

Harley Francisco de Araújo

**Caracterização de Tráfego Entre Filas M/G/c/c
Dependentes do Estado**

**Dissertação apresentada ao
Departamento de Estatística do
Instituto de Ciências Exatas da
Universidade Federal de Minas
Gerais, como requisito parcial à
obtenção do título de Mestre em
Estatística.**

Universidade Federal de Minas Gerais

Belo Horizonte, novembro de 2004

Folha de Aprovação

Agradecimentos

Desejo expressar meus sinceros agradecimentos a todos aqueles que, de alguma forma, contribuíram para o êxito deste trabalho. Em especial, agradeço meus pais, irmão e filhos, Francisco, Belmira, Daniel, Camila e André que, mesmo sentindo por minha ausência, sempre me incentivaram, dividindo alegrias e tristezas por toda minha caminhada e não mediram esforços para que eu atingisse essa vitória.

Agradeço também ao Professor Frederico R. B. Cruz pela pronta disposição em orientar-me nesse trabalho, não medindo esforços, paciência e a atenção necessária para atingirmos nossa meta. Aos Professores Gregório S. Atuncar, Enrico Antônio Colosimo, Luiz Henrique Duczmal, Sueli Aparecida Mingoti vão os meus agradecimentos por proporcionarem o conhecimento necessário ao meu enriquecimento e a satisfação de tê-los como mestres.

Agradeço, também, a Luciana pelos incentivos, apoios e compreensão neste meu percurso.

Agradeço a Deus que me proporcionou vencer mais esta etapa e por ter colocado em minha vida todas estas pessoas que doaram um pouco de si e me auxiliaram nesta jornada.

*“Tudo posso Naquele que me fortalece”
Filipenses -4,13*

Sumário

RESUMO	IX
ABSTRACT	XI
CAPÍTULO 1.....	1
INTRODUÇÃO	1
1.1 PRELIMINARES	1
1.1.1 Tráfego de Pedestres	2
1.1.2 Congestionamento	2
1.2 ESCOPO DA DISSERTAÇÃO E PRINCIPAIS CONTRIBUIÇÕES	4
1.3 ORGANIZAÇÃO DO TEXTO.....	4
CAPÍTULO 2.....	5
SOBRE FILAS FINITAS E MODELAGEM DE TRÁFEGO	5
2.1 REVISÃO SOBRE PROCESSOS ESTOCÁSTICOS.....	5
2.2 ANÁLISE DE FILAS ÚNICAS	5
2.2.1 Probabilidades de Equilíbrio	5
2.2.2 Filas Markovianas.....	7
2.2.3 Filas Não Markovianas	7
2.3 MODELO ANALÍTICO PARA TRÁFEGOS COM CONGESTIONAMENTO.....	10
2.3.1 Modelo de Congestionamento para Tráfego de Pedestres	11
2.4 EXPERIMENTOS COMPUTACIONAIS	13
2.5 PRINCIPAIS CONTRIBUIÇÕES DESTE CAPÍTULO.....	16
CAPÍTULO 3.....	17
SOBRE FILAS M/G/C/C CONFIGURADAS EM REDES.....	17
3.1 INTRODUÇÃO	17
3.2 MÉTODO DA EXPANSÃO GENERALIZADO PARA FILAS M/G/C/C.....	19
3.4 UMA NOVA IMPLEMENTAÇÃO PARA O GEM.....	21
3.5 EXPERIMENTOS COMPUTACIONAIS	24
3.6 PRINCIPAIS CONTRIBUIÇÕES DESTE CAPÍTULO.....	26
CAPÍTULO 4.....	27
VALIDAÇÃO DA MODELAGEM POR FILAS M/G/C/C CONFIGURADAS EM REDE.....	27
4.1 UM SISTEMA DE SIMULAÇÃO A EVENTOS DISCRETOS.....	27
4.1.1 Introdução	27
4.1.2 Algoritmo de Simulação	29
4.2 AJUSTE DE MODELOS.....	32
4.3 EXPERIMENTOS COMPUTACIONAIS	33
4.3.1 Sobre o Tempo de Serviço	33
4.3.2 Sobre o Tráfego Entre Filas	35
4.4 PRINCIPAIS CONTRIBUIÇÕES DESTE CAPÍTULO.....	42
CAPÍTULO 5.....	43
CONCLUSÕES E OBSERVAÇÕES FINAIS	43
REFERÊNCIAS BIBLIOGRÁFICAS.....	45

Resumo

Nesta dissertação, abordamos a questão da modelagem do tráfego entre filas em sistemas de filas $M/G/c/c$ dependentes do estado, configuradas em redes. Topologias série, divisão (do inglês *split*) e junção (*merge*) foram consideradas. Para a avaliação dos sistemas observamos principalmente os tempos entre partidas, mas também as tradicionais medidas de desempenho, isto é, a probabilidade de bloqueio, a taxa efetiva de chegada (também conhecida como a taxa de atendimento, em inglês *throughput*), o número médio de usuários no sistema (ou o *work-in-process*) e o tempo médio no sistema. Os resultados indicam que o modelo com chegadas exponenciais é uma boa aproximação na modelagem do tráfego entre filas, em quase todos os casos simulados. Eles apontam, também, para uma surpreendente similaridade existente entre topologias diversas. Estes resultados são comprovados pelas simulações.

Abstract

This dissertation deals with issues related to modeling inter-queue traffic in network of $M/G/c/c$ state-dependent queues. We considered queues in series, merges, and splits. In order to evaluate the systems, we observed mainly the inter-departure time, but also regarded were the traditional performance measurements such as blocking probability, throughput, work-in-process, and sojourn time. Our results point to a good fit of the exponential distributions in modeling the inter-queue inter-departure time, in almost all cases tested.

Capítulo 1

Introdução

“FOGO MATA 11 EM INCÊNDIO NA CALIFÓRNIA (Jornal o Globo, 27/10/2003) (...) incêndio iniciou no segundo andar do hotel, se alastrando rapidamente. A fumaça e a precariedade da estrutura das instalações foram as principais causas que levaram onze pessoas à morte. O chefe do departamento de bombeiros (...)”

“CONCLUÍDAS OBRAS DE CONSERVAÇÃO NA RODOVIA BR-101 (Jornal do Brasil, 23/07/2002) (...) BR-101, entre Angra dos Reis e Tarituba, em Parati (...) consideradas estratégicas, pois a BR-101 é a única rota rodoviária de evacuação da população da região em caso de acidente nas Usinas de Angra I e II. As obras (...) Foram construídas cinco passarelas e uma passagem subterrânea de pedestres e (...)”

“BHTRANS PLANEJA OBRAS NA PAMPULHA” (Jornal Estado de Minas, 2003) (...) preocupados com o fluxo do tráfego na região, os engenheiros estudam maneiras de contornar os problemas que (...)”

“CONGESTIONAMENTO FAZ RECEITA ALTERAR SITE” (Globo OnLine, 10/06/2004) (...) Não duraram nem um dia as exigências da Receita Federal para acesso às informações na internet sobre a restituição do Imposto de Renda da Pessoa Física. Diante do congestionamento no site ontem, as autoridades da Receita foram obrigadas a voltar atrás e dispensar a necessidade de informação do número do recibo de entrega da declaração para (...)”

1.1 Preliminares

Diariamente deparamo-nos com manchetes por todo o mundo que anunciam enormes congestionamentos, causados por evacuações devido a catástrofes, grandes filas de atendimentos e problemas causados por planejamentos deficientes. As grandes cidades brasileiras, assim como muitas cidades de países desenvolvidos e em desenvolvimento, estão se adaptando nas últimas décadas para o uso eficiente das vias públicas. A engenharia de tráfego está cada vez melhor e mais ágil na busca por soluções para o tráfego.

Muitos problemas do mundo real, incluindo aqueles da área de telecomunicações, transportes, fluxo de pedestres, dentre outros, podem ser modelados como redes de filas com capacidade finita e taxa de serviço dependente do estado. Nosso objetivo é apresentar uma metodologia geral para análise aproximada de filas $M/G/c/c$ dependentes do estado, configuradas em redes. De acordo com a notação de Kendall (1953), o M indica que a chegada é um processo markoviano, o G que o tempo de serviço segue uma distribuição geral, o c refere-se ao número máximo de servidores simultaneamente disponíveis e, por fim, assume-se que a capacidade total do sistema (espaço para a fila, incluindo os servidores) está restrita a c usuários. Assumimos que os tempos de serviço são dependentes do número de usuários no sistema.

Uma aplicação que pode ser tratada por esta metodologia está relacionada com o fluxo de pedestres. Estudos clássicos sobre a velocidade de locomoção de pedestres incluem os trabalhos de Fruin (1971) e Tregenza (1976). Nestes trabalhos é ressaltado que existem muitos fatores comuns que podem ser associados com as diferentes velocidades de locomoção de pedestres, mas o mais importante parece mesmo ser a densidade de aglomeração, isto é, o estado do sistema.

1.1.1 Tráfego de Pedestres

A eficiência do sistema de circulação interno de um edifício é determinada principalmente pela sua forma básica. Um projeto detalhado das escadas, corredores, elevadores e escadas rolantes é importante, porém, a qualidade do edifício depende principalmente da análise básica do fluxo de pessoas dentro do seu sistema de circulação e da conseqüente decisão sobre a forma fundamental da sua estrutura. Se longos tempos de espera para os elevadores ocorrem com freqüência, isto é um reflexo da ineficiência do sistema de circulação. Análises do fluxo de pedestres são especialmente importantes no projeto de rotas de escape de emergência e no projeto da circulação de edifícios públicos, *shopping centers*, terminais de transporte *etc* (Cheah & McGregor Smith, 1994).

No caso de emergências, o movimento o mais rápido e tranqüilo possível da multidão ao longo das rotas de escape é extremamente vital. Para atingir tal fluxo de pedestres, no caso de uma emergência, o projeto de rotas de escape, como também o estudo do tráfego de pedestres, é essencial.

Quando se trata de rotas de pedestre em grandes edifícios públicos, não há muita informação em que um projetista ou um arquiteto possa confiar. Portanto, o estudo e, conseqüentemente, o entendimento da natureza ou característica do fluxo de pedestres podem ajudar os arquitetos a planejar melhor um sistema de circulação em edifícios públicos.

Quando um grande número de usuários ou pedestres busca ocupar o espaço limitado em um sistema de circulação, o congestionamento acontece. Então, não é surpreendente que congestionamentos aconteçam freqüente e severamente dentro de edifícios públicos. As propriedades inerentes de um sistema de circulação que causam congestionamento são:

- A taxa de serviço do movimento do tráfego no sistema decai com o aumento do tráfego de pedestres;
- A quantidade de espaço disponível dentro do sistema é finita.

Estas características conduzem uma pessoa a se perguntar como alguém deveria modelar o congestionamento dentro de sistemas de circulação. O objetivo deste trabalho é examinar os problemas de congestionamento em sistemas de circulação, corredores em particular, a partir de uma fila e da perspectiva de otimização de rede em que o tráfego em sistemas de circulação é modelado como filas $M/G/c/c$ dependentes do estado, configuradas em redes.

1.1.2 Congestionamento

Congestionamentos ocorrem principalmente devido a tráfegos pesados de usuários em espaços limitados. A velocidade de locomoção varia com a população e com o ambiente físico. De acordo com Tregenza (1976), a velocidade de locomoção tende a ser normalmente distribuída com médias que variam mais entre diferentes situações do que o desvio padrão.

Muitas pesquisas sobre a velocidade de caminhada têm sido conduzidas (Fruin, 1971; Tregenza, 1976). A maioria tem sido estudos de projetos de ruas e cruzamentos de pedestres. Em particular, aqueles em interiores de edifícios em estudos de rotas de fugas de incêndios. Disparidades de condições entre as pesquisas introduzem perigos na comparação entre um conjunto de resultados e outro. Contudo, alguém pode identificar

muitos fatores, por exemplo, idade, sexo, propósito da viagem *etc*, associados com variações na média da velocidade de locomoção de um pedestre.

O fator que determina mais significativamente a velocidade de um pedestre, porém, é a densidade do tráfego (Cheah & McGregor Smith, 1994). Uma caminhada normal requer área suficiente para um ritmo irrestrito, reconhecimento sensorial e reação a obstáculos potenciais. Com o aumento da densidade e o número de pessoas por unidade de área, a velocidade média diminui devido à redução da área disponível para locomoção. Como resultado, a velocidade de locomoção para todos os pedestres tende a ter menor variabilidade em decorrência de o incremento de a densidade populacional restringir a habilidade de ultrapassar um pedestre mais lento e retomar a velocidade desejada de caminhada. O efeito da população na velocidade de locomoção tem sido medido em muitas pesquisas. De acordo com Tregenza (1976), a velocidade média de locomoção reduz para um *arrastar de pés* a uma densidade média de 3 ped/m² e o movimento à frente é bloqueado em 5 ped/m². Resultados empíricos obtidos por Tregenza (1976), relacionando a velocidade média de locomoção e a densidade populacional, são apresentados na Figura 1.1.

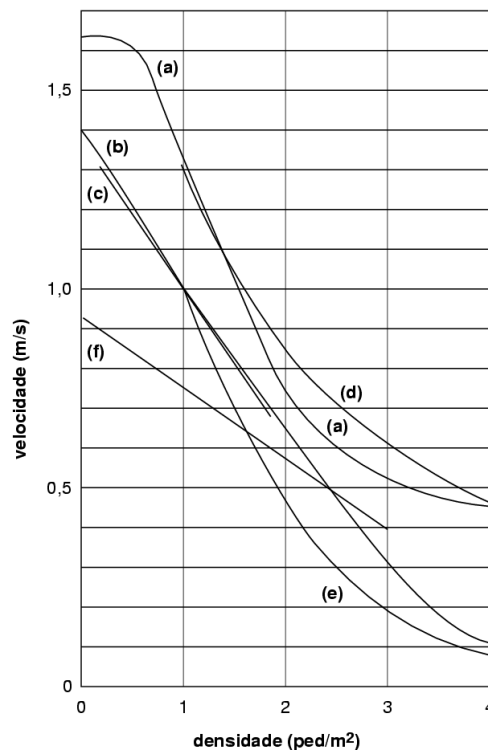


Figura 1.1: Velocidade média de locomoção por densidade populacional, para diversos estudos empíricos – curvas de (a) a (f) – conduzidos por Tregenza (1976).

O volume de pedestres, ou o número de pessoas passando por um dado ponto numa unidade de tempo, é o mais importante parâmetro no projeto de áreas de circulação. Se a demanda do tráfego excede a capacidade de uma seção da área de circulação, aglomerando pedestres, resultará numa locomoção arrastada e desconfortável, além de demora, produzindo um ambiente ruim aos pedestres.

Fruin (1971) estabeleceu relações entre fluxo e volume para tráfegos unidirecionais, bidirecionais e multidirecionais. Estas relações, representando as condições médias de três tipos distintos de tráfego de pedestres, mostraram um intervalo relativamente

pequeno de variação, o que sugere fortemente que conflitos com tráfegos contrários ou cruzamentos não reduzem drasticamente o volume do tráfego de pedestres ou sua velocidade. Assim, com a ajuda destes dados empíricos de Fruin (1971) e Tregenza (1976), podem-se modelar congestionamentos de fluxos de tráfego unidirecionais, bidirecionais e multidirecionais.

1.2 Escopo da Dissertação e Principais Contribuições

Nesta dissertação tratamos de problemas decorrentes da modelagem de sistemas com fluxos sujeitos a congestionamento, via filas com capacidade restrita (isto é, espaço para formação de filas), configuradas em redes (Gross & Harris, 1998). Questões relevantes são relativas à caracterização do tráfego entre filas configuradas em redes.

Contribuições desta dissertação incluem:

- Levantamento detalhado da bibliografia recentemente publicada sobre o assunto;
- Verificação empírica da equivalência estocástica entre filas finitas markovianas e filas com taxas de serviço gerais dependentes do estado;
- Apresentação detalhada de uma implementação do método da expansão generalizado, uma técnica aproximada para análise de filas finitas com taxas de serviço gerais dependentes do estado, configuradas em redes;
- Condução de extensivos experimentos computacionais que comprovam a eficácia da implementação proposta para o método da expansão generalizado;
- Criação de uma metodologia para validação da modelagem de sistemas de tráfego de pedestres via filas finitas dependentes do estado, configuradas em redes;
- Apresentação de resultados de validação em diversas situações.

