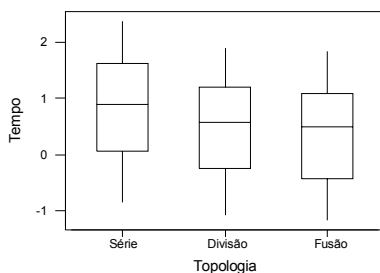


Observando a Tabela 2, notamos que para as duas seqüências geradas com a divisão, a soma da capacidade dos corredores é maior que a capacidade do corredor onde ocorre a divisão, o que não era o esperado. Em outras palavras, para  $\lambda=1$  e  $V=3$  nós, a um corredor com capacidade 42 seguem-se dois com capacidade 26, cuja soma 52 é maior que 42. Após a divisão, essas seqüências apresentam um comportamento de uma topologia série.

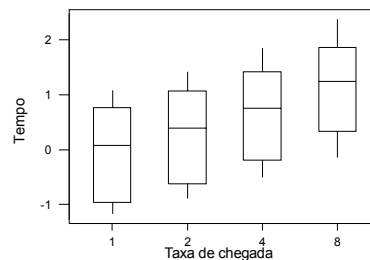
**Tabela 3: Resultados para a topologia fusão.**

$\lambda$	c (ped)	w (m)	max $p(c_i)$	IC (95%)	tempo (s)
0,5	(24;24;44)	(0,61;0,61;1,11)	0,0435	(-0,0043;0,0912)	0,31
	(24;24;25;25;45)	(0,61;0,61;0,64;0,64;1,14)	0,0052	(-0,0007;0,0112)	1,0
	(24;24;25;25;27;27;46)	(0,61;0,61;0,64;0,64;0,69;0,69;1,16)	0,0024	(0,0000;0,0047)	2,1
1	(42;42;81)	(1,06;1,06;2,04)	0,0154	(-0,0148;0,0457)	0,41
	(42;42;45;45;83)	(1,06;1,06;1,14;1,14;2,09)	0,0010	(0,0006;0,0013)	1,4
	(42;42;45;45;46;46;83)	(1,06;1,06;1,14;1,14;1,16;1,16;2,09)	0,0165	(-0,0158;0,0487)	2,9
2	(79;79;152)	(1,99;1,99;3,81)	0,0004	(-0,0002;0,0011)	0,60
	(79;79;79;79;154)	(1,99;1,99;1,99;1,99;3,86)	0,0055	(-0,0011;0,0121)	2,0
	(79;79;79;79;82;82;155)	(1,99;1,99;1,99;1,99;2,06;2,06;3,89)	0,0055	(-0,0011;0,0121)	4,1
4	(151;151;297)	(3,79;3,79;7,44)	0,0008	(-0,0008;0,0023)	0,86
	(151;151;152;152;300)	(3,79;3,79;3,81;3,81;7,51)	0,0019	(-0,0018;0,0055)	3,0
	(151;151;152;152;155;155;302)	(3,79;3,79;3,81;3,81;3,89;3,89;7,56)	0,0019	(-0,0018;0,0055)	6,3

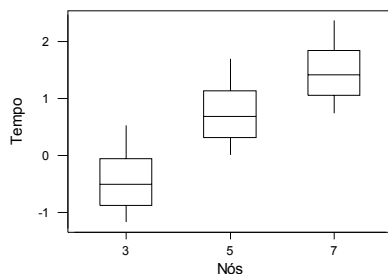
Pela Tabela 3, verificamos que os dois ramos que antecedem a fusão comportam-se como se fossem a topologia série. A capacidade da fila que recebe a fusão é menor ( $c=44$ , para  $\lambda=1$  e  $V=3$ ) que a soma das capacidades das antecessoras ( $24+24=48$ , para  $\lambda=1$  e  $V=3$ ), resultado também surpreendente, evidenciando aqui um efeito de economia de escala.



