Ministério da
Ciência e Tecnologia

INPE-15350-TDI/1386

UM ESTUDO DE MÉTODOS DE SOLUÇÃO DO MODELO HIPERCUBO DE FILAS PARA SISTEMAS DE GRANDE PORTE

Leandro Luque

Dissertação do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Dr. Solon Venâncio de Carvalho, aprovada em 10 de agosto de 2007.

Registro do documento original:
<http://urlib.net/sid.inpe.br/mtc-m17@80/2007/12.07.11.42>

INPE
São José dos Campos
2008

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GB)
Serviço de Informação e Documentação (SID)
Caixa Postal 515 - CEP 12.245-970
São José dos Campos - SP - Brasil
Tel.:(012) 3945-6911/6923
Fax: (012) 3945-6919
E-mail: pubtc@sid.inpe.br

Ministério da
Ciência e Tecnologia

INPE-15350-TDI/1386

UM ESTUDO DE MÉTODOS DE SOLUÇÃO DO MODELO HIPERCUBO DE FILAS PARA SISTEMAS DE GRANDE PORTE

Leandro Luque

Dissertação do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Dr. Solon Venâncio de Carvalho, aprovada em 10 de agosto de 2007.

Registro do documento original:
<http://urlib.net/sid.inpe.br/mtc-m17@80/2007/12.07.11.42>

INPE
São José dos Campos
2008

Dados Internacionais de Catalogação na Publicação (CIP)

153m Luque, Leandro.
Um estudo de métodos de solução do modelo hipercubo de filas para sistemas de grande porte/ Leandro Luque. – São José dos Campos: INPE, 2008.
147p. ; (INPE-15350-TDI/1386)

Dissertação (Mestrado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2007.

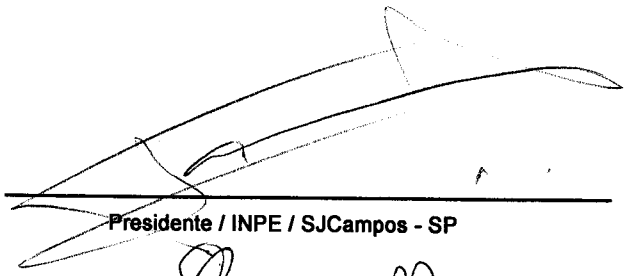
1. Modelo hipercubo de filas. 2. Sistemas de grande porte. 3. Decomposição. 4. Métodos aproximados. 5. Precisão. I. Título.

CDU 519.872

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de Mestre em

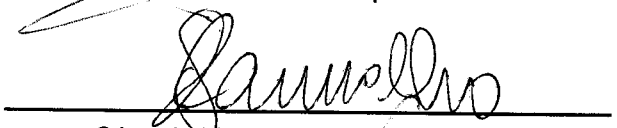
Computação Aplicada

Dr. Horacio Hideki Yanasse



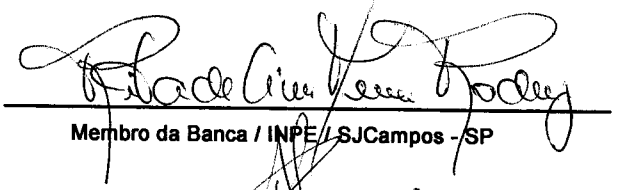
Presidente / INPE / SJCampos - SP

Dr. Solon Venâncio de Carvalho



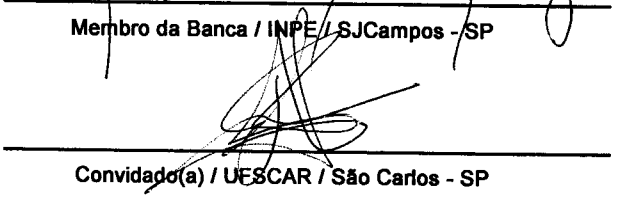
Orientador(a) / INPE / SJCampos - SP

Dra. Rita de Cássia Meneses Rodrigues



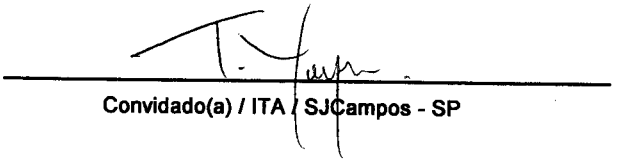
Membro da Banca / INPE / SJCampos - SP

Dr. Reinaldo Morábito Neto



Convidado(a) / UESCAR / São Carlos - SP

Dr. Takashi Yoneyama



Convidado(a) / ITA / SJCampos - SP

Aluno (a): Leandro Luque

São José dos Campos, 10 de Agosto de 2007

“Estou convencido das minhas próprias limitações – e esta convicção é minha força”.

Mahatma Gandhi

“Viver envolve apostar. Apostamos em nossa memória quando lembramos, no passado quando aprendemos, no futuro quando plantamos, em outras pessoas quando confiamos, em nossos sentimentos quando amamos... Apostando, ganharemos algumas vezes e, certamente, perderemos outras. Aprender a viver compreende, portanto, aprender quando e no que apostar, quando parar e, principalmente, saber perder.”

Dedico este trabalho,
a meu pai, Darcy Luque,
a minhã mãe, Celina Luque
e a meu irmão, Alexandre Luque,
referências de vida pessoal, profissional e ética.

AGRADECIMENTOS

Agradeço a todos aqueles que são diretamente ou indiretamente responsáveis pela minha existência.

A todas as pessoas que me ajudaram a vencer mais esta etapa.

A meus familiares por sempre acreditarem na importância do estudo e por sempre acreditarem em mim.

Ao Instituto Nacional de Pesquisas Espaciais – INPE, pela oportunidade de estudos e utilização de suas instalações.

Ao Laboratório de Matemática e Computação Aplicada – LAC, pela oportunidade de estudos e utilização de suas instalações.

Ao meu orientador Prof. Dr. Solon Venâncio de Carvalho pelo voto de confiança, pelo conhecimento passado e pela orientação e apoio na realização deste trabalho.

Aos professores do INPE pelo conhecimento compartilhado.

Aos membros da banca de defesa pelos valiosos comentários e recomendações.

Aos meus amigos do INPE e aos meus inumeráveis amigos que suportaram pacientemente momentos de mau humor, ausências e preocupações retribuindo com incentivo e compreensão.

RESUMO

O planejamento de sistemas públicos e privados de atendimento à população é essencial para a garantia e manutenção da qualidade dos serviços prestados. Um modelo que tem sido amplamente utilizado para o planejamento de sistemas de atendimento nos quais servidores se deslocam até clientes para prestar serviços é o modelo Hipercubo de Filas. O modelo Hipercubo de Filas é um modelo analítico estocástico que permite a avaliação de diferentes cenários de configuração de sistemas através de diversas medidas de desempenho numéricas. O cálculo de valores exatos para essas medidas de desempenho através do modelo envolve a solução de um sistema de 2^N equações lineares, o que dificulta ou inviabiliza, em alguns casos, o uso de métodos diretos ou iterativos tradicionais para a solução de modelos de sistemas de grande porte. Métodos alternativos de solução, como alguns métodos de decomposição de cadeias de Markov, podem ser aplicados ao modelo, mas as condições necessárias para sua aplicação são restritivas e seus resultados nem sempre são satisfatórios. Procurando superar estas limitações, foram desenvolvidos diversos métodos aproximados de solução do modelo que envolvem, de uma forma geral, a solução de um sistema de N equações não-lineares. Porém, os testes realizados com estes métodos foram incompletos e apenas observações gerais referentes à sua precisão foram apresentadas. Completando este cenário, foram propostas algumas modificações nos métodos aproximados com o objetivo de garantir sua convergência, mas não foram realizados testes de precisão para estas versões modificadas. Portanto, a identificação do método aproximado mais apropriado para determinado sistema é hoje baseada em conclusões fundamentadas em um pequeno conjunto de resultados sobre os quais poucos detalhes foram publicados. Neste trabalho, alguns métodos de decomposição que podem ser aplicados ao modelo Hipercubo de Filas são estudados, os métodos aproximados de solução do modelo e suas versões modificadas são revisados em relação a sua precisão e são apresentados novos resultados que estendem e completam aqueles encontrados na literatura. A relevância deste trabalho está relacionada à apresentação de novos resultados que permitem uma melhor avaliação da precisão dos métodos aproximados e suas versões modificadas.

A STUDY OF HYPERCUBE QUEUEING MODEL SOLUTION METHODS FOR LARGE SCALE SYSTEMS

ABSTRACT

The planning of public and private service systems is essential to assure the quality of the services realized by these systems. A model that has been widely used for planning server-to-customer service systems is the Hypercube Queueing model. The Hypercube Queueing model is an analytical stochastic model that allows the evaluation of different configuration scenarios of systems throught numerical performance measures. The calculation of accurate values for these performance measures through the model involves the solution of a system of 2^N linear equations, what makes it difficult or unfeasible, in some cases, the use of traditional direct or iterative methods in the solution of models for large scale systems. Alternative solution methods, as some Markov chain decompositional methods, can be applied to the model, but the necessary conditions for its application are restrictive and its results aren't always satisfactory. Looking for to surpass these limitations, many approximate procedures have been developed that involve, in general, the solution of a system of N nonlinear equations. However, the tests carried out with these methods were incomplete and only general comments about its accuracy have been made. Completing this scene, some modifications to these methods were proposed with the objective to guarantee its convergence, but have not been carried out tests of accuracy with these modified versions. Consequently, the identification of the more appropriate approximate procedure is based today on conclusions about a small set of results on which few details have been published. In this work, some decompositional methods that can be applied to the Hypercube model are studied, the approximate procedures and its modified versions are revised in despite to its accuracy and are presented new results that extends and completes those published in the literature. For the attainment of these results, 25650 cases of tests with variations in diverse model parameters were generated. The relevance of this work is related to the presentation of new results that allows a more accurate analysis of the approximate procedure and its modified versions.

SUMÁRIO

Pág.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

A/S/m	Sistema de filas onde A representa o processo de chegadas, S representa o processo de atendimento e m representa o número de servidores
BACOP	Backup Coverage
EAM	Erro Absoluto Médio
EMR	Erro Médio Relativo
FCFS	First Come, First Served
GTH	Grassman-Taksar-Heyman
LCFS	Last Come, First Served
MALP	Maximal Availability Location Problem
MCLP	Maximal Covering Location Problem
MEXCLP	Maximal Expected Covering Location Problem
M/M/N	Sistema de filas com processos de chegada e atendimento sem memória e N servidores
M/M/N/B	Sistema de filas com processos de chegada e atendimento sem memória, N servidores e com capacidade de B usuários (incluindo filas de espera)
PCAM	Patrol Car Allocation Model (Modelo de Alocação de Patrulha)
PIA	Patrol Initiated Activities
SAMU	Serviço de Atendimento Móvel de Urgência
SIATE	Serviço Integrado de Atendimento ao Trauma em Emergência
SIG	Sistema de Informações Geográficas
SIRO	Service In Random Order (service em ordem aleatória)
VSA	Veículo de Suporte Avançado
VSB	Veículo de Suporte Básico

LISTA DE SÍMBOLOS

a_{mk}	K-ésimo servidor preferencial para o átomo m.
α_m^{k-1}	Fator de ajuste utilizado para convergência do procedimento de cálculo de frações de despacho.
B_i	Evento que indica que o servidor i está ocupado.
D_n	Desvio médio da demanda atendida pelos servidores.
E_{nj}	Conjunto dos estados que resultam obrigatoriamente na alocação do servidor n para qualquer chamado originado no átomo geográfico j.
f_I	Fração de todos os atendimentos realizados fora da área de cobertura primária dos servidores.
f_{im}	Fração dos despachos que enviam o servidor n ao átomo geográfico j e que não implicam em espera em fila
f_{im}^{Q}	Fração dos despachos que enviam o servidor n ao átomo geográfico j e que implicam em espera em fila
f_{In}	Fração dos atendimentos do servidor n realizados fora de sua área de cobertura primária.
f_{nj}	Fração de todos os despachos do sistema que resultam no envio do servidor n ao átomo geográfico j.
$f_{nj}^{[1]}$	Fração dos despachos que enviam o servidor n ao átomo geográfico j e que não implicam em espera em fila
$f_{nj}^{[2]}$	Fração dos despachos que enviam o servidor n ao átomo geográfico j e que implicam em espera em fila
F_i	Evento que indica que o servidor i está livre.
λ	Taxa total de chegada de solicitações de serviço.
λ_D	Taxa de solicitações que aguardam em fila.
λ_m	Taxa de chegada de solicitações originadas no átomo m.
$\lambda_m^{normalizado}$	Taxa de chegada de solicitações de serviço normalizada originadas no átomo m
λ_{ij}	Taxa de transição entre os estados i e j.
l_{nj}	Fração de tempo durante o qual o servidor n está localizado no átomo j.
μ	Taxa total de serviço.
μ_s	Taxa de serviço por servidor.
N	Número de servidores.
N_A	Número de átomos geográficos.
π	Vetor de probabilidades estacionárias.
ρ	Taxa de ocupação ou intensidade de tráfego.
ρ_i	Carga de trabalho do servidor i.
ρ_i^t	Carga de trabalho do servidor i calculada na etapa t.

P_i	Probabilidade estacionária do estado *i*. Para modelos de fila, P_i representa a probabilidade de que *i* clientes estejam no sistema.
P_Q	Probabilidade de formação de fila.
Q	Matriz dos coeficientes.
Q(N, ρ, j)	Fator de correção.
R	Taxa ou fator médio de utilização.
$R^{(i)}$	Cadeia de Markov da disponibilidade do servidor i.
R_i^F	Taxa de solicitações atendidas pelo servidor i quando o servidor está livre
R_n	Átomo geográficos que fazem parte da área de cobertura primária do servidor n.
τ	Tempo médio de serviço do sistema
τ_{im}	Tempo médio de viagem entre os átomos i e j.
T_i	Tempo de permanência no estado *i*.
V_i	Taxa de solicitações atendidas pelo servidor i quando o servidor está livre
X_t	Processo estocástico *X* indexado pelo parâmetro *t*.
Y_N	Densidade da matriz de coeficientes de uma modelo com N servidores.

1 INTRODUÇÃO

A Pesquisa Operacional[1] é uma área de conhecimento relativamente nova. Apesar de suas origens históricas poderem ser traçadas a partir de alguns séculos atrás, o princípio da atividade que ficou conhecida com este nome é geralmente atribuído ao serviço militar inglês nos anos que antecederam a Segunda Guerra Mundial (1939-1945)[2]. Durante estes anos, cientistas de diferentes áreas analisaram problemas operacionais, tais como: controle de defesa antiaérea e de sistemas de radar, dimensionamento e roteamento de comboios, entre outros.

Após o término da guerra, as técnicas e ferramentas até então desenvolvidas passaram a ser formalizadas e aplicadas para fins não-militares, principalmente no setor privado. Esta concentração das primeiras aplicações no setor privado deveu-se, em grande parte, a similaridade entre os problemas tratados durante o período de guerra e problemas industriais. Apenas a partir da década de 60 as aplicações voltadas ao setor público passaram a receber maior atenção.

Desde essa época, uma das aplicações que tem recebido especial atenção é a localização de facilidades. Uma facilidade pode ser entendida como uma unidade ou um conjunto de unidades prestadoras de serviço (p.ex.: ambulâncias, viaturas de polícia, escolas, fábricas).

Diversos modelos de localização de facilidades têm sido propostos como ferramentas de apoio à decisão no planejamento de sistemas que prestam serviços a alguma população usuária. Estes modelos têm como característica a determinação do número de facilidades, sua localização e/ou política de operação, com o objetivo de otimizar algum critério ou medida de desempenho de um sistema, tal como: tempo de espera por serviço, área de cobertura etc.

---

[1] Pesquisa Operacional (*Operations Research* ou *Operational Research* em inglês) pode ser definida como uma disciplina de aplicação de métodos analíticos avançados que ajudam na tomada de melhores decisões (INFORMS, 2004).

[2] Para maiores detalhes sobre a história da Pesquisa Operacional, consulte Pollock e Maltz (1994) e Whitehouse e Wechsler (1976).

Esta otimização é importante principalmente no planejamento de sistemas nos quais atrasos na prestação de serviços podem provocar danos significativos à população usuária, como é o caso dos sistemas de atendimento emergencial[1] (GONÇALVES, 1994) (p.ex.: combate a incêndios, atendimento médico pré-hospitalar e patrulhamento policial).

A primeira publicação notória para a área de localização de facilidades foi escrita por Weiszfeld (1937), citado por Gass (2004), e intitulava-se “O ponto cuja soma das distâncias em relação a n pontos é minimizada”. Neste trabalho, Weiszfeld analisou o problema formulado por Pierre de Fermat em 1643: “Dados n pontos, encontre um ponto tal que a soma das distâncias deste ponto aos n pontos seja a menor possível”.

São exemplos de modelos de localização de facilidades: *Maximal Covering Location Problem* - MCLP (CHURCH; REVELLE, 1974), *Patrol Car Allocation Model* – PCAM (CHAIKEN; DORMONT, 1978a,b), *Maximal Expected Covering Location Problem* - MEXCLP (DASKIN, 1983), *Backup Coverage* - BACOP (HOGAN; REVELLE, 1986), *Maximal Availability Location Problem* - MALP (REVELLE; HOGAN, 1989), entre outros.

Como as decisões de planejamento associadas a modelos de localização geralmente envolvem custos altos (p.ex.: compra de novas facilidades, preparação de infra-estrutura para reposicionamento de facilidades etc.), a validação das configurações propostas por estes modelos é uma fase importante que deve preceder a tomada de decisão.

Esta validação pode ser feita através de métodos e modelos analíticos e de simulação. Com este propósito, o modelo Hipercubo de Filas, um modelo analítico estocástico proposto por Richard C. Larson (1973, 1974b), tem sido utilizado com sucesso desde a década de 1970 (BATTA et al., 1989;

[1] Para maiores detalhes sobre a aplicação de modelos de Pesquisa Operacional para o planejamento de sistemas de atendimento emergencial, consulte Goldberg (2004), Larson (2004) e Swersey (1994).

BENVENISTE, 1985; BERMAN et al., 1987; BERMAN; LARSON, 1982; BRANDEAU; LARSON, 1986; CHAIKEN, 1978; CHIYOSHI et al., 2000, 2003; GALVÃO et al., 2003, 2005; GOLDBERG et al., 1990; IANNONI; MORABITO, 2006b; SAYDAM et al., 1994; SAYDAM; AYTUG, 2003).

O modelo é baseado na teoria de cadeias de Markov em tempo contínuo e na teoria de filas, e considera tanto aspectos espaciais quanto temporais na modelagem de sistemas nos quais servidores viajam até clientes para prestar serviços (*server-to-customer*), como é o caso de muitos sistemas de atendimento emergencial.

Apesar de não sugerir mudanças nas configurações do sistema modelado como, por exemplo, o reposicionamento de servidores, o modelo Hipercubo possibilita a avaliação de diferentes cenários de configuração através de medidas numéricas de desempenho do sistema, como as cargas de trabalho dos servidores e o tempo médio de viagem dos servidores para atendimento às solicitações de serviço.

Alguns exemplos de aplicação do modelo no Brasil são a análise de interrupções na distribuição de energia elétrica em Santa Catarina (ALBINO, 1994), a determinação de zonas de atendimento e a localização de ambulâncias do Serviço Integrado de Atendimento ao Trauma em Emergência - SIATE da cidade de Curitiba-PR (COSTA, 2003), a localização de ambulâncias em um trecho da BR-111 (GONÇALVES et al., 1994, 1995), o balanceamento da carga de trabalho de ambulâncias no sistema “Anjos do Asfalto” da Rodovia Presidente Dutra (MENDONÇA; MORABITO, 2000), a avaliação do centro de emergência da polícia militar de Santa Catarina (OLIVEIRA, 2003) e a configuração do Serviço de Atendimento Móvel de Urgência - SAMU de Campinas-SP (TAKEDA, 2000; TAKEDA et al., 2004, 2007).

Nos Estados Unidos, são exemplos de aplicação do modelo: a localização de ambulâncias em Boston (BRANDEAU; LARSON, 1986) e Greenville

(BURWELL et al., 1993) e o patrulhamento policial em New Haven (CHELST; BARLACH, 1981) e Orlando (SACKS; GRIEF, 1994).

1.1 Definição do problema

Para poder incorporar políticas de atendimento complexas e calcular certas medidas de desempenho, o estado de cada servidor do sistema é preservado no espaço de estados do modelo Hipercubo.

Desta forma, um aumento no número de servidores está relacionado a um crescimento exponencial do espaço de estados do modelo, um problema bastante conhecido em estudos com modelos baseados em cadeias de Markov. De uma forma geral, o cálculo dos valores numéricos exatos das medidas de desempenho para sistemas que operam com N servidores e não permitem a formação de filas envolve a solução de um sistema de 2^N equações lineares.

Para sistemas de tamanho moderado, a solução deste sistema de equações pode ser obtida através de métodos diretos e iterativos tradicionais, como o método de eliminação de Gauss, Grassman-Taksar-Heyman - GTH, Gauss-Jordan ou os métodos iterativos de Gauss-Jacobi e Gauss-Seidel (apesar da convergência destes dois últimos métodos não poder ser garantida para o modelo Hipercubo).

Porém, a dificuldade de armazenamento e manipulação do grande volume de dados gerados pela descrição de sistemas de grande porte ($N > 25$) e, conseqüentemente, do cálculo de medidas de desempenho destes sistemas dificulta ou, em alguns casos, inviabiliza o uso deste modelo para a avaliação de sistemas maiores.

Este é o caso de muitos sistemas de atendimento emergencial brasileiros, como o Serviço de Atendimento Móvel de Urgência – SAMU de diversos municípios. Na cidade de Fortaleza, por exemplo, o SAMU opera com mais de

27 ambulâncias. Em Salvador são mais de 54 ambulâncias e em São Paulo são em média 90 ambulâncias.

Em alguns casos, métodos de decomposição que reduzem as exigências de memória e processamento para a solução do modelo podem ser empregados. Porém, como será visto no capítulo 4, as condições para a aplicação destes métodos são bastante restritivas e alguns deles não apresentam bons resultados quando aplicados ao modelo.

Visando contornar estas limitações, diversos métodos aproximados de solução do modelo Hipercubo foram propostos (HALPERN, 1977; LARSON, 1974a, 1975a; LARSON; ODONI, 1981; JARVIS, 1975, 1985; GOLDBERG; PAZ, 1991; BURWELL et al., 1985, 1993; CHELST; BARLACH, 1981; GAU; LARSON, 1986). Estes métodos envolvem, de uma forma geral, a solução de um sistema de N equações não-lineares, ao invés de um sistema com 2^N equações lineares do método exato.

Entretanto, apenas algumas observações gerais referentes à precisão destes métodos foram feitas e poucos detalhes foram fornecidos sobre os testes nos quais estas observações foram baseadas. Entre outras coisas, estas observações não incluíram a quantificação de alguns erros, que foram considerados apenas “significativos”. Desta forma, existem poucos parâmetros que definem quando a utilização de cada um dos métodos é mais apropriada.

A convergência destes métodos foi estudada recentemente por Goldberg e Szidarovszky (1991a,b). Os autores mostraram que se pode provar a existência e unicidade de solução dos métodos para dois conjuntos de soluções iniciais a partir da modificação de algumas de suas etapas. Entretanto, não foram apresentados resultados para a precisão destas versões modificadas dos métodos.

1.2 Objetivos

Nesse contexto, o presente trabalho procura contribuir com o estudo de dois métodos de decomposição que permitem a redução das exigências de memória e processamento para a solução do modelo, com uma revisão dos principais métodos aproximados de solução e com informações que auxiliem a decisão de qual método é mais apropriado para a solução do modelo Hipercubo.

Como ainda não foram encontrados limites analíticos para a precisão destes métodos, a análise apresentada neste trabalho é baseada nos resultados obtidos para um grande conjunto de casos de teste.

Para a realização destes testes foi desenvolvida uma biblioteca de classes orientada a objetos que permite a solução do modelo Hipercubo e de suas principais extensões. O desenvolvimento desta biblioteca foi motivado pela carência de soluções extensíveis e de domínio público para o modelo. Espera-se que esta biblioteca possa ser utilizada em trabalhos futuros com o modelo.