1.3 Organização do Texto

O texto está organizado como se segue. O capítulo 2 analisa filas únicas e apresenta as equações para as suas probabilidades de equilíbrio. Apresenta também uma demonstração da equivalência estocástica entre sistemas $M/M/c/c$ e $M/G/c/c$, além de fornecer resultados computacionais da comparação dos modelos de tráfego de pedestres, linear e exponencial, desenvolvidos por Yuhaski e MagGregor Smith (1989). O capítulo 3 trata de filas $M/G/c/c$ configuradas em redes e discorre sobre o método da expansão generalizado para análise de tais filas, descrevendo detalhadamente suas etapas. Além disso, são mostrados resultados computacionais de experimentos envolvendo redes de filas. O capítulo 4 apresenta um sistema de simulação a eventos discretos, ilustra seus módulos e descreve com mais detalhes o módulo de simulação deste pacote computacional. São apresentados também resultados computacionais obtidos através deste sistema, relevantes ao estudo de tráfego entre filas. O capítulo 5 encerra o texto com as principais conclusões e propostas de trabalhos futuros.

Capítulo 2

Sobre Filas Finitas e Modelagem de Tráfego

2.1 Revisão sobre Processos Estocásticos

Uma seqüência de variáveis aleatórias indexadas por um mesmo parâmetro, por exemplo, o tempo, forma um processo estocástico. Por exemplo, o número de usuários em uma fila de um sistema em função do tempo é um processo estocástico.

Os processos mais comuns utilizados nos sistemas de fila são:

- **processos estocásticos com estado discreto:** dependendo dos valores que a variável aleatória pode assumir, temos processos com estados discretos ou contínuos. O número de usuários em uma fila é um processo discreto.
- **processo de Markov:** se os estados futuros do processo dependem exclusivamente do estado atual, o processo é chamado de processo de Markov, como no caso de variáveis aleatórias com distribuição exponencial que possuem a propriedade de falta de memória. Neste caso, o tempo em que o sistema se encontra no estado atual não afeta seu próximo estado.
- **cadeia de Markov:** um processo de Markov com estados discretos é chamado de cadeia de Markov.
- **processo de nascimento e morte:** os processos de Markov que possuem um espaço de estado discreto e no qual as transições de estado só podem ocorrer entre estados vizinhos são denominados de processos de nascimento e morte. Neste caso, podem-se representar os estados por um número inteiro N e as possíveis mudanças de estado são para $(N+1)$ ou $(N-1)$. O número de usuários, em um sistema de filas que apresenta somente chegadas individuais, pode ser modelado por um processo de nascimento e morte.
- **processo de Poisson:** se os intervalos entre chegadas são independentes e identicamente distribuídos e seguem uma distribuição exponencial então o número de chegadas num intervalo de tempo $(t; t+\Delta t)$ possui uma distribuição de Poisson.

2.2 Análise de Filas Únicas

Seja um sistema de fila única como indicado na Figura 2.1. Este sistema pode ser descrito como um processo de nascimento e morte em que o estado é representado pelo número de usuários no sistema. Na Figura 2.2 temos um diagrama que representa as transições entre os estados e suas taxas.

2.2.1 Probabilidades de Equilíbrio

Consideremos que em um instante t o sistema está no estado j , isto é, existem j usuários no sistema. $p_j(t)$ é a probabilidade do sistema estar no estado j no instante t , isto é, de possuir j usuários. O cálculo de $p_j(t)$ na situação de equilíbrio (se houver equilíbrio) pode ser feito através de uma equação de balanceamento de fluxo de probabilidades em

cada estado da cadeia de Markov. A última equação necessária é que a soma das probabilidades de todos os estados seja igual à unidade.



Figura 2.1: Representação esquemática de um sistema de fila única, com chegadas a uma taxa λ_i e serviço a uma taxa μ_i .

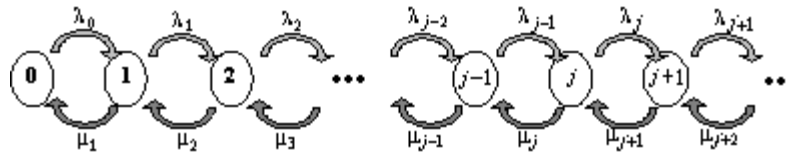


Figura 2.2: Diagrama de transição de estados.

Em um intervalo de tempo Δt , o sistema poderá continuar no estado j ou mudar para o estado $j-1$ ou $j+1$ de acordo com as probabilidades:

$$\Pr\{\text{ir para o estado } j+1 \mid \text{está no estado } j\} = \lambda_j \Delta t + o(\Delta t)$$

$$\Pr\{\text{ir para o estado } j-1 \mid \text{está no estado } j\} = \mu_j \Delta t + o(\Delta t)$$

$$\Pr\{\text{continuar no estado } j\} = 1 - \lambda_j \Delta t - \mu_j \Delta t + o(\Delta t)$$

O intervalo de tempo Δt deve ser pequeno o suficiente para não haver dois eventos simultâneos no seu transcorrer. Podemos montar um conjunto de equações para as transições de estado:

$$\begin{aligned} p_0(t + \Delta t) &= [1 - \lambda_0 \Delta t + o(\Delta t)]p_0(t) - [\mu_1 \Delta t + o(\Delta t)]p_1(t) \\ p_1(t + \Delta t) &= [\lambda_0 \Delta t + o(\Delta t)]p_0(t) + [1 - \lambda_1 \Delta t - \mu_1 \Delta t + o(\Delta t)]p_1(t) + [\mu_2 \Delta t + o(\Delta t)]p_2(t) \\ p_2(t + \Delta t) &= [\lambda_1 \Delta t + o(\Delta t)]p_1(t) + [1 - \lambda_2 \Delta t - \mu_2 \Delta t + o(\Delta t)]p_2(t) + [\mu_3 \Delta t + o(\Delta t)]p_3(t) \\ &\vdots \\ p_j(t + \Delta t) &= [\lambda_{j-1} \Delta t + o(\Delta t)]p_{j-1}(t) + [1 - \lambda_j \Delta t - \mu_j \Delta t + o(\Delta t)]p_j(t) + [\mu_{j+1} \Delta t + o(\Delta t)]p_{j+1}(t) \end{aligned} \quad (1)$$

Reescrevendo as equações, aplicando o limite quando $\Delta t \rightarrow 0$ e obtendo as equações diferenciais, encontramos a probabilidade de equilíbrio de um processo de nascimento e morte estar em um determinado estado:

$$p_n = p_0 \prod_{j=0}^{n-1} \frac{\lambda_j}{\mu_{j+1}}, \quad \text{para } n = 1, 2, \dots, \infty, \quad (2)$$

em que p_n é a probabilidade de que existam n usuários no sistema e

$$p_0 = \left[1 + \sum_{n=1}^{\infty} \prod_{j=0}^{n-1} \frac{\lambda_j}{\mu_{j+1}} \right]^{-1} \quad (3)$$

é a probabilidade de não haver usuários no sistema, observando que $\sum_{n=0}^{\infty} p_n = 1$.

2.2.2 Filas Markovianas

Filas podem ser markovianas ou não, e finitas ou infinitas. Filas markovianas infinitas com servidor único, as filas $M/M/1/\infty$, são um caso particularmente interessante, pela análise simples e elucidativa. O primeiro M indica que a chegada é um processo markoviano, o segundo M representa um serviço também markoviano, 1 refere-se ao número de servidores disponíveis e, por fim, assume-se que a capacidade total do sistema (espaço para a fila adicionado do número de servidores) está irrestrita. Assumimos que o sistema $M/M/1/\infty$ encontra-se na situação de equilíbrio em que os λ e μ não se alteram conforme o número de usuários no sistema. Nas Figuras 2.3 e 2.4 temos as representações da fila e seu diagrama de transições de estados.

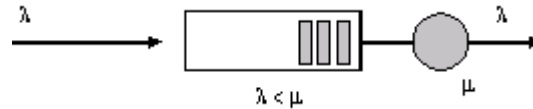


Figura 2.3: Representação de uma fila com um único servidor, com chegadas a uma taxa λ e serviço a uma taxa μ .

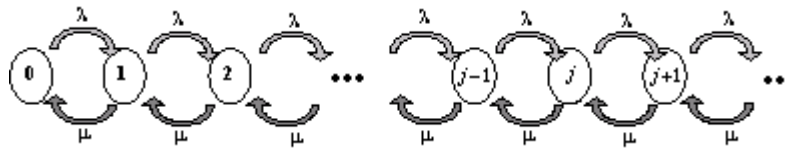


Figura 2.4: Diagrama de transição de estados de uma fila com apenas um servidor.

Para o sistema $M/M/1/\infty$ tem-se que $\lambda_n = \lambda$ e $\mu_n = \mu$. Aplicando a forma geral de solução do processo de nascimento e morte, chega-se a:

$$P_n = \left(\frac{\lambda}{\mu} \right)^n P_0, \quad (4)$$

em que a razão λ/μ é chamada intensidade de tráfego.

As markovianas finitas são aquelas com capacidade finita, com chegadas e atendimentos pelo processo de Poisson, indicadas de forma genérica por $M/M/c/k$, em que c refere-se ao número de servidores disponíveis e a capacidade total do sistema está restrita a $k \geq c$ usuários. Mais comum neste caso é considerar que o sistema encontra-se na situação de equilíbrio em que a taxa de chegada não se altera, sendo igual a λ , e as taxas de serviço dos servidores também não se alteram, sendo que todos os servidores têm a mesma taxa de atendimento μ .

Um caso particular de filas markovianas finitas é o modelo $M/M/c/c$, conhecido também como modelo de perda de Erlang (Ross, 1996), por não haver espaço para formação de filas de espera, uma vez que $c=c$. Este modelo é bastante importante, pelas suas inúmeras aplicações práticas, tais como no planejamento de centrais telefônicas, estacionamentos, sistemas de computação e similares.

2.2.3 Filas Não Markovianas

Uma extensão muito natural para o modelo de perda de Erlang são as filas $M/G/c/c$, em que a distribuição do tempo de serviço seja geral dependente de estado. O foco desta sessão é, então, derivar um modelo analítico para filas do tipo $M/G/c/c$ dependentes do

estado, isto é, em que as chegadas seguem um processo de Poisson, a taxa de serviço é geral dependente do estado do sistema e o sistema é de perdas (isto é, como $c=c$, não há espaço para formação de filas de espera). O conceito de cadeia reversa será aplicado na derivação das equações correspondentes a este tipo de fila. Em seguida, o modelo será aplicado ao estudo de tráfegos sujeitos a congestionamento (isto é, que apresentam redução na taxa de serviço, com o aumento do número de usuários no sistema).

Para aplicar a aproximação de cadeia reversa ao modelar um sistema, é necessário o seguinte teorema:

Teorema 1 (Ross, 1996): *Sejam q_{ij} as taxas de transição de uma cadeia de Markov irreduzível com tempo contínuo. Se pudermos achar um conjunto de q_{ij}^* não negativos, $i, j \geq 0$, $i \neq j$, e outro conjunto de números não negativos p_i , $i \geq 0$, cuja soma seja igual à unidade, de maneira que:*

$$p_i q_{ij} = p_j q_{ji}^*, \quad i \neq j,$$

e

$$\sum_{i \neq j} q_{ij} = \sum_{i \neq j} q_{ij}^*, \quad i \geq 0,$$

então os q_{ij}^* são as taxas de transição para a cadeia reversa e os p_i são as probabilidades limite para ambas as cadeias.

Ross (1996) aplicou o conceito de cadeia reversa para modelar sistemas de perda de Erlang, bem como sistemas de processadores compartilhados $M/G/1$, e descobriu que a distribuição limite do número de usuários em ambos os sistemas depende de G (a distribuição geral dos tempos de serviço) somente através de suas médias, que é dada por:

$$p_n = \frac{[\lambda E(S)]^n}{1 + \sum_{i=1}^k \frac{[\lambda E(S)]^i}{i!}}, \quad (5)$$

em que $n=1,2,\dots,k$, e

$$p_n = [\lambda E(S)]^n [1 - \lambda E(S)], \quad (6)$$

para $n \geq 0$, em que $E(S) = \int \tilde{G}(x) dx$.

A aproximação de Ross (1996) será usada para analisar filas $M/G/c/c$ dependentes do estado. Suponha que usuários cheguem de acordo com um processo de Poisson com taxa λ . Qualquer chegada que encontre todos os c servidores ocupados não entra e é desconsiderada (perdida) pelo sistema. Assumimos que os tempos de serviços dos usuários são distribuídos de acordo com uma distribuição geral G dependente do número de usuários no sistema. Assim, dado que existem n pessoas no sistema, cada servidor processa o trabalho a uma taxa $f(n)$. Em outras palavras, se existir uma chegada, a taxa de serviço mudará para $f(n+1)$. Se existir uma partida, a taxa de serviço mudará para $f(n-1)$.

Supomos que G é uma distribuição contínua que tem densidade g e taxa de falha $\mu(t) = g(t)/\bar{G}(t)$. Isto é, $\mu(t)$ é a probabilidade instantânea de que o serviço das t unidades antigas seja concluído.

Fazendo o estado a qualquer instante ser a quantidade de serviço já prestado aos usuários que ainda estão no sistema, o estado será $\bar{x} = (x_1, x_2, \dots, x_n)$, $x_1 \leq x_2 \leq \dots \leq x_n$. Se existem n usuários no sistema ($n \leq c$) e x_1, x_2, \dots, x_n é a quantidade de serviço já prestado a estes usuários, o processo de estados sucessivos será um processo de Markov no sentido que a distribuição condicional de qualquer estado futuro somente dependerá do estado presente.

Assumimos primeiro uma cadeia inversa e as probabilidades limites, para então usarmos o Teorema 1 para validar nossa suposição. Faça $P(\bar{x})$ denotar a densidade de probabilidade limite e p_0 denotar a probabilidade limite que o sistema esteja vazio.

Conjectura 1 (Cheah, 1990): *O processo reverso é do mesmo tipo que o processo direto, com usuários chegando por um processo de Poisson com taxa λ , tendo as cargas de tarefas distribuídas de acordo com G e com os estados representando as cargas residuais de serviço dos usuários presentes no sistema.*

Prova: Para verificar a Conjectura 1 e ao mesmo tempo obter as probabilidades limites, façamos $e_i(\bar{x}) = (x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_{n-1}, x_n)$, em que $\bar{x} = (x_1, x_2, \dots, x_n)$, $x_1 \leq x_2 \leq \dots \leq x_n$. Note que o sistema passa do estado \bar{x} para um estado $e_i(\bar{x})$ com intensidade de probabilidade $\mu(x_i)f(n)$ e num sentido inverso, o sistema passa do estado $e_i(\bar{x})$ para o estado \bar{x} com intensidade de probabilidade (conjunta) $\lambda g(x_i)$. Se $P(\bar{x})$ é a densidade limite, então de acordo com o Teorema 1:

$$P(\bar{x})\mu(x_i)f(n) = P[e_i(\bar{x})]\lambda g(x_i),$$

$$P(\bar{x}) = \frac{P[e_i(\bar{x})]\lambda g(x_i)}{\mu(x_i)f(n)} = \frac{P[e_i(\bar{x})]\lambda g(x_i)}{\frac{g(x_i)}{G(x_i)}f(n)} = \frac{\lambda P[e_i(\bar{x})]G(x_i)}{f(n)}.$$

Fazendo $i = 1$ e integrando sobre todo o vetor \bar{x} obtemos

$$p_n = \frac{[\lambda E(S)]^n}{n!f(n)\dots f(2)f(1)} p_0, \quad (7)$$

em que $n = 1, 2, \dots, c$ e $p_0 = \left[1 + \sum_{i=1}^c [\lambda E(S)]^i / i!f(i)\dots f(2)f(1) \right]$. Estes resultados estão baseados na suposição de que a hipótese seja válida. Para completar a prova, devemos verificar que

$$P(\bar{x})\lambda = P(0, \bar{x})f(n+1)$$

é a equação pertinente, desde que o processo reverso, quando no estado (ε, \bar{x}) , irá para o estado \bar{x} no instante $\varepsilon/f(n+1)$.

$$P(\bar{x})\lambda = P(0, \bar{x})f(n+1)$$

$$\frac{\lambda^n \prod_{i=1}^n \bar{G}(x_i)}{f(n)\dots f(2)f(1)} \lambda = \frac{\lambda^{n+1} \prod_{i=1}^{n+1} \bar{G}(x_i)}{f(n+1)\dots f(2)f(1)} f(n+1)$$

Desde que $\bar{G}(0) = 1$,

$$\frac{\lambda^{n+1} \prod_{i=1}^n \bar{G}(x_i)}{f(n) \cdots f(2) f(1)} = \frac{\lambda^{n+1} \prod_{i=1}^n \bar{G}(x_i)}{f(n) \cdots f(2) f(1)}.$$

c.q.d.