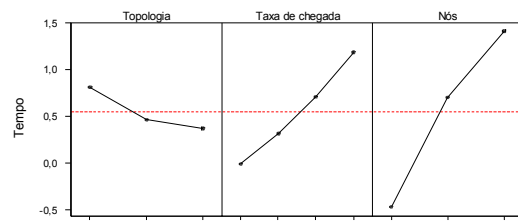
**Figura 8: Box-plot para a topologia.**



**Figura 9: Box-plot para a taxa de chegada  $\lambda$ .**



**Figura 10: Box-plot para o número de nós.**



**Figura 11: Gráfico de efeitos principais.**

Com o objetivo de obter indicações iniciais sobre o efeito que a topologia, a taxa de chegada e o número de nós exercem sobre o tempo de processamento do algoritmo de otimização, foram construídos box-plots estratificados por estes fatores, bem como o gráfico de efeitos principais. A partir da análise da Figura 8 foi possível notar que o tempo de processamento parece não ser afetado pela topologia, o que é confirmado pelo gráfico de efeitos principais da topologia, apresentado na

Figura 11, no qual a topologia série parece se diferir das demais, tendendo a consumir um tempo maior de execução. Por outro lado, observando a Figura 9, a taxa de chegada parece influenciar o tempo de processamento, sendo que quanto maior essa taxa, maior o tempo de processamento. O número de nós (Figura 10) também parece afetar o tempo de processamento, o qual possui um comportamento ascendente à medida que aumenta o número de nós. Entretanto, não se nota um crescimento exponencial que pudesse comprometer a utilização do algoritmo em instâncias com um maior número de nós.

#### 4. Estudo de caso

O sistema escolhido para análise foi uma caixa de escadas de um prédio de quatro andares. Esta caixa de escadas é composta por três lances de escada, conforme mostra a Figura 12. Para ser analisada, a caixa de escadas foi dividida em vários segmentos. A sua estrutura é composta por três *lobbies* ( $w_i \times 1,5 \text{ m}^2$  cada), nos quais ocorrem chegadas do 4º, 3º e 2º andar. Cada escada que liga dois andares adjacentes foi modelada como uma fila ( $w_i \times 13,0 \text{ m}^2$  cada um). A saída se dá na escada que liga o segundo ao primeiro andar, que é quando a pessoa acaba de descer todos os degraus da escada. O modelo de redes de filas resultante é apresentado na Figura 13. É de interesse avaliar o desempenho deste sistema estocástico, principalmente sob condições de emergências, tais como incêndios.

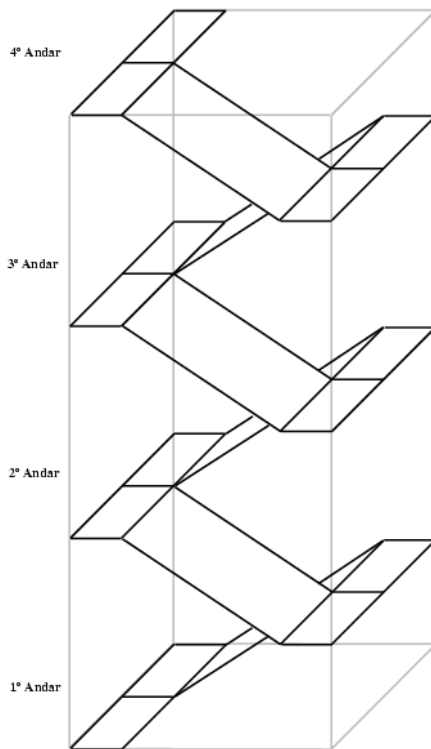


Figura 12: Caixa de escada analisada.

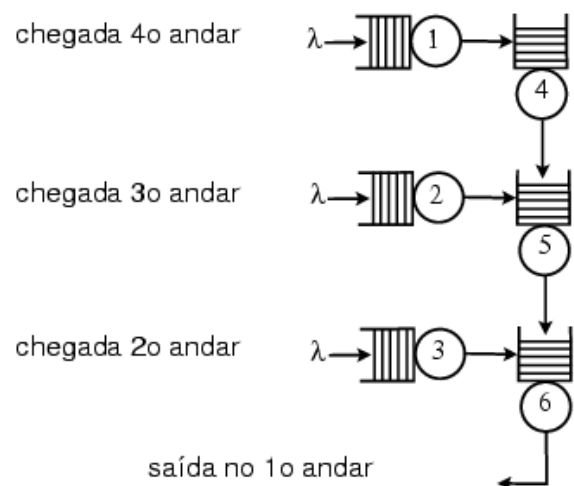


Figura 13: Modelo em rede de filas.

##### 4.1. Alocação de Capacidades

Nesta seção serão apresentados os resultados obtidos na alocação de capacidades para os diversos segmentos da caixa de escadas. Esta alocação leva em consideração que a melhor capacidade é aquela na qual a probabilidade de bloqueio  $\varepsilon$  seja menor que 0,001 (0,1%), para todos os corredores, ou seja, para todos os segmentos e *lobbies* da escada. Utilizaremos nesta seção as taxas de chegada 0,25, 0,50, 1,00, 2,00 e 4,00 ped/s.