1.3 Estrutura do trabalho

Este trabalho foi organizado em 6 capítulos, além da introdução, e 1 apêndice:

Capítulo 2 – Conceitos Básicos

Neste Capítulo, são abordados alguns conceitos básicos relacionados aos assuntos tratados nesta dissertação. Como o modelo Hipercubo de Filas é baseado em teoria de cadeias de Markov em tempo contínuo e teoria de filas, são apresentados definições e conceitos básicos para cada uma destas teorias.

Capítulo 3 – Modelo Hipercubo de Filas

Este Capítulo trata da descrição do modelo Hipercubo de Filas proposto para permitir a avaliação de cenários alternativos de configuração de sistemas nos quais servidores se deslocam até clientes para realizar serviços (server-to-customer). São apresentadas suas principais características, as hipóteses assumidas em sua aplicação, a construção de seu espaço de estados, o método exato de solução e suas principais extensões. Por fim, é apresentado um exemplo ilustrativo de sua aplicação.

Capítulo 4 – Métodos de decomposição

Este Capítulo trata da aplicação de métodos de decomposição de cadeias de Markov ao modelo Hipercubo de Filas. Procura-se destacar a possibilidade de aplicação de diversos de métodos de decomposição ao modelo Hipercubo de Filas.

Capítulo 5 – Métodos aproximados de solução

Este Capítulo trata dos métodos aproximados de solução do modelo Hipercubo de Filas propostos para viabilizar a aplicação do modelo a sistemas de grande porte. São revisados dois dos principais métodos aproximados e são apresentados resultados para a precisão destes métodos.

Capítulo 6 – Avaliação de modificações nos métodos aproximados de solução

Este Capítulo apresenta uma análise de modificações nos métodos aproximados de solução do modelo Hipercubo de Filas. Algumas destas modificações foram retiradas da literatura, enquanto outras foram propostas neste trabalho.

Capítulo 7 – Conclusões

Neste Capítulo, são apresentadas as principais conclusões e, para finalizar, são discutidas algumas perspectivas para trabalhos futuros.

No Apêndice é apresentada a modelagem de uma biblioteca orientada a objetos para o modelo Hipercubo.

2 CONCEITOS BÁSICOS

Neste capítulo, são abordadas algumas definições e conceitos básicos sobre Teoria de Cadeias de Markov e Teoria de Filas necessários para a descrição do modelo Hipercubo de Filas e de seus métodos de solução.

Nas seções 2.1 e 2.2, alguns conceitos de Processos Estocásticos e de Processos de Markov são revisados, com atenção especial para as cadeias de Markov em tempo contínuo.

Na seção 2.3, é apresentada uma visão geral da Teoria de Filas e alguns detalhes sobre os modelos M/M/N e M/M/N/B.

2.1 Processos estocásticos

Em muitas situações reais, é possível representar o comportamento de um sistema a partir da descrição dos diferentes estados que o sistema pode ocupar e da forma como o sistema se movimenta de um estado para outro no tempo. Se esse comportamento não for determinístico, o sistema pode ser estudado através de processos estocásticos.

Um processo estocástico pode ser definido como um conjunto de variáveis aleatórias (variáveis cujos valores são o resultado de um experimento aleatório) indexadas por um parâmetro, que geralmente representa o tempo. A representação usual de um processo estocástico é $X=\{X_t, t \in T\}$, onde X é o nome do processo, X_t é o nome das variáveis aleatórias desse processo e t é o seu parâmetro indexador.

O parâmetro indexador (t) pertence a um conjunto (T), geralmente tomado como o conjunto dos números naturais ou como o conjunto dos números reais. Se o conjunto T for finito ou infinito enumerável, o processo estocástico será conhecido como um processo estocástico em tempo discreto; por outro lado, se

T for infinito não-enumerável, o processo será conhecido como um processo estocástico em tempo contínuo.

O valor assumido por uma variável aleatória X_t é conhecido como estado do processo e o conjunto de todos os estados possíveis como espaço de estados. De forma similar ao conjunto T, o espaço de estados também pode ser classificado em discreto ou contínuo.

Como exemplo ilustrativo de um processo estocástico $X=\{X_t, t \in T\}$, pode-se tomar a observação de lances sucessivos de uma moeda. O resultado possível de cada lance é “cara” ou “coroa”. Na impossibilidade de prever com exatidão o resultado de um lance qualquer, pode-se estimá-lo probabilisticamente através da distribuição de probabilidades mais adequada para a moeda utilizada. O resultado de cada lance é uma variável aleatória indexada pelo parâmetro $t \in \{1, 2, 3,...\}$ que especifica o número do lance associado a cada lance da moeda. Como os resultados possíveis e o parâmetro indexador pertencem a um conjunto discreto, pode-se modelar o exemplo apresentado como um processo estocástico em tempo discreto e com espaço de estados discreto. Assim, a variável aleatória X_8 desse processo representa o resultado do oitavo lance da moeda e pode assumir os valores “cara” ou “coroa”.

Um dos mais importantes tipos de processo estocástico, com aplicações na Engenharia, Administração, Ciências Físicas, Biológicas etc. é o processo de Markov. A seguir, é apresentada uma breve revisão sobre este tipo de processo.

2.2 Processos de Markov

Um processo de Markov[1] é um processo estocástico $X=\{X_t, t \in T\}$ cujo valor de qualquer variável X_{t+1} é independente dos valores das variáveis com índice

[1] Andrei Andreevich Markov (1865-1922). Matemático russo, aluno de Pafnuty Lvovich Chebyshev, criou o que hoje é conhecido como Processos de Markov, procurando provar que P. A. Nekrasov estava errado ao assumir que a independência é condição necessária para a lei fraca dos grandes números. Para mais detalhes, consulte Basharin et al. (2004).

menor que t se o valor da variável X_t for conhecido. Esta propriedade fundamental de um processo de Markov, designada “propriedade markoviana”, especifica que o futuro do processo é condicionalmente independente do passado, dado o presente.

Matematicamente isso significa que para qualquer número inteiro n, quaisquer estados x_0, x_1, x_2, ..., x_{n+1} e qualquer seqüência t_0, t_1, t_2, ..., t_n, t_{n+1}, tal que $t_0 < t_1 < t_2 < ... < t_n < t_{n+1}$:

$$P(X_{t_{n+1}} = x_{n+1} \mid X_{t_0} = x_0, X_{t_1} = x_1, ..., X_{t_n} = x_n) = P(X_{t_{n+1}} = x_{n+1} \mid X_{t_n} = x_n) \qquad (2.1)$$

Para satisfazer esta propriedade, a distribuição do tempo de permanência nos estados do processo deve apresentar a propriedade de falta de memória (*memoryless*).

Os processos estocásticos markovianos que possuem espaço de estados discreto são denominados simplesmente “Cadeias de Markov”. Como o modelo Hipercubo de Filas, apresentado no Capítulo 3, é baseado na teoria de cadeias de Markov em tempo contínuo, a seguir serão revisados alguns aspectos desse tipo de processo.

2.2.1 Cadeias de Markov em tempo contínuo

As cadeias de Markov em tempo contínuo são processos de Markov que possuem espaço de estados discreto e tempo contínuo, com tempos de permanência nos estados distribuídos exponencialmente. A distribuição exponencial especifica que, dado que o processo permaneceu s unidades de tempo em um estado i, a probabilidade de que o processo permaneça mais t unidades de tempo nesse estado é igual à probabilidade de que o processo permaneça t unidades de tempo no estado i, independente de s, isto é, a distribuição apresenta a propriedade de falta de memória. Assim sendo:

$$P(T_i > s + t \mid T_i > s) = P(T_i > t) \quad \forall\, s,t \in \mathrm{R}^+ \qquad (2.2)$$

O tempo de permanência nos estados de cadeias de Markov é parametrizado por taxas de transição. As taxas de transição são parâmetros da distribuição exponencial que especificam o número médio de transições entre os estados da cadeia que ocorrem por unidade de tempo.

O estudo de sistemas através de processos de Markov tem como objetivo, em geral, a observação do sistema em regime estacionário (ou estado estacionário). O estado estacionário pode ser entendido como o estado do sistema no qual nenhuma variação significativa é percebida nas distribuições de probabilidade associadas as suas medidas de operação.

2.2.1.1 Estado estacionário de cadeias de Markov

O estado estacionário de cadeias de Markov é obtido a partir da solução de um sistema de equações lineares que representam o equilíbrio entre o fluxo de entrada e o fluxo de saída dos estados (TIJMS, 1994). Para derivar estas equações de equilíbrio, para cada estado, deve-se igualar o fluxo de entrada ao fluxo de saída do estado.

Para uma cadeia de Markov com três estados, conforme ilustrado na Figura 2.1, a equação de equilíbrio do estado 0 pode ser derivada identificando-se o fluxo de saída do estado 0 para outros estados, multiplicado pela probabilidade do estado 0, e o fluxo de entrada nesse estado a partir de qualquer outro estado que esteja a ele conectado, multiplicado pelas probabilidades dos respectivos estados. Assim, para o exemplo apresentado, tem-se:

$$(\lambda_{01} + \lambda_{02})\, p_0 = \lambda_{10}\, p_1 + \lambda_{20}\, p_2 \qquad (2.3)$$

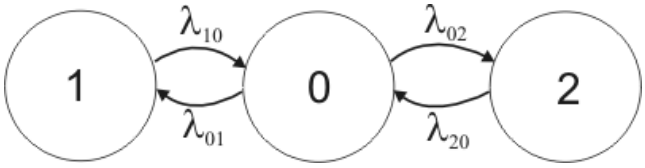


Figura 2.1 – Espaço de estados de uma cadeia de Markov.

As equações para os outros estados são:

$$(\lambda_{10})P_1 = \lambda_{01}P_0 \tag{2.4}$$

$$(\lambda_{20})P_2 = \lambda_{02}P_0 \tag{2.5}$$

Estas equações podem ser postas em forma matricial, de tal forma que as probabilidades dos estados formem um vetor linha (π) e as taxas de transição formem uma matriz retangular (Q):

$$\pi Q = 0 \quad \text{ou} \quad Q^T \pi = 0 \tag{2.6}$$

Para o exemplo apresentado na Figura 2.1, o vetor π e a matriz Q, conhecida como matriz dos coeficientes, podem ser definidos como:

$$\pi = \begin{bmatrix} P_0 & P_1 & P_2 \end{bmatrix} \tag{2.7}$$

$$Q = \begin{array}{c} 0 \\ 1 \\ 2 \end{array} \begin{bmatrix} -(\lambda_{01}+\lambda_{02}) & \lambda_{10} & \lambda_{20} \\ \lambda_{01} & -\lambda_{10} & 0 \\ \lambda_{02} & 0 & -\lambda_{20} \end{bmatrix} \tag{2.8}$$

As equações de estados de cadeias de Markov impõem condições de equilíbrio para cada estado do sistema, mas nada especificam sobre a forma como a massa total de probabilidades se distribui entre estes estados, tornando o sistema possível e indeterminado.

Para eliminar esta indeterminação, deve-se introduzir uma equação de normalização, que leva em consideração que a soma das probabilidades de todos os estados possíveis do sistema é igual a 1:

$$P_0 + P_1 + ... + P_n = 1 \tag{2.9}$$

Esta equação, associada a Equação 2.6, forma o sistema que deve ser resolvido para o cálculo das probabilidades do sistema em regime estacionário.

A seguir, um tipo de sistema que geralmente é modelado através de cadeias de Markov é discutido.

2.3 Teoria de filas

Em sistemas nos quais servidores (p.ex.: caixas de banco ou supermercados, pedágios etc.) prestam serviços para clientes (p.ex.: pedidos, pessoas, veículos etc.) pode ocorrer a formação de filas de espera. As filas formam-se principalmente por flutuações aleatórias na chegada de clientes e nos tempos de serviço. Sistemas desta natureza podem ser estudados através da Teoria de Filas.

A Teoria de Filas[1] é uma área da Pesquisa Operacional que auxilia o planejamento de sistemas de filas através de uma análise matemática que permite que diversas medidas de desempenho dos sistemas estudados sejam calculadas.

De uma forma geral, um sistema de filas pode ser caracterizado através de informações sobre a população usuária do sistema, o processo de chegada de indivíduos desta população (solicitações) ao sistema, o número de servidores do sistema, o processo de atendimento destes servidores, sua disciplina de serviço e o tamanho máximo da fila do sistema.

Em muitos sistemas reais, a população usuária do sistema é finita. Entretanto, se ela é muito grande, é vantajoso para a análise considerá-la infinita. O processo de chegadas especifica como os indivíduos da população usuária chegam ao sistema. Em termos quantitativos, este processo pode ser expresso pelo número médio de solicitações que chegam ao sistema por unidade de tempo (ou pelo tempo médio entre a chegada de solicitações sucessivas).

[1] Os fundamentos da Teoria de Filas podem ser encontrados em Gross e Harris (1998), Mussen (1987), Cooper (1981), Saaty (1961), entre outros.

Em sistemas nos quais não ocorrem variações aleatórias no processo de chegadas (determinísticos), estes valores são suficientes para descrever o processo. Por outro lado, se o sistema apresenta natureza estocástica (o que ocorre em grande parte dos sistemas reais), é necessária a definição de uma distribuição de probabilidades para determinar o processo de chegadas, ou ainda, se os tempos entre chegadas consecutivas não são estocasticamente independentes, é necessária uma descrição completa do processo.

O número de servidores, por sua vez, representa o número de unidades que podem prestar serviços simultâneos no sistema. Para um supermercado, este número seria igual ao número de caixas de atendimento em operação.

Assim como o processo de chegadas, o processo de atendimento dos servidores pode ser expresso pelo número médio de solicitações atendidas por unidade de tempo (ou pelo tempo médio de atendimento das solicitações). As mesmas observações feitas em relação à natureza determinística ou estocástica do processo de chegadas podem ser estendidas ao processo de atendimento.

A disciplina de serviço define como as solicitações que chegam ao sistema são atendidas pelos servidores. As disciplinas mais comuns são: FCFS (*first-come*, *first-served*), ou seja, os clientes recebem atendimento em ordem de chegada; LCFS (*last-come*, *first-served*), ou seja, o último cliente a entrar no sistema será o primeiro a receber atendimento; SIRO (*service in random order*), na qual os clientes são escolhidos aleatoriamente; filas com prioridade, entre outras.

Por fim, o tamanho máximo da fila do sistema representa o número máximo de solicitações que podem aguardar por atendimento no sistema quando todos os servidores estão ocupados realizando serviços.

Para facilitar a descrição destas características dos sistemas de filas, David Kendall (1951) propôs uma notação que envolve a especificação de três informações: A/S/m, onde A representa o processo de chegadas; S representa

o processo de atendimento; e m representa o número de servidores que operam no sistema.

Entre os valores que A e S podem assumir, destacam-se: D – processo determinístico, M – processo sem memória e G – processo com distribuição geral.

Além dessas informações, podem ser especificados a capacidade do sistema (considerando o número máximo de solicitações em fila), a disciplina de serviço, entre outros, na ordem que aparecem no texto anterior.

Diversos modelos de sistemas de fila foram estudados. Os modelos de interesse deste trabalho são: M/M/N e M/M/N/B.

Estes modelos representam sistemas de filas com N servidores, processo de chegada e atendimento sem memória e disciplina de serviço FCFS. Estes dois modelos podem ser especificados através de cadeias de Markov em tempo contínuo. A seguir, as principais características de cada um destes modelos são apresentadas.

2.3.1 M/M/N

No modelo M/M/N, não existe limite para o tamanho da fila e da população usuária do sistema. Assim sendo, para que um sistema possa ser modelado, a taxa de chegada λ deve ser menor que a taxa total de serviço ($\mu = N \cdot \mu_s$), com μ_s igual à taxa de serviço por servidor, pois, caso contrário, o tamanho da fila aumentaria indefinidamente. Em outras palavras, a razão $\rho=\lambda/\mu$ (conhecida como intensidade de tráfego ou taxa de ocupação) deve ser menor que a unidade. Caso contrário, o sistema nunca entrará em equilíbrio.

A utilização média dos servidores (r) para este tipo de sistema é igual a taxa de ocupação do sistema (ρ).

$$r = \rho \tag{2.10}$$

Os estados do modelo representam o número de clientes que estão sendo atendidos e em espera em um determinado momento. A figura seguinte representa alguns estados deste modelo e suas possíveis transições.

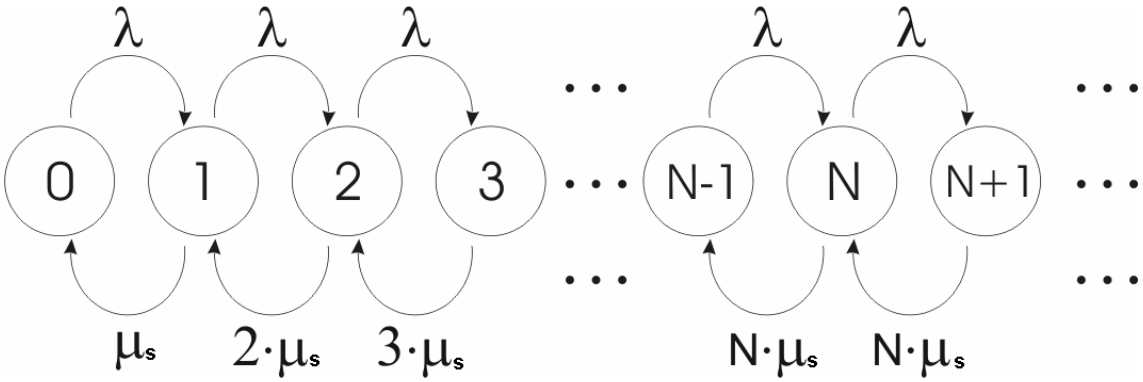


Figura 2.2 – Espaço de estados do modelo de filas M/M/N.

A probabilidade de que k clientes estejam no sistema é igual a probabilidade estacionária do estado correspondente da cadeia de Markov. Esta probabilidade tem dois comportamentos distintos: um para quando o sistema está saturado (todos servidores ocupados) e outro para quando pelo menos um servidor está livre:

$$P_k = P_0 \frac{(N\rho)^k}{k!}, \text{ para k<N} \tag{2.11}$$

$$P_k = P_0 \frac{\rho^k N^N}{N!}, \text{para k}\geq\text{N} \tag{2.12}$$

onde:

$$P_0 = \frac{1}{1 + \frac{(N\rho)^N}{N!(1-\rho)} + \sum_{k=1}^{N-1} \frac{(N\rho)^k}{N!}} = 1 - \rho \tag{2.13}$$

A partir destas probabilidades, diversas medidas de desempenho do sistema podem ser calculadas, como a probabilidade de que uma solicitação que chega ao sistema aguarde em fila:

$$P_Q = \frac{(N\rho)^N}{N!(1-\rho)} P_0 \qquad (2.14)$$

2.3.2 M/M/N/B

A diferença entre o modelo M/M/N e o M/M/N/B está na fila do sistema. O modelo M/M/N/B permite a formação de fila de espera finita de tamanho máximo (B-N). Como o tamanho máximo da fila é finito, o sistema sempre entra em equilíbrio e não há restrição para a intensidade de tráfego. O espaço de estados deste modelo é similar ao do modelo M/M/N com exceção de que os estados que representam fila são finitos.

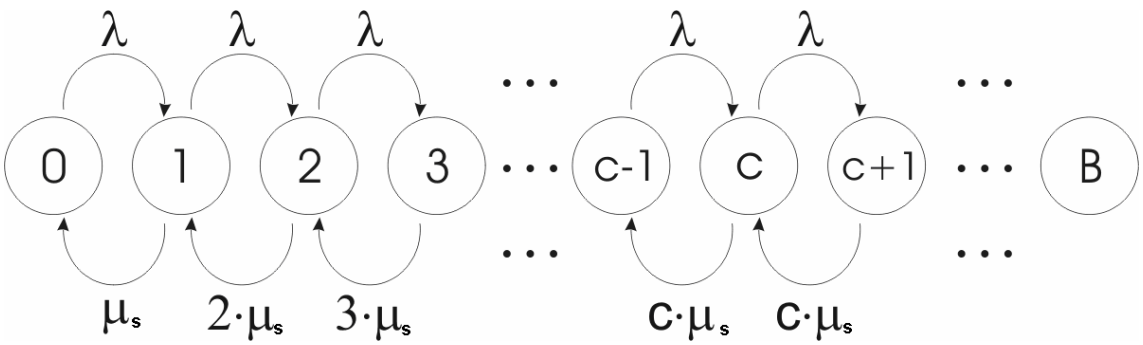


Figura 2.3 – Espaço de estados do modelo de filas M/M/N/B.

A probabilidade de que os servidores estejam ociosos tem dois comportamentos distintos: um para intensidades de tráfego iguais à unidade e outro, caso contrário.

$$P_0 = \frac{1-\rho}{1-\rho^{B+1}}, \qquad \text{para } \rho \neq 1 \qquad (2.32)$$

$$P_0 = \frac{1}{B+1}, \qquad \text{para } \rho = 1 \qquad (2.15)$$

A probabilidade de que existam k serviços em andamento no sistema pode ser calculada a partir das equações 2.11 e 2.12 (do sistema de filas M/M/N), sendo que a equação 2.12 se aplica para N <= k <= B.

Quando não é permitida a formação de filas, o sistema comporta-se de acordo com um modelo M/M/N/N. Neste caso, a probabilidade de que k clientes estejam no sistema pode ser calculada a partir da equação 2.11 (do modelo de filas M/M/N), com P_0 substituído pela equação 2.16:

$$P_0 = \frac{1}{\sum_{i=0}^{N} \frac{(Np)^i}{i!}} \tag{2.16}$$

A probabilidade de formação de fila é zero ($P_Q = 0$) e a fração de tempo durante a qual os servidores estão livres é menor que ρ, já que as solicitações que chegam quando todos servidores estão ocupados são perdidas.

$$r = \rho(1 - P_N) \tag{2.17}$$

Revisados estes conceitos básicos, o modelo Hipercubo de Filas, baseado em cadeias de Markov em tempo contínuo e em teoria de filas será revisado.

3 MODELO HIPERCUBO DE FILAS

Este capítulo trata da descrição do modelo Hipercubo de Filas e de suas principais extensões, valiosas ferramentas para o planejamento de sistemas nos quais servidores se deslocam até clientes para realizar serviços (*server-to-customer*).

Após algumas considerações iniciais serem feitas na seção 3.1, nas seções subseqüentes são apresentadas as principais características do modelo, as hipóteses que validam sua aplicação, a descrição de seu espaço de estados, seu método exato de solução, as principais medidas de desempenho que podem ser calculadas através do modelo, suas principais extensões e um exemplo ilustrativo de sua aplicação.

3.1 Considerações iniciais

O modelo Hipercubo de Filas (LARSON, 1973, 1974b) é um modelo analítico estocástico baseado na teoria de cadeias de Markov em tempo contínuo e na teoria de filas. O modelo considera tanto aspectos espaciais quanto temporais de sistemas nos quais servidores se deslocam até clientes para realizar serviços.

Estes sistemas são caracterizados no modelo como uma extensão do modelo de filas M/M/N na qual a identidade de cada servidor é preservada no espaço de estados e políticas de atendimento complexas são permitidas.

Entre as diversas aplicações do modelo está o planejamento de áreas de atuação e patrulhamento policial, de áreas de cobertura para ambulâncias ou veículos de reparo, de visitas do serviço social, entre outras (LARSON; ODONI, 1981).

Historicamente, o modelo Hipercubo pode ser visto como uma evolução do modelo GEOQUEUE proposto por Gregory Lewis Campbell em 1972 (LARSON, 1973, p. 5,1974b, p. 72).

O modelo Hipercubo baseia-se na partição da área de atendimento de um sistema em um conjunto finito de fontes independentes geradoras de solicitações de serviço, conhecidas como átomos geográficos.