Portanto $\lambda g(x_i)$ são as taxas de transição para a cadeia reversa e $P(\bar{x})$ são as probabilidades limites. Note que a distribuição limite do número de usuários no sistema depende de G somente através de sua média. Isto significa que filas $M/G/c/c$ e $M/M/c/c$ dependentes do estado são estocasticamente equivalentes se a carga de tarefas requerida pelas chegadas dos dois modelos é a mesma. Verificada a hipótese, temos o seguinte teorema:

Teorema 2 (Cheah & MacGregor Smith, 1994): Para o modelo de fila $M/G/c/c$ dependente de estado, o número de usuários no sistema tem distribuição:

$$p_n = \frac{[\lambda E(S)]^n}{n! f(n) \cdots f(2) f(1)} p_0, \quad (8)$$

em que $n=1,2,\dots,c$, e $p_0^{-1} = \left[1 + \sum_{i=1}^c [\lambda E(S)]^i / [i! f(i) \cdots f(2) f(1)] \right]$.

Também, o processo reverso de fila $M/G/c/c$ dependente de estado é do mesmo tipo que o processo normal, com usuários chegando por um processo de Poisson com taxa λ , tendo carga de tarefas distribuídas de acordo com G e com o estado representando a carga residual de tarefas dos usuários presentes no sistema.

Corolário 1 (Cheah, 1990): No modelo $M/G/c/c$ dependente de Estado, o processo de partida (incluindo ambos os usuários que completaram o serviço e aqueles que foram perdidos) é um processo de Poisson com taxa λ .

Prova (Cheah, 1990): Segue de que o processo reverso de chegadas de todos os usuários (incluindo aqueles perdidos) constitui um processo de Poisson.

c.q.d.

2.3 Modelo Analítico para Tráfegos com Congestionamento

Para modelar o tráfego em uma via de acesso, para automóveis ou pedestres, devemos tratar a sua capacidade como a de um sistema em que o número de servidores é c . Além disso, devemos assumir chegadas de Poisson e taxas de serviço dependentes do número de usuários.

Se deixarmos a via comportar-se como um sistema com c servidores para seus ocupantes e se evitarmos que uma chegada encontre todos os c servidores ocupados, podemos modelá-la como um fila tendo c servidores e capacidades c com taxas de serviço dependentes do estado. Este modelo pode então ser descrito como uma fila $M/G/c/c$ dependente do estado.

Com as filas $M/G/c/c$ dependente do estado, descritas na seção anterior, segue que a probabilidade de se ter n usuários em uma via pode ser encontrada pela seguinte expressão:

$$p_n = \frac{[\lambda E(S)]^n}{n! f(n) \cdots f(2) f(1)} p_0, \quad n=1,2,\dots,c, \quad (9)$$

em que $p_0^{-1} = \left[1 + \sum_{i=1}^c [\lambda E(S)]^i / [i! f(i) \cdots f(2) f(1)] \right]$ e $f(n) = V_n / V_1$.

Pode-se observar que a distribuição limite do número de usuários na via de acesso dependerá da média do tempo esperado de serviço para um único ocupante, $E(S)$. A taxa de serviço de cada servidor será a função $f(n) = V_n / V_1$, em que V_n é a função do fluxo dada pelo modelo de congestionamento a ser desenvolvido na próxima seção.

2.3.1 Modelo de Congestionamento para Tráfego de Pedestres

Nesta seção, por comodidade particularizaremos o modelo $M/G/c/c$ para o caso de fluxo de pedestres. Entretanto, resultados bastante satisfatórios já foram obtidos com a utilização do modelo na análise de tráfego de veículos (Jain & MacGregor, 1997).

Como suposições para o modelo tem-se que a chegada de pedestres segue um processo de Poisson com taxa λ . A densidade do tráfego determina a velocidade média dos pedestres, o que nos leva a um modelo de taxa de serviço dependente do estado. A densidade do tráfego, em um sistema de circulação, é assumida ser aproximadamente uniformemente distribuída no sistema.

Como já mencionado aqui, existem muitos fatores, tangíveis ou intangíveis, que podem afetar a velocidade de locomoção de um pedestre, por exemplo, a forma, a cor e a iluminação do corredor, o padrão do material do piso, bagagem, a idade dos pedestres *etc.* Entretanto, como hipótese, a densidade do tráfego será usada como determinante principal da velocidade de um pedestre. A velocidade média de locomoção como uma função do número de pedestres será uma variável fundamental. Além disso, o fluxo de tráfego em um corredor simples será o foco central.

Para um corredor de comprimento l e largura constante w , está claro que existe um limite para o número de pedestres que podem mover-se através dele num dado momento. Referência será feita a este limite superior como a capacidade c do corredor. De acordo com Tregenza (1976), o fluxo de pedestres pára quando a densidade populacional se aproxima de 5 ped/m². Assim, a capacidade do corredor será expressa como:

$$c = \lfloor 5 \times l \times w \rfloor,$$

em que l e w são as dimensões de um corredor, comprimento e largura, em metros e $\lfloor x \rfloor$ é o maior inteiro não superior a x .

De acordo com Fruin (1971), a densidade do tráfego tem um impacto relativamente pequeno na velocidade de locomoção individual até ser atingida uma ocupação média de 0,27 ped/m². Tregenza (1976), por outro lado, encontrou que uma velocidade típica para um ocupante sozinho em um corredor é de 1,5 m/s. Modelos para velocidade de locomoção unidirecional, linear e exponencial, foram propostos por Yuhaski e MacGregor Smith (1989). O modelo linear é expresso por:

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

e o modelo exponencial por

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

em que

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

$$\beta = (a-1)/\log(V_1/V_a)^{\gamma} = (b-1)/\log(V_1/V_b)^{\gamma}, \quad (13)$$

$V_1=1,5$ m/s é a velocidade típica para um ocupante sozinho no sistema, $V_a=0,64$ m/s é a velocidade média para uma densidade de 2 ped/m² (isto é, $a=2lw$ pedestres) e $V_b=0,25$ m/s é a velocidade média para uma densidade de 4 ped/m² (isto é, $b=4lw$ pedestres).

A partir do gráfico empírico de Tregenza (1976) apresentado na Figura 1.1, parece que o modelo exponencial é melhor para descrever o comportamento da velocidade de locomoção de pedestres. Embora as relações empíricas de Fruin (1974) estejam em termos de volume de pedestre, pode-se obter a velocidade de locomoção, uma vez que o volume de pedestres é igual à velocidade média, em pés/min, multiplicada pela densidade, dada em ped/pés². As relações empíricas de Fruin (1971) para o fluxo de tráfegos bi e multidirecionais poderiam ser utilizados para desenvolver modelos adequados a novas situações, conforme as apresentadas na Figura 2.5.

Fruin (1971) constatou via experimentos que o tráfego é paralisado aproximadamente a 3,5 ped/m², ao passo que Tregenza (1976) conclui que o tráfego paralisa aproximadamente a 5 ped/m². Embora os referidos autores não concordem na densidade exata para a qual o tráfego paralisa, isto não é essencial desde que as pesquisas foram conduzidas sob diferentes condições e com diferentes populações. Os resultados são somente aproximações. As relações entre diferentes fluxos são de maior interesse.

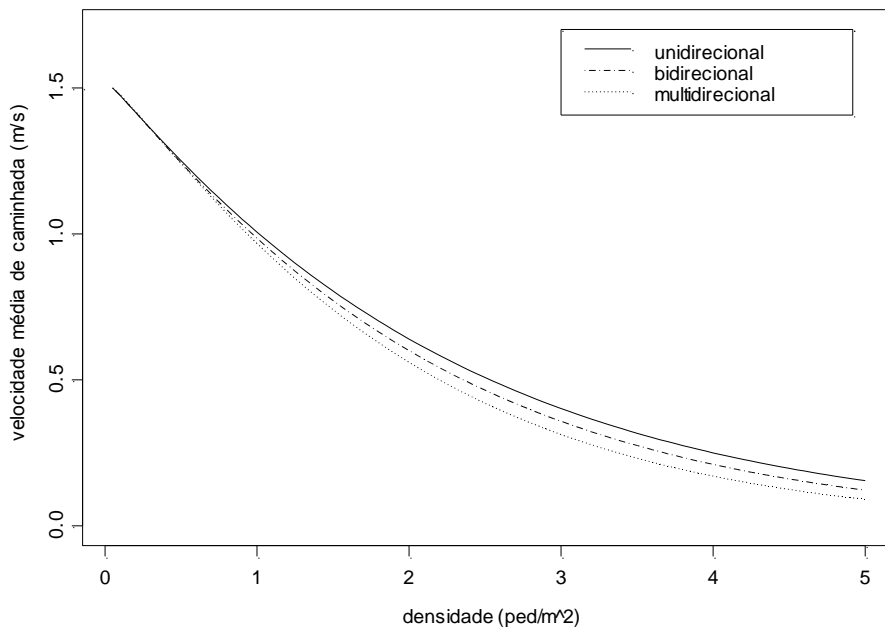


Figura 2.5: Modelos para fluxos uni, bi e multidirecionais.

Como observação final, uma vez que filas $M/G/c/c$ e $M/M/c/c$ dependente do estado são estocasticamente equivalentes, o fluxo do tráfego modelado como uma fila $M/M/c/c$ dependente do estado, desenvolvido por Yuhaski e MacGregor Smith (1989), é:

$$p_n = \frac{\lambda^n p_0}{\mu_1 \mu_2 \cdots \mu_n} \quad (14)$$

em que $\mu_i = i \times (1,5/l) \exp[-((i-1)/\beta)^\gamma]$ e $p_0^{-1} = \left[1 + \sum_{n=1}^c \lambda^n / (\mu_1 \cdots \mu_n) \right]$. Note que (14) é equivalente ao modelo analítico $M/G/c/c$ dependente do estado definido por (9), fazendo-se $E(S)=l/1,5$ e $f(n)=V_n/V_1$.

2.4 Experimentos Computacionais

Utilizando-se (9) e (14) é possível calcular medidas de desempenho para filas $M/G/c/c$ e $M/M/c/c$ dependentes do estado. Essas medidas incluem, entre outras, a probabilidade de bloqueio, a taxa de chegada efetiva (ou taxa de atendimento, no inglês, *throughput*), o número médio de usuários no sistema (*work-in-process*) e o tempo médio no sistema. O bloqueio ocorre quando n for igual à capacidade do sistema c :

$$P[n = c] = p_c. \quad (15)$$

A taxa de atendimento, também conhecida como taxa de chegada efetiva ao sistema, pode ser calculada em função da taxa de chegada λ :

$$\theta = \lambda(1 - p_c). \quad (16)$$

O número médio de usuários no sistema segue da definição de esperança:

$$L = E[N] = \sum_{n=1}^c n p_n. \quad (17)$$

O tempo médio no sistema pode ser dado pela lei de Little:

$$W = E[T] = \frac{L}{\theta}. \quad (18)$$

A Tabela 2.1 e a Figura 2.6 apresentam uma comparação destas medidas de desempenho, em um corredor para pedestres de $8 \times 2,5 \text{ m}^2$ (isto é, com capacidade igual a 100), dadas pelos modelos de filas $M/G/c/c$ e $M/M/c/c$ dependentes de estado. Podemos observar pelos resultados obtidos que as probabilidades de bloqueio, as taxas de chegadas efetivas (taxas de atendimento), os números médios de usuários no sistema e os tempos médios nos sistemas são idênticos, para uma mesma taxa de chegada, para os sistemas $M/G/c/c$ e $M/M/c/c$, comprovando a equivalência estocástica entre os dois sistemas.

O comportamento dos sistemas está de acordo com o esperado. Para taxas de chegada baixas a probabilidade de bloqueio também é baixa e a taxa de chegada efetiva cresce linearmente com a taxa de chegada, até $\lambda \approx 2,7$ ped/s, o que aparenta ser o limite deste corredor. A partir deste ponto, com o aumento da taxa de chegada, a taxa de bloqueio aumenta, a taxa de chegada efetiva diminui, o número médio de usuários aumenta, até a capacidade do sistema (isto é, 100 pedestres), e o tempo médio no sistema também aumenta, atingindo um valor máximo ($w \approx 51$ s), próximo a dez vezes o tempo médio para um único ocupante (isto é, $E(S) = 8/1,5 \approx 5,3$ s). Este ponto de transição pode ser facilmente identificado nos gráficos da Figura 2.6.

Tabela 2.1: Comparação das medidas de desempenho pelos modelos *M/M/c/c* e *M/G/c/c*.

λ	Medida	<i>M/M/c/c</i>	<i>M/G/c/c</i>		Erro
1,667	p_c	0,000000	0,000000	<	1,00E-07
	θ	1,667000	1,667000	<	1,00E-07
	L	11,23229	11,23229	<	1,00E-06
	W	6,738024	6,738024	<	1,00E-07
2,000	p_c	0,000000	0,000000	<	1,00E-07
	θ	2,000000	2,000000	<	1,00E-07
	L	14,48751	14,48751	<	1,00E-06
	W	7,243754	7,243754	<	1,00E-07
2,667	p_c	0,005950	0,005950	<	1,00E-07
	θ	2,651131	2,651131	<	1,00E-07
	L	26,06751	26,06751	<	1,00E-06
	W	9,832599	9,832599	<	1,00E-07
3,333	p_c	0,407453	0,407453	<	1,00E-07
	θ	1,974961	1,974961	<	1,00E-07
	L	98,45008	98,45008	<	1,00E-06
	W	49,84913	49,84913	<	1,00E-06
4,000	p_c	0,510172	0,510172	<	1,00E-07
	θ	1,959314	1,959314	<	1,00E-07
	L	99,01137	99,01137	<	1,00E-06
	W	50,53370	50,53370	<	1,00E-06
5,000	p_c	0,610020	0,610020	<	1,00E-07
	θ	1,949898	1,949898	<	1,00E-07
	L	99,35067	99,35067	<	1,00E-06
	W	50,95173	50,95173	<	1,00E-06
10,00	p_c	0,806137	0,806137	<	1,00E-07
	θ	1,938629	1,938629	<	1,00E-07
	L	99,75848	99,75848	<	1,00E-06
	W	51,45826	51,45826	<	1,00E-06

Legenda:

λ = taxa de chegada (pedestres/s)

p_c = probabilidade de bloqueio

θ = taxa de atendimento (pedestres/s)

L = número médio de usuários no sistema (pedestres)

W = tempo médio no sistema (s)

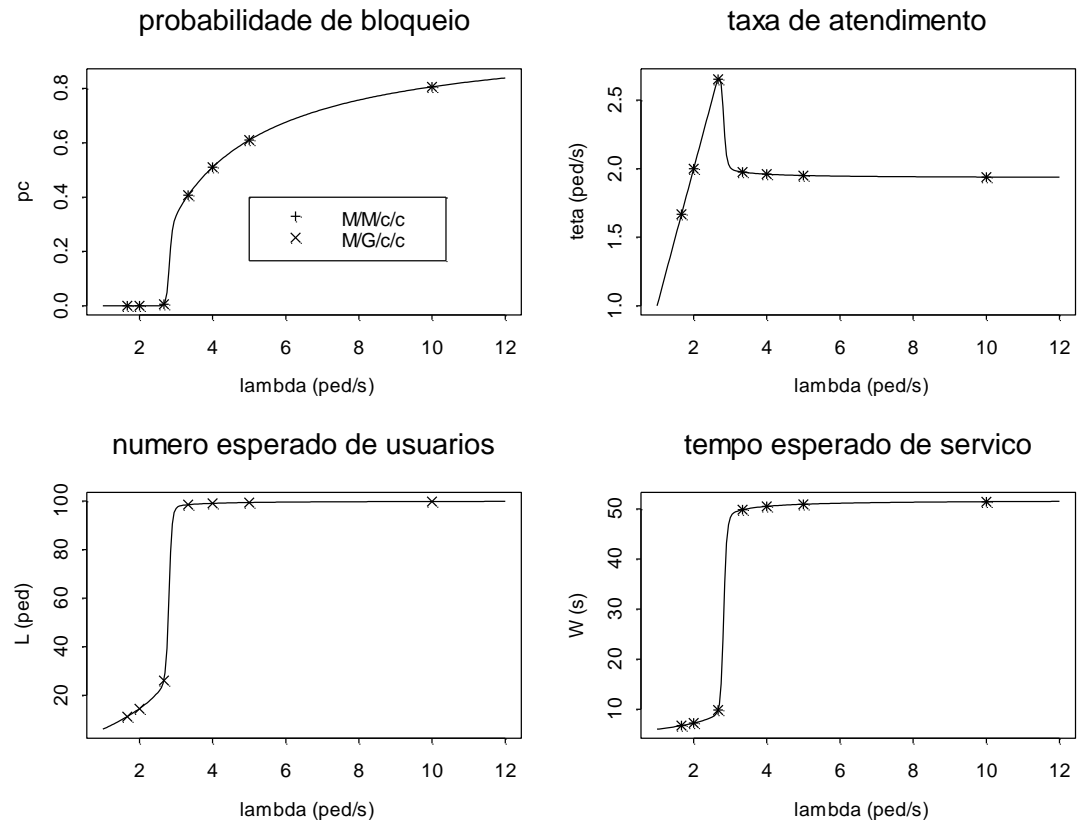


Figura 2.6: Comparação das medidas de desempenho pelos modelos $M/M/c/c$ e $M/G/c/c$.