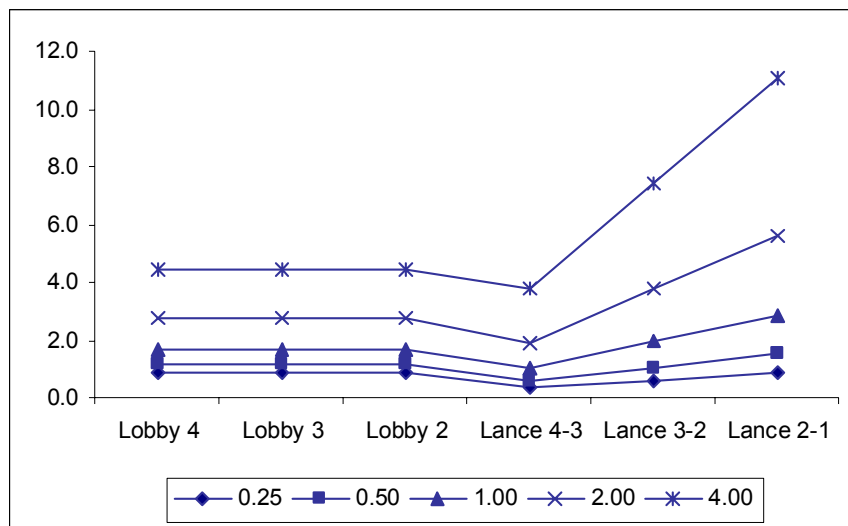
A Tabela 4 mostra as capacidades e as larguras, respectivamente, obtidas para cada taxa de chegada. Otimizamos a largura uma vez que esta foi a que influenciou mais fortemente a probabilidade de bloqueio. Através da análise da Tabela 4 e da Figura 14, nota-se que os lances entre o 4º e o 3º andares têm a menor capacidade alocada. Analisando-se a alocação percebe-se que, à medida que se desce, a

capacidade ótima da escada aumenta. O aumento de capacidade se dá porque o fluxo de pessoas nos andares inferiores é maior. Em outras palavras, quem entra no 4º andar, por exemplo, só sai no 1º andar. Assim os andares inferiores acumulam as pessoas que os deixam acrescidas daquelas que deixam os andares superiores.

**Tabela 4: Capacidades ótimas, em número de pedestres (c) e em metros (w).**

$\lambda$	lobby 4		lobby 3		lobby 2		lance 4-3		lance 3-2		lance 2-1		max p(c)		
	c	w	c	w	c	w	c	w	c	w	c	w	GEM	simulação*	IC 95%*
0,25	6	<b>0,87</b>	6	<b>0,87</b>	6	<b>0,87</b>	21	<b>0,33</b>	36	<b>0,56</b>	57	<b>0,89</b>	0.0010	0.0087	(-0.0024; 0.0198)
0,50	8	<b>1,13</b>	8	<b>1,13</b>	8	<b>1,13</b>	37	<b>0,58</b>	67	<b>1,04</b>	99	<b>1,53</b>	0.0010	0.0111	(-0.0082; 0.0305)
1,00	12	<b>1,67</b>	12	<b>1,67</b>	12	<b>1,67</b>	67	<b>1,04</b>	126	<b>1,95</b>	186	<b>2,87</b>	0.0010	0.0009	(0.0007; 0.0010)
2,00	20	<b>2,73</b>	20	<b>2,73</b>	20	<b>2,73</b>	124	<b>1,92</b>	243	<b>3,75</b>	362	<b>5,58</b>	0.0009	0.0004	(0.0003; 0.0004)
4,00	33	<b>4,47</b>	33	<b>4,47</b>	33	<b>4,47</b>	244	<b>3,76</b>	480	<b>7,39</b>	718	<b>11,05</b>	0.0010	0.0010	(0.0008; 0.0011)

\*Via modelo de Cruz et al. (2004).



**Figura 14: Comportamento das larguras.**

## 5. Conclusões

Uma metodologia baseada em redes de filas  $M/G/c/c$  dependentes do estado, apropriada para análise e síntese de sistemas sujeitos a congestionamento foi apresentada, enfocando-se aplicações em redes de pedestres, embora muitas outras aplicações sejam possíveis em sistemas estocásticos com tempo de serviço sujeito a congestionamento. Ressaltamos a importância do modelo e fizemos uma breve revisão dos resultados recentes na área. Discutimos em detalhes a aplicação do modelo ao problema de planejamento de redes de circulação de pedestres. Resultados computacionais apresentados atestaram a qualidade da abordagem. Especificamente em relação ao problema de análise e síntese de uma caixa de escadas sob demanda de emergência, pudemos notar que a largura ótima aumenta nos andares inferiores e é bem sensível à taxa de chegada.