As solicitações geradas nestes átomos geográficos são atendidas por servidores (p.ex.: patrulhas, ambulâncias etc.) que podem estar fixos (limitados a um átomo geográfico) ou móveis dentro de uma determinada região (conjunto de átomos geográficos). Os átomos geográficos para os quais um servidor é despachado (quando disponível), mesmo quando os outros servidores estão disponíveis, compreendem a área de cobertura primária do servidor.

A Figura 3.1 exemplifica uma região particionada em 5 átomos geográficos e a distribuição espacial de 3 servidores.

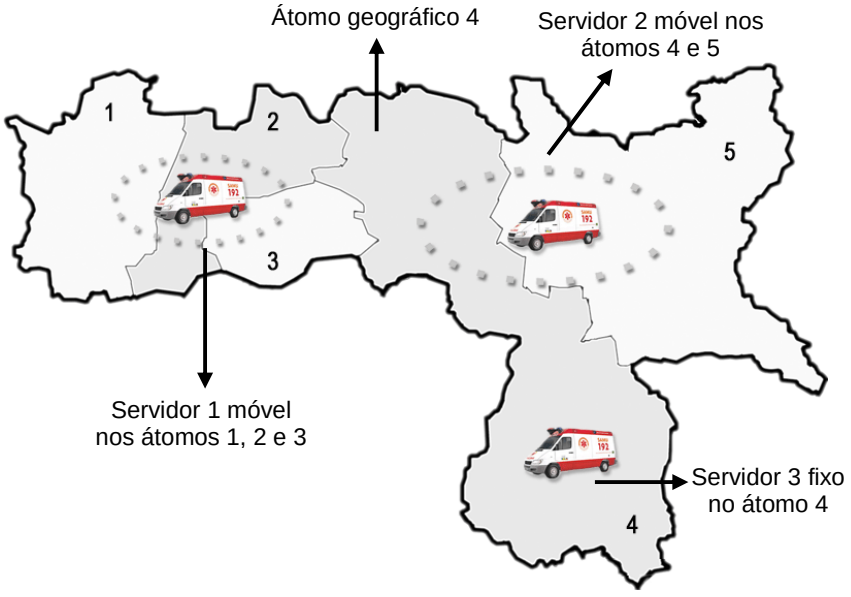


Figura 3.1 – Representação da região de um sistema hipotético.

O nome Hipercubo é derivado da representação do espaço de estados do modelo. Neste espaço de estados, cada servidor pode encontrar-se, em um instante determinado, disponível (0) ou ocupado (1). Um estado particular do modelo é descrito pela lista (geralmente, representada da direita para a esquerda) dos servidores que estão disponíveis e daqueles que estão ocupados. Por exemplo, para um sistema com 3 servidores, um estado possível do modelo seria a disponibilidade de todos os três servidores (000), outro seria a disponibilidade do servidor 1 e a indisponibilidade dos servidores 2 e 3 (110) etc.

Para sistemas com 3 servidores, o espaço de estados do modelo pode ser representado através de um cubo (Figura 3.2). À medida que o número de servidores cresce, a representação pode ser feita através de um hipercubo de dimensão N (com N igual ao número de servidores do modelo).

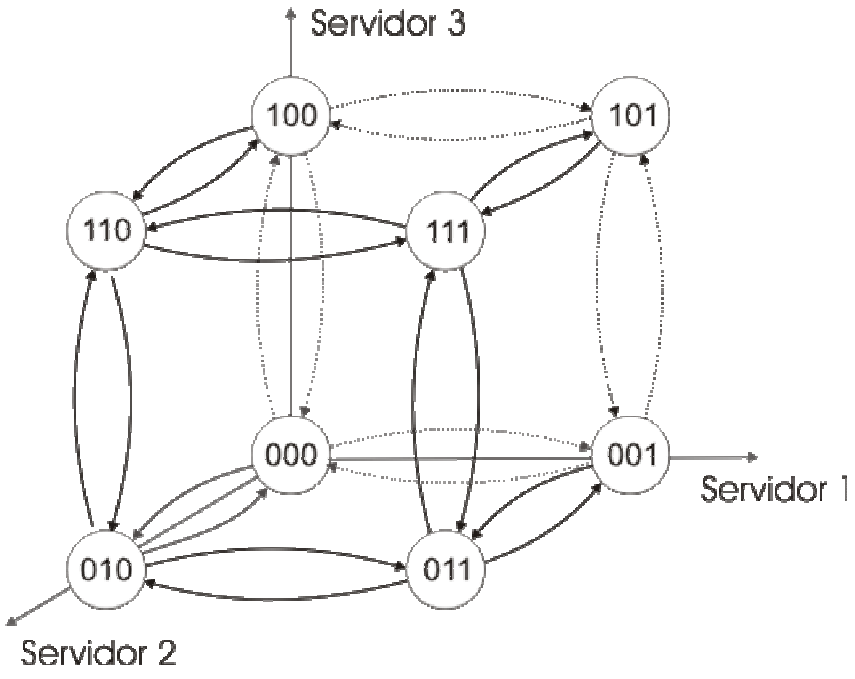


Figura 3.2 – Espaço de estados para sistemas com três servidores (com todas transições possíveis) sem formação de fila.

Apesar de terem sido representadas todas as transições possíveis em um modelo com 3 servidores, alguns modelos podem não permitir certas transições por restrições na política de despacho dos servidores. Como exemplo, para um sistema com 3 servidores que possui política de despacho idêntica para todos os átomos geográficos nunca ocorrerá alocação do terceiro servidor preferencial quando apenas o primeiro estiver ocupado, ou seja, não existirá transição entre os estados 001 e 101.

Este espaço de estados é modificado caso o sistema permita a formação de fila de espera, quando solicitações chegam ao sistema e não existem servidores disponíveis. Neste caso, o espaço de estados é acrescido de uma "cauda" (finita ou infinita, dependente do tamanho da fila) a partir do estado no qual todos os servidores estão ocupados (1…1), que representa as solicitações que aguardam serviço (Figura 3.3). Estas solicitações são atendidas segundo a disciplina FCFS, apresentada no capítulo anterior.

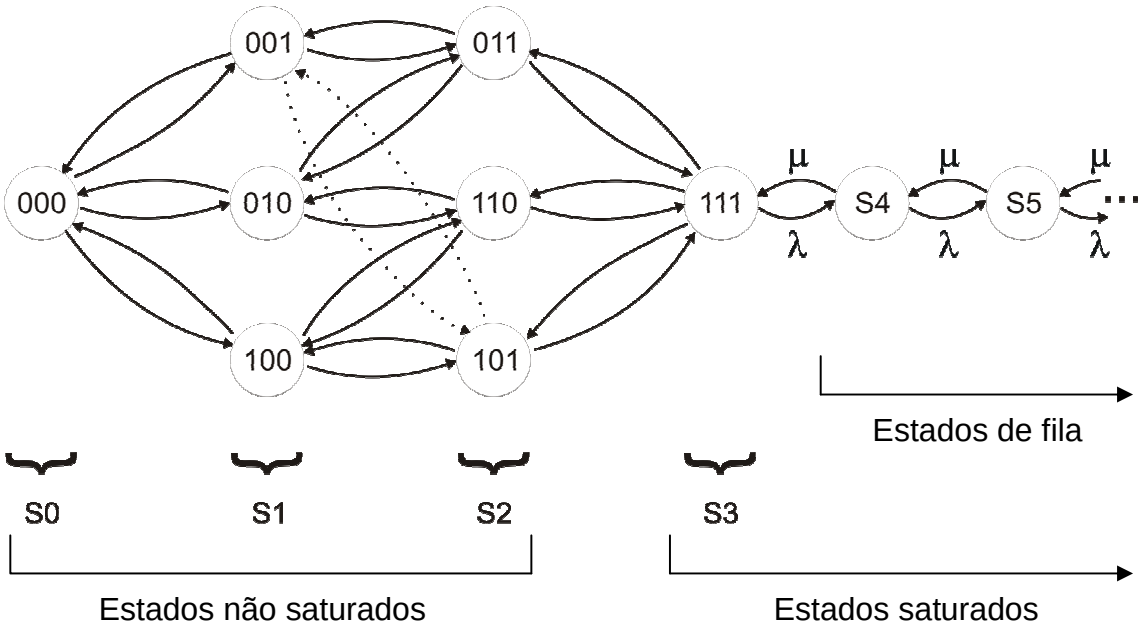


Figura 3.3 – Espaço de estados para sistemas com três servidores (com todas transições permitidas) com formação de fila. Nesta figura, μ representa a taxa total de serviço e λ representa a taxa total de chegada de solicitações ao sistema.

Fonte: Adaptada de Larson e Odoni (1981).

3.2 Hipóteses do modelo

A versão original do modelo, conforme descrita por Larson e Odoni (1981), supõe as seguintes hipóteses que devem ser satisfeitas para que se possa aplicá-lo a um sistema qualquer:

1) Átomos geográficos: A área de cobertura do sistema pode ser particionada em um número N_A de regiões conhecidas como “átomos geográficos” ou “células”, cada qual representando uma fonte de solicitações de serviço;

2) Solicitações de serviço independentes e distribuídas de acordo com processos de Poisson: As solicitações de serviço originadas em cada átomo geográfico j (j = 1, 2,..., N_A) são independentes das solicitações geradas em outros átomos geográficos e estão distribuídas de acordo com um processo de Poisson com taxa média λ_j, constante no tempo e conhecida;

3) Tempo de viagem: Os tempos médios de viagem τ_{ij} entre o átomo geográfico i e o átomo geográfico j (i, j=1, 2,..., N_A) são conhecidos. Quando não existem dados históricos que permitem estimar empiricamente os tempos de viagem, pode-se considerá-los proporcionais a métricas de distância, como a métrica de distância Manhattan ou ângulo-reto (LARSON, 1975b).

4) Servidores: O sistema é composto por um número N de servidores espacialmente distribuídos ao longo da área de cobertura, que podem se deslocar e atender às solicitações de serviço originadas em qualquer um dos átomos geográficos;

5) Localização dos servidores: A localização de cada servidor, quando disponível para atender às solicitações de serviço, é conhecida. Os servidores estão fixos em um átomo geográfico (p.ex.: ambulâncias em bases) ou móveis dentro de uma determinada região (p.ex.:

viaturas de patrulhamento policial). A localização dos servidores é representada por uma matriz L, cujo elemento l_{nj} representa a probabilidade de que o servidor n esteja localizado no átomo geográfico j em um determinado momento;

6) Alocação de servidores: Em resposta a cada solicitação de serviço, exatamente um servidor é despachado para o local da solicitação. Se não houver servidor disponível, a solicitação poderá entrar em fila com política de atendimento FCFS;

7) Política de despacho de servidores: Há uma política fixa de preferência de despacho para cada átomo geográfico. Se o primeiro servidor desta lista estiver disponível, ele é despachado para atender às solicitações de serviço originadas no átomo geográfico, caso contrário, o próximo servidor disponível na lista (*backup*) é despachado. A construção de políticas fixas de preferências de despacho é difícil porque muitos sistemas operam sem listas ou com listas parcialmente definidas. Nestes casos, alguma política de despacho que melhor represente a política real de operação deve ser escolhida (CHELST, 1975, p. 14).

8) Tempo de atendimento: O tempo total de atendimento de uma solicitação de serviço inclui: o tempo de preparo do servidor (*setup*), tempo de viagem até o local, tempo de execução do serviço junto ao usuário e o tempo de retorno à base. Os servidores podem ter tempos médios de atendimento distintos. Para sistemas com servidores homogêneos, a hipótese dos tempos de serviço exponenciais torna-se menos restritiva, pois os resultados do sistema de filas M/M/N são iguais aos resultados do sistema de filas M/G/N (MENDONÇA; MORABITO, 2000, p.77);

9) Dependência do tempo de atendimento em relação ao tempo de viagem: Variações no tempo de atendimento causadas por variações

no tempo de viagem são de ordem secundária, quando comparadas com as variações de tempo de execução e/ou tempo de preparação.

Apesar de algumas destas hipóteses serem restritivas, diversas extensões que possibilitam relaxá-las foram propostas (BRANDEAU; LARSON, 1986; BURWELL et al., 1993; CHELST; BARLACH, 1981; CHELST; JARVIS, 1979; GAU; LARSON, 1988; HALPERN, 1977; IANNONI, 2005; IANNONI; MORABITO, 2006a; JARVIS, 1985; LARSON; ODONI, 1981; LARSON; MCKNEW, 1982; MENDONÇA; MORABITO, 2000; SWERSEY, 1994). Mais adiante, algumas destas extensões serão revisadas.

3.3 Transições entre estados

As transições entre os estados do modelo se dão de modo similar ao das transições de modelos de fila clássicos, ou seja, a probabilidade de que duas solicitações cheguem simultaneamente ao sistema é nula, como também é nula a probabilidade de que dois servidores ocupados tornem-se livres simultaneamente.

Em síntese, qualquer transição de um passo é permitida e não são permitidas transições de mais de um passo. As taxas de transição entre estes estados estão relacionadas aos tempos de atendimento, às taxas de solicitação de serviço e à política de despacho dos servidores, verificadas nas hipóteses 2, 7 e 8.

Para derivar a taxa de transição de um estado x para um estado y (com $x \neq y$), deve-se verificar quais servidores estão ocupados no estado x e analisar, através da tabela de preferências de despacho, quais átomos devem solicitar serviço ou quais servidores devem concluir o serviço para que o modelo passe para o estado y.

3.4 Equações de equilíbrio

A solução exata do modelo pode ser obtida a partir da construção das equações de equilíbrio, assumindo-se que o estado estacionário (*steady state*) é atingido. Conforme visto anteriormente para cadeias de Markov em tempo contínuo, pode-se construir estas equações através do estabelecimento de igualdade entre os fluxos de entrada e saída de cada estado do modelo.

Ao sistema de equações formado, deve ser adicionada a equação de normalização das probabilidades.

$$P_{0\cdots0} + P_{0\cdots1} + \cdots + P_{1\cdots1} = 1 \qquad (3.1)$$

Os sistemas que permitem a formação de filas, tanto finita quanto infinita, exigem a inclusão das equações de equilíbrio dos estados de fila no sistema, bem como a alteração da equação de normalização anterior.

Em sistemas nos quais todos os servidores apresentam o mesmo tempo médio de atendimento (homogêneos), pode-se estabelecer uma relação entre o modelo e um sistema de filas M/M/N e, desta forma, reduzir o número de incógnitas nas equações do método exato.

Estas equações podem ser resolvidas através de métodos diretos e iterativos tradicionais, como o método de eliminação de Gauss, Grassman-Taksar-Heyman (GTH), Gauss-Jordan ou os métodos iterativos de Gauss-Jacobi e Gauss-Seidel. Chiyoshi et al. (2001) realizaram um estudo comparativo entre alguns destes métodos e mostraram que, para o conjunto de casos de testes estudado, o método de Gauss-Seidel é mais apropriado que o método de Gauss-Jacobi para sistemas com um grande número de servidores.

Apesar de Larson (1973, 1974b) ter afirmado que a convergência do método de Gauss-Seidel poderia ser garantida para o modelo Hipercubo (dada a matriz do modelo ser diagonalmente dominante), não é difícil perceber que a matriz do

modelo Hipercubo não é diagonalmente dominante e que, portanto, não há como garantir a convergência do método. Em conversa recente com Larson, foi confirmado o erro na afirmação feita nestes dois trabalhos (LARSON, 2007).

Na solução exata do modelo, as exigências de memória tornam-se restritivas antes das exigências de processamento. Mesmo quando alguma estrutura especial é utilizada para armazenar a matriz de coeficientes, como matrizes esparsas, a solução do modelo pode tornar-se proibitiva para sistemas com mais de 25 servidores, dada às limitações de memória dos computadores atuais.

A densidade da matriz (Y_N) do modelo (medida em número de elementos não-nulos dividido pelo número total de elementos) pode ser obtida através de (LARSON, 1973):

$$Y_N = \frac{(N+1)2^N}{2^{2N}} = \frac{N+1}{2^N} \qquad (3.2)$$

Para N=10, por exemplo, a densidade Y_{10}= 0,0107, o que mostra que a matriz realmente deve ser armazenada em um formato esparso.

Para o caso de servidores homogêneos, Larson propôs uma forma de armazenamento que pode reduzir significativamente a memória necessária para armazenar a matriz de coeficientes.

A forma proposta por Larson consiste em armazenar apenas as taxas de transição de alocação da matriz de coeficientes, já que as taxas de transição de conclusão de serviço seguem um padrão e não precisam ser armazenadas.

A seguir, são listadas as exigências de memória para armazenar a matriz de coeficientes do modelo Hipercubo (considerando 4 bytes para armazenar os índices e 4 bytes para armazenar os valores) para sistemas com diferentes valores de N e diferentes formas de armazenamento:

Tabela 3.1 – Memória (em MB) necessária para armazenar a matriz de coeficientes do modelo Hipercubo, considerando 4 *bytes* para armazenar os índices e 4 *bytes* para armazenar os valores.

Número de Servidores (N)	Densa (M_D)	Linha/Coluna Esparsa Compactada ($M_{CRS/CRC}$)	Formato proposto por Larson (M_{Larson})
15	4096	3,62	1,18
16	16384	7,75	2,49
17	65536	16,5	5,24
18	262144	35	11
19	1048576	74	23
20	4194304	156	48
21	1,68E+07	328	100
22	6,71E+07	688	208
23	2,68E+08	1440	432
24	1,07E+09	3008	896
25	4,29E+09	6272	1856
	Memória necessária em Megabytes (MB)		

Estes valores foram obtidos a partir das seguintes equações para o armazenamento em matrizes densas e esparsas nos formatos Linha/Coluna Esparsa Compactada (Compressed Row/Column Storage), respectivamente.

$$M_D = (2^N)^2 \cdot tipo\_val \tag{3.3}$$

$$M_{CRS/CRC} = (N \cdot 2^N + 1) \cdot tipo\_ind + ((N-1) \cdot 2^N) \cdot tipo\_val \tag{3.4}$$

onde tipo_ind e tipo_val representam a quantidade de bytes necessária para armazenar o índice e os valores da matriz, respectivamente.

Apesar de Larson (1973, 1974b) não ter derivado uma fórmula para calcular a quantidade de memória necessária para armazenar a matriz de transições de um modelo com N servidores, não é difícil verificar que essa fórmula é igual a:

$$M_{Larson} = \frac{N \cdot N!}{\left(\prod_{i=1}^{N} i!(N-i)!\right)^2} \cdot tipo\_val + 2^N \cdot tipo\_ind \tag{3.5}$$

Esta fórmula foi derivada do número de alocações que podem ser realizadas para o conjunto de estados com i servidores ocupados (que é igual ao produto de *N-i* pela combinação de N elementos i a i) adicionada ao número de estados do modelo (para armazenar a matriz de valores).

3.5 Medidas de desempenho do sistema

Satisfeitas as hipóteses apresentadas, o modelo pode ser resolvido e diversas medidas de desempenho do sistema modelado podem ser calculadas, como a carga de trabalho (tempo médio de ocupação) dos servidores, as frações de despacho, o tempo médio de viagem etc. Estas medidas de desempenho são derivadas das probabilidades de equilíbrio dos estados do modelo.

A seguir, as principais medidas de desempenho que podem ser calculadas através do modelo são revisadas. Nestas medidas de desempenho, λ representa a taxa total de chegada no sistema de solicitações de serviço e μ representa a taxa total de serviço.

3.5.1 Carga de trabalho dos servidores

A carga de trabalho (*workload*) ρ_n do servidor n (n=1,2,...,N), definida como sendo a fração da unidade de tempo durante a qual o servidor está ocupado, pode ser calculada a partir da soma das probabilidades dos estados nos quais o servidor n está ocupado, ou seja:

$$\begin{aligned} \rho_1 &= P_{0\cdots01} + P_{0\cdots11} + P_{1\cdots11} \\ \rho_2 &= P_{0\cdots10} + P_{0\cdots11} + P_{1\cdots11} \\ &\vdots \\ \rho_n &= P_{1\cdots00} + P_{11\cdots0} + P_{1\cdots11} \end{aligned} \tag{3.6}$$

3.5.2 Fração de despacho dos servidores

Para calcular outras medidas de desempenho do sistema, a fração (f_{nj}) de todos os despachos do sistema que resultam no envio do servidor n (n=1,2,...,N) a um determinado átomo geográfico j (j=1,2,...,N_A), deve ser calculada.

Esta fração é igual à soma de dois termos: (1) a fração dos despachos que enviam o servidor n ao átomo geográfico j e que não implicam em espera em fila ($f_{nj}^{[1]}$); e (2) a fração desses despachos que implicam em espera em fila ($f_{nj}^{[2]}$).

Sendo E_{nj} o conjunto dos estados que resultam obrigatoriamente na alocação do servidor n para qualquer chamado originado no átomo geográfico j, pode-se definir f_{nj} algebricamente como:

$$f_{nj} = f_{nj}^{[1]} + f_{nj}^{[2]} = \frac{\lambda_j}{\lambda}\left(\sum_{e \in Enj} P_e + (P_{1...1} + P_Q)\frac{\mu_n}{\mu} \right) \quad (3.7)$$

A partir do valor de f_{nj}, podem ser calculadas outras medidas de desempenho, como a fração de todos os atendimentos realizados fora da área de cobertura primária dos servidores (f_I), esta fração para os atendimentos realizados por um servidor específico (f_{In}), entre outras.

A fração dos atendimentos realizados fora da área de cobertura primária dos servidores é igual à soma, para cada servidor, das frações de despacho que resultam no envio do servidor em questão aos átomos que não fazem parte da área de cobertura primária (R_n) do servidor em questão. Algebricamente:

$$f_I = \sum_{n=1}^{N} \sum_{j \notin Rn} f_{nj} \quad (3.8)$$

Analogamente, a fração dos atendimentos realizados fora da área de cobertura primária de um servidor n qualquer pode ser calculada algebricamente por:

$$f_{In} = \frac{\sum_{j \notin Rn} f_{nj}}{\sum_{j=1}^{N_A} f_{nj}} \quad (3.9)$$

3.5.3 Tempo de viagem

As medidas relacionadas aos tempos de viagem são obtidas a partir da matriz de localizações dos servidores (L) e da matriz de tempo médio de viagem entre os átomos geográficos (T).

Quando um servidor n qualquer está disponível, a probabilidade de que ele esteja localizado em um átomo geográfico i qualquer é igual a l_{ni} (l_{ni} é um elemento da matriz de localização dos servidores. Ver hipótese 5 do modelo).

Dado que o servidor n está localizado no átomo i, o tempo médio de viagem deste servidor ao átomo geográfico i é igual a τ_{ij}.

Assim sendo, o tempo médio de viagem necessário para que o servidor n, quando disponível, viaje ao átomo geográfico j é igual à soma, para todos os átomos geográficos, da probabilidade de que este servidor esteja em cada átomo geográfico multiplicada pelo tempo médio de viagem deste átomo ao átomo geográfico j:

$$t_{nj} = \sum_{i=1}^{N_A} l_{ni} \tau_{ij} \tag{3.10}$$

Se o sistema está em fila, todos os servidores estão ocupados atendendo chamados de algum átomo geográfico do sistema. Portanto, assim que um servidor n qualquer terminar o atendimento, ele terá que viajar ao átomo geográfico j. Como a probabilidade de que o servidor n esteja atendendo um chamado no átomo i é igual a λ_i/λ e a probabilidade de que o próximo chamado em fila tenha sido originado no átomo j é igual a λ_j/λ, o valor do tempo médio necessário para que o servidor n, quando o sistema está em fila, viaje ao átomo geográfico j é igual a:

$$\overline{T}_Q \equiv \sum_{i=1}^{N_A} \sum_{j=1}^{N_A} \frac{\lambda_i \cdot \lambda_j}{\lambda^2} \cdot \tau_{ij} \tag{3.11}$$

3.5.3.1 Tempo médio de viagem para todo o sistema

A partir destes valores, o tempo médio de viagem para todo o sistema pode ser obtido através de:

$$\overline{T} = \sum_{n=1}^{N} \sum_{j=1}^{N_A} f_{nj}^{[1]} \cdot t_{nj} + P_Q \cdot \overline{T}_Q \tag{3.12}$$

3.5.3.2 Tempo médio de viagem para cada átomo geográfico

$$\overline{T}_j = \frac{\sum_{n=1}^{N} f_{nj}^{[1]} \cdot t_{nj}}{\sum_{n=1}^{Ns} f_{nj}^{[1]}} (1 - P_Q) + \sum_{i=1}^{N_A} \left(\frac{\lambda_i}{\lambda} \right) \tau_{ij} P_Q \qquad (3.13)$$

Outras interessantes medidas de desempenho que podem ser calculadas pelo modelo, estão descritas em Larson (1973, 1974b) e Larson e Odoni (1981).