2.5 Principais Contribuições deste Capítulo

As principais contribuições deste capítulo incluem a apresentação detalhada das propriedades estocásticas das filas $M/G/c/c$ dependentes do estado, a discussão detalhada da aplicabilidade dos modelos $M/G/c/c$ na análise de tráfegos sujeitos ao fenômeno de congestionamento (isto é, sujeitos ao decrescimento da taxa de serviço, com o aumento do número de usuários no sistema), a implementação computacional dos modelos (9) e (14) e a apresentação de experimentos computacionais extensivos, que demonstram empiricamente a equivalência estocástica entre os modelos $M/G/c/c$ e $M/M/c/c$ dependentes do estado.

Capítulo 3

Sobre Filas M/G/c/c Configuradas em Redes

Muitos dos resultados a serem apresentados neste capítulo foram apresentados no artigo de Araújo et al. (2004), publicado recentemente no XV Congresso Brasileiro de Automática.

3.1 Introdução

O estudo do fluxo de pedestres através dos modelos desenvolvidos por Yuhaski e MacGregor Smith (1989), restrito a apenas em um único corredor, esconderia a enorme capacidade de tais modelos em retratar muitas outras situações de interesse. Como mencionado, existem várias aplicações em que a teoria de filas pode ser um instrumento de grande eficácia, tanto para uma melhor compreensão (descrição) como para o aperfeiçoamento (prescrição) do sistema em estudo.

Não depreciando a importância de um sistema de corredor único, é óbvio que sistemas como rotas de emergência, telecomunicações, processamento de dados, fluxo de veículos e outros, são mais complexos. Estes sistemas podem ser imaginados com a união de filas únicas agrupadas convenientemente (rede de filas), com o objetivo de fornecer uma representação que melhor assemelhe-se ao sistema que desejamos estudar. A união de filas únicas forma três configurações básicas, nomeadas topologia em série, junção e divisão, conforme apresentado nas Figuras 3.1, 3.2 e 3.3.

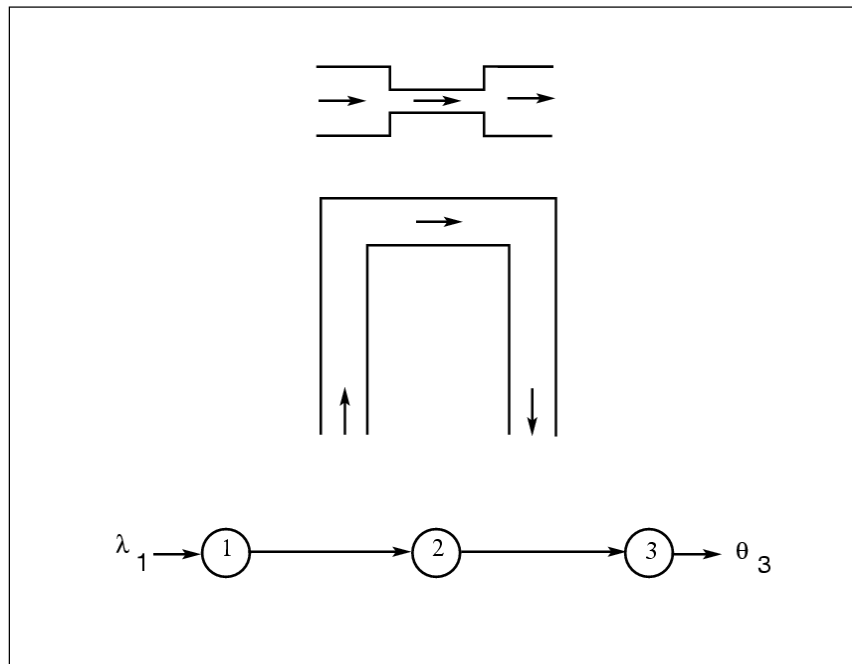


Figura 3.1: Rede configurada em topologia série.

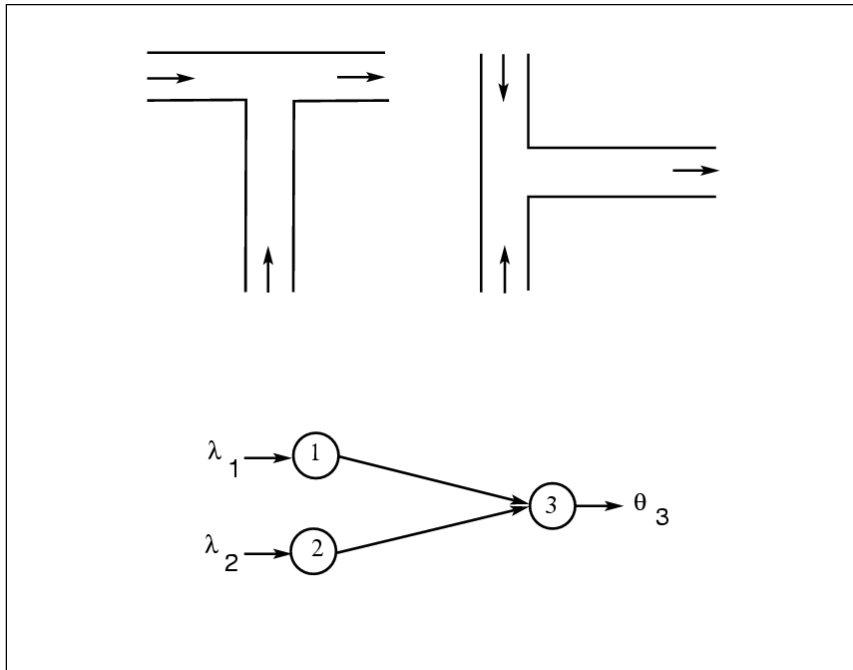


Figura 3.2: Rede configurada em topologia junção.

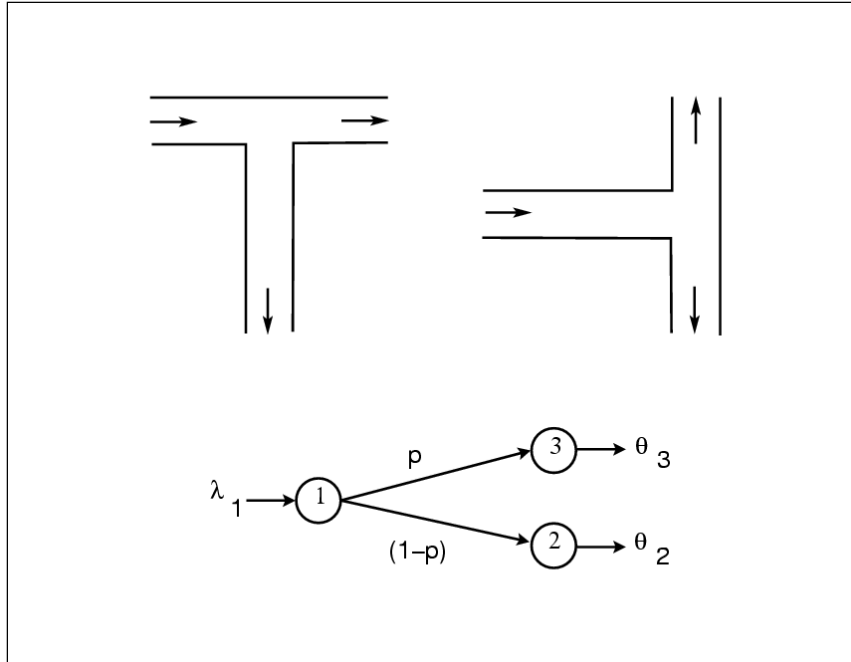


Figura 3.3: Rede configurada em topologia divisão.

Certamente, o problema de determinação das medidas de desempenho fica bem mais complexo, se estamos interessados em filas configuradas em rede. É fácil concluir que a probabilidade de bloqueio na i -ésima fila de uma rede depende de todo o fluxo anterior

e também das probabilidades de bloqueio das filas seguintes. Um usuário do sistema pode ser bloqueado imediatamente após ser atendido em um nó, pois o nó seguinte pode encontrar-se em sua capacidade máxima, criando assim uma situação de difícil controle. Métodos aproximados são considerados uma boa alternativa para tratar estes sistemas em redes. Passemos à apresentação de um método bastante popular.

3.2 Método da Expansão Generalizado para Filas M/G/c/c

O método da expansão generalizado (GEM) foi proposto por Kerbache & MacGregor Smith (1987), como forma de determinar aproximadamente medidas de desempenho de sistemas de filas finitas configuradas em redes. O GEM é uma combinação de tentativas repetidas e métodos de aproximação por decomposição nó-a-nó, com uma característica fundamental de que um nó artificial é adicionado precedendo cada fila finita na rede, com o objetivo de registrar usuários bloqueados que tentam entrar em um nó finito quando sua capacidade está esgotada (vide Figura 3.2). Pela adição de nós de bloqueio, a rede de filas é “expandida” em uma rede de Jackson equivalente na qual cada nó pode ser decomposto e analisado separadamente. A Figure 3.2 ilustra um exemplo de corredores configurados em série e a rede de filas correspondente.

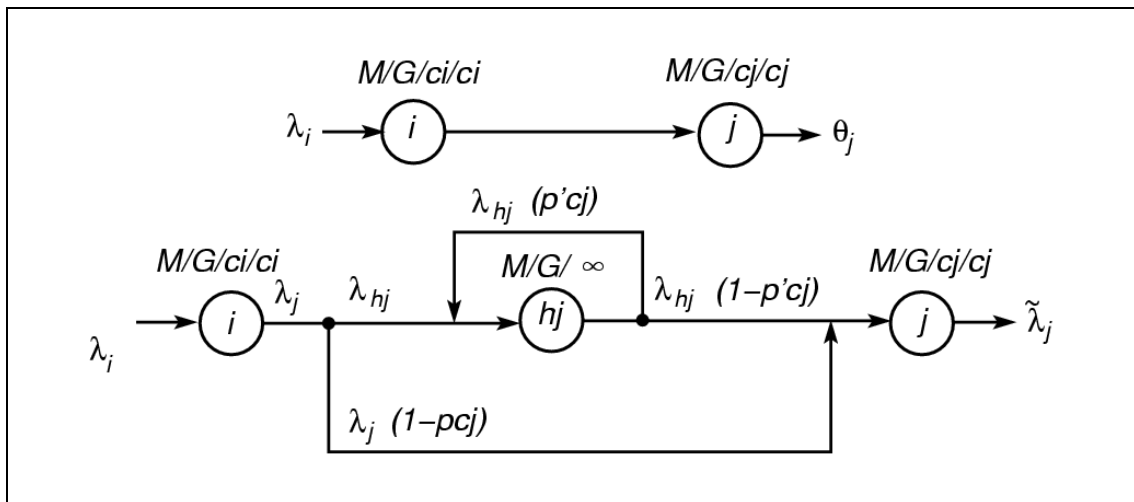


Figura 3.2: Método da expansão generalizado.

O GEM consiste em três fases que são executadas para cada nó finito na rede de filas original.

Fase 1: Reconfiguração da Rede

Para cada nó com capacidade finita, um nó artificial é adicionado imediatamente antes, como mostrado em Figura 3.2. Usuários que tentam mover-se para um novo nó, mas são barrados devido à capacidade desde nó estar esgotada, são redirecionados ao nó artificial. A probabilidade que um usuário recém chegado seja bloqueado pelo nó j é igual a p_{c_j} . Assim, com probabilidade $(1 - p_{c_j})$, entrará no nó j , e com probabilidade p_{c_j} entrará no nó de bloqueio h_j . O nó de bloqueio é modelado como uma fila $M/G/\infty$, de forma que não haverá nenhuma espera para entrar nele.

Atendido no nó de bloqueio, o usuário poderá ser novamente bloqueado, com uma nova probabilidade p'_{c_j} . Com probabilidade $(1 - p'_{c_j})$, prosseguirá ao nó seguinte. Caso contrário, deverá ser redirecionado novamente pelo laço de realimentação ao nó artificial h_j .

Fase 2: Estimação de parâmetros

O valor de p_{c_j} pode ser determinado através resultados analíticos conhecidos. Para filas $M/G/c/c$ dependentes do estado, tal valor pode ser determinado diretamente por (9), i.e., $p_{c_j} = \Pr\{N = c_j\}$. O valor de p'_{c_j} é determinado de resultados de aproximações. Depois que o atendimento a um usuário no nó de bloqueio h_j estiver terminado, ele é forçado a retornar com probabilidade p'_{c_j} , para outro serviço de espera. Uma aproximação que usa técnicas de difusão é apresentada por Labetoulle & Pujolle (1980):

$$p'_{c_j} = \left\{ \frac{\mu_j + \mu_{h_j}}{\mu_{h_j}} - \frac{\lambda \left[(r_2^{c_j} - r_1^{c_j}) - (r_2^{c_j-1} - r_1^{c_j-1}) \right]}{\mu_{h_j} \left[(r_2^{c_j+1} - r_1^{c_j+1}) - (r_2^{c_j} - r_1^{c_j}) \right]} \right\}^{-1}, \quad (19)$$

em que r_1 e r_2 são as raízes para:

$$\lambda_{ext} - (\lambda_{ext} + \mu_{h_j} + \mu_j)x + \mu_{h_j}x^2 = 0. \quad (20)$$

Definida com ajuda de Figura 3.3, a taxa de chegada externa λ_{ext} , usada em (20), é:

$$\begin{cases} \lambda_{ext} = \tilde{\lambda}_j - \lambda_{h_j} (1 - p'_{c_j}), \\ \tilde{\lambda}_j = \lambda_j (1 - p_{c_j}), \\ \lambda_{h_j} = \lambda_j (p_{c_j}), \\ \lambda_j = \lambda_i (1 - p_{c_i}) = \tilde{\lambda}_i, \end{cases} \quad (21)$$

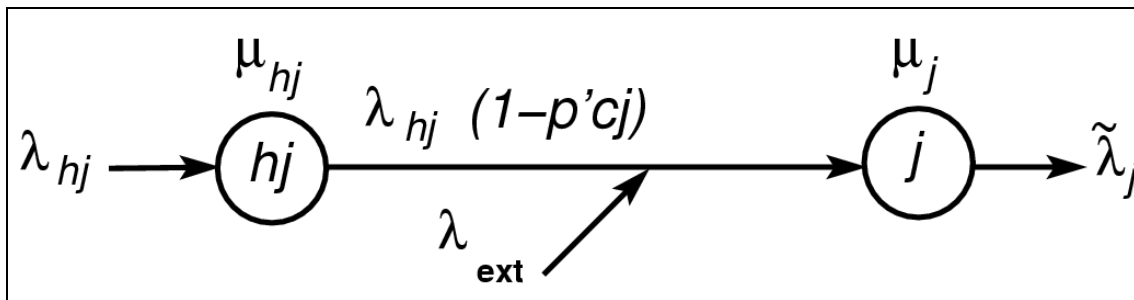


Figura 3.3: Definição da taxa de chegada externa λ_{ext} .

Usando a teoria da renovação, pode ser mostrado que a taxa de serviço do nó de bloqueio é, no caso exponencial, como segue (Kleinrock, 1975):

$$\mu_{h_j} = \frac{2\mu_j}{1 + \sigma_j^2 \mu_j^2}, \quad (22)$$

em que σ_j^2 é a variância do tempo de serviço. Porém, desde que a taxa de serviço é dependente do estado para o sistema de fila que esta sendo estudado, uma suposição razoável é considerar o pior caso:

$$\mu_{h_j} = \mu_j \approx \frac{c_j}{E[T_1]/f(c_j)}, \quad (23)$$

em que c_j é o número máximo de servidores em paralelo e $E[T_1]/f(c_j)$ é o tempo de atendimento para c_j ocupantes.

Fase 3: Eliminação da Realimentação e Atualização da Taxa de Serviço

Uma reconfiguração do nó de bloqueio é executada, de forma que as fortes dependências causadas nos processos de chegada devido às repetidas visitas ao nó artificial sejam removidas. O arco de realimentação é removido do nó de bloqueio através da atualização da taxa de serviço deste nó como segue:

$$\mu'_h = (1 - p'_{c_j}) \mu_{h_j}. \quad (24)$$

Finalmente, o tempo de serviço médio que um usuário despense no nó i que precede o nó j é determinado por:

$$\tilde{\mu}_i^{-1} = \mu_i^{-1} + p_{c_j} (\mu'_h)^{-1}. \quad (25)$$

A atualização apresentada em (25) representa o passo final do GEM, no qual a principal meta é prover um esquema de aproximação para atualizar a taxa de serviço dos nós que precedem um nó de capacidade finita, para compensar o fenômeno de bloqueio após serviço.