Questões ainda permanecem em aberto. Testes adicionais devem ser feitos para diferentes topologias e probabilidades de bloqueio. Na área de estudos sobre evacuação, diferentes alturas e disposições de caixas de escadas devem ser analisadas. Também de interesse é saber se a complexidade computacional do problema SCA aumentaria muito com a inclusão de restrições que levem em consideração a dificuldade prática de se construir caixas de escada com larguras crescentes. Outra possibilidade é estender a utilização destes modelos ao problema de alocação de capacidades em redes de tráfego de veículos.

## Agradecimentos

Paula C. Oliveira é aluna de mestrado financiada pela CAPES. Flávia G. Amorim e Maria Luíza G. Toledo são bolsistas de iniciação científica pela FAPEMIG e CNPq, respectivamente. Frederico R. B. Cruz é financiado pelo *Conselho Nacional de Desenvolvimento Científico e Tecnológico*, CNPq, processos 301809/96-8 e 201046/94-6, pela *Fundação de Amparo à Pesquisa do Estado de Minas Gerais*, FAPEMIG, processos CEX-289/98 e CEX-855/98, e pela Pró-Reitoria de Pesquisa da UFMG, PRPq-UFMG.

## Referências bibliográficas

- (1) Cheah, J. & Macgregor Smith, J. (1990). Generalized M/G/C/C state dependent queueing networks, *Queueing Systems*, 15:365-386.
- (2) Chaudhry, M. L.; Gupta, U.C. & Goswami, V. (2001). Modelling and analysis of discrete-time multiserver queues with batch arrival:  $GI^X/Geom/m$ , *INFORMS Journal on Computing* 13:172-180.
- (3) Araújo, H. F.; Toledo, M. L. G.; Cruz, F. R. B. & Macgregor Smith, J. (2004) Generalized M/G/c state dependent queueing network analysis, In: *XV Congresso Brasileiro de Automática* (to appear). URL: <ftp://ftp.est.ufmg.br/pub/fcruz/publics/cba04.pdf>.
- (4) Cruz, F. R. B.; MacGregor Smith, J. & Medeiros, R. O. (2004) An M/G/c state dependent network simulation model, *Computers & Operations Research* (forthcoming).
- (5) Enginarlar, E.; Li, J.; Meerkov, S. M. & Zhang, R. Q. (2002). Buffer capacity for accommodating machine downtime in serial production lines, *International Journal of Production Research* 40(3): 601-624.
- (6) Gross, D. & Harris. C. M. (1998). *Fundamentals of queueing theory*, 3<sup>a</sup> ed., John Wiley & Sons, New York.
- (7) Harris, J. H. & Powell, S. G. (1999). An algorithm for optimal buffer placement in reliable serial lines, *IIE Transactions* 31: 287-302.
- (8) Huang, M. G.; Chang, P. L. & Chou, Y. C. (2002). Buffer allocation in flow-shop-type production systems with general arrival and service patterns, *Computers & Operations Research* 29(2): 103-121.
- (9) Jain, R. & MacGregor Smith, J. (1997) Modeling vehicular traffic flow using M/G/C/C state dependent queueing models, *Transportation Research*, 31(4):324-336.
- (10) Kimura, T. (1996). A transform-free approximation for the finite capacity M/G/s queue, *Operations Research* 44(6): 984-988.
- (11) Li, Y. J.; Cai, X. Q.; Tu, F. S. & Shao, X. L. (2004). Optimization of tandem queue systems with finite buffers. *Computers & Operations Research* 31:963-984.
- (12) MacGregor Smith, J. & Li, W. (2001). Quadratic assignment problems and M/G/C/C state dependent network flows, *Journal of Combinatorial Optimization*, 5:421-443.
- (13) McGeoch, C. C. (2001). Experimental analysis of algorithms, *Notices of the AMS* 48(3): 304-311.
- (14) Rubin, I. & Zhang, Z. (1991). Message delay and queue-size analysis for circuit-switched TDMA systems, *IEEE Transactions on Communications* 39: 905-913.
- (15) Tregenza, P. R. (1976). *The Design of Interior Circulation*, Van Nostrand Reinhold Company, New York, USA.
- (16) Yuhaski, S. J. e MacGregor Smith, J. (1989). Modeling circulation systems in buildings using state dependent models. *Queueing Systems*, 4:319-338.