3.6 Exemplo de aplicação

Como exemplo de aplicação do modelo, será considerada a modelagem de um sistema hipotético cuja geografia foi apresentada na Figura 3.1. Neste sistema, existem cinco átomos geográficos e três servidores que possuem taxas de chegada e serviço especificadas nas Tabelas 3.2 e 3.3, respectivamente.

Tabela 3.2 – Taxas de chegada de solicitações de serviço originadas nos átomos geográficos do sistema hipotético.

Número do Átomo Geográfico	Taxa de Chegada de Solicitações
1	0,5
2	0,4
3	0,2
4	1,0
5	2,0

Tabela 3.3 – Taxas de serviço dos servidores do sistema hipotético.

Número do Servidor	Taxa de Serviço
1	1,0
2	2,0
3	2,0

3.6.1 Políticas de despacho

Apesar de o modelo Hipercubo suportar qualquer política fixa de despacho único, existe um conjunto de políticas que geralmente são adotadas. Uma dessas políticas (*Expected Modified Center of Mass* – EMCM) representa a melhor política possível de despacho quando não há informação em tempo-real sobre os servidores disponíveis para despacho (LARSON, 1975b, p. 54).

Estas políticas são: *Strict Center of Mass* – SCM, *Modified Center of Mass* – MCM, *Expected Modified Center of Mass* – EMCM e *Expected Strict Center of Mass* – ESCM.

Estas políticas envolvem o cálculo dos tempos de viagem dos servidores até os átomos geográficos. A partir destes tempos de viagem, a política de despacho é definida atribuindo aos servidores com o menor tempo de viagem as posições preferenciais de despacho. A diferença entre estas políticas está na forma de cálculo dos tempos de viagem dos servidores até os átomos geográficos.

A partir destas considerações, o tempo de viagem de cada um dos servidores para atendimento a chamados de cada um dos átomos geográficos é calculado e a política de despacho é criada ordenando-se os servidores com menor tempo de viagem para atendimento a cada um dos átomos geográficos.

A tabela de preferências de despacho está listada a seguir:

Tabela 3.4 – Preferências de despacho.

Número do Átomo Geográfico	1	2	3
1	2	1	3
2	2	1	3
3	1	2	3
4	3	2	1
5	1	2	3

A distância entre os átomos geográficos está definida na Tabela 3.5.

Tabela 3.5 – Distância entre os átomos geográficos.

Número do Átomo Geográfico	1	2	3	4	5
1	0	2	3	5	6
2	2	0	3	4	5
3	3	3	0	6	8
4	5	4	6	0	9
5	6	5	8	9	0

Completando os dados de entrada do modelo, as localizações dos servidores estão definidas na Tabela 3.6.

Tabela 3.6 – Localização dos servidores.

	Número dos átomos geográficos				
Número dos servidores	1	2	3	4	5
1	0,4	0,3	0,3	0	0
2	0	0	0	0,5	0,5
3	0	0	0	1,0	0

Os possíveis estados do modelo para este sistema são: {000}, {001}, {010}, {011}, {100}, {101}, {110} e {111}. As taxas de transição entre estes estados podem ser observadas nas Figuras 3.4 e 3.5.

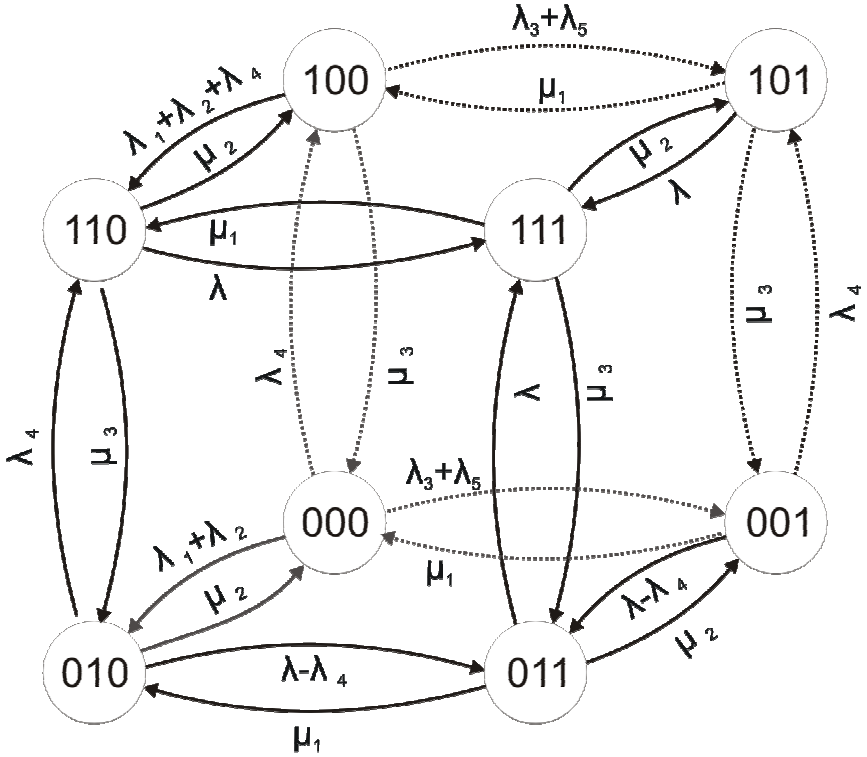


Figura 3.4 – Taxas de transições para o modelo hipotético estudado.

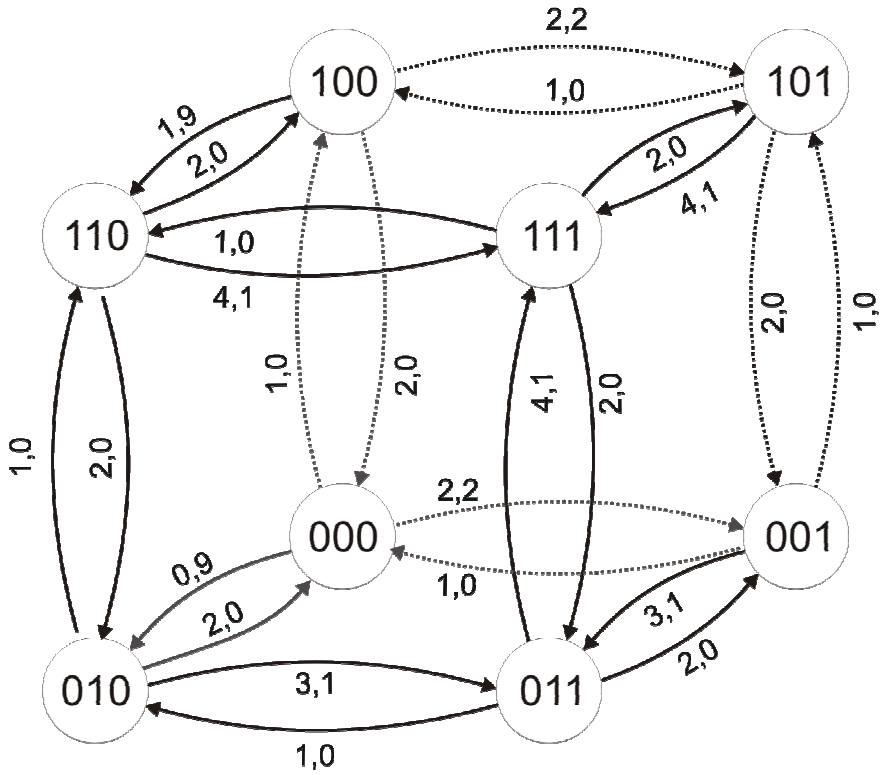


Figura 3.5 – Valores numéricos das taxas de transições para o modelo hipotético estudado.

Para obter as medidas de desempenho, foram construídas as equações de equilíbrio do modelo, listadas a seguir:

$$4{,}1 \cdot P_{000} = 1{,}0 \cdot P_{001} + 2{,}0 \cdot P_{010} + 2{,}0 \cdot P_{100}$$
$$5{,}1 \cdot P_{001} = 2{,}2 \cdot P_{000} + 2{,}0 \cdot P_{011} + 2{,}0 \cdot P_{101}$$
$$6{,}1 \cdot P_{010} = 0{,}9 \cdot P_{000} + 1{,}0 \cdot P_{011} + 2{,}0 \cdot P_{110}$$
$$7{,}1 \cdot P_{011} = 3{,}1 \cdot P_{001} + 3{,}1 \cdot P_{010} + 2{,}0 \cdot P_{111}$$
$$6{,}1 \cdot P_{100} = 1{,}0 \cdot P_{000} + 1{,}0 \cdot P_{101} + 2{,}0 \cdot P_{110}$$
$$7{,}1 \cdot P_{101} = 1{,}0 \cdot P_{001} + 2{,}2 \cdot P_{100} + 2{,}0 \cdot P_{111}$$
$$8{,}1 \cdot P_{110} = 1{,}0 \cdot P_{010} + 1{,}9 \cdot P_{100} + 1{,}0 \cdot P_{111}$$
$$5{,}0 \cdot P_{111} = 4{,}1 \cdot P_{011} + 4{,}1 \cdot P_{101} + 4{,}1 \cdot P_{110}$$
$$P_{000} + P_{001} + P_{010} + P_{011} + P_{100} + P_{101} + P_{110} + P_{111} = 1$$

Após serem resolvidas, as seguintes probabilidades de estado foram obtidas:

$$P_{000} = 0{,}09349$$
$$P_{001} = 0{,}15586$$
$$P_{010} = 0{,}06070$$
$$P_{011} = 0{,}17539$$
$$P_{100} = 0{,}05302$$
$$P_{101} = 0{,}11922$$
$$P_{110} = 0{,}05536$$
$$P_{111} = 0{,}28697$$

A partir destas probabilidades, as medidas de desempenho do sistema podem ser calculadas. As duas medidas bases para o cálculo das restantes são:

$$\rho_1 = P_{001} + P_{011} + P_{101} + P_{111} = 0{,}73744$$
$$\rho_2 = P_{010} + P_{011} + P_{110} + P_{111} = 0{,}57842$$
$$\rho_3 = P_{100} + P_{101} + P_{110} + P_{111} = 0{,}51457$$

As frações de despacho f_{nj} dos servidores para os átomos geográficos estão listadas na tabela seguinte:

Tabela 3.7 – Frações de despacho f_{nj} dos servidores para os átomos geográficos.

	Número do Servidor			
Número do Átomo Geográfico	1	2	3	Total
1	0,0195	0,0711	0,0295	0,1201
2	0,0156	0,0568	0,0236	0,0962
3	0,0177	0,0185	0,0118	0,048
4	0,0186	0,0580	0,1637	0,2403
5	0,1770	0,1855	0,1182	0,4809
Total	0,2484	0,3901	0,347	1,0000

A partir destes valores, outras medidas de desempenho podem ser calculadas.

3.7 Principais extensões do modelo

Um grande número de extensões do modelo Hipercubo foram propostas. Estas extensões estão relacionadas, em sua grande maioria, a relaxação de algumas das hipóteses exigidas para sua aplicação.

3.7.1 Calibração dos tempos de atendimento

A hipótese 9 do modelo determina que variações no tempo de atendimento causadas por variações no tempo de viagem são de ordem secundária, quando comparadas com as variações de tempo de execução e/ou tempo de preparação. No entanto, em muitos sistemas reais, como os médico-emergenciais, o tempo de viagem representa uma fração significativa do tempo total de atendimento. Para estes sistemas, é necessário ajustar separadamente os tempos de viagem de cada servidor de forma a considerar os fatores geográficos que o influenciam.

O processo de ajuste consiste em calcular, através do modelo, o tempo médio de viagem para cada servidor e verificar a diferença entre as taxas de serviço calculadas utilizando este tempo de viagem e aquelas que foram utilizadas como entrada do modelo. Se a diferença entre estes valores for significativa, o modelo deve ser resolvido novamente, utilizando como parâmetro as taxas de serviço calculadas a partir deste novo tempo médio de atendimento.

Este processo deve se repetir até que a diferença entre os valores admitidos como parâmetro e os tempos médios de atendimento calculados através do modelo sejam suficientemente próximos (LARSON; ODONI, 1981).

3.7.2 Servidores que possuem mais de dois estados

Na aplicação do modelo Hipercubo a certos tipos de sistemas reais (principalmente policiais), a representação de disponibilidade/indisponibilidade

dos servidores no espaço de estados do modelo é insuficiente para representar as características destes sistemas. Em diversas cidades da Califórnia (EUA), por exemplo, uma fração significativa de tempo é gasto pelas viaturas de polícia em atividades conhecidas como PIAs (*Patrol Initiated Activities*) (LARSON; MCKNEW, 1982). Estas atividades são geralmente identificadas durante rondas e envolvem, entre outras coisas, ocorrências de violação de tráfego, inspeção de carros etc.

Procurando tornar o modelo Hipercubo mais realista para a modelagem destes sistemas, Larson e Mcknew (1982) propuseram uma extensão que assume a existência de três estados possíveis para os servidores: disponível, ocupado atendendo uma solicitação de serviço ou ocupado em uma PIA. Com essa modificação do espaço de estados, o modelo passa a ter 3^N estados e sua solução exata torna-se ainda mais proibitiva. No entanto, um método aproximado que envolve a solução de um sistema de 2^N equações lineares foi proposto.

A consideração de servidores com mais de dois estados também foi estudada em outros trabalhos com o modelo Hipercubo (IANNONI, 2005; MENDONÇA; MORABITO, 2000, 2001).

Ainda que tais modelos sejam importantes em várias aplicações, o escopo do presente estudo foi limitado à modelos de servidores com dois estados.

3.7.3 Distribuição de probabilidade dos tempos de atendimento

As medidas de desempenho relacionadas aos tempos de viagem calculadas pelo modelo Hipercubo são apenas valores médios. Estas medidas podem ser utilizadas, por exemplo, para estimar o efeito de atrasos no resultado do serviço prestado. Chelst e Jarvis (1979) mostraram que, em muitos casos, o relacionamento entre o tempo de viagem dos servidores até os clientes e os efeitos provocados pela variação neste tempo tem um relacionamento não linear. Desta forma, a média passa a não ser uma medida suficiente. Neste

trabalho, os autores derivam algumas fórmulas que permitem o cálculo da distribuição de probabilidades dos tempos de viagem.

3.7.4 Despacho de múltiplos servidores

Uma das limitações do modelo Hipercubo original é a restrição de despacho de apenas um servidor para o atendimento a solicitações. No entanto, em muitos sistemas emergenciais pode ocorrer o despacho de dois ou mais servidores, por exemplo, para solicitações graves. O despacho múltiplo de servidores foi primeiramente tratado por Chelst (1975), em seu estudo com o departamento de polícia de New Haven, através do aumento das taxas de solicitações que necessitassem mais de um servidor (por exemplo, as taxas de transição que necessitam dois servidores foram dobradas). Esta modificação permite que o despacho de múltiplos servidores seja sentido nas cargas de trabalho, mas não permite que outras medidas de desempenho relacionadas aos despachos múltiplos sejam calculadas. Para permitir que medidas específicas para estes despachos fossem calculadas, Chelst e Barlach (1981) estenderam o modelo Hipercubo e propuseram um método exato e um método aproximado de solução. Entre as medidas de desempenho que podem ser calculadas através desta extensão estão: o tempo médio de viagem para solicitações que necessitam dois servidores; o tempo médio de viagem da unidade primária e secundária para este tipo de solicitação; o tempo médio do servidor que primeiro chegou ao local etc.

Procurando permitir o múltiplo despacho de servidores e as atividades iniciadas em patrulha (PIA), Gau e Larson (1988) criaram uma versão do modelo que incorpora estas duas extensões.

3.7.5 Prioridades de solicitações

Apesar de o modelo não tratar diretamente a existência de solicitações com diferentes prioridades de atendimento, Larson (1973, p. 49,1974, p. 94) sugeriu uma estratégia que permite considerar a existência de prioridades. A estratégia

consiste em definir, para cada átomo geográfico que pode gerar solicitações com prioridades diferentes, as frações f_{ik} de solicitações geradas no átomo i que possuem prioridade k. Desta forma, o átomo pode ser considerado como fontes múltiplas solicitadoras de serviço.

Em Takeda (2000) e Takeda et al. (2000, 2004, 2007) tal estratégia foi utilizada para a modelagem do Serviço de Atendimento Móvel de Urgência (SAMU) de Campinas-SP. Neste sistema, algumas ambulâncias são especializadas no atendimento de solicitações com alta gravidade (Veículo de Suporte Avançado – VSA), enquanto outras ambulâncias são despachadas para atender outros tipos de solicitações (Veículo de Suporte Básico – VSB).

Na modelagem deste sistema, cada átomo geográfico foi considerado uma dupla fonte solicitadora de serviço, ou seja, para cada átomo i foram definidas as taxas de chegada λ_i^a (correspondente às solicitações de alta prioridade) e λ_i^b (correspondente às solicitações de menor prioridade). Para as solicitações de alta prioridade, os servidores VSA foram definidos como primários e os VSB como secundários. De forma similar, para as solicitações de menor prioridade, os servidores VSB foram definidos como primários e os VSA como secundários. Outro exemplo de aplicação desta estratégia pode ser encontrado em Iannoni (2005).

3.7.6 Política de despacho particular

A hipótese 4 do modelo, que assume que todo servidor pode atender às solicitações de qualquer átomo geográfico, não corresponde à realidade de alguns sistemas de atendimento. Em sistemas de atendimento emergencial em rodovias, por exemplo, devido a limitações de distância, algumas solicitações podem ser atendidas apenas por um conjunto de servidores que se encontram próximos do local da solicitação e que tenham condição de prestar assistência em um intervalo de tempo aceitável.

Em Mendonça e Morabito (2000, 2001) e Iannoni (2005) foi proposta uma extensão para o modelo Hipercubo que permite que sejam definidas políticas de despacho particulares e foram definidas algumas novas medidas de desempenho para este tipo de sistema.

3.7.7 Despacho de servidores co-localizados

Em sistemas emergenciais de grandes centros urbanos, podem existir dois ou mais servidores, localizados na mesma base (co-localizados), por exemplo, que sejam igualmente preferenciais para o atendimento a solicitações de serviço. Burwell et al. (1993) propuseram uma extensão para o modelo Hipercubo, baseada no método aproximado de Jarvis, que é menos custosa do que outras alternativas para considerar servidores com preferências iguais. Como será visto no próximo capítulo, a versão exata do modelo Hipercubo pode ser reduzida quando o sistema emergencial apresenta esta característica.

4 MÉTODO DE DECOMPOSIÇÃO

Este capítulo trata da aplicação de métodos de decomposição de cadeias de Markov ao modelo Hipercubo de Filas. Após algumas considerações iniciais serem feitas na seção 4.1, as principais estruturas de cadeias de Markov que podem direcionar a escolha de métodos apropriados de decomposição são descritas na seção 4.2 e a aplicação de dois métodos de decomposição que permitem a redução da complexidade da solução do modelo Hipercubo é estudada nas seções 4.3 e 4.4.

4.1 Considerações iniciais

As restrições observadas no capítulo anterior quanto à aplicação de métodos diretos e iterativos tradicionais para a solução das equações de equilíbrio de modelos de sistemas de grande porte motivam a pesquisa por métodos alternativos de solução.

Como os recursos de memória tornam-se restritivos antes dos recursos de processamento necessários para a solução do modelo, esta pesquisa deve ser direcionada em busca de métodos que permitam a redução da quantidade de informação armazenada para a descrição do sistema modelado.

Uma técnica que tem sido amplamente utilizada com este fim em estudos com cadeias de Markov é a técnica de decomposição. A técnica de decomposição explora, em procedimentos exatos ou aproximados, alguma característica da estrutura de cadeias de Markov para reduzir as exigências de memória e/ou processamento para a obtenção de medidas de interesse. (KIM; SMITH, 1995).

Alguns exemplos de estruturas que podem ser exploradas em cadeias de Markov são: aglutinação exata (*exact lumpability*), aglutinação fraca/aproximada (*weak lumpability*), único estado de saída ou entrada (*single input/exit state*), conjuntos obrigatórios (*mandatory sets*) e quase completamente particionável (*nearly completely decomposable*).

Entre os diferentes tipos de métodos de decomposição, estão os métodos iterativos de agregação/desagregação, os métodos multiníveis, entre outros.

A aplicação de técnicas de decomposição em cadeias de Markov pode ser considerada como uma metodologia de dividir para conquistar. A idéia por detrás da decomposição é a divisão da cadeia de Markov em estudo em cadeias menores que possam ser resolvidas separadamente e reunidas novamente para a construção do resultado da cadeia inicial.

A seguir, as principais estruturas de cadeias de Markov que podem direcionar a escolha dos métodos de decomposição apropriados são definidas.

4.2 Estruturas especiais de cadeias de Markov

Dada uma cadeia de Markov $X=\{X_t, t \in T\}$, com matriz de probabilidades de transição P, com vetor de probabilidades estacionárias π e seja $A = \{A_1, A_2, ..., A_N\}$ uma partição mutuamente exclusiva e exaustiva do espaço de estados desta cadeia de Markov.

a) Aglutinação exata (*Exact lumpability*): A cadeia de Markov X é dita exatamente/fortemente aglutinável com relação à partição A se, para qualquer par de conjuntos A_i e A_j desta partição, todos os estados de A_i possuem transições iguais para todos os estados de A_j.

a) Aglutinação fraca (*Weak lumpability*): A cadeia de Markov X é dita fracamente aglutinável com relação a uma partição A se, para algum vetor de probabilidades iniciais, o processo estocástico $Y=\{Y_n, n=1, 2, ...\}$, tal que $Y_n=A_i$ quando $X_n=a_i \in A_i$, for uma cadeia de Markov;

b) Particionável com um único estado de entrada (*Single input state*): A cadeia de Markov X é classificada como particionável com um único estado de entrada com respeito a uma partição A se, para todo $A_i \in A$, o processo só pode entrar em A_i através de um estado;

c) Particionável com um único estado de saída (*Single exit state*): A cadeia de Markov X é classificada como particionável com um único estado de saída com respeito a uma partição A se, para todo $A_i \in A$, o processo só pode sair de A_i através de um estado;

d) Conjuntos obrigatórios (*Mandatory sets*): A cadeia de Markov X é dita particionável em conjuntos obrigatórios com respeito a uma partição A se, para todo $A_i \in A$, existe um subconjunto de estados tal que sempre que o processo entrar em A_i, alguns estados neste conjunto serão visitados antes da saída, e, para todo estado no conjunto obrigatório, a probabilidade estacionária condicional é conhecida;

e) Quase completamente particionável (*Nearly Completely Decomposable – NCD*): A cadeia de Markov X é classificada como quase completamente particionável quando os elementos externos aos blocos da diagonal P_{ii} da matriz de probabilidade de transição contêm probabilidades relativamente pequenas em comparação aos elementos dos blocos da diagonal;

Quando a matriz de transições da cadeia de Markov não se encontra apropriadamente ordenada para a aplicação de métodos de decomposição, algoritmos que procuram por uma ordenação apropriada da matriz devem ser utilizados (DAYAR; STEWART, 1995; DAYAR; STEWART, 2000; O'NEIL; SZYLD, 1990; SEZER; ŠILJAK, 1986; SUMITA; RIEDERS, 1990; TANAKA; SHIOYAMA, 1995).

Para cada uma destas estruturas, diversos métodos de decomposição foram estudados (CAO; STEWART, 1985; FEINBERG; CHIU, 1987; HAVIV, 1987; MEYER, 1989; SIMON; ANDO, 1961; STEWART et al., 1984; STEWART; WU, 1992; SUMITA; RIEDERS, 1990; TANAKA; SHIOYAMA, 1995;).