O GEM em Topologias Gerais

As topologias junção e divisão também podem ser tratadas pelo GEM. Em topologias divisão, as probabilidades do roteamento precisam ser conhecidas, então, o processamento tem apenas que ser decomposto de acordo com as probabilidades de ramificação. Mudanças semelhantes devem ser feitas nos casos de topologias junção. Em tais casos, a taxa de chegada para o nó de junção é a soma do processamento dos nós prévios.

Concluimos que qualquer topologia de rede pode ser analisada, uma vez que se saiba como aplicar o GEM a estas topologias básicas. A seguir será descrita uma nova implementação computacional para (25), desenvolvida especialmente para redes de filas $M/G/c/c$ dependentes do estado (Araújo et al., 2004).

3.4 Uma Nova Implementação para o GEM

Seja a rede sob estudo definida em um grafo $G(V,A)$, em que V é um conjunto finito de nós (corredores) e A é um conjunto finito de arcos (conexões entre pares de corredores). A implementação proposta, apresentada na Figura 3.4, é baseada no fato de que bloqueios nos nós a jusante causam bloqueios após serviço nos nós a montante. Como resultado, as taxas de serviço efetivas nos nós a montante devem ser reduzidas, de modo semelhante aos sistemas markovianos, de acordo com (25). Em redes de filas $M/G/c/c$ dependentes do estado, um efeito semelhante deve estar presente, indiferentemente destas apresentarem mudanças dinâmicas nas taxas de serviço, com usuários entrando

ou deixando o sistema. Assim, o algoritmo ajusta a curva de taxa de serviço dependente do estado progressivamente. Os ajustes são feitos simplesmente considerando que a conservação do fluxo ao longo das redes é uma condição de contorno que deve ser satisfeita.

Assim, o algoritmo para avaliação de desempenho da Figura 3.4 é composto por duas partes. Primeiro, uma pré-avaliação é executada a todo nó da rede, desconsiderando neste primeiro momento qualquer possível efeito de bloqueio entre nós. Em outras palavras, o procedimento de pré-avaliação aproxima a rede de filas finitas real por uma rede de Jackson (isto é, rede de filas infinitas sem interferência), em que cada nó pode ser analisado separadamente. Note que o procedimento de pré-avaliação é uma variante do algoritmo de Dijkstra (1959) para determinação de caminhos mínimos em grafos. Note também que sob tráfegos muito leves (isto é, com efeitos desprezíveis de bloqueio entre nós), o procedimento de pré-avaliação por si só fornecerá uma aproximação bastante precisa das todas as medidas de desempenho.

Por exemplo, na rede ilustrada na Figura 3.5, uma possível seqüência válida para executar a pré-avaliações é $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 6$. A razão de um nó só poder ser pré-avaliado após todos os seus antecessores é o fato de que é necessário conhecer a taxa de chegada global a este nó antes de proceder com qualquer cálculo e, obviamente, a taxa de chegada total é uma função das saídas de todos os nós antecessores.

A segunda parte do algoritmo avalia as medidas de desempenho, mas agora buscando a conservação do fluxo. Por conservação de fluxo quer-se dizer que $\theta_{k_1} \leq \theta_{k_2} \leq \dots \leq \theta_{k_n}$ para todo caminho distinto (k_1, k_2, \dots, k_n) , de uma origem k_1 até um destino k_n , na rede de filas. As estimativas são melhoradas por meio de ajustes no tempo esperado de serviço para um usuário único em cada nó k , $E[T_1]_k$. Para descrever o procedimento, seja θ_k^{\max} o máximo fluxo possível através do nó k , isto é, o fluxo máximo que pode ser remetido ao conjunto de sucessores de k . Assim, se k é um nó terminal (isto é, não possuir nenhum sucessor), obviamente $\theta_k^{\max} = \infty$. Caso contrário, a atualização de θ_k^{\max} dependerá do conhecimento da capacidade de processamento de todos os sucessores.

Inicialmente, é assumido que todos os nós são ilimitados e θ_k^{\max} é fixado como infinito. A partir dos nós finais, $E[T_1]_k$ é minimizada, sujeita a prover um processamento menor ou igual a θ_k^{\max} e não sendo inferior a l_k/V_1 , em que l_k é o comprimento do k -ésimo corredor e V_1 , a velocidade para um único usuário ($V_1 \approx 1,5$ m/s, para a aplicação de pedestres). Então, todas as medidas de desempenho de k são atualizadas, considerando-se a nova $E[T_1]_k$. Finalmente, θ_k^{\max} é atualizado para o conjunto de antecessores de k . A atualização é realizada atribuindo-se aos antecessores uma parte igual de θ_k . Se algum antecessor não puder prover sua parte, o fluxo é redirecionado ao outro antecessor, assegurando a conservação do fluxo.

Note que, para a rede apresentada na Figura 3.5, uma possível sucessão válida para executar as avaliações é $6 \rightarrow 4 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3$, uma vez que um nó só pode ser avaliado quando todos seus antecessores já tiverem sido previamente avaliados. Como observação final, ressalte-se que o procedimento de avaliação foi projetado para corrigir estimativas em casos de gargalos nos nós à jusante. Assim, o efeito da etapa de avaliação do algoritmo, naquelas redes submetidas a tráfego leves e moderados e sem quaisquer gargalos, é mínimo.

```

algorithm
  read  $G(V, A)$  and arrival rates,  $\lambda_i, \forall i \in V$ 
  read routing probabilities,  $p_{ij}, (i, j) \in A$ 
  /* pre-evaluate every node */
   $P \leftarrow \emptyset$ 
  while  $P \neq V$ 
    choose  $k \in (V \setminus P)$ 
    if  $\{i | (i, k) \in A\} \subseteq P$  then
      /* compute performance measures */
       $E[T_1]_k \leftarrow l_k/V_1$ 
      compute  $\Pr\{N = c_k\}, \theta_k, L_k, W_k$ 
       $P \leftarrow P \cup \{k\}$ 
      /* forward information */
       $\lambda_j \leftarrow \lambda_j + p_{kj}\theta_k, \forall (k, j) \in A$ 
    end if
  end while
to be continued ...

```

a) pré-avaliação

```

algorithm (continued)
  /* evaluate every node */
   $P \leftarrow \emptyset$ 
   $\theta_i^{\max} \leftarrow \infty, \forall i \in V$ 
  while  $P \neq V$ 
    choose  $k \in (V \setminus P)$ 
    if  $\{j | (k, j) \in A\} \subseteq P$  then
      /* update performance measures */
       $E[T_1]_k \leftarrow \min E[T_1]_k$ 
      s.t.:
       $\theta_k \leq \theta_k^{\max},$ 
       $E[T_1]_k \geq l_k/V_1$ 
      compute  $\Pr\{N = c_k\}, \theta_k, L_k, W_k$ 
       $P \leftarrow P \cup \{k\}$ 
      /* back-propagate information */
      update  $\theta_i^{\max}, \forall (i, k) \in A$ 
    end if
  end while
  write  $\Pr\{N = c_i\}, \theta_i, L_i, W_i, \forall i \in V$ 
end algorithm

```

b) avaliação

Figura 3.4: Algoritmo de avaliação de desempenho.

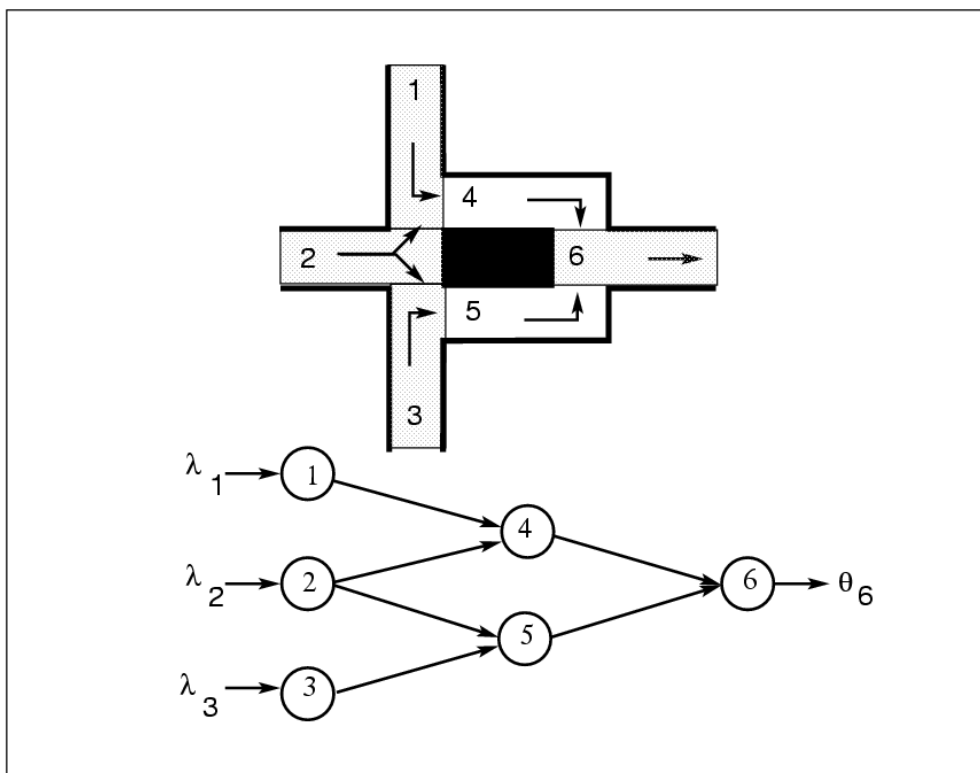


Figura 3.5: Corredores em topologia mista.

3.5 Experimentos Computacionais

Para avaliar a precisão das medidas de desempenho obtidas através desta metodologia, são apresentados os resultados de experimentos computacionais. Todas as experiências foram realizadas em um PC, com CPU Pentium II 400 MHz, 256 MB RAM e sistema operacional Windows NT 4.0. Para nossa conveniência, apenas o modelo exponencial de congestionamento (11), proposto por Yuhaski e MacGregor Smith (1989), foi utilizado. Para verificação da exatidão dos resultados, foi utilizado um modelo digital de simulação a eventos discretos (Cruz et al., 2005a), cuja descrição detalhada adiaremos até o próximo capítulo.

A Tabela 3.1. apresenta os resultados para uma versão de três nós (Figura 3.6) do sistema de seis nós apresentado na Figura 3.5, em que o corredor #2 não recebe fluxos, os corredores #1 e #4 são fundidos em um único corredor #1 e os corredores #3 e #5 são fundidos em outro corredor #2. O nó #6 passou a ser indexado como nó #3. A Figura 3.6 ilustra a nova rede de três nós. É assumido que todos os nós possuem um comprimento de 8,5 m, os corredores fundidos #1 e #2 possuem uma largura de 2,4 m e o corredor #3 uma largura de apenas 1,2 m, o que o caracteriza como um gargalo.

Neste ponto podemos confirmar a precisão do novo modelo analítico em quase todas as situações testadas. Nos casos extremos de alto desequilíbrio entre λ_1 e λ_2 , o erro pode atingir 92% no número esperado de usuários L e nos tempos médios de serviço W . A situação foi intencional para ilustrar um cenário em que ocorre uma inadequação do modelo analítico. Porém nota-se que a taxa de chegada efetiva, medida de maior interesse, sempre é precisa. Observa-se também que a probabilidade de bloqueio no nó #3 é sempre elevada e que o método analítico tende a superestimar tal medida.

Tabela 3.1: Medidas de desempenho para rede de três nós configurados em junção.

λ_1	λ_2		nó 1				nó 2				nó 3				tempo simulação					
			anal.	aver.	[ic]	erro	anal.	aver.	[ic]	erro	anal.	aver.	[ic]	erro						
2.9	0.1	p_c	0,7050	0,7135	0,7130	0,7139	-1,2%	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0%	0,5267	0,4994	0,4994	0,4995	5,5%
		θ	0,8554	0,8302	0,8294	0,8310	3,0%	0,1000	0,1006	0,0998	0,1014	-0,6%	0,9554	0,9308	0,9308	0,9308	2,6%			
		L	101,6	101,5	101,5	101,5	0,1%	0,573	70,66	70,01	71,31	-99,2%	50,05	51,00	51,00	51,00	-1,9%			
		W	118,8	122,3	122,2	122,4	-2,9%	5,726	70,24	70,00	70,48	-91,8%	52,39	54,79	54,79	54,79	-4,4%	(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	-20,1%	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	1,9%	0,9428	0,9308	0,9308	0,9308	1,3%			
		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	-0,9%			
		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	-2,2%	(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	1,5%	0,6857	0,4996	0,4996	0,4996	37,2%			
		θ	0,4714	0,4537	0,4519	0,4556	3,9%	0,4714	0,4771	0,4752	0,4789	-1,2%	0,9428	0,9308	0,9308	0,9308	1,3%			
		L	101,7	101,6	101,5	101,7	0,1%	100,9	98,79	98,68	98,91	2,1%	50,53	51,00	51,00	51,00	-0,9%			
		W	215,8	223,9	222,9	224,8	-3,6%	214,0	207,1	206,3	207,9	3,3%	53,60	54,79	54,79	54,79	-2,2%	(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	-0,3%	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	0,5%	0,9428	0,9308	0,9308	0,9308	1,3%			
		L	101,6	101,2	101,2	101,3	0,4%	101,4	100,8	100,7	100,9	0,6%	50,53	51,00	51,00	51,00	-0,9%			
		W	215,6	219,2	218,4	220,1	-1,6%	215,1	215,0	214,2	215,8	0,0%	53,60	54,79	54,79	54,79	-2,2%	(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	-0,6%	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	1,3%	0,9428	0,9308	0,9308	0,9308	1,3%			
		L	101,5	101,1	101,0	101,2	0,4%	101,5	101,1	101,0	101,2	0,4%	50,53	51,00	51,00	51,00	-0,9%			
		W	215,4	217,2	216,2	218,2	-0,8%	215,4	217,3	216,4	218,2	-0,9%	53,60	54,79	54,79	54,79	-2,2%	(1h 58m 35s)		

Legenda: λ_i = taxa de chegada (pedestres/s)
 p_c = probabilidade de bloqueio
 θ = taxa de atendimento (pedestres/s)
L = número médio de usuários no sistema (pedestres)
W = tempo médio no sistema (s)

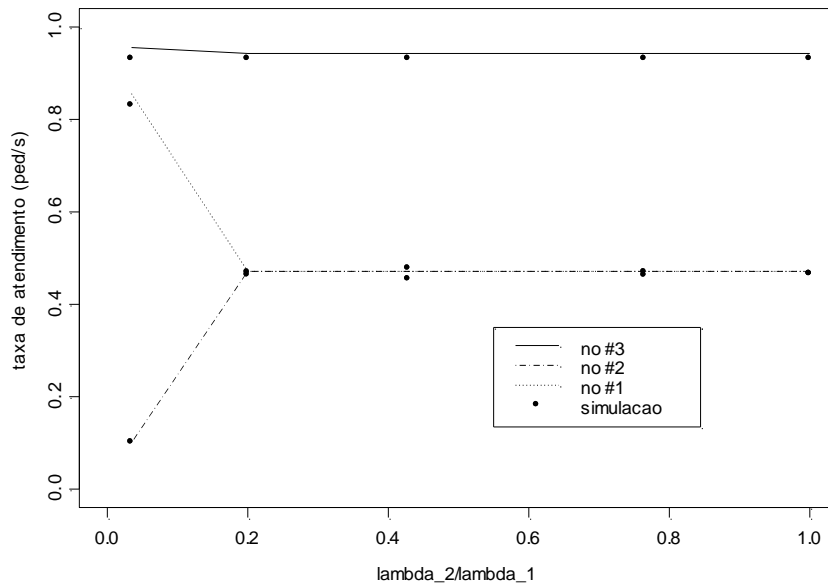


Figura 3.6: Taxa de atendimento para rede de três nós configurados em junção.

A Figura 3.6 apresenta a taxa de atendimento θ_i para todos os nós i da rede apresentada na Figura 3.7, em função da razão das taxas de chegada λ_2/λ_1 . Note que a concordância entre os dois resultados, analítico e de simulação, é grande. Também notável é que o fluxo vem em igual quantidade dos dois nós 1 e 2, mesmo que a taxa de entrada deles, λ_1 e λ_2 , seja desbalanceada. Claro que este fenômeno deixa de ser válido naqueles casos em que $\lambda_2 < \theta_3/2$, ou seja, a taxa de chegada ao nó imediatamente anterior à junção é insuficiente para prover a respectiva metade da taxa de atendimento do nó após a junção. Veja na Tabela 3.1 os resultados correspondentes a esta situação, na primeira configuração testada (primeiras quatro linhas).

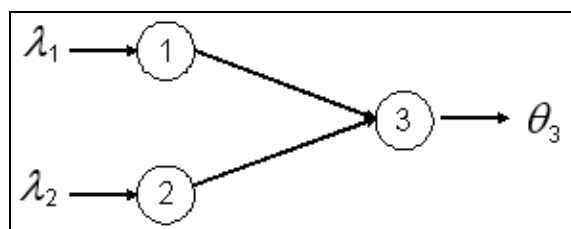


Figura 3.7: Sistema de três nós em topologia junção.

3.6 Principais Contribuições deste Capítulo

Dentre as contribuições deste capítulo, podemos citar a apresentação detalhada de uma nova implementação do método da expansão generalizado para análise aproximada de redes de filas finitas gerais dependentes do estado, configuradas em redes. Experimentos computacionais foram realizados e atestaram a qualidade das medidas de desempenho previstas pelo modelo analítico.

Capítulo 4

Validação da Modelagem por Filas M/G/c/c Configuradas em Rede

4.1 Um Sistema de Simulação a Eventos Discretos

4.1.1 Introdução

No capítulo anterior, utilizamos um modelo de simulação a eventos discretos que foi desenvolvido recentemente por Medeiros (2001) e Cruz et al. (2005a), para simular filas $M/G/c/c$ configuradas em redes. O modelo de simulação faz parte de um pacote mais amplo que segue a estrutura apresentada na Figura 4.1. O módulo *congestion models* é composto pela classe virtual `GenCM`, que provê métodos de propósito geral, tais como a taxa de serviço V_n , a capacidade c , o tempo de serviço esperado para um único usuário no sistema $E[T_1]$, e a velocidade média para um único usuário V_1 . As classes derivadas `LinCM` e `ExpCM` implementam os serviços linear e exponencial e o módulo *user congestion models* ajusta os modelos lineares e exponenciais para a aplicação. O módulo *analytical model* implementa o método da expansão generalizado (GEM), descrito no Capítulo 3. Já o módulo *optimization system*, que implementa métodos para otimização de redes de filas $M/G/c/c$, não será abordado neste trabalho (para maiores detalhes, ver artigo de Cruz et al., 2005b).

O principal interesse nesta sessão é o módulo *simulation model*, que essencialmente implementa o objeto `MgccSim`, apresentado na Figura 4.2, a qual descreve todas as estruturas de dados dentro do objeto e como elas se inter-relacionam. Basicamente, este objeto tem o número de nós na rede $M/G/c/c$ dependente do estado, `nOfNodes`, o tempo de simulação total, `totalTime`, a matriz origem-destino, os arcos, um vetor de objetos `MgccResource` e uma fila de eventos `MgccEventQueue`. Os objetos `MgccResource` mantêm o histórico de todas as estatísticas que podem ser de interesse para cada fila $M/G/c/c$, que é a soma de bloqueios, as chegadas e partidas, a soma do tempo global dentro do sistema e o número atual de usuários. Também faz parte de cada objeto `MgccResource` o modelo de congestionamento `GenCM`, que possibilita ter acesso à velocidade média (taxa de serviço) V_n , a capacidade c e o tempo esperado de serviço para um único usuário $E[T_1]$.

A parte crítica do objeto `MgccSimul` é o objeto `MgccEventQueue`, que implementa a fila de eventos. A fila de eventos é implementada como uma lista dinâmica, criada em tempo real e que mantém todos os eventos discretos. Surpreendentemente, depois de muita experimentação, foi visto que o melhor modo para assegurar baixos tempos de processamento era manter os eventos na fila de eventos não ordenados, com o custo adicional de precisar inspecionar toda ela, para recuperar o evento mais recente (e que, portanto, seja o próximo a ser processado).

Cada um dos objetos `MgccEvent` da fila de eventos contém a indicação de a qual fila $M/G/c/c$ estão relacionados, `whichQueue`, o tempo esperado de ocorrência, `occurTime`, o tipo de evento (`arrival`, `departure`, ou `end_simulation`) e a

entidade a qual estão relacionados, *MgccEntity*. O objeto *MgccEntity* representa cada usuário (pedestre) da rede *M/G/c/c* dependente do estado e tem um número identificador de usuário único, *id*, o tempo de chegada ao sistema, *sisArrival*, o tempo de chegada à fila atual, *queueArrival*, o tempo de ocorrência da última mudança de estado (um usuário que entra na fila ou a deixa), *lastChange*, e a posição do usuário quando da última mudança de estado, *lastPosition*. A arquitetura usada é livremente inspirada na do Arena (Kelton et al., 2001).

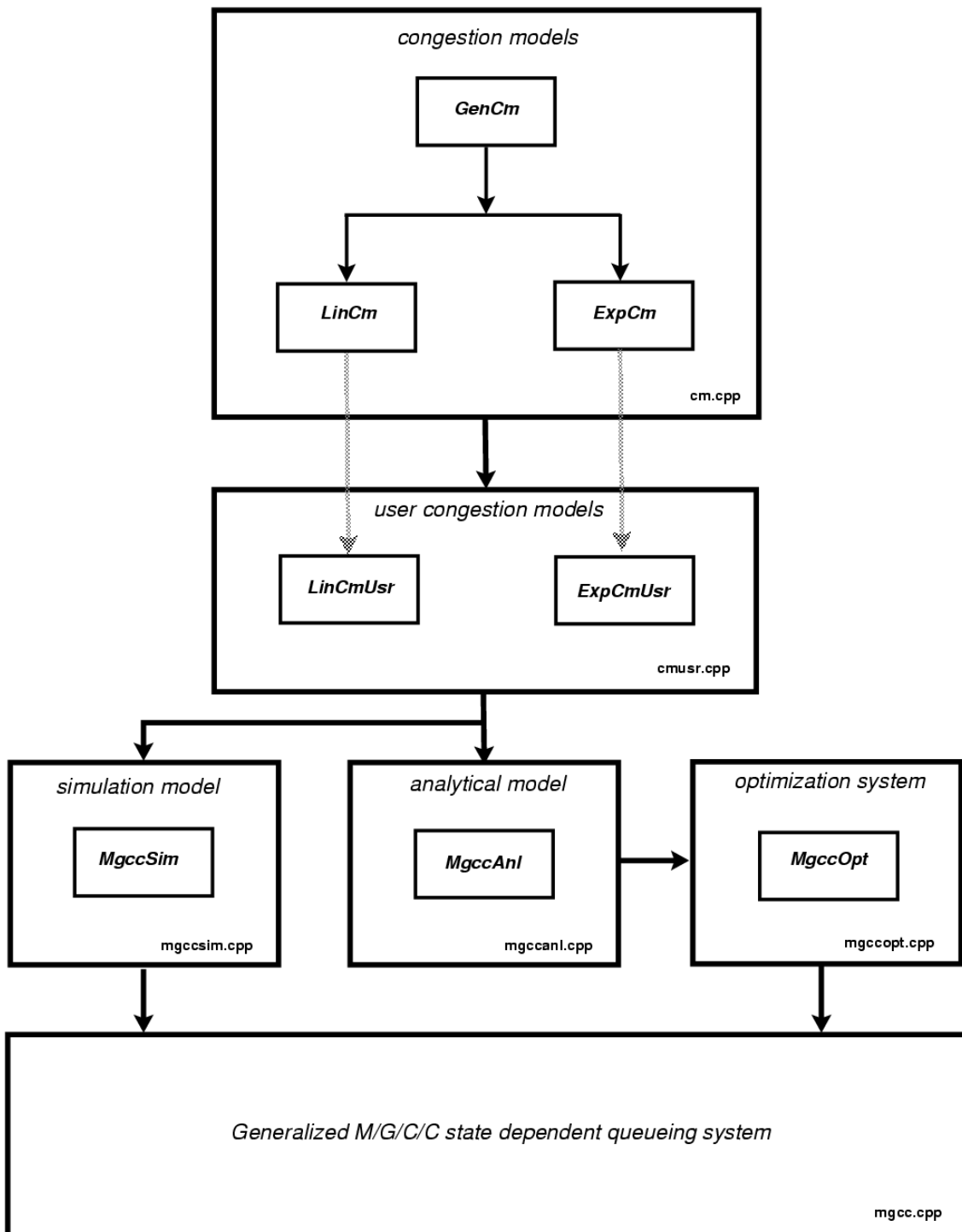


Figura 4.1: Sistema para filas M/G/c dependentes do estado (Cruz et al., 2005a).

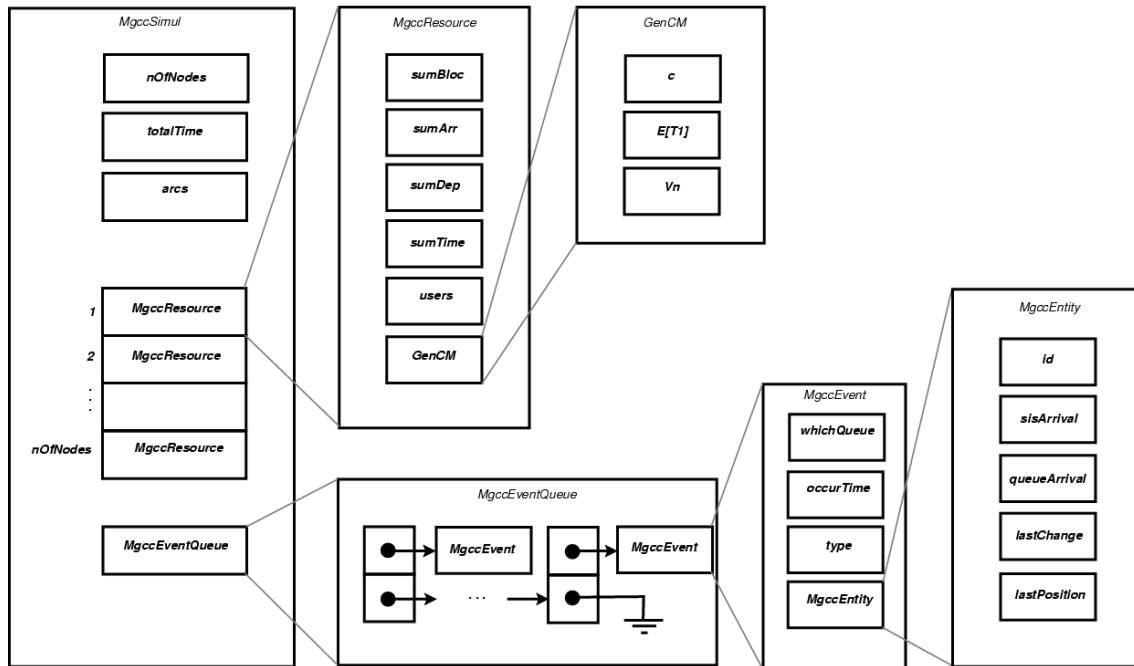


Figura 4.2: Objeto MgccSim (Cruz et al., 2005a).

4.1.2 Algoritmo de Simulação

O algoritmo de simulação, em pseudolinguagem, é ilustrado em Figura. 4.3. Tudo que algoritmo faz é iniciar a fila de eventos `MgccEventQueue`, programando o último evento (`end_simulation`) e os primeiros eventos (`arrival`). Então, o algoritmo repetidamente procura pelo evento mais recente (em ordem de ocorrência), processa-o, gera os respectivos novos eventos e repete a operação até que o evento final (`end_simulation`) seja alcançado.

As chegadas devem ser tratadas pelo algoritmo mostrado na Figura 4.4. No momento da chegada, é possível programar a próxima chegada àquela fila e incluir este novo evento na fila de eventos. Como é assumido que as chegadas estão vindo de um processo markoviano, os tempos entre chegadas seguem uma distribuição exponencial, isto é:

$$t \sim \text{Exp}(\lambda_i).$$

A chegada atual gera um evento de partida que também deve ser inserido na fila de eventos. A ocorrência desta partida é após uma demora de l_i/V_n , em que l_i é o comprimento do i -ésimo corredor e V_n , a velocidade média para os n ocupantes. A operação mais complexa no algoritmo `ProcessArrival` está em atrasar a partida de todos os usuários na fila, ou seja, levar em conta a dependência de estado do sistema. Para todas as entidades naquela fila, o algoritmo `DelayDeparture` (não apresentado) precisa atualizar a posição desde a última mudança, `lastPosition`, o

tempo desde a última mudança, `lastChange`, e o tempo de ocorrência do novo evento, `occurTime` (Figura 4.2), como segue:

$$\begin{aligned} \text{lastPosition} &\leftarrow \text{lastPosition} + V_{n-1} \times (t - \text{lastChange}), \\ \text{lastChange} &\leftarrow t, \\ \text{occurTime} &\leftarrow t + (l - \text{lastPosition}) / V_n. \end{aligned}$$

em que t é o tempo atual. Note que como é possível que o usuário seja bloqueado depois do serviço, é necessário ter certeza que `lastPosition` não vá além do comprimento do i -ésimo corredor, l_i .

```

algorithm Simulate
    /* initialize event queue */
    Initialize(MgccEventQueue);
    /* create and insert 'last' event */
    MgccEvent ← new();
    MgccEvent.occurTime ← totalTime;
    MgccEvent.type ← end_simulation;
    Insert(MgccEventQueue, MgccEvent);
    /* create and insert 'first' events */
    for  $\forall n \mid \lambda_n \neq 0$  do
        MgccEvent ← new();
        MgccEvent.whichQueue ←  $n$ ;
        MgccEvent.occurTime ← 0.0;
        MgccEvent.type ← arrival;
        Insert(MgccEventQueue, MgccEvent);
    end for
    /* simulate */
    MgccEvent ← GetNext(MgccEventQueue);
    while MgccEvent.type  $\neq$  end_simulation do
        if MgccEvent.type = arrival then
            ProcessArrival(MgccEventQueue, MgccEvent);
        else if MgccEvent.type = departure then
            ProcessDeparture(MgccEventQueue, MgccEvent);
        else
            error, unknown event
        end if
        MgccEvent ← GetNext(MgccEventQueue);
    end while
    print results
end algorithm

```

Figura 4.3: Algoritmos de simulação a eventos discretos (Cruz et al., 2005a).

O tratamento das partidas, Figura 4.4, envolve a determinação cuidadosa do novo destino do usuário. Se a fila destino estiver lotada, então a partida da fila de origem deve ser adiada (bloqueio após o atendimento) até que haja lugar na fila de destino. Caso contrário, o novo tempo de partida desta entidade é atualizado, somando-se um atraso

de l/V_n , adiantando-se as partidas de todas as entidades da fila de origem, e atrasando-se todas as partidas da fila de destino.

```

algorithm ProcessArrival(EventQueue,Event)
    /* program next arrival */
    newEvent ← new();
    newEvent.whichQueue ← Event.whichQueue;
    newEvent.occureTime ∼ Exp( $\lambda$ ) + Event.occureTime;
    newEvent.type ← arrival;
    Insert(EventQueue,newEvent);
    /* process this arrival */
    if queue is blocked then
        reject arrival and count it
    else
        newEvent ← new();
        newEvent.whichQueue ← Event.whichQueue;
        newEvent.occureTime ←  $l/V_n$  + Event.occureTime;
        newEvent.type ← departure;
        Insert(EventQueue,newEvent);
        count arrival
        /* delay departure of every user in this queue */
        DelayDeparture(EventQueue,Event.whichQueue);
    end if
end algorithm

```

a) ProcessaChegada

```

algorithm ProcessDeparture(EventQueue,Event)
    generate next destination queue  $d$ 
    if final destination then
        count departure
        /* advance departure of users */
        AdvanceDeparture(EventQueue,Event.whichQueue);
    else if queue is blocked then
        postpone departure and count it
    else
        /* create and insert departure */
        newEvent ← new();
        newEvent.whichQueue ←  $d$ ;
        newEvent.occureTime ←  $l/V_n$  + Event.occureTime;
        newEvent.type ← departure;
        Insert(EventQueue,newEvent);
        count departure
        count arrival
        /* advance departure of users */
        AdvanceDeparture(EventQueue,Event.whichQueue);
        /* delay departure of users */
        DelayDeparture(EventQueue, $d$ );
    end if
end algorithm

```

b) ProcessaPartida

Figura 4.4: Algoritmos ProcessaChegada e ProcessaPartida (Cruz et al., 2005a).

O algoritmo `AdvanceDeparture` (não apresentado) segue a mesma filosofia do algoritmo `DelayDeparture`. A posição, o tempo e o novo tempo de ocorrência do evento, desde a última mudança, são atualizados como segue:

$$\begin{aligned} \text{lastPosition} &\leftarrow \text{lastPosition} + V_{n+1} \times (t - \text{lastChange}), \\ \text{lastChange} &\leftarrow t, \\ \text{occurTime} &\leftarrow t + (l + \text{lastPosition}) / V_n. \end{aligned}$$

Note que, atualizando (adiantando ou atrasando) os tempos de ocorrência de todas as entidades de uma fila particular, uma lista de eventos (por exemplo, a lista `MgccEventQueue`, Figura 4.2) previamente ordenada (pelo tempo de ocorrência) acabaria desordenada. Esta é a explicação para um menor tempo computacional observado em um modelo de simulação a eventos discretos baseado em filas não ordenadas. Para este modelo, é trabalhoso recuperar o próximo evento em uma fila não ordenada, pois no pior caso há a necessidade e o custo extra de inspecionar todos os elementos da lista de eventos. Entretanto, computacionalmente ainda mais caro seria manter a fila de eventos ordenada, a cada nascimento ou morte em uma fila.