Outros métodos que não exigem estruturas especiais podem ser encontrados em Semal (1995), Schweitzer (1986), Sheskin (1985), Feinberg e Chiu (1987).

Praticamente todas estas estruturas podem ocorrer no modelo Hipercubo. Porém, a ocorrência de cada uma delas está condicionada a certas características que o modelo deve apresentar, como certas políticas de preferência de despacho etc.

Como exemplo de ocorrência de uma destas estruturas, será considerada uma modificação do exemplo estudado no capítulo anterior. A modificação consiste na alteração das preferências de despacho do modelo para a preferência descrita na Tabela 4.1.

Tabela 4.1 – Preferências de despacho.

Número do Átomo Geográfico	1	2	3
1	1	2	3
2	1	2	3
3	1	2	3
4	1	2	3
5	1	2	3
	Números dos Servidores		

Para esta alteração, a estrutura de elementos não-nulos da matriz de transições da cadeia de Markov do modelo Hipercubo pode ser observada na Tabela 4.2.

Tabela 4.2 – Elementos não-nulos (X) da matriz de transições da cadeia de Markov.

	000	001	010	011	100	101	110	111
000	X	X						
001	X	X		X				
010	X		X	X				
011		X	X	X				X
100	X				X	X		
101		X			X	X		X
110			X		X		X	X
111				X		X	X	X

Se a cadeia de Markov for particionada em dois subconjuntos de acordo com A={{000, 001, 010, 011}, {100, 101, 110, 111}} e, dependendo da magnitude dos valores escolhidos para as taxas de chegada e de serviço, a cadeia de Markov pode ser considerada “quase completamente particionável”. Entretanto, a ocorrência de tais características depende de condições muito restritivas nos dados de entrada do problema.

A seguir, duas das técnicas de decomposição que podem ser aplicadas ao modelo Hipercubo são estudadas. A possibilidade de uso da primeira destas técnicas é conhecida (BURWELL et al., 1985, 1993), mas nenhum dos trabalhos encontrados na literatura formaliza sua aplicação.

4.3 Aglutinação de estados no modelo Hipercubo de Filas

A aglutinação de estados é uma técnica de decomposição que permite reduzir o espaço de estados de modelos markovianos, de tal forma que seja mantido um nível de detalhe suficiente para o cálculo das medidas de interesse. (SOUZA E SILVA; MUNTZ, 1992).

Quando a aglutinação é exata, nenhuma informação é perdida mediante a aplicação da técnica e, conseqüentemente, a solução do modelo original é completamente especificada pela solução do modelo reduzido. Na aglutinação fraca, algumas informações são perdidas.

Para ilustrar a idéia básica da técnica de aglutinação, será utilizado um exemplo descrito em Souza e Silva e Muntz (1992) com modificações.

Considere uma cadeia de Markov em tempo contínuo utilizada para modelar um sistema computacional simples composto por um processador e duas unidades de memória, conforme ilustrado na Figura 4.1.

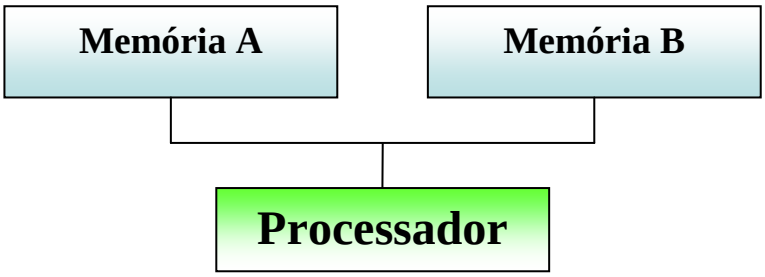


Figura 4.1 – Componentes de um sistema computacional simples.

Neste sistema, tanto o processador quanto cada unidade de memória podem sofrer falhas. O processador falha com taxa λ_p e cada unidade de memória falha com taxa λ_m. É suposto que as falhas ocorrem independentemente umas das outras e o tempo entre falhas é exponencialmente distribuído. Este sistema está disponível apenas quando o processador e pelo menos uma das unidades de memória estão disponíveis.

Um estado do modelo deste sistema pode ser representado por uma tripla (d_p, d_{ma}, d_{mb}), onde cada variável d_x indica se o processador, unidade de memória A e unidade de memória B estão operacionais, respectivamente. O número total de estados deste modelo é 8, a saber: (0,0,0), (1,0,0), (0,1,0), (0,0,1), (1,1,0), (1,0,1), (0,1,1), (1,1,1), onde 1 indica que o componente está falho e 0 que o componente está funcionando corretamente.

Intuitivamente pode-se observar que, para o cálculo da disponibilidade do sistema, não é necessário saber qual das unidades de memória não está operacional, mas apenas quantas unidades estão falhas. Por conseguinte, pode-se aglutinar os estados que indicam a unidade particular de memória que está com defeito em um único estado que representa apenas o número de unidades falhas. Em outras palavras, pode-se aglutinar os estados (0,1,0) com (0,0,1) em (0,1) e os estados (1,1,0) com (1,0,1) em (1,1). O modelo final terá 6 estados, 2 a menos que o original. Para modelos com um número grande de estados, este método pode representar uma economia significativa.

Nem sempre é possível obter todas as medidas de desempenho da cadeia original através da cadeia reduzida. Em casos nos quais os estados de cada partição A_i não interagem entre si, pode-se obter as probabilidades dos estados originais a partir dos estados aglutinados, através da divisão das probabilidades igualmente pelos estados da cadeia original. Porém, para modelos nos quais os estados de cada partição Ai interagem entre si, a forma de interação destes estados pode não permitir a obtenção das probabilidades dos estados originais.

Para provar que é possível aplicar a técnica de aglutinação de estados no modelo Hipercubo de Filas, basta provar que, para alguns tipos de sistemas reais, os estados do modelo Hipercubo podem ser particionados de tal forma que a estrutura de “aglutinação fraca” descrita na seção 4.2 ocorre no modelo.

Em alguns sistemas emergenciais de grandes cidades, como o SAMU de São Paulo, vários servidores estão co-localizados na mesma base de atendimento e

possuem políticas de atendimento idênticas. Quando estes servidores prestam o mesmo tipo de serviço, o tempo médio de serviço deles tende a ser igual.

Na construção das equações de equilíbrio do modelo Hipercubo, tanto as taxas de chegada quanto às taxas de saída de um determinado estado são baseadas na preferência de despacho, no tempo médio de atendimento dos servidores e nas taxas de chegada de solicitações de serviço.

Como os servidores co-localizados destes sistemas apresentam preferências de despacho iguais, tempos médios de atendimento iguais e respondem a mesma demanda, as taxas de transição de seus estados com os estados vizinhos são iguais. Assim sendo, se os estados que representam a disponibilidade destes servidores (para um cenário específico de disponibilidade dos outros servidores) forem aglutinados em uma partição A_l, a cadeia de Markov original estará de acordo com o teorema descrito na seção 4.2 e, portanto, a cadeia de Markov do modelo Hipercubo poderá sofrer aglutinação.

4.4 O método aproximado de Birge e Pollock

O método aproximado de Birge e Pollock (1989) é uma alternativa (que permite que os servidores assumam mais de dois estados) para o método aproximado de Larson. O método de Birge e Pollock é baseado na aplicação de técnicas de decomposição e envolve a solução de um sistema de N equações não-lineares. Cada equação deste método representa a disponibilidade de um dos servidores do sistema e é derivada de uma cadeia de Markov que possui apenas dois estados: o servidor em questão está livre ou está ocupado.

As transições entre estes estados são estimadas supondo-se que os servidores são independentes.

Para um sistema com dois servidores (um primário e outro reserva), por exemplo, a taxa de chegada de solicitações de serviço para o servidor reserva

é calculada a partir de $\rho_1 \cdot \lambda$, onde ρ_1 é a taxa de ocupação do servidor primário e λ a taxa total de solicitações que chegam a este servidor. Se ρ_1 for considerado como sendo igual a $P(B_2/F_1)$ (B_2 - evento que determina que o servidor 2 está ocupado; F_1 - evento que determina que o servidor 1 está livre) ou seja, a probabilidade de que o servidor 2 esteja ocupado dado que o servidor 1 está livre, o resultado seria exato. No entanto, dada a hipótese de independência, é assumido que $P(B_2/F_1)=P(B_2)$.

Para sistemas com 3 ou mais servidores, as taxas de chegada de cada servidor preferencial são distribuídas igualmente entre todos servidores reservas.

4.4.1 Precisão

Enquanto os resultados apresentados por Birge e Pollock (1989), Pollock e Birge (1983) e Pollock et al. (1982; 1985) indicaram que o método pode calcular medidas de desempenho com boa precisão, a precisão do método tende a reduzir com o aumento da interação entre os servidores. Como o efeito da interação entre os servidores cresce com N, a hipótese da independência torna-se mais restritiva e o método tende a apresentar erros maiores para sistemas com muitos servidores.

A seguir, um exemplo de aplicação do método a um sistema com 3 servidores e 1 átomo geográfico é apresentado. A matriz de preferências de despacho está listada na tabela:

Tabela 4.3 – Preferências de despacho para o sistema exemplo.

Número do Átomo Geográfico	Preferência		
	1	2	3
1	1	2	3

As cadeias de Markov que representam a disponibilidade de cada servidor possuem matriz de transição representada a seguir:

$$R^{(1)} = \begin{pmatrix} 0 & \lambda \\ \mu_1 & 0 \end{pmatrix}$$

$$R^{(2)} = \begin{pmatrix} 0 & \frac{\lambda}{2} \cdot \rho_1 \cdot (1-\rho_3) + \lambda \cdot \rho_1 \cdot \rho_3 \\ \mu_2 & 0 \end{pmatrix}$$

$$R^{(3)} = \begin{pmatrix} 0 & \frac{\lambda}{2} \cdot \rho_1 \cdot (1-\rho_2) + \lambda \cdot \rho_1 \cdot \rho_2 \\ \mu_3 & 0 \end{pmatrix}$$

As cargas de trabalho destes servidores podem ser obtidas a partir do cálculo do estado estacionário da matriz de transições de cada servidor. Assim sendo, as cargas de trabalho dos três servidores podem ser obtidas a partir de:

$$\rho_1 = \frac{\lambda}{\lambda + \mu_1}$$

$$\rho_2 = \frac{\frac{\lambda}{2} \cdot \rho_1 \cdot (1-\rho_3) + \lambda \cdot \rho_1 \cdot \rho_3}{\frac{\lambda}{2} \cdot \rho_1 \cdot (1-\rho_3) + \lambda \cdot \rho_1 \cdot \rho_3 + \mu_2}$$

$$\rho_3 = \frac{\frac{\lambda}{2} \cdot \rho_1 \cdot (1-\rho_2) + \lambda \cdot \rho_1 \cdot \rho_2}{\frac{\lambda}{2} \cdot \rho_1 \cdot (1-\rho_2) + \lambda \cdot \rho_1 \cdot \rho_2 + \mu_3}$$

Assumindo um sistema homogêneo com $\mu_1=\mu_2=\mu_3=0,25$ e $\lambda=0,5$, os valores de ρ_1, ρ_2 e ρ_3 podem ser calculados iterativamente e os resultados obtidos são, respectivamente: 0,666666667, 0,50 e 0,50. Enquanto os resultados exatos para estas cargas de trabalho são: 0,6666666666666667, 0,5333333333327961 e 0,37894736842158994, o que representa um erro significativo, principalmente para o terceiro servidor que sofre maior influência dos outros servidores e, portanto, torna-se mais distante da hipótese de independência.

5 MÉTODOS APROXIMADOS DE SOLUÇÃO

Este capítulo trata dos métodos aproximados de solução do modelo Hipercubo de Filas propostos para viabilizar a aplicação do modelo a sistemas de grande porte.

Após algumas considerações iniciais serem feitas na seção 5.1, os principais métodos aproximados de solução do modelo Hipercubo são revisados nas seções 5.2 e 5.3. Por fim, nas seções 5.4 e 5.5 são apresentados e discutidos resultados referentes à precisão dos métodos revisados.

5.1 Considerações iniciais

As limitações do método exato do modelo Hipercubo na modelagem de sistemas com muitos servidores, descritas anteriormente, incentivaram a pesquisa por métodos de solução menos custosos em termos de processamento e, principalmente, memória.

O primeiro método a apresentar estas características foi proposto por Richard C. Larson (1974a, 1975a) e ficou conhecido como *A-Hypercube* (ou método aproximado de Larson). Este método envolve a solução de um sistema de N equações não-lineares, ao invés de um sistema com 2^N equações lineares do método exato.

Desde sua proposição, o método aproximado tem sido utilizado em lugar do método exato para a modelagem de sistemas de atendimento emergencial em diversas cidades, como Boston (BRANDEAU; LARSON, 1986), Dallas, Wilmington, Caracas, entre outras (LARSON, 2002).

Apesar dessa ampla utilização, apenas algumas observações gerais referentes à precisão do método foram feitas e não foi apresentado nenhum detalhe sobre como foram conduzidos os resultados que motivaram estas observações. Estas observações, por exemplo, não incluíram a quantificação de alguns erros.

Entre as hipóteses assumidas por Larson para derivar o método, a da homogeneidade dos servidores (todos os servidores têm o mesmo tempo médio de serviço) é a mais restritiva. Os resultados de testes apresentados por Larson (1974a, 1975a), Larson e Odoni (1981) e Chaiken (1975) indicaram erros pequenos e convergência rápida na solução do método para sistemas com servidores homogêneos. Para o caso de sistemas com servidores não-homogêneos, Larson (1975a) citou que o método poderia ser utilizado para o cálculo de uma aproximação inicial para as medidas de desempenho do sistema.

Não demorou muito tempo, contudo, para que a hipótese da homogeneidade fosse relaxada. Jarvis (1975, 1985) desenvolveu um método baseado no método de Larson que permite distribuições gerais dos tempos de serviço dependentes não apenas dos servidores, mas também da localização onde o serviço será realizado. Entretanto, as mesmas observações feitas anteriormente sobre a precisão do método de Larson podem ser estendidas ao método de Jarvis.

Desde o desenvolvimento destes dois métodos, diversas variantes que permitem a relaxação de algumas das hipóteses do modelo foram propostas, tais como: o suporte a servidores co-localizados (BURWELL et al., 1985, 1993), o despacho de múltiplos servidores (CHELST; BARLACH, 1981), a consideração de servidores com mais de dois estados (MENDONÇA; MORABITO, 2001; IANNONI, 2005; LARSON; MCKNEW, 1982), entre outras.

Estas variantes conservam, em sua maioria, a mesma estrutura básica dos métodos de Larson e Jarvis. Exceções são alguns métodos que seguem caminhos diferentes de derivação, como o método apresentado por Birge e Pollock (1989), descrito no Capítulo anterior, baseado em técnicas de decomposição.

Apesar da quantidade de métodos (considerando variantes) aproximados, neste trabalho serão estudados apenas os métodos de Larson e de Jarvis. A escolha destes métodos justifica-se pela sua ampla divulgação e utilização (comparativamente aos outros) e pela razoável consideração de que os resultados apresentados para estes métodos possam ser estendidos para as variantes que mantêm a mesma estrutura básica dos dois métodos.

Nas próximas seções, estes dois métodos são revisados e são apresentados os resultados considerados até então como referências para a precisão e convergência dos métodos.

5.2 O método aproximado de Larson

O método aproximado de Larson (1974a, 1975a) é derivado do modelo de filas M/M/N e envolve a solução de um sistema de N equações não-lineares cujas incógnitas são as cargas de trabalho (ρ_i, i = 1, 2,..., N) dos servidores.

O caráter aproximado do método está relacionado a duas hipóteses consideradas durante seu desenvolvimento. A primeira hipótese é de que todos os servidores são homogêneos, e a segunda é de que a seleção dos servidores para alocação se dá de forma aleatória (enquanto, de acordo com a sétima hipótese do modelo Hipercubo, se dá a partir de preferências de despacho fixas) (CHIYOSHI et al., 2000).

Estas hipóteses podem ser consideradas apropriadas para sistemas cujas cargas de trabalho não diferem significativamente e nos quais diferentes vetores de preferências de despacho simulam a seleção aleatória dos servidores (LARSON; ODONI, 1981).

Resumidamente, o método conforme descrito em Larson (1974a, 1975a) e Larson e Odoni (1981) envolve os seguintes passos:

Inicialização

a) Normalização da unidade de tempo: a unidade de tempo deve ser igualada ao tempo médio de serviço (τ). Esta alteração na unidade de tempo pode provocar alterações nas taxas de chegada de solicitações de serviço que devem ser normalizadas conforme Equação 5.1;

$$\lambda_m^{\ normalizado} = \lambda_m \cdot \tau \tag{5.1}$$

b) Definição dos valores iniciais das cargas de trabalho (ρ_i^0): os valores de ρ_i são inicializados com o valor do fator médio de utilização do sistema (r), calculado a partir do modelo de filas M/M/N (ver Capítulo 2). Para sistemas sem limite para o tamanho da fila, o seguinte valor deve ser utilizado:

$$r = \frac{\lambda}{N} \tag{5.2}$$

Iterações

c) Cálculo dos novos valores de ρ_i: os valores de ρ_i são atualizados a cada iteração *t* a partir do cálculo de uma aproximação da taxa de solicitações atendidas pelo servidor i quando o servidor está livre (R_i^F).

$$\rho_i^{\ t} = \frac{R_i^F + \lambda_Q P_Q}{1 + R_i^F} \tag{5.3}$$

O valor de R_i^F pode ser obtido a partir da Equação 5.4, derivada do modelo de filas M/M/N assumindo a independência entre os servidores e utilizando um fator de correção para esta hipótese geralmente falsa.

$$R_i^F = \sum_{k=1}^{N} \sum_{m:a_{mk}=i} \lambda_m \cdot \tau \cdot Q(N, \rho, k-1) \cdot \prod_{j=1}^{k-1} \rho_{a_{mj}}^{\ t-1} \tag{5.4}$$

onde a_{mk} representa o k-ésimo servidor preferencial para o átomo m e $\lambda_m \cdot \tau$ é a taxa de solicitação normalizada do átomo m. Neste passo, o fator de correção

descrito pela Equação 5.5 (para sistemas sem limite para o tamanho da fila) é utilizado.

$$Q(N,\rho,k-1) = \left[\sum_{j=k}^{N-1}\left\{\frac{(N-k-1)!(N-j)\cdot\left(\rho^{j-k}\right)}{(j-k)!}\right\}N^{j}\right]\cdot\frac{P_0}{N!(1-\rho)} \qquad (5.5)$$

Normalização

d) Normalização dos valores de ρ_i^t: a soma dos valores de ρ_i^t deve ser normalizada para se igualar ao fator médio de utilização do sistema.

$$N^{-1}\sum_{i=1}^{N}\rho_i^{\,t} = r \qquad (5.6)$$

Teste de parada

e) O teste de parada pode ser realizado a partir da precisão absoluta ou relativa das cargas de trabalho, do número máximo de iterações ou de qualquer outro critério. Se o teste não indicar a parada, deve-se retornar ao passo (c).

Para que outras medidas de desempenho (como as medidas de tempos de viagem) possam ser calculadas, as frações de despacho (f_{im}) dos servidores para os átomos geográficos devem ser calculadas. Enquanto no método exato estas frações são calculadas a partir das probabilidades de estado do modelo, no método aproximado elas são calculadas a partir das cargas de trabalho dos servidores. Em alguns casos, é possível calcular valores exatos para estas frações a partir das cargas de trabalho[1], porém, na maioria dos casos, apenas estimativas destas frações podem ser obtidas.

Para estimar estas frações de despacho, Larson (1974a, 1975a) propôs alguns procedimentos que diferem entre si pela precisão e pelo custo computacional.

[1] Para um exemplo de modelo que permite o cálculo dos valores exatos das frações de despacho, consulte CHIYOSHI et al. (2000).

A seguir, o procedimento que apresenta melhor precisão no cálculo das frações de despacho (LARSON, 1975a; LARSON; ODONI, 1981) é apresentado:

Inicialização

a) Inicialização de f_{im}: as frações de despacho devem ser inicializadas a partir da Equação 5.6. Esta equação não inclui os termos necessários para representar despachos para atendimento de solicitações em filas.

$$f_{im} = \frac{\lambda_m}{\lambda} \cdot Q(N, \rho, k-1) \cdot (1-\rho_i) \cdot \prod_{j=1}^{k-1} \rho_{a_{mk}} \tag{5.6}$$

Normalização

b) Normalização das frações de despacho: os valores de f_{im} devem ser normalizados para que algumas condições derivadas do modelo M/M/N sejam satisfeitas. As condições (para sistemas sem limite para o tamanho da fila) podem ser expressas pela Equação 5.7;

$$\sum_{i=1}^{N} f_{im} = \frac{\lambda_m}{\lambda} \cdot (1 - P_N), \quad k = 1, 2, ..., N_A \tag{5.7}$$

onde P_N é igual a probabilidade de saturação derivada do sistema de filas M/M/N.

Esta normalização pode ser realizada de diferentes formas. Das formas testadas por Larson, aquela que apresentou resultados mais precisos é apresentada na Equação 5.8. Nesta equação, o fator α_m pode ser calculado por tentativa-e-erro procurando reduzir o erro na normalização.

$$f_{im} = \frac{\lambda_m}{\lambda} \cdot Q(N, \rho, k-1) \cdot (1-\rho_i) \cdot \prod_{j=1}^{k-1} \rho_{a_{mk}} \cdot \alpha_m{}^{k-1} \tag{5.8}$$

c) Inclusão das frações de despacho que envolvem espera em fila (quando apropriado): para os sistemas que suportam a formação de

filas de espera, às frações de despacho deve ser acrescido o valor das frações de despacho que envolvem espera em fila.

$$f_{im}^{\ Q} = \frac{\lambda_m}{\lambda} \cdot \frac{P_N}{N} \tag{5.9}$$

Apesar de não terem sido abordados nos trabalhos de Larson, o método aproximado pode ser aplicado a sistemas que suportam a formação de fila finita. Para tanto, algumas das etapas anteriores devem ser alteradas para refletir as equações do modelo M/M/N/B.

5.2.1 Precisão e convergência

Os resultados apresentados por Larson (1974a, 1975a), Larson e Odoni (1981) e Chaiken (1975) indicaram que a aproximação das medidas de desempenho calculada pelo método aproximado de Larson apresenta erros, em porcentagem, geralmente entre 1% e 2% e que são necessárias de 7 a 8 iterações para a solução do método com vários dígitos significativos de precisão para sistemas com servidores homogêneos. No entanto, nenhum dos trabalhos anteriormente citados apresentou detalhes sobre como foram executados os testes que sugeriram estes resultados.

Larson (1975a) também citou que o desbalanceamento das demandas e da área de cobertura dos servidores poderia produzir erros significativos (que não foram quantificados) nas medidas de desempenho e que, para o caso de sistemas com servidores não-homogêneos (com tempos de serviço diferentes), o método pode ser utilizado para calcular uma aproximação inicial das medidas de desempenho do sistema.

Quanto à convergência, Goldberg e Szidarovszky (1991a,b) apresentaram uma prova de convergência e existência de unicidade de solução desenvolvida para um método proposto pelos autores no trabalho e citaram que esta prova é extensível para o método de Larson.

A prova de convergência apresentada pelos autores é garantida para dois conjuntos de taxas de ocupação iniciais (ρ_i^0=0 e ρ_i^0=1), o que envolve a modificação da etapa de inicialização do método. A etapa de normalização também deve ser modificada para que a convergência possa ser garantida.

Para garantir a convergência, os autores provam que a seqüência $\{\rho_i^t\}_{t=0}^{\infty}$ calculada pelo método de Larson é decrescente (para o caso de ρ_i^0=1) ou crescente (para o caso de ρ_i^0=0) e permanece no intervalo [0, 1].

Para o caso de ρ_i^0=1, pode-se mostrar que a seqüência de taxas de ocupação calculadas pelo método é decrescente observando que, se $\rho_i^0 = 1$, então $\rho_i^1 \leq \rho_i^0$, i = 1, 2, ..., N, dado que o valor de R_i^F utilizado no cálculo de ρ_i^1 é sempre finito e, por definição, $\lambda_Q < 1$. Esta redução no valor de ρ_i^1, i = 1, 2, ..., N, por sua vez, provoca uma redução no valor de R_i^F que será utilizado no cálculo de ρ_i^2, dado que os valores de λ_m e Q(N, ρ, k-1) são fixos. Como ρ_i e R_i^F são diretamente proporcionais, uma redução em R_i^F provoca uma redução em ρ_i e, portanto, $\rho_i^2 \leq \rho_i^1$. O mesmo raciocínio pode ser seguido para provar que $\rho_i^{t+1} \leq \rho_i^t$, ou seja, que a seqüência é decrescente.

Para que ρ_i^t seja menor que zero, R_i^F necessariamente deve ter um valor negativo. Porém, para que R_i^F assuma um valor negativo, os valores de ρ_i^{t-1} utilizados no calculo de R_i^F devem ser negativos. Como $\rho_i^0 = 1$ e a seqüência é decrescente, nunca R_i^F terá um valor negativo, o que garante que a seqüência $\{\rho_i^t\}_{t=0}^{\infty}$ permanece no intervalo [0, 1].

Mais detalhes sobre esta prova e sobre a prova de existência de unicidade de solução podem ser encontrados em Goldberg e Szidarovszky (1991a,b).

5.3 O método aproximado de Jarvis

O método aproximado de Jarvis diferencia-se do método de Larson por permitir distribuições gerais dos tempos de serviço dependentes não apenas dos servidores, mas também da localização onde o serviço será realizado. Em

relação às etapas do método aproximado de Larson, a principal diferença do método de Jarvis está na normalização das taxas de chegada e nos valores de ρ, τ e P_N, atualizados a cada iteração.

Enquanto no método aproximado de Larson as taxas de chegada são normalizadas a partir do tempo médio de serviço antes da aplicação do método, no método de Jarvis a normalização é atualizada a cada iteração, e se dá apenas em função do tempo médio de serviço dos servidores para cada átomo geográfico.

O método conforme descrito por Jarvis (1985) pode ser resumido nos seguintes passos:

Inicialização

a) Definição do valor do tempo médio de serviço (τ): o tempo médio de serviço do sistema é inicializado a partir dos tempos médios de serviço dos servidores primários de cada átomo geográfico, conforme Equação 5.11;

$$\tau = \sum_{m=1}^{N_A} \left(\frac{\lambda_m}{\lambda} \right) \tau_{a_{m1} m} \tag{5.11}$$

b) Definição dos valores iniciais de ρ_i: os valores de ρ_i são inicializados com o produto das taxas de solicitação dos átomos geográficos para os quais o servidor i é preferencial pelos tempos de serviço dos servidores nestes átomos, conforme Equação 5.12;

$$\rho_i^{\,0} = \sum_{m:a_{m1}=i} \lambda_m \tau_{im} \tag{5.12}$$

Iterações

c) Cálculo dos novos valores de ρ_i: os valores de ρ_i são atualizados a cada iteração a partir do cálculo de uma aproximação da taxa de

solicitações atendidas pelo servidor i quando o servidor está livre (V_i), similar ao valor de R_i^F.

$$\rho_i^{\,t} = \frac{V_i}{1+V_i} \tag{5.13}$$

O valor de V_i pode ser obtido a partir da Equação 5.14, derivada do modelo de filas M/M/N assumindo a independência entre os servidores e utilizando um fator de correção para esta hipótese geralmente falsa.

$$V_i = \sum_{k=1}^{N} \sum_{m:a_{mk}=i} \lambda_m \cdot \tau_{im} \cdot Q(N,\rho,k-1) \cdot \prod_{j=1}^{k-1} \rho_{a_{mj}}^{\,t-1} \tag{5.14}$$

onde a_{mk} é o k-ésimo servidor preferencial para o átomo m e $\lambda_m \cdot \tau$ é a taxa de solicitação normalizada do átomo m. No cálculo do fator de correção (Equação 5.5), o valor de ρ deve ser calculado a partir da Equação 5.15.

$$\rho = \frac{\lambda \cdot \tau}{N} \tag{5.15}$$

Teste de Parada

d) O teste de parada pode ser realizado a partir da precisão absoluta ou relativa das cargas de trabalho, do número máximo de iterações ou de qualquer outro critério. Se o teste não indicar a parada, deve-se retornar ao passo (e).

Atualização de valores

e) Atualização do tempo médio de serviço, de f_{im} e P_N: os valores do tempo médio de serviço, das frações de despacho dos servidores aos átomos geográficos e da probabilidade de que o sistema esteja

ocupado devem ser atualizados a partir das Equações 5.16, 5.17 e 5.18, respectivamente:

$$\tau = \frac{1}{(1-P_N)} \sum_{m=1}^{N_A} \left(\frac{\lambda_m}{\lambda} \right) \cdot \sum_{i=1}^{N} \tau_{a_{im}} \cdot f_{im} \tag{5.16}$$

$$P_N = 1 - \frac{\sum_{i=1}^{N} \rho_i^{\ t}}{\rho} \tag{5.17}$$

$$f_{im} = Q(N, \rho, k-1) \cdot (1-\rho_i^{\ t}) \cdot \prod_{j=1}^{k-1} \rho_{a_{mj}}^{\ t} \tag{5.18}$$

Sob alguns aspectos, o trabalho de Jarvis (1985) não deixou claro se o valor de P_N, calculado a partir da Equação 5.17, deve ser utilizado apenas no calculo do valor de τ ou se deve ser também utilizado no cálculo dos fatores de correção.

5.3.1 Precisão e convergência

Os resultados dos testes apresentados por Jarvis (1985) indicaram que a aproximação das cargas de trabalho e das frações de despacho dos servidores primários (f_{im}) obtida a partir do método apresenta erros inferiores a 3%, e que são necessárias entre 4 a 6 para sistemas com 10 servidores com tempos de serviço distribuídos de acordo com a distribuição exponencial ou de Erlang.

Para o método de Jarvis, a mesma prova de convergência derivada para o método de Larson é possível, se o valor de ρ for mantido fixo durante as iterações.

Apesar de Goldberg e Szidarovszky (1991a,b) terem apresentado resultados para a convergência do método de Jarvis, não foram apresentados resultados sobre a precisão da versão modificada dos métodos.

5.4 Resultados para os métodos aproximados de Larson e Jarvis

A realização de novos testes com os métodos aproximados é importante por diversos motivos. O primeiro deles está relacionado à falta de detalhes sobre como os testes, cujos resultados foram publicados por Larson (1974a, 1975a), Larson e Odoni (1981), Chaiken (1975) e Jarvis (1985), foram realizados e ao comentário dos autores de que os testes realizados não foram completos.

Outro motivo é a falta de quantificação de muitos erros que foram considerados “significativos”, mas que não tiveram seus valores numéricos apresentados. Algumas modificações sugeridas para garantir a convergência dos métodos não foram testadas apropriadamente e comparadas com os resultados produzidos pelos métodos conforme definidos originalmente. Por fim, não foram apresentados resultados que mostrassem o relacionamento entre erros nas cargas de trabalho e nas outras medidas de desempenho, dela derivadas.

Como ainda não foram encontrados limites analíticos para a precisão destes métodos, o estudo da precisão será conduzido neste trabalho a partir da análise da precisão dos métodos para um grande conjunto de casos de teste com variações em diversos parâmetros do modelo.

A precisão dos métodos aproximados de Larson e Jarvis está relacionada à combinação de diferentes fatores cujos efeitos podem se anular, como o balanceamento das demandas, áreas de cobertura e taxas de serviço. Alguns dados que poderiam indicar o resultado da combinação destes efeitos (como as frações de despacho) são obtidos apenas após a solução do modelo.

Neste estudo de precisão dos métodos serão consideradas 6 medidas de desempenho calculadas a partir do modelo. A escolha destas medidas foi motivada pela sua importância no planejamento de sistemas emergenciais. São elas: carga de trabalho dos servidores, fração de todos os despachos entre áreas, frações de despachos entre áreas dos servidores, tempo médio de

viagem do sistema, tempo médio de viagem para os átomos geográficos e tempo médio de viagem para os servidores.

Procurando superar a dificuldade para captar a interação entre os fatores citados anteriormente, os testes serão executados para dois grupos distintos de problemas teste: com demanda e preferência de despacho balanceada e com demanda e preferência de despacho não balanceadas.

Para tanto, foram gerados 25650 problemas testes com variações em diversos parâmetros do modelo, tais como: número de servidores (N), número de átomos geográficos (NA), taxa de ocupação, entre outros. Na próxima seção serão apresentados os principais resultados destes testes.

Para a realização dos testes com o método aproximado, a biblioteca descrita no Apêndice A foi utilizada.

5.4.1 Gerador de problemas testes

Os problemas testes para os métodos aproximados foram obtidos a partir de um gerador implementado em Java que recebe como dados de entrada a taxa de ocupação do sistema (ρ), o desvio padrão das taxas de serviço (σ_S), o número de servidores (N) e de átomos geográficos (N_A), a forma de distribuição da demanda entre os átomos geográficos (aleatória ou balanceada) e de geração das preferências de despacho (balanceada e aleatória de acordo com a política *Expected Modified Center of Mass*) e o tamanho máximo para a fila do sistema.

O gerador de números pseudo-aleatórios utilizado foi o *Mersenne Twister* (MATSUMOTO; NISHIMURA, 1998). Este gerador apresenta período de $2^{19937}-1$ e apresenta resultado positivo para muitos testes estatísticos de aleatoriedade, como o teste de Diehard.

Como saída, o gerador de problemas testes produz o modelo de um sistema que não permite a formação de filas de espera, com seus átomos geográficos e

demandas, servidores e taxas de serviço, distâncias entre átomos, localização dos servidores e matriz de despachos.

As taxas de serviço são geradas aleatoriamente (distribuídas de acordo com uma distribuição normal – a escolha da distribuição normal justifica-se pela maior probabilidade, em sistemas reais, dos servidores terem tempos de serviço próximos) a partir de um valor de desvio padrão, e as demandas são geradas de forma que o sistema tenha a taxa de ocupação especificada e distribuída igualmente (balanceadas) ou aleatoriamente entre os átomos geográficos.

Para preferências de despacho balanceadas, o parâmetro N_A é desconsiderado pelo gerador, o número de átomos geográficos é igualado ao número de servidores e cada servidor é definido como primário para um dos átomos geográficos e secundário para outro. Para preferências de despacho aleatórias, sorteia-se aleatoriamente a posição dos átomos em um plano cartesiano, define-se aleatoriamente a distância referente à unidade no plano, calcula-se o tempo médio de viagem entre os átomos a partir destas posições e da distância referente à unidade, sorteiam-se os átomos onde os servidores estão localizados e, por fim, determina-se a preferência de despacho para cada átomo em relação a distância mínima dos servidores (todos distribuídos de acordo com uma distribuição uniforme).

Os testes de precisão do método foram realizados variando-se o número de servidores de N=2 a N=20 (com incrementos de 1), as demandas e preferências de despacho (75 combinações por problema para servidores homogêneos e 5 para servidores não-homogêneos), a taxa de ocupação de ρ=0,1 a ρ=0,9 (com incrementos de 0,1) e, para sistemas com servidores não-homogêneos, o desvio padrão das taxas de serviço de σ_S=1 a σ_S=15 (com incrementos de 1), totalizando 12825 problemas testes para sistemas com servidores homogêneos e 12825 para não-homogêneos.

A seguir, os resultados obtidos a partir destes testes são apresentados e analisados.

5.4.2 Resultados obtidos

Os problemas testes gerados foram resolvidos através do método exato (foi utilizado o método de Gauss-Seidel para a solução do sistema de equações lineares) e dos métodos aproximados de Larson e de Jarvis. Foram adotados dois critérios de convergência e um critério de parada definidos pelas Equações 5.11, 5.12 e 5.13, respectivamente: o erro absoluto máximo, o erro relativo máximo e o número de iterações.

$$\max\left|x_n^{\,i+1} - x_n^{\,i}\right| \leq 1{,}0E^{-10} \tag{5.11}$$

$$\max\left|\frac{x_n^{\,i+1} - x_n^{\,i}}{x_n^{\,i+1}}\right| \leq 1{,}0E^{-6} \tag{5.12}$$

$$iter \leq 500 \tag{5.13}$$

onde x_n^{i+1} e x_n^{i} representam as variáveis estudadas cujos valores são calculados pelos métodos nas etapas i+1 e i, respectivamente, e *iter* representa o número de iterações do método.

Após a solução dos problemas, os erros das medidas de desempenho foram calculados. As medidas de erro adotadas para as medidas de desempenho analisadas foram o erro absoluto médio (EAM) e o erro médio relativo (EMR). De um modo geral, para uma determinada medida de desempenho x, as seguintes fórmulas foram utilizadas para calcular os erros:

$$EAM = \sum_i \left|x_i - x_i^{\,*}\right| \Big/ M \tag{5.14}$$

$$EMR = \sum_i \left|x_i - x_i^{\,*}\right| \Big/ \sum_i x_i \tag{5.15}$$

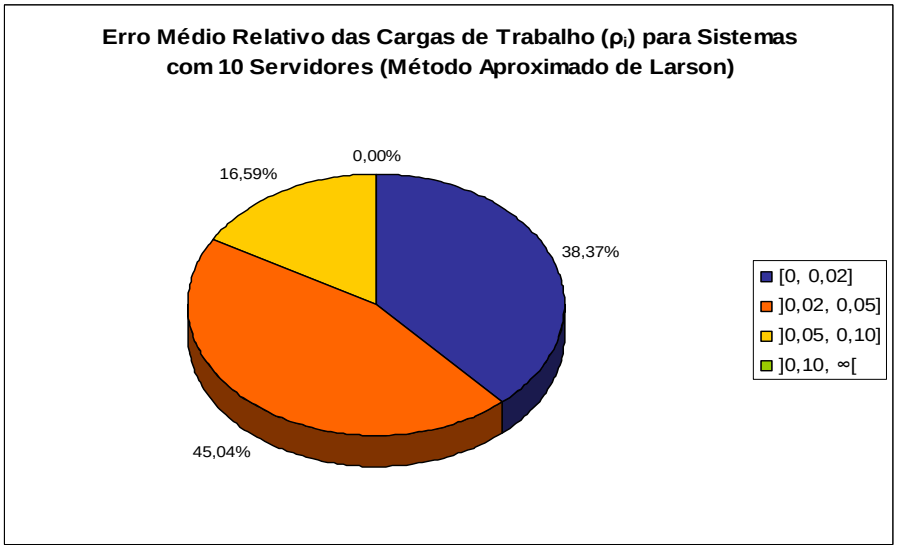
onde x_i e x_i* representam a medidas de desempenho exata e aproximada, respectivamente, e M o número de observações para a medida de desempenho x (p.ex.: para as cargas de trabalho de sistemas com 20 servidores, o erro média absoluto é igual a $\sum_i \left|\rho_i - \rho_i^*\right| \Big/ 20$).

Para a apresentação dos resultados, os problemas testes foram separados em dois grupos. O primeiro grupo envolveu testes realizados com sistemas com servidores homogêneos e o segundo grupo envolveu testes realizados com sistemas com servidores não-homogêneos, variando-se os parâmetros conforme intervalos definidos na seção anterior.

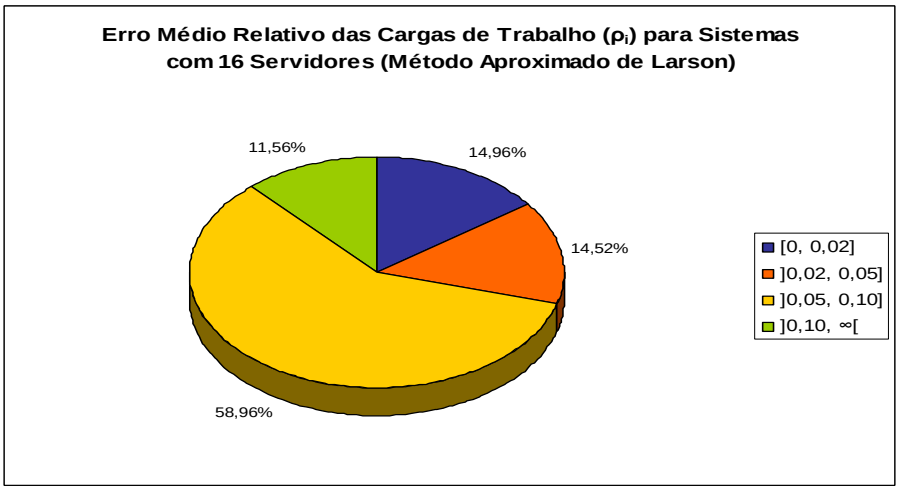
5.4.2.1 Resultados para sistemas com servidores homogêneos

Os resultados do método aproximado de Larson, obtidos para este primeiro grupo, foram ligeiramente diferentes das observações publicadas por Larson (1974a, 1975a), Larson e Odoni (1981) e Chaiken (1975) no que diz respeito ao cálculo de cargas de trabalho com erros relativos geralmente entre 1% e 2%. Para sistemas com 10 servidores ou mais, a maior parte dos problemas testes apresentou erros superiores a 2% (Figura 5.1).

Para alguns problemas testes, foram obtidos erros relativos superiores a 10% para as cargas de trabalho dos servidores (como pode ser observado na Figura 5.1b para sistemas com 16 servidores). As outras medidas de desempenho (mais agregadas) calculadas pelo método, no entanto, apresentaram erros inferiores aos erros calculados para as cargas de trabalho e, geralmente, entre 1% e 2%.



(a)



(b)

Figura 5.1 – Porcentagem de problemas para faixas de erros das cargas de trabalho do sistema.

Como exemplo dos erros calculados pelo método de Larson, a seguir estão listados os resultados obtidos para um dos sistemas gerados com 19 servidores.

Tabela 5.1 – Erros nas medidas de desempenho calculadas através do método aproximado de Larson para um dos sistemas gerados com N=19 e ρ=0,3.

Medida de Desempenho	Erro Absoluto	Método Aproximado de Larson
Carga de Trabalho	Mínimo	0,002
	Médio	0,026 (Erro Médio Relativo: 8%)
	Máximo	0,073
Fração Total de Despacho Entre Áreas	Exato	0,008 (Exato: 0,71313; Aproximado: 0,72211)
Frações de Despacho Entre Áreas dos Servidores	Médio	0,006
Tempo Médio de Viagem do Sistema	Médio	0,001 (Exato: 0,31888; Aproximado: 0,31754)
Tempo Médio de Viagem dos Servidores	Médio	0,005
Tempo Médio de Viagem para os Átomos Geográficos	Médio	0,001

Quanto às observações feitas por Larson referentes à relação entre os erros do método e o desbalanceamento das demandas e áreas de cobertura, realmente

pôde-se observar que, para os problemas com demandas e preferências de despacho aleatórias, os erros foram maiores do que os erros obtidos para os problemas com demandas e preferências de despacho balanceadas (que se aproximaram de zero).

Para avaliar com maior clareza esta relação entre o balanceamento das demandas e preferências de despacho e a precisão do método, para cada problema teste foi calculado o desvio médio da demanda atendida pelos servidores (D_n) através da Equação 5.16.

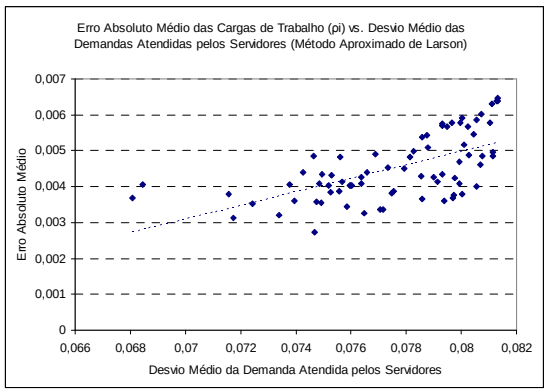
$$D_n = \sum_{a=1}^{N_A} \lambda_a \cdot f_{na} \qquad (5.16)$$

Os resultados indicam que os erros das cargas de trabalho dos servidores apresentam uma correlação linear positiva em função do desvio médio da demanda atendida pelos servidores (Figura 5.2).

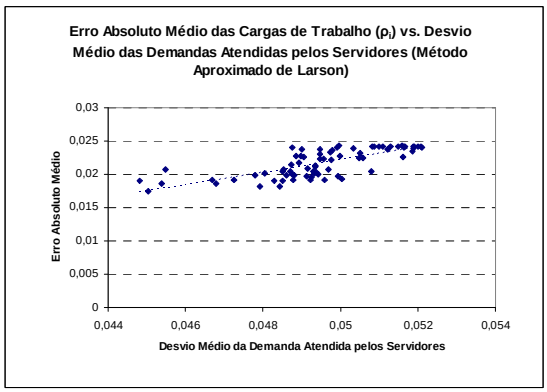
Apesar de variações serem perdidas quando valores médios são utilizados para o estabelecimento de correlações, a correlação observada é válida para identificar o comportamento do método aproximado de Larson em função do desbalanceamento da demanda e da área de cobertura.

Para as outras medidas de desempenho, não foi possível observar uma relação clara entre o desvio médio das demandas atendidas pelos servidores e os erros calculados.

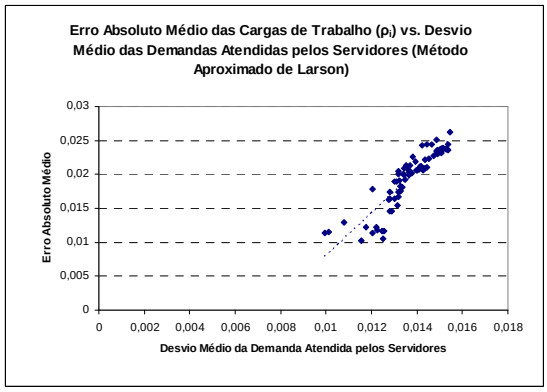
Como o uso dos métodos aproximados para a solução do modelo Hipercubo é importante principalmente para o caso de sistemas com muitos servidores, é interessante analisar o comportamento dos métodos em função do número de servidores. Segundo Larson (1975a), a precisão do seu método aproximado parece ser diretamente proporcional ao número de servidores do sistema. Entretanto, os resultados encontrados neste trabalho apontam para outra direção.



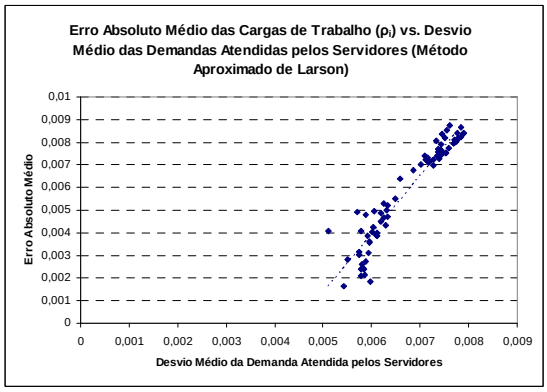
(a)



(b)



(c)



(d)

Figura 5.2 – Relação entre o erro das cargas de trabalho e o desvio da demanda atendida pelos servidores para sistemas com 15 servidores e taxa de ocupação (a) 0,1; (b) 0,3; (c) 0,7; (d) 0,9. Cada gráfico compreende o resultado obtido para 75 diferentes modelos.

Porém, para os problemas testes estudados, o método de Larson apresentou, diferentemente do que Larson havia observado, precisão das cargas de trabalho inversamente proporcional ao número de servidores (Figura 5.3). Para as outras medidas de desempenho mais agregadas, esta relação na ficou tão clara.

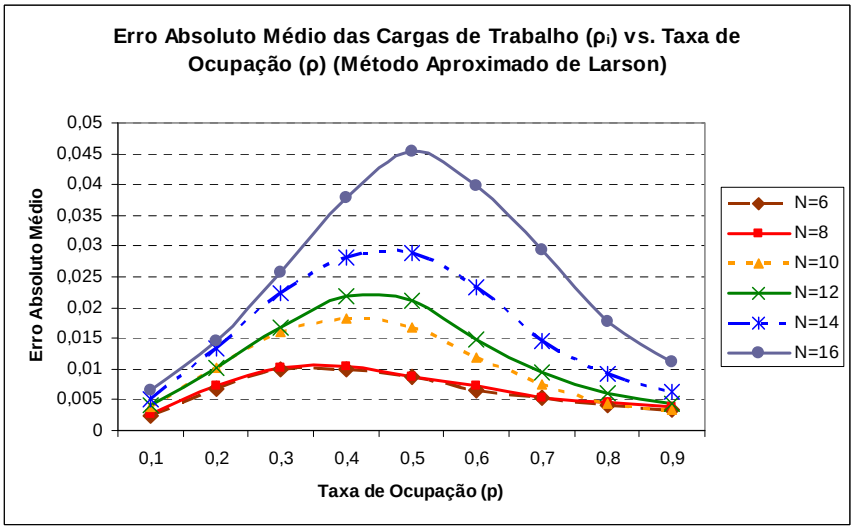


Figura 5.3 – Erro absoluto médio da carga de trabalho sistemas com 6, 8, 10, 12, 14 e 16 servidores. (Para cada ponto do gráfico foram considerados 75 problemas).

O método de Jarvis, por sua vez, apresentou erros absolutos das cargas de trabalho de até 1,8 para muitos problemas testes, resultado bastante divergente do esperado, já que Jarvis (1985) encontrou erros relativos geralmente inferiores a 4%. Estes erros foram superiores a 10% a partir de sistemas com $p \geq 0,2$.

Para as frações de despachos entre áreas dos servidores, os erros absolutos para sistemas com $p \geq 0,2$ superaram, em alguns casos, 0,90. Os erros absolutos no tempo médio de viagem do sistema não ultrapassaram 0,2 e os erros dos tempos de viagem para os átomos geográficos e para os servidores apresentaram, raramente, erros superiores a 0,10.

Quanto ao desvio médio das demandas atendidas pelos servidores, diferentemente do método de Larson, não foi possível observar um padrão claro para o método de Jarvis.

A relação entre o número de servidores do modelo e a precisão do método de Jarvis foi mais clara do que a observada para o método de Larson. Para o

método de Jarvis, o erro de praticamente todas as medidas de desempenho estudadas mostrou-se diretamente proporcional ao número de servidores do modelo (Figura 5.4)

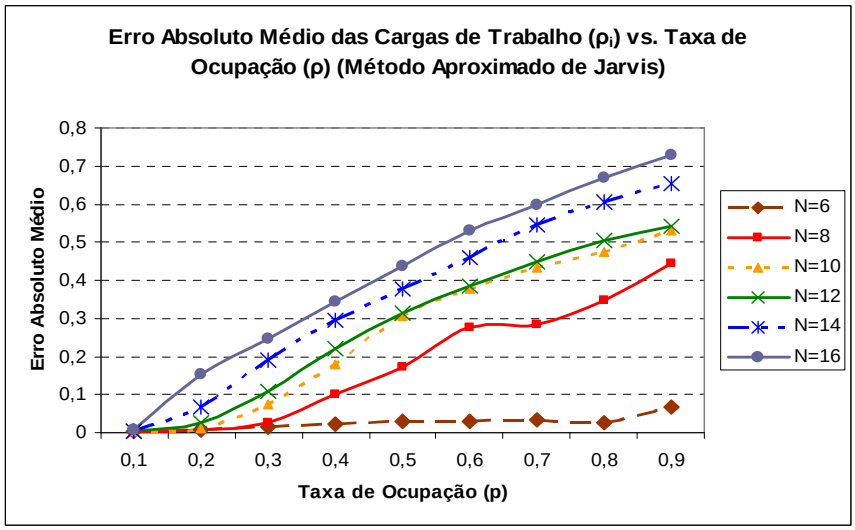


Figura 5.4 – Erro absoluto médio da carga de trabalho para sistemas com 6, 8, 10, 12, 14 e 16 servidores resolvidos através do método aproximado de Jarvis.

Em relação à taxa de ocupação do sistema, os métodos aproximados de Larson e Jarvis comportam-se de forma distinta. Enquanto o método de Larson apresenta erros maiores para sistemas com taxa de ocupação de baixa à média, o método de Jarvis apresenta erros crescentes em função da taxa de ocupação.

Jarvis (1985) observou com curiosidade que os erros relativos calculados pelo seu método são maiores para sistemas com taxas de ocupação baixas, mas não é difícil perceber que, para sistemas com baixa taxa de ocupação, as cargas de trabalho dos servidores assumem valores pequenos e, portanto, qualquer variação é sentida significativamente em medidas relativas.

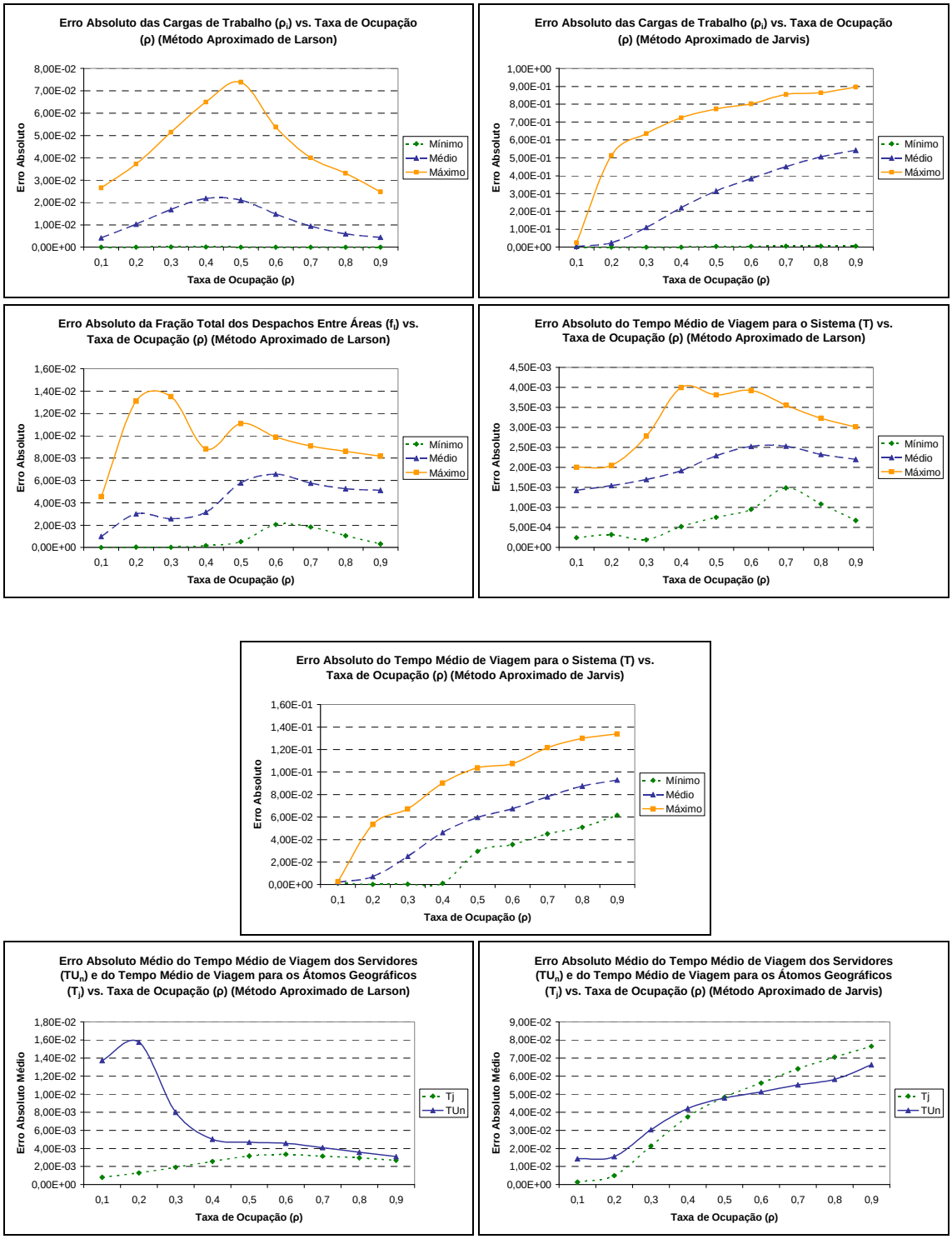


Figura 5.5 – Erros de algumas das medidas de desempenho estudadas para sistemas com 12 servidores resolvidos através do método aproximado de Larson e de Jarvis.

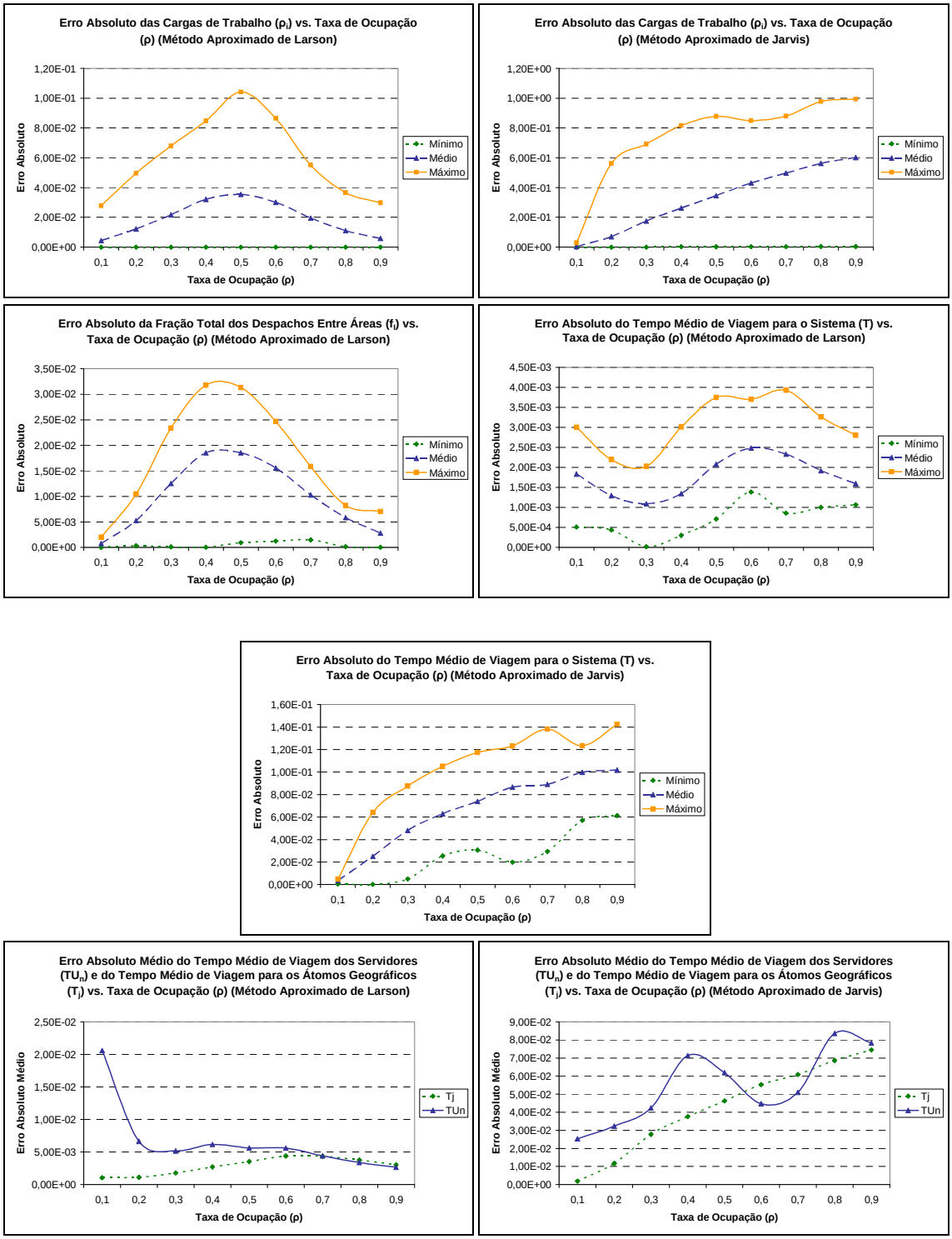


Figura 5.6 – Erros de algumas das medidas de desempenho estudadas para sistemas com 15 servidores resolvidos através do método aproximado de Larson e de Jarvis.

5.4.2.2 Resultados para sistemas com servidores não-homogêneos

O segundo grupo de testes envolveu sistemas que, além de apresentarem variações nos parâmetros citados para o primeiro grupo, apresentam servidores não-homogêneos.

Os erros resultantes da aplicação do método Larson a estes sistemas representam a diferença destes resultados em relação aos que seriam obtidos se os servidores fossem homogêneos. Deste modo, os erros encontrados por Morabito et al. (2008) podem ser vistos como limites inferiores (ou próximos aos limites inferiores) dos erros calculados pelo método de Larson.

Conforme observado pelos autores, a modelagem de servidores não-homogêneos como homogêneos pode produzir erros significativos, mesmo para um baixo de nível de não homogeneidade. Estes resultados foram confirmados por Luque e Carvalho (2006), que identificaram erros significativos para as cargas de trabalho de modelos com baixo nível de não homogeneidade.

Entretanto, como nosso objetivo é avaliar a utilidade do método como estimativa inicial das cargas de trabalho para estes sistemas, consideramos que, mesmo apresentando erros, uma característica desejável é que seja mantida a relação de ordem entre as cargas de trabalho dos servidores. Caso essa relação não seja mantida, interpretações equívocas podem ser produzidas a partir dos dados aproximados.

O número de inversões foi calculado a partir do número necessário de substituições na relação de ordem das cargas de trabalho aproximadas, para que fosse obtida a mesma relação de ordem das cargas exatas. Apesar de não considerar a distância entre as cargas invertidas (p.ex. uma inversão entre a primeira e a última carga é contada de forma igual a uma inversão entre a penúltima e a última carga), este cálculo é suficiente para o contexto deste trabalho.

Os resultados obtidos neste segundo grupo indicam que, mesmo para sistemas com desvios médios das taxas de serviço moderados, a relação de ordem das cargas de trabalho pode sofrer inversão total. Também se pode verificar que, com o balanceamento das demandas e área de cobertura, o erro passa a apresentar uma tendência linear crescente em função do desvio médio das taxas de serviço.

O cálculo do número de inversões também foi realizado para o caso de servidores homogêneos e verificou-se que em mais de 50% dos problemas testes, houve pelo menos uma inversão nas cargas de trabalho.

As mesmas relações observadas sobre a precisão do método e o número de servidores (N), e taxas de ocupação (ρ) para o caso de servidores homogêneos, foram observadas para o caso de servidores não-homogêneos.

5.5 Discussões

Os resultados obtidos indicam que, para sistemas com servidores homogêneos, o método aproximado de Larson é mais apropriado que o método de Jarvis. Para todas as medidas desempenho, os valores mínimo, médio e máximo são raramente superiores às obtidas através do método Jarvis.

Mesmo para sistemas com servidores heterogêneos, o método de Larson apresentou uma média de erros relativos inferior a media apresentada pelo método de Jarvis para todas as medidas de desempenho.

Diferentemente daquilo que Larson observou, a precisão do método parece ser inversamente proporcional ao número de servidores do modelo, o que coloca em questão o uso dos métodos em situações nas quais são mais úteis: modelagem de sistemas de grande porte.

Na próxima seção serão estudadas algumas modificações, principalmente no método de Jarvis, que podem garantir resultados melhores que aqueles obtidos pelo método original.

6 AVALIAÇÃO DE MODIFICAÇÕES NOS MÉTODOS APROXIMADOS

Este capítulo apresenta uma análise de modificações nos métodos aproximados de solução do modelo Hipercubo de Filas estudados no capítulo anterior. Estas modificações estão direcionadas a garantia de existência e unicidade de solução para os métodos e em algumas alternativas sugeridas pelos resultados apresentados no capítulo anterior. Os mesmos testes realizados no capítulo anterior foram repetidos com as modificações estudadas neste capítulo e uma comparação entre os resultados das versões modificadas e originais dos métodos é apresentada.

6.1 Considerações iniciais

No capítulo anterior, foram apresentados resultados para a precisão dos métodos aproximados de Larson (1974a, 1975a) e de Jarvis (1975, 1985), conforme definidos originalmente. Estes resultados mostraram um cenário, em geral, diferente daquele até então conhecido principalmente no que se refere à relação entre a precisão dos métodos e o número de servidores do modelo e à precisão, como um todo, do método aproximado de Jarvis.

No que diz respeito à relação entre a precisão de algumas medidas de desempenho calculadas pelos métodos aproximados e o número de servidores do modelo, pôde-se observar que, para os problemas testes gerados neste trabalho, a precisão do método é inversamente proporcional ao número de servidores para duas importantes medidas de desempenho.

Em relação ao método de Jarvis, foram encontrados erros muito superiores àqueles observados por Jarvis (1985).

No Capítulo anterior, foram comentadas algumas modificações que podem ser feitas nos métodos aproximados para que seja garantida a existência e unicidade de solução. Estas modificações, propostas por Goldberg e Szidarovszky (1991a,b), estão relacionadas à etapa de inicialização e de

normalização para o método aproximado de Larson e à etapa de inicialização e de atualização dos valores de ρ para o método de Jarvis.

Apesar de Goldberg e Szidarovszky (1991a,b) terem apresentado resultados referentes à convergência da versão do método proposta pelos autores e do método de Jarvis, não foram apresentados detalhes sobre a precisão das versões modificadas estudadas pelos autores.

Assim sendo, neste capítulo, os mesmos testes realizados no capítulo anterior com os métodos de Larson e de Jarvis serão realizados com as versões modificadas destes métodos.

A escolha das modificações que foram consideradas foi orientada pelas modificações necessárias para garantir a existência e unicidade de solução dos métodos. Além disso, foram testadas algumas alternativas sugeridas por Goldberg e Szidarovszky (1991a,b), como a solução das equações não-lineares através da substituição dos valores mais recentes disponíveis de ρ_i (o que denotamos de processo de iteração Seidel) e a solução do método em duas etapas.

A solução em duas etapas envolve a utilização do método aproximado de solução duas vezes seguidas. Na primeira delas, o modelo é resolvido assumindo que Q(N, ρ, j) é igual a 1 para qualquer combinação de valores de N, ρ e j. Após a primeira solução, a taxa de ocupação do sistema é calculada a partir da Equação 6.1 e o método é resolvido novamente, agora com os valores desta nova taxa de ocupação e os valores dos fatores de correção conforme originalmente definidos.

$$\rho = \frac{\sum_{j=1}^{N} \rho_j}{N(1 - P_N)} \qquad (6.1)$$

Para o método de Larson, as modificações testadas neste capítulo envolveram a modificação da etapa de inicialização (com valores de p_i^0=1 e p_i^0=0, para i=1,2,...,N), a modificação da etapa de normalização do método (normalização realizada apenas após o término do processo), a utilização do processo de iteração Seidel e a solução em duas etapas, todas propostas no trabalho de Goldberg e Szidarovszky (1991a,b), conforme combinações da Tabela 6.1.

Tabela 6.1 – Modificações estudadas para o método aproximado de Larson.

Modificação	Opções	Quantidade
Inicialização	p_i^0=0; p_i^0=1 ou p_i^0=r, para i=1,2,...,N	3
Normalização	Apenas no fim do processo.	1
Substituição	Normal/Seidel	2
Etapas	Normal/Duas	2
Total de modificações (combinações)		12

Para o método de Jarvis foi considerado um número maior de modificações. As modificações envolveram, além daquelas necessárias para a garantia de convergência do método e da modificação do processo de iteração e do número de etapas, algumas modificações testadas pelos autores deste trabalho (p.ex.: a solução do método com o valor de ρ fixo, mas mantendo a etapa de inicialização conforme definida originalmente ou inicializando os valores de ρ_i com os valores obtidos a partir do método de Larson para o mesmo problema).

As modificações para o método de Jarvis estão detalhadas na Tabela 6.2.

Tabela 6.2 – Modificações estudadas para o método aproximado de Jarvis.

Modificação	Opções	Quantidade
Inicialização	p_i^0=0; p_i^0=1 ou p_i^0=normal, p_i^0=Larson para i=1,2,...,N	4
ρ fixo	Sim/Não	2
Substituição	Normal/Seidel	2
Etapas	Normal/Duas	2
Total de modificações (combinações – excluída aquela que resulta no modelo original)		31

A seguir, os resultados obtidos para estas modificações são apresentados e analisados.

6.1.1 Resultados obtidos

Os resultados para todas as combinações que utilizaram o processo de iteração Seidel apresentaram, de uma forma geral, erros maiores ou iguais aqueles obtidos pelo processo normal de substituição, tanto para o método aproximado de Larson quanto para o método aproximado de Jarvis.

Com a alteração da etapa de inicialização do método de Larson (tanto para ρ_i^0=1 quanto para ρ_i^0=0), o método convergiu para a mesma solução obtida quando ρ_i^0=r e, geralmente, no mesmo número de iterações.

Para o método de Jarvis, entretanto, a modificação na etapa de inicialização fez com que o método convergisse para soluções com todas as cargas de trabalho próximas de 1 e com valores similares, o que exigiu do método poucas iterações para $\rho_i^0=1$, mas muitas iterações para $\rho_i^0=0$. Os erros apresentados para esta modificação são muito maiores do que aqueles apresentados pelo método original.

Isto mostra que, apesar de garantir a convergência e unicidade de solução do método, o valor calculado para as cargas de trabalho através da alteração da etapa de inicialização do método de Jarvis difere significativamente daquele calculado pelo método conforme originalmente definido e, portanto, apresenta erros muitos maiores.

A inicialização do método com os valores de cargas de trabalho calculadas pelo método de Larson garantiu a convergência do método em um número menor de iterações.

Por fim, as modificações referentes a fixar o valor de ρ e a realização de duas etapas de solução do método com valores fixos de p apresentaram excelentes resultados para a inicialização normal do método de Jarvis (modificação sugerida neste trabalho).

Com estas modificações, o método sempre convergiu para soluções válidas e com erros comparativamente pequenos em relação ao método conforme definido originalmente. Para um sistema com 18 servidores gerado, os seguintes resultados foram obtidos para o método de Jarvis e suas duas versões modificadas citadas.

Tabela 6.3 – Erros nas medidas de desempenho para o método de Jarvis original, com ρ fixo (em 1 e 2 etapas) para N=18.

Medida de Desempenho	Erro Absoluto	Jarvis	Jarvis com ρ fixo	Jarvis com ρ fixo (2 etapas)
Carga de Trabalho	Mínimo	1,64E-07	2,07E-08	2,46E-09
	Médio	0,316	0,022	0,024
	Máximo	0,921	0,133	0,137
Fração Total de Despacho Entre Áreas	Mínimo	1,59E-03	1,60E-03	1,60E-03
	Médio	0,135	0,010	0,010
	Máximo	0,272	0,027	0,026
Frações de Despacho Entre Áreas dos Servidores	Médio	0,601	0,008	0,008
Tempo Médio de Viagem do Sistema	Mínimo	2,52E-06	5,57E-06	3,36E-06
	Médio	0,013	0,001	0,001
	Máximo	0,231	0,006	0,005
Tempo Médio de Viagem dos Servidores	Médio	2,65E-02	2,19E-03	1,87E-03
Tempo Médio de Viagem para os Átomos Geográficos	Médio	4,19E-02	4,05E-03	3,93E-03

Para as medidas de fração de despacho e tempos de viagem, a versão com 2 etapas do método apresentou resultados melhores do que a versão apenas com ρ fixo. Para as cargas de trabalho, está última modificação foi melhor.

Entretanto, as mesmas relações verificadas entre a precisão do método e o número de servidores e a relação com as cargas de trabalho pôde ser observada para as versões modificadas.

6.2 Discussões

Os resultados apresentados mostraram que as modificações propostas por Goldberg e Szidarovszky (1991a,b) para garantir a convergência do método de Jarvis não apresentam bons resultados, convergindo quase sempre para resultados próximos de 1 e com cargas de trabalho similares. Mesmo a versão com 2 etapas para valores iniciais 1 apresentou resultados ruins.

Entretanto, foi testada neste trabalho uma modificação que mantém a etapa de inicialização conforme originalmente definida e mantém fixos os valores de ρ. A versão em uma ou duas etapas com iteração normal apresentou excelentes resultados, garantindo a convergência dos métodos para resultados válidos e com erros muito inferiores aos produzidos pelo método conforme originalmente definido. Quando comparado ao método de Larson, esta modificação do método de Jarvis apresentou melhores resultados para o caso de sistemas com servidores não-homogêneos.

As modificações para o método de Larson, por sua vez, não apresentaram melhorias no resultado do método. A modificação na etapa de inicialização e normalização convergiu sempre para os mesmo resultados do método conforme originalmente definido. A modificação do método com 2 etapas não se mostrou uma boa alternativa e apresentou erros muito superiores a versão original.

7 CONCLUSÕES E PERSPECTIVAS

Este trabalho apresentou um estudo de alguns dos principais métodos de solução do modelo Hipercubo de Filas para sistemas de grande porte. Neste capítulo, são apresentadas as principais conclusões obtidas durante o desenvolvimento deste trabalho e algumas perspectivas de pesquisas para trabalhos futuros.

7.1 Conclusões

Os resultados publicados a respeito dos principais métodos aproximados de solução do modelo Hipercubo de Filas, algumas vezes, generalizam incorretamente seu comportamento.

Os resultados para um grande conjunto de casos de teste mostraram que a versão original do método de Jarvis não é a mais apropriada para a solução do modelo Hipercubo. Nestes casos, o método aproximado de Larson apresenta erros significativos proporcionais ao desbalanceamento das taxas de serviço, demanda e área de cobertura.

No entanto, apesar deste cenário desfavorável para a solução de sistemas não-homogêneos com muitos servidores, foram estudadas modificações no método de Jarvis cujos resultados sempre convergiram para soluções válidas e com precisão boa.

As modificações propostas por Goldberg e Szidarovszky (1991a, 1991b) para garantir a convergência dos métodos de Larson e Jarvis mostraram resultados distintos. Para o método de Larson, não se pode observar alterações nos resultados para a versão modificada em relação à garantia de convergência. Já para o método de Jarvis, o método modificado convergiu geralmente para soluções com cargas de trabalho muito próximas de 1,0 e, portanto, produziu erros superiores a 100%.

Nas comparações realizadas entre os métodos aproximados e variantes, pode-se observar que os métodos aproximados comportam-se de forma distinta em função do número de servidores e das taxas de ocupação do sistema.

Para o caso de sistemas com servidores homogêneos, o método de Larson apresentou resultados melhores do que aqueles apresentados pelo método de Jarvis.

No caso dos servidores não-homogêneos, o método de Jarvis modificado (com p fixo e inicialização normal) apresentou resultados muito mais precisos que o método original de Jarvis. Foi testada também uma forma de solução proposta por Goldberg e Szidarovszky (1991a, 1991b) que consiste na solução do método em 2 etapas. Entretanto, os resultados desta modificação não apresentam grandes diferenças em relação à versão modificada com ρ fixo.

Pode-se observar que os erros de algumas das medidas de desempenho calculadas pelos métodos aproximados é diretamente proporcional ao número de servidores do modelo, o que reduz a utilidade dos métodos para os casos em que seriam necessários, mas para as medidas de desempenho mais agregadas esta relação não ficou tão clara. Assim sendo, os resultados obtidos neste trabalho, sugerem que, no cálculo de cargas de trabalho, o uso dos métodos aproximados não é indicado para sistemas com muitos servidores.

7.1.1 Perspectivas para futuras pesquisas

Da análise dos resultados, pode-se observar que o uso do modelo Hipercubo de Filas para a modelagem de sistemas com muitos servidores ainda é um problema, dadas as limitações do método exato e a proporcionalidade dos erros de algumas medidas de desempenho, calculadas pelos métodos aproximados, e do número de servidores.

Assim sendo, o estudo da aplicação de métodos de decomposição ainda constitui um tema interessante para pesquisa. Principalmente o estudo dos métodos de decomposição para matrizes quase completamente particionáveis.

REFERÊNCIAS BIBLIOGRÁFICAS

ALBINO, J. C. C. **Quantificação e locação de unidades móveis de atendimento de emergência e interrupção de redes de distribuição de energia elétrica:** aplicação do modelo Hipercubo. 1994. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Catarina (UFSC), Florianópolis, 1994.

ANDRADE, V. M. B.; CARVALHO, S. V.; VIJAYKUMAR, N. L. **Desenvolvimento de um software para análise de sistemas através de modelos markovianos** - uma abordagem orientada a objetos. São José dos Campos, Instituto Nacional de Pesquisas Espaciais (INPE), 1996.

BASHARIN, G. P.; LANGVILLE, A. N.; NAUMOV, V. A. The life and work of A. A. Markov. **Linear Algebra and Applications**, v. 386, p. 3–26, 2004. Disponível em: <http://citeseer.ist.psu.edu/670557.html>. Acesso em: 30 junho 2007.

BATTA, R.; DOLAN, J. M.; KRISHNAMURTHY, N. N. The maximal expected covering location problem: revisited. **Transportation Science**, v. 23, p. 277-287, 1989.

BENVENISTE, R. Solving the combined zoning and location problem for several emergency units. **The Journal of the Operational Research Society**, v. 36, n. 5, p. 433-450, 1985.

BERMAN, O.; LARSON, R. C. The median problem with congestion. **Computers and Operations Research**, v. 9, p. 119-126, 1982.

BERMAN, O.; LARSON, R. C.; PARKAN, C. The stochastic queue p-median problem. **Transportation Science**, v. 21, p. 207-216, 1987.

BIRGE, J. R.; POLLOCK, S. M. Using parallel iteration for approximate analysis of a multiple server queueing system. **Operations Research**, v. 37, n. 5, p. 769-779, 1989.

BLAHA, M.; RUMBAUGH, J. **Modelagem e projetos baseados em objetos com UML 2:** tradução. Rio de Janeiro: Campus, 2006.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML - guia do usuário:** tradução. Rio de Janeiro: Campus, 2000.

BRANDEAU, M.; LARSON, R. C. Extending and applying the Hypercube Queueing model to deploy ambulances in Boston. **TIMS Studies in the Management Sciences**, v. 22, p. 121-153, 1986.

BURWELL, T. H.; JARVIS, J. P.; MCKNEW, M. A. Ambulance location using a spatially distributed queueing model. In: Annual Meeting of The American Institute for Decision Science, 17., 1985, [S.l.]. **Proceedings...** [S.l.]: [s.n.], 1985. p. 705-707.

BURWELL, T. H.; JARVIS, J. P.; MCKNEW, M. A. Modeling co-located servers and dispatch ties in the hypercube model, **Computers and Operations Research**, v. 20, n. 2, 1993, p. 113-119.

CAMPBELL, G. L. **A spatially distributed queueing model for police patrol sector design**. 1972. Thesis (Master of Science) – Massachusetts Institute of Technology (MIT), Cambridge, 1972.

CAO, W.; STEWART, W. J. Iterative aggregation/disaggregation techniques for nearly uncoupled Markov chains. **Journal of the Association for Computing Machinery**, v. 32, n. 3, p. 702-719, 1985.

mCHAIKEN, J. M. **Hypercube queuing model:** executive summary. New York: The Rand Corporation, 1975. 17 p. Disponível em: <http://www.rand.org/pubs/reports/2006/R1688.1.pdf>. Acesso em: 30 jun. 2007.

CHAIKEN, J. M. Transfer of emergency service deployment models to operating agencies. **Management Science**, v. 24, p. 719-731, 1978.

CHAIKEN, J. M.; DORMONT, P. A patrol car allocation model: background. **Management Science**, v. 24, n. 12, p. 1280-1290, 1978a.

CHAIKEN, J. M.; DORMONT, P. A patrol car allocation model: capabilities and algorithms. **Management Science**, v. 24, n. 12, p. 1291-1300, 1978b.

CHELST, K. R. **Implementing the hypercube queueing model in the New Haven department of police services:** a case study in technology transfer. New York: The Rand Corporation, 1975. 73 p. Disponível em: <http://www.rand.org/pubs/reports/2006/R1566.6.pdf>. Acesso em: 30 jun. 2007.

CHELST, K.; JARVIS, J. P. Estimating the probability distribution of travel times for urban emergency service systems. **Operations Research**, v. 27, n. 1, p.199-205, 1979.

CHELST, K. R.; BARLACH, Z. Multiple unit dispatches in emergency services: models to estimate system performance, **Management Science**, v. 27, n.12, 1981, p. 1390–1409.

CHIYOSHI, F. Y.; GALVÃO, R. D.; MORABITO NETO, R. O uso do modelo hipercubo na solução de problemas de localização probabilísticos. **Gestão &**

Produção, v. 7, p. 146-174, 2000. Disponível em: <http://www.scielo.br/pdf/gp/v7n2/a05v7n2.pdf>. Acesso em: 30 jun. 2007.

CHIYOSHI, F. Y.; GALVÃO, R. D.; MORABITO NETO, R. Modelo hipercubo: análise e resultados para o caso de servidores não-homogêneos. **Pesquisa Operacional**, v. 21, n. 2, p. 199-218, 2001. Disponível em: <http://www.scielo.br/pdf/pope/v21n2/a05v21n2.pdf >. Acesso em: 17 set. de 2007.

CHIYOSHI, F. Y.; GALVÃO, R. D.; MORABITO NETO, R. A note on solutions to the maximal expected covering location problem. **Computers and Operations Research**, v. 30, n. 1, p. 87-96, 2003.

CHURCH, R. L.; REVELLE, C. S. The maximal covering locations problem. **Papers of the Regional Science Association**, v. 32, p. 101-120, 1974.

COOPER, R. B. **Introduction to queueing theory**. 2 ed. London: Edward Arnold, 1981.

COSTA, D. M. B. **Uma metodologia iterativa para determinação de zonas de atendimento de serviços emergenciais**. 2003. Tese (Doutorado em Engenharia de Produção) – Universidade Federal de Santa Catarina (UFSC), Florianópolis, 2003.

DASKIN, M. S. A maximum expected covering location model. **Transportation Science**, v. 17, n. 1, p. 48-70, 1983.

DAYAR, T.; STEWART, W. J. **Exploring states of sparse, large Markov chains**. Ankara, Turkey: Bilkent University - Department of Computer Engineering and Information Science, Apr. 1995.

DAYAR, T.; STEWART, W. J. Comparison of partitioning techniques for two-level iterative solvers on large, sparse Markov chains. **SIAM Journal on Scientific Computing**, v. 21, n. 5, p. 1691-1705, 2000.

EXECUTIVE guide to operations research. Linthicum, 2004. 16 p. Disponível em: <http://www.scienceofbetter.org/or_executive_guide.pdf>. Acesso em: 16 set. de 2007.

FEINBERG, B. N.; CHIU, S. S. A method to calculate steady-state distributions of large Markov chains by aggregating states. **Operations Research**, v. 35, n. 2, p. 282-290, 1987.

GALVÃO, R. D.; CHIYOSHI, F. Y.; ESPEJO, L. G. A.; RIVAS, M. P. A. Solução do problema de localização de máxima disponibilidade utilizando o modelo Hipercubo. **Pesquisa Operacional**, v. 23, p. 61-78, 2003. Disponível em: < http://www.scielo.br/pdf/pope/v23n1/a06v23n1.pdf>. Acesso em: 30 jun. 2007.

GALVÃO, R. D.; CHIYOSHI, F. Y.; MORABITO NETO, R. Towards unified formulations and extensions of two classical probabilistic location models. **Computers and Operations Research**, v. 32, p. 15-33, 2005.

GASS, S. In memoriam: Andy Vazsonyi. **OR/MS Today**, v. 31, 2004.

GAU, S.; LARSON, R. C. **Hypercube model with multiple-unit dispatches and police patrol-initiated activities.** Massachusetts Institute of Technology (*MIT*), Operations Research Center Working Paper #OR 188-88, 1988.

GOLDBERG, J.; DIETRICH, R.; CHEN, J.; MITWASI, M.; VALENZUELA, T.; CRISS, E. Validating and applying a model for locating emergency medical vehicles in Tucson, AZ. **European Journal of Operational Research**, v. 49, p. n.. 3, 308-324, 1990.

GOLDBERG, J.; PAZ, L. Locating emergency vehicle bases when service time depends on call location. **Transportation Science**, v. 25, n. 4, p. 264-280, 1991.

GOLDBERG, J.; SZIDAROVSZKY, F. A general model and convergence results for determining vehicle utilization in emergency systems. **Communications in Statistics – Stochastic Models**, v. 7, n. 1, p. 137-160, 1991a.

GOLDBERG, J.; SZIDAROVSZKY, F. Methods for solving nonlinear equations used in evaluating emergency vehicle busy probabilities. **Operations Research**, v. 39, n. 6, p. 903-916, 1991b.

GOLDBERG, J. Operations Research models for the deployment of emergency services vehicles. **EMS Management Journal**, v. 1, n. 1, 2004.

GONÇALVES, M. B. Métodos de pesquisa operacional em serviços emergenciais. In: Simpósio Brasileiro de Pesquisa Operacional, 26., 1994, Florianópolis. **Anais...** Rio de Janeiro: SOBRAPO, 1994, p. 597-601.

GONÇALVES, M. B.; NOVAES, A. G. N.; ALBINO, J. C. C. Modelos para localização de serviços emergenciais em rodovias. In: Simpósio Brasileiro de Pesquisa Operacional, 26., 1994, Florianópolis. **Anais...** Rio de Janeiro: SOBRAPO, 1994, p. 591-596.

GONÇALVES, M. B.; NOVAES, A. G. N.; SCHMITZ, R. Um modelo de otimização para localizar unidades de serviços emergenciais em rodovias. In: Congresso de Pesquisa e Ensino em Transportes, 9., 1995, São Carlos. **Anais...** São Carlos: ANPET,1995, p. 962-972.

GROSS, D.; HARRIS, C. **Fundamentals of queueing theory.** 3 ed. New York: Wiley-Interscience, 1998. 464 p.

HALPERN, J. The accuracy of estimates for the performance criteria in certain emergency service queueing systems, **Transportation Science**, v. 11, n. 3, p. 223–242, 1977.

HAVIV, M. Aggregation/disaggregation methods for computing the stationary distribution of a Markov chain. **SIAM Journal on Numerical Analysis**, v. 24, n. 4, p. 952-966, 1987.

HOGAN, K.; REVELLE, C. S. Concepts and applications of backup coverage. **Management Science**, v. 32, p. 1434-1444, 1986.

IANNONI, A. P. **Otimização da configuração e operação de sistemas médico emergencial em rodovias utilizando o modelo Hipercubo.** 2005. Tese (Doutorado em Engenharia de Produção) – Universidade Federal de São Carlos (UFSCAR), São Carlos, 2005.

IANNONI, A.; MORABITO, R. Modelo de fila hipercubo com múltiplo despacho e backup parcial para análise de sistemas de atendimento médico emergenciais em rodovias, **Pesquisa Operacional**, v. 26, n. 3, p. 493-519, 2006a.

IANNONI, A.; MORABITO, R. Modelo hipercubo integrado a um algoritmo genético para analise de sistemas médicos emergenciais em rodovias. **Gestão & Produção**, v. 13, n. 1, p. 93-104, 2006b.

JARVIS, J. P. **Optimization in stochastic service systems with distinguishable servers.** 1975. Dissertation (Doctor of Philosophy), Massachusets Institute of Technology (MIT), Massachusets, 1975.

JARVIS, J. P. Approximating the equilibrium behavior of multi-server loss systems, **Management Science**, v. 31, n. 2, p. 235–239, 1985.

KIM, D. S.; SMITH, R. L. An exact aggregation/disaggregation algorithm for large scale Markov chains. **Naval Research Logistics**, v. 42, p. 1 115-1128, 1995.

CHELST, K. R. **Implementing the hypercube queueing model in the New Haven department of police services:** a case study in technology transfer. New York: The Rand Corporation, 1975. 73 p. <http://www.rand.org/pubs/reports/2006/R1566.6.pdf>. Acesso em: 30 jun. 2007.

LARSON, R. C. **A Hypercube queueing model for facility location and redistricting in urban emergency services.** New York: Rand Corporation,

1973. xx p. Disponível em: <http://www.rand.org/pubs/reports/2006/R1238.pdf>. Acesso em: 30 jun. 2007.

_____. **Urban emergency service systems:** an iterative procedure for approximating performance characteristics. New York: Rand Corporation, 1974a. 73 p. Disponível em: <http://www.rand.org/pubs/reports/2006/R1493.pdf>. Acesso em: 20 abr. 2007.

_____. A hypercube queuing model for facility location and redistricting in urban emergency services. **Computers and Operations Research**, v. 1, p. 67-95, 1974b.

_____. Approximating the performance of urban emergency service systems. **Operations Research**, v. 23, n. 5, p. 845-868, 1975a.

_____. **Hypercube queueing model:** user's manual. New York: The Rand Corporation, 1975b. 93 p. Disponível em: <http://www.rand.org/pubs/reports/2007/R1688.2.pdf>. Acesso em: 30 jun. 2007. (**********)

_____. **Hypercube queueing model:** program description. New York: The Rand Corporation, 1975c. 54 p. Disponível em: <http://www.rand.org/pubs/reports/2007/R1688.3.pdf>. Acesso em: 30 jun. 2007.

LARSON, R. C.; ODONI, A. R. **Urban operations research.** New Jersey: Prentice-Hall, 1981. 573 p. ISBN 0-13-939447-8.

LARSON, R. C.; MCKNEW, M.A. Police patrol-initiated activities within a systems queuing model. **Management Science**, v. 28, n. 7, p. 759-774, 1982.

LARSON, R. C. Public sector Operations Research: a personal journey. **Operations Research**, v. 50, n. 1, p. 135-145, 2002.

_____. Decision models for emergency response planning. In: Kamien, D. G. (ed.). **The McGraw-Hill homeland security book**. New York, 2004.

LUQUE, L.; CARVALHO, S. V. **Modelo hipercubo de filas:** análise e resultados para o método aproximado de solução. In: CONGRESO LATINO-IBEROAMERICANO DE INVESTIGACIÓN OPERATIVA, 13., 2006, Montevideo. **Proceedings…** Montevideo: Universidad de la República, 2006.

MORABITO, R.; CHIYOSHI, F.; GALVÃO, R. **Non-homogeneous servers in emergency medical systems:** practical applications using the hypercube queuing model. **Socio-Economic Planning Sciences**, doi: 10/1016/j.seps.2007.04.002.

MATSUMOTO, M.; NISHIMURA, T. Mersenne Twister: a 623-dimensionally equidistributed uniform pseudorandom number generator. **ACM Transactions on Modeling and Computer Simulation**, v. 8, n. 1, p. 3-30, 1998.

MENDONÇA, F. C. **Aplicação do modelo hipercubo, baseado em teoria de filas, para análise de um sistema médico-emergencial em rodovia.** 1999. Tese (Mestrado em Engenharia de Produção) - Universidade Federal de São Carlos (UFSCAR), São Carlos, 1999.

MENDONÇA, F. C.; MORABITO, R. Aplicação do modelo hipercubo para análise de um sistema médico-emergencial em rodovia. **Gestão & Produção**, v. 7, n. 1, p. 73–91, 2000.

MENDONCA, F. C.; MORABITO, R. Analysing emergency medical service ambulance deployment on a brazilian highway using the Hypercube Model. **Journal of the Operational Research Society**, v. 52, p. 261-270, 2001.

MEYER, C. D. Stochastic complementations, uncoupling Markov chains, and the theory of nearly reducible systems. **SIAM Review**, v. 32, n. 2, p. 240-272, 1989.

MUSSEN, S. **Applied probability and queues.** Chichester, England: John Wiley, 1987. 318 p.

OLIVEIRA, L. K. **Uma aplicação do modelo hipercubo de filas para avaliação do centro de emergências da polícia militar de Santa Catarina, Florianópolis.** 2003. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Catarina (UFSC), Santa Catarina, 2003.

O'NEIL, J.; SZYLD, D. B. A block ordering method for sparse matrices. **SIAM Journal on Scientific and Statistical Computing**, v. 11, n. 5, p. 811-823, 1990.

POLLOCK, S. M.; BIRGE, J. R. **Parallel iteration for multiple servers.** Michigan: Department of Industrial and Operations Engineering, University of Michigan, 1983.

POLLOCK, S. M.; MALTZ, M. D. Operations Research in the public sector: an introduction and a brief history. In: POLLOCK, S. M.; ROTHKOPF, M. H.; BARNETT, A. (ed.). **Operations Research and the public sector**. Amsterdam: Elsevier, 1994.

REVELLE, C. S.; HOGAN, K. The maximum avaliability location problem. **Transportation Science**, v. 23, p. 192-200, 1989.

RIAÑO, G.; GÓEZ, J. jMarkov: an object oriented framework for modeling and analyzing Markov Chains and QBDs. In: SMCtools'06, 2006, Italy. **Proceedings…** Pisa, Italy: ACM Press, 2006.

SAATY, T. L. **Elements of queuing theory:** with applications. New York: Mcgraw-Hill, 1961. 423p.

SACKS, S. R.; GRIEF. S., Orlando police department uses OR/MS methodology, new software to design patrol districts. **OR/MS Today**, p. 30-32, 1994.

SACKS, S. R. A tool for evaluation of police patrol patterns. In: ANNUAL ESRI INTERNATIONAL USER CONFERENCE, 23., 2003. San Diego. **Proceedings…** San Diego, California: ESRI, 2003.

SAYDAM, C.; REPEDE, J.; BURWELL, T. Accurate estimation of expected coverage: a comparative study. **Socio-Economic Planning Sciences**, v. 28, n. 2, p. 113-120, 1994.

SAYDAM, C., AYTUG, H. Accurate estimation of expected coverage: revisited. **Socio-Economic Planning Sciences**, v. 37, p. 69-80, 2003.

SEMAL, P. Refinable bounds for large Markov chains. **IEEE Transactions on Computers**, v. 44, n. 10, p. 1216-1222, 1995.

SCHWEITZER, P. J. An iterative aggregation-disaggregation algorithm for solving linear equations. **Applied Mathematics and Computation**, v. 18, p. 313-353, 1986.

SEZER, M. E.; ŠILJAK, D. D. Nested ε-decompositions and clustering of complex systems. **Automatica**, v. 22, n. 3, p. 321-331, 1986.

SHESKIN, T. J. A Markov chain partitioning algorithm for computing steady state probabilities. **Operations Research**, v. 33, n. 1, p. 228-235, 1985.

SIMON, H. A.; ANDO, A. Aggregation of variables in dynamic systems. **Econometrica**, v. 29, n. 2, p. 111-138, 1961.

SOUZA E SILVA, E. A.; MUNTZ, R. R. Métodos computacionais de solução de cadeias de Markov: aplicações a sistemas de computação e comunicação. In: ESCOLA DE COMPUTAÇÃO, 8., 1992, Gramado. UFRGS,1992.

STEWART, G. W.; STEWART, W. J.; MCALLISTER, D. F. **A two-stage iteration for solving nearly uncoupled Markov chains.** Maryland: Department of Computer Science, University of Maryland, College Park, MD, 1984. Technical Report CSC TR138. Disponível em: <http://citeseer.ist.psu.edu/stewart91twostage.html>. Acesso em: 30 jun. 2007.

STEWART, W. J.; WU, W. Numerical experiments with iteration and aggregation for Markov chains. **ORSA J. Comput.**, v. 4, p. 336-350, 1992. Disponível em: http://citeseer.ist.psu.edu/stewart96numerical.html. Acesso em: 30 jun. 2007.

SUMITA, U.; RIEDERS, M. A new algorithm for computing the ergodic probability vector for large Markov chains: replacement process approach. **Probability in the Engineering and Informational Sciences**, v. 4, p. 89-116, 1990.

SWERSEY, A. J. The deployment of police, fire, and emergency medical units. In: POLLOCK