4.2 Ajuste de Modelos

Em diversas situações, têm-se observações de uma variável aleatória cuja distribuição na população é desconhecida. Em algumas delas é possível incorporar informações de outras variáveis que descrevam fenômenos aleatórios similares e que tenham distribuição conhecida, proporcionando uma distribuição candidata a modelo, bastando estabelecer procedimentos para aceitá-la ou não. Contudo em vários outros casos não se tem a menor idéia do comportamento da variável. Um procedimento inicial de análise é a confecção de um diagrama, com as freqüências de ocorrência, nos moldes do histograma. Desta representação gráfica, pode sair uma sugestão de modelos adequados aos dados. Em qualquer caso, o modelo proposto pode ser testado através dos testes de aderência. As hipóteses a serem testadas são:

$$\begin{aligned} H_0 &: X \text{ segue o modelo proposto} \\ H_A &: X \text{ não segue o modelo proposto} \end{aligned}$$

Geyer & Willianson, em seu recente artigo na *Communications in Statistics* (Geyer & Willianson, 2004), utilizam um dos mais simples e bem conhecidos testes de aderência, o teste qui-quadrado, para uma bem sucedida detecção de fraude em conjuntos de dados do imposto de renda dos Estados Unidos, que supostamente deveriam seguir uma certa distribuição, conhecida como lei de Benford.

A lei de Benford é um fenômeno numérico em que conjuntos de dados que enumeram ou mensuram algum evento seguem uma certa distribuição, diferente da uniforme. Assim, se questionados qual a probabilidade de um preço acionário ter o primeiro dígito significativo igual a 1, incorretamente podemos responder que seria 1/9. Esta resposta considera que todos os dígitos possuem a mesma probabilidade de ocorrer.

A lei de Benford tem sua origem nas observações de Simon Newcomb, um astrônomo e matemático, que notou que as primeiras páginas das tábuas de logaritmos eram muito mais sujas e que as páginas subseqüentes iam clareando para o fim do livro. Ele observou que os valores com o dígito significativo 1 eram mais procurados que os valores com o dígito significativo 2 e estes, com maior freqüência que os valores iniciados pelo algarismo 3, e assim sucessivamente. Infelizmente o artigo de Newcomb no *American Journal of Mathematics* (Newcomb, 1881, citado por Geyer & Willianson,

2004) não foi notado. Em 1938, Benford publicou um artigo no *Proceedings of the American Philophical Society* (Benford, 1938, citado por Geyer & Willianson, 2004) descrevendo o fenômeno e incluindo mais de 20.000 observações para provar sua teoria.

Segundo a lei de Benford, a probabilidade de que um número tenha o primeiro dígito significativo igual a um certo algarismo é dada por:

$$P(\text{primeiro dígito significativo} = d) = \log_{10}(1 + d^{-1}), \quad d = 1, \dots, 9. \quad (26)$$

No conhecido teste qui-quadrado, aplicado de maneira bem sucedida por Geyer & Willianson (2004), no problema de detecção de fraude, a quantidade utilizada para a tomada da decisão será baseada no quadrado das diferenças entre os valores observados na amostra, o_i , e os valores esperados sob H_0 , e_i . A diferença $o_i - e_i$ proporciona uma idéia da compatibilidade entre os valores observados e o modelo proposto. Assim, se as diferenças forem muito grandes, é razoável admitir que o modelo não deve ser adequado. Por outro lado, pequenas diferenças podem ser aceitas. A quantidade utilizada no teste será:

$$\chi^2 = \sum_{i=1}^k \frac{(o_i - e_i)^2}{e_i}, \quad (27)$$

na qual k representa o número de categorias. É possível demonstrar que a quantidade (27) pode ser aproximada por um modelo qui-quadrado com parâmetro $k-1$ (graus de liberdade), a qual representamos por $\chi_{(k-1)}^2$.

A decisão será baseada no comportamento de χ^2 . A região crítica é constituída de valores grandes de χ^2 , isto é, $RC = \{\chi^2 : \chi^2 \geq q_c\}$, com q_c sendo determinado pelo nível de significância do teste, ou seja, $\alpha = P(\chi^2 \geq q_c | H_0 \text{ verdadeiro})$. Se H_0 for verdadeiro, X segue o modelo proposto (Magalhães & Lima, 2002).

Em seguida, utilizaremos técnicas de simulação a eventos discretos e ajuste de modelos para caracterizar o tráfego entre filas e validar a modelagem via filas $M/G/c/c$ dependentes do estado, configuradas em redes.

4.3 Experimentos Computacionais

Os algoritmos foram programados em C++, os gráficos foram gerados no S-PLUS (S-PLUS, 2000) e os ajustes dos modelos foram feitos no Arena (Kelton et al., 2001). Todos os experimentos computacionais foram realizados em um PC, Pentium II, 400 MHz, 256 MB RAM, sistema operacional Windows NT 4.0. Para as simulações foram adotadas configurações em série, divisão e junção, variando-se as taxas de entrada do sistema e em alguns casos as dimensões de alguns corredores. Foi considerado também um tempo para “warm-up”, a fim de que o sistema atingisse o regime.

4.3.1 Sobre o Tempo de Serviço

A Figura 4.5 ilustra um sistema de dois corredores com dimensões distintas, configurando um sistema com “gargalo”. Pelas simulações feitas, podemos observar que os tempos de serviços seguem um comportamento peculiar, justificando a utilização de um modelo com taxa de serviço geral.

Nas Figuras 4.6 e 4.7 temos uma comparação entre os tempos de serviços nos nós #1 e #2, para duas taxas de chegada, $\lambda=1$ ped/s e $\lambda=3$ ped/s. Vemos que com o aumento da

taxa de chegada o tempo de serviço tende a crescer e a reduzir sua variabilidade. Os histogramas apresentados nas Figuras 4.6 e 4.7 indicam que para $\lambda=1$ ped/s a variabilidade dos tempos de serviço no nó #2 é maior que no nó #1. Por outro lado, quando $\lambda=3$ ped/s a situação inverte-se, sendo que o tempo de serviço no nó #2 cresce e a sua variabilidade diminui drasticamente, ficando menor que a variabilidade no nó #1, que também aumenta de valor médio.

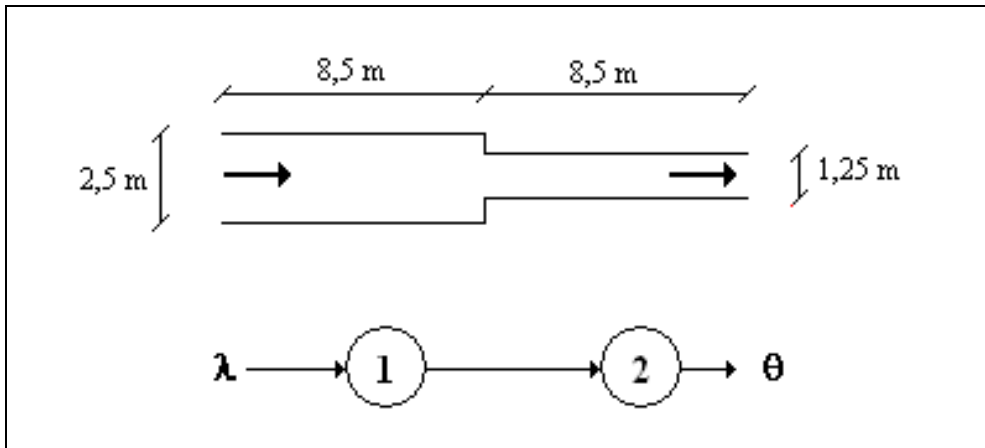


Figura 4.5: Sistema de duas filas configuradas em série.

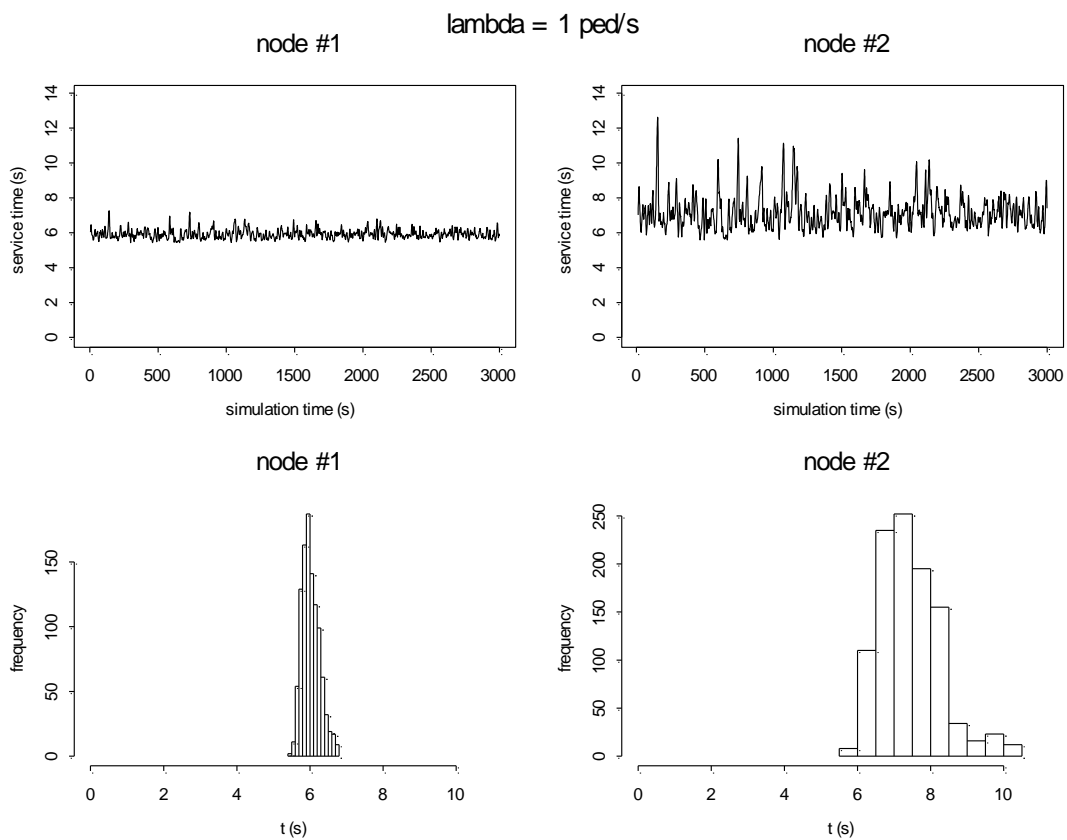


Figura 4.6: Tempo de serviço em um sistema de duas filas em série com taxa de chegada $\lambda=1$ ped/s.

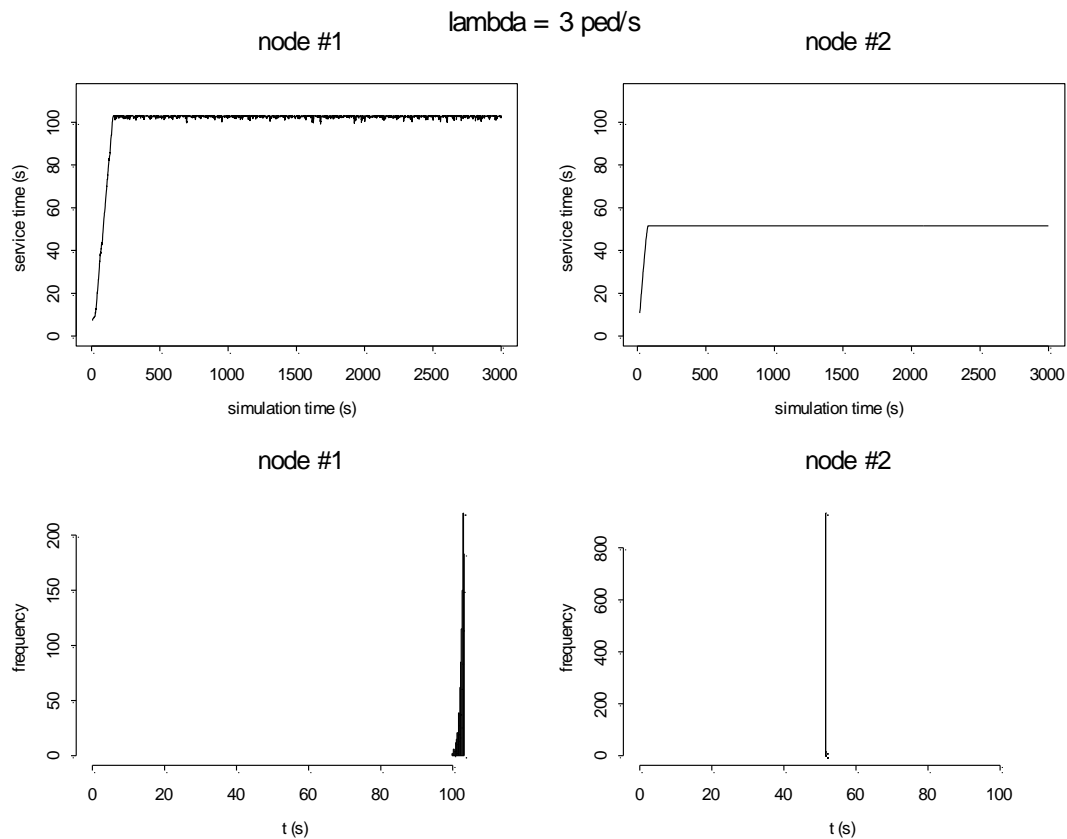


Figura 4.7: Tempo de serviço em um sistema de duas filas em série com taxa de chegada $\lambda=3$ ped/s.

4.3.2 Sobre o Tráfego Entre Filas

a) Configuração Série – Caso Homogêneo

O sistema considerado nesta simulação é similar ao da Figura 4.5, diferindo-se apenas no fato de que os dois corredores têm dimensões iguais a $8,5 \times 2,5 \text{ m}^2$, isto é, não existe gargalo no sistema. Na Tabela 4.1 temos as medidas de desempenho do sistema, bem como do ajuste da distribuição exponencial. Em relação às medidas de desempenho, como esperado as probabilidades de bloqueio aumentam com o acréscimo da taxa de chegada ao sistema e no nó #1 atingimos a capacidade máxima do sistema (100 pedestres) para taxas de chegada mais elevadas. Os resultados provenientes do ajuste de modelo indicam que a distribuição exponencial parece ser uma boa aproximação na maioria dos casos testados, apresentando um ajuste com alto valor-p e autocorrelação desprezível, conforme pode ser visto na Figura 4.8. Rejeições ocorrem apenas em taxas de chegada elevadas e mesmo nesta situação, apesar de a autocorrelação dos dados parecer significativa, o modelo exponencial não aparenta ser tão discrepante, conforme pode ser observado na Figura 4.9.

b) Configuração Série – Caso Heterogêneo

Neste caso, tratamos exatamente o sistema da Figura 4.5, configuração em série, em que o segundo corredor apresenta uma largura inferior à do primeiro, caracterizando um “gargalo”. Pelos resultados na Tabela 4.2 vemos que a aproximação através do modelo exponencial piora consideravelmente devido à inserção do gargalo no sistema, pois começamos a observar perda de aderência do modelo exponencial bem mais cedo, já a partir de uma taxa de chegada $\lambda=2$ ped/s. Testando a autocorrelação dos tempos entre partidas notamos valores significativos, à semelhança dos apresentados na Figura 4.9. Entretanto, apesar deste fato, similarmente ao caso *a*, aqui também o modelo exponencial não parece ser assim tão ruim.

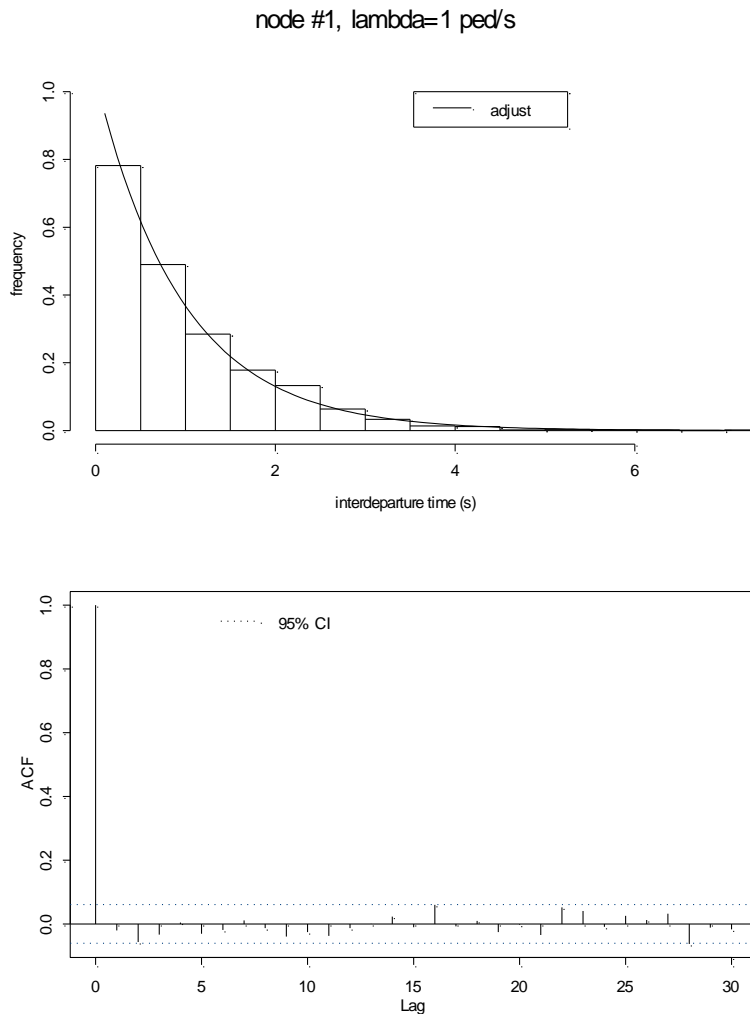


Figura 4.8: Ajuste para o tempo entre partidas no nó #1 em um sistema de duas filas em série com taxa de chegada $\lambda=1$ ped/s.

c) Configuração Divisão – Caso Homogêneo

No caso de um sistema configurado em divisão como o apresentado na Figura 3.3, para corredores de dimensões $8,5 \times 2,5 \text{ m}^2$, as medidas de desempenho e de ajuste, apresentadas na Tabela 4.3, indicam um desempenho similar ao caso de corredores homogêneos (isto é, mesmas dimensões, portanto sem gargalo) configurados em série, ocorrendo rejeição apenas em taxas de chegada elevadas.

d) Configuração junção – Caso Homogêneo

Para o caso de um sistema configurado em junção como aquele apresentado na Figura 3.2, para corredores de dimensões $8,5 \times 2,5 \text{ m}^2$, temos caracterizado um gargalo na entrada do nó #3. Este gargalo leva o sistema a comportar-se como no caso de corredores heterogêneos configurados em séries, isto é, com um gargalo inserido. Os resultados das medidas de desempenho e do modelo de ajuste podem ser vistos na Tabela 4.4.

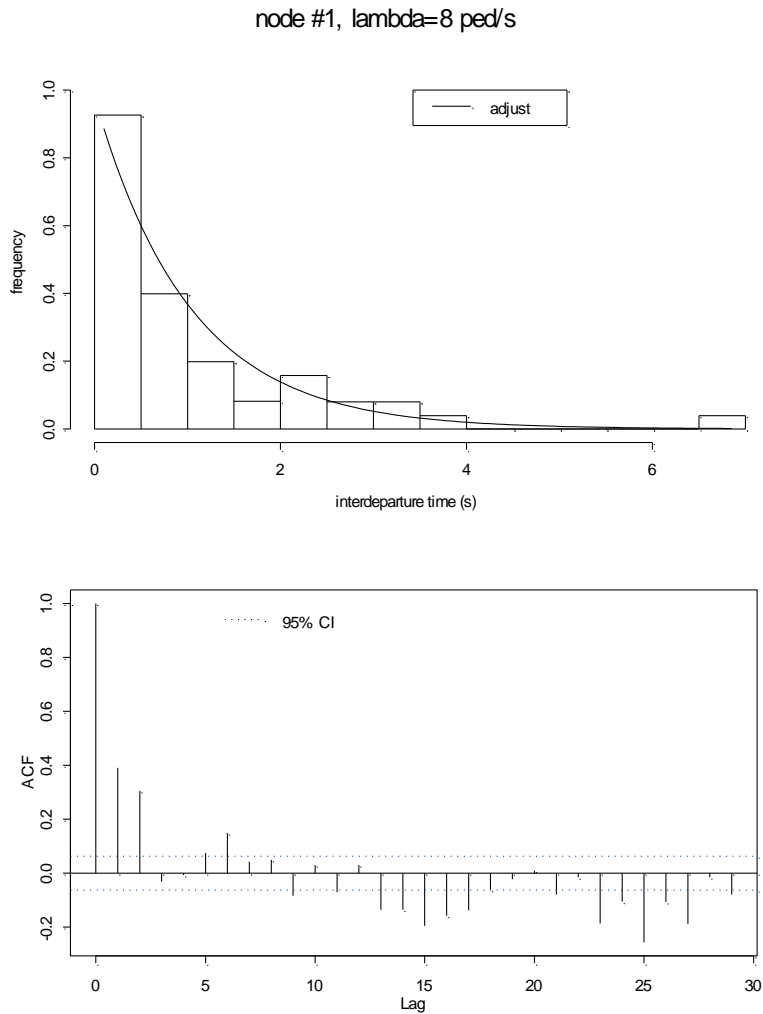


Figura 4.8: Ajuste para o tempo entre partidas no nó #1 em um sistema de duas filas em série com taxa de chegada $\lambda=8 \text{ ped/s}$.

Tabela 4.1: Medidas de desempenho e de ajuste de modelo para o sistema de dois corredores de 8,5 x 2,5 m² configurados em série.

λ	Nó 1				Nó 2				Entre Filas		
	p_c	θ	L	W	p_c	θ	L	W	Modelo Exp(λ)		
									λ	Erro Quadrático	Valor-p
0,5	0,000000	0,516000	2,909482	5,638532	0,000000	0,517000	2,915609	5,639476	1,94	0,000636	> 0,75
1,0	0,000000	1,041000	6,268951	6,022047	0,000000	1,039000	6,255234	6,020437	0,96	0,000760	0,197
2,0	0,000000	2,034000	14,83056	7,291329	0,000000	2,028000	14,75877	7,277501	0,49	0,000195	> 0,75
3,0	0,341052	1,992000	97,74885	49,07071	0,000000	1,990000	14,38008	7,226169	0,50	0,000588	0,0443
5,0	0,605870	1,947000	99,19081	50,94546	0,000000	1,944000	13,76991	7,083285	0,51	0,000501	0,2
8,0	0,754152	1,954000	100,3300	51,3460	0,000000	1,935000	14,70145	7,597646	0,51	0,001340	< 0,005

Legenda: λ = taxa de chegada (pedestres/s)
 p_c = probabilidade de bloqueio
 θ = taxa de atendimento (pedestres/s)
 L = número médio de usuários no sistema (pedestres)
 W = tempo médio no sistema (s)

Tabela 4.2: Medidas de desempenho e de ajuste de modelo para o sistema de dois corredores de 8,5 x 2,5 m² e 8,5 x 1,25 m², configurados em série.

λ	Nó 1				Nó 2				Entre Filas		
									Modelo Exp(λ)		
	p_c	θ	L	W	p_c	θ	L	W	λ	Erro Quadrático	Valor-p
0,5	0,000000	0,516000	2,909482	5,638532	0,000000	0,518000	3,164780	6,109614	1,94	0.000636	> 0,75
1,0	0,000000	1,041000	6,268951	6,022047	0,000000	1,040000	7,726757	7,429574	0,96	0,000760	0,197
2,0	0,522527	0,975000	99,62154	102,1759	0,499743	0,975000	50,27177	51,56079	1,03	0,017497	< 0,005
3,0	0,681217	0,964000	98,82693	102,5176	0,500000	0,964000	49,70460	51,56079	1,04	0,007803	< 0,005
5,0	0,804819	0,964000	99,12663	102,8285	0,500000	0,964000	49,70460	51,56079	1,04	0,022763	< 0,005
8,0	0,876950	0,978000	100,6918	102,9569	0,499744	0,978000	50,42645	51,56079	1,02	0,031364	< 0,005

Legenda: λ = taxa de chegada (pedestres/s)
 p_c = probabilidade de bloqueio
 θ = taxa de atendimento (pedestres/s)
 L = número médio de usuários no sistema (pedestres)
 W = tempo médio no sistema (s)

Tabela 4.3: Medidas de desempenho e de ajuste de modelo para o sistema de três corredores de 8,5 x 2,5 m² configurados em divisão.

λ	Nó 1				Nó 2				Nó 3				Entre Filas		
													Modelo Exp(λ)		
	p_c	θ	L	W	p_c	θ	L	W	p_c	θ	L	W	λ	Erro Quadrático	Valor-p
0,5	0,000000	0,516000	2,909482	5,638532	0,000000	0,255000	1,391772	5,457929	0,000000	0,262000	1,439471	5,494164	1,94	0,000636	> 0,75
1,0	0,000000	1,041000	6,268951	6,022047	0,000000	0,548000	3,100839	5,658466	0,000000	0,492000	2,762089	5,614003	0,96	0,000760	0,197
2,0	0,000000	2,034000	14,83056	7,291329	0,000000	1,025000	6,206142	6,054772	0,000000	1,001000	6,038021	6,031989	0,49	0,000195	> 0,75
3,0	0,341052	1,992000	97,74885	49,07071	0,000000	0,993000	5,967703	6,009772	0,000000	1,000000	6,021855	6,021855	0,50	0,000588	0,0443
5,0	0,605870	1,947000	99,19081	50,94546	0,000000	0,978000	5,859149	5,990950	0,000000	0,966000	5,779744	5,983172	0,51	0,000501	0,2
8,0	0,754152	1,954000	100,3300	51,34598	0,000000	0,915000	5,522380	6,035388	0,000000	1,025000	6,312867	6,158894	0,51	0,001340	< 0,005

Legenda:
 λ = taxa de chegada (pedestres/s)
 p_c = probabilidade de bloqueio
 θ = taxa de atendimento (pedestres/s)
 L = número médio de usuários no sistema (pedestres)
 W = tempo médio no sistema (s)

Tabela 4.4: Medidas de desempenho e de ajuste de modelo para o sistema de três corredores de 8,5 x 2,5 m² configurados em junção.

λ	Nó 1				Nó 2				Nó 3				Entre Filas		
													Modelo Exp(λ)		
	p_c	θ	L	W	p_c	θ	L	W	p_c	θ	L	W	λ	Erro Quadrático	p-Valor
1,0	0,000000	0,541000	3,053321	5,643847	0,000000	0,499000	2,806011	5,623269	0,000000	1,039000	6,258987	6,024049	1,85	0,000994	0,611
2,0	0,000000	0,976000	5,858740	6,002807	0,000000	1,064000	6,459302	6,070773	0,000000	2,028000	14,75849	7,277363	1,03	0,000603	> 0,75
4,0	0,532791	0,976000	99,81070	102,2651	0,508543	0,978000	99,70065	101,9434	0,499872	1,954000	101,1396	51,76026	1,03	0,005983	< 0,005
6,0	0,686153	0,961000	98,94134	102,9566	0,665856	0,962000	99,04463	102,9570	0,499479	1,923000	99,53499	51,76026	1,04	0,016344	< 0,005
10	0,806172	0,961000	99,15941	103,1836	0,807576	0,955000	98,65714	103,3059	0,499478	1,916000	99,17266	51,76026	1,04	0,008237	< 0,005
16	0,878894	0,972000	100,4404	103,3337	0,881595	0,956000	98,79259	103,3395	0,499741	1,928000	99,79379	51,76026	1,03	0,034588	< 0,005

Legenda:
 λ = taxa de chegada (pedestres/s)
 p_c = probabilidade de bloqueio
 θ = taxa de atendimento (pedestres/s)
 L = número médio de usuários no sistema (pedestres)
 W = tempo médio no sistema (s)

4.4 Principais Contribuições deste Capítulo

As principais contribuições apresentadas neste capítulo incluem uma revisão bibliográfica sobre técnicas de simulação a eventos discretos, que são bastante apropriadas para a verificação da qualidade de modelos analíticos aproximados de filas configuradas em redes. Também foram apresentados resultados de experimentos computacionais que atestaram a qualidade da implementação do método da expansão generalizado, uma técnica bem estabelecida para determinação aproximada de importantes medidas de desempenho em sistemas de filas finitas configuradas em redes.

Capítulo 5

Conclusões e Observações Finais

Para o desenvolvimento deste trabalho foram realizados estudos sobre processos estocásticos, incluindo a teoria de filas, a obtenção das probabilidades de equilíbrio de um sistema, a demonstração da equivalência estocástica entre filas $M/M/c/c$ e $M/G/c/c$ e o modelo analítico para tráfego com congestionamento. Também se fez necessário o entendimento do método da expansão generalizado para filas $M/G/c/c$. Para a validação da modelagem de filas $M/G/c/c$ de filas configuradas em redes foi estudado um sistema de simulação a eventos discretos e um método de ajuste de modelos.

Para a obtenção dos resultados foi realizada uma modificação no algoritmo de simulação a fim de obtermos as medidas dos tempos entre partidas dos nós. Com estes resultados podemos observar que o tráfego entre filas pode ser modelado através de um modelo exponencial com boa aproximação, apesar de encontrarmos que este modelo pode ser rejeitado em casos em que a taxa de entrada no sistema é elevada.

Em pesquisas futuras podemos avaliar o tráfego entre filas em sistemas de redes com configurações mais complexas, utilizando outros métodos de ajuste de modelos menos propícios à rejeição sob grandes amostras e até mesmo avaliando o ajuste para outros modelos da família exponencial.

Referências Bibliográficas

- Araújo, H. F.; Toledo, M. L. G.; Cruz, F. R. B. & Macgregor Smith, J. (2004). Generalized M/G/c/c state dependent queueing network analysis, *In: XV Congresso Brasileiro de Automática*, Gramado. Anais [CD-ROM], 1-6.
- Benford, F. (1938). The law of anomalous numbers, *Proceedings of the American Philosophical Society*, 78:551-572.
- Cheah, J. (1990). Modeling traffic as M/G/c/c state dependent queues, Dissertação de Mestrado, University of Massachusetts, Amherst, MA.
- Cheah, J. & 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. & Medeiros, R. O. (2005a). An M/G/c/c state dependent network simulation model, *Computers & Operations Research*, 32(4), 919-941.
- Cruz, F. R. B., MacGregor Smith J, & Queiroz, D.C. (2005b). Service and capacity allocation in M/G/c/c state dependent queueing networks, *Computers & Operations Research*, 32(6), 1545-1563.
- Dijkstra, E. W. (1959). A note on two problems in connection with graphs, *Numerical Mathematics*, 1:269-271.
- Fruim, J. J. (1971). *Pedestrian Planning and Design*. Metropolitan Association of Urban Designers and Environmental Planners, Inc., New York, USA.
- Geyer, C. L. & Willianson, P. P. (2004). Detecting fraud in data sets using Benford's Law. *Communications in Statistics – Simulation and Computation*, 33(1):229-246.
- Gross, D. & Harris, C. M. (1998) *Queueing Theory*, John Wiley & Sons, Inc., New York.
- Jain, R. & MacGregor Smith, J. (1997). Modeling Vehicular Traffic Flow Using M/G/c/c State Dependent Queueing Models, *Transportation Science*, 31(4):324-336.
- Kelton, D., Sadowski, R. P. & Sadowski, D. A. (2001) *Simulation with Arena*, MacGraw Hill College Div., Nova York.
- Kendall, D. G. (1953). Stochastic Processes occurring in the theory of queues and their analysis by the method of imbedded Markov chains, *Annals Mathematical Statistics*, 24:338-354.
- Kerbache, L. & 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, Nova York.
- Labetoulle, J. & Pujolle, G. (1980). Isolation method in a network of queues, *IEEE*

Transactions on Software Engineering, SE-6(4):373-381.

- Magalhães, M. N. & Lima, A. C. P. (2002). *Noções de Probabilidade e Estatística*, IME-USP, Edusp.
- Medeiros, R. O., (2001). Simulação de Redes de Filas *M/G/C/C* Dependentes do Estado, *Dissertação de Mestrado*, Departamento de Estatística - ICEX – UFMG.
- Newcomb, S. (1881). Note of the frequency of the use of the different digits in natural numbers, *American Journal of Mathematics*, 4:39-40.
- Ross, S. M., (1996). *Stochastic Processes*, 2^a Ed., John Wiley & Sons, Inc., New York
- Tregenza, P. R. (1976). *The Design of Interior Circulation*, Van Nostrand Reinhold Company, New York, USA.
- S-PLUS (2000). *Programmer's Guide*, Data Analysis Products Division, MathSoft, Seattle, USA.
- Wagner, H. M. (1986). *Pesquisa Operacional*, 2^a Ed., Prentice-Hall do Brasil Ltda., Rio de Janeiro.
- Yuhaski, S. J. & MacGregor Smith, J. (1989). Modeling circulation systems in buildings using state dependent models. *Queueing Systems*, 4:319-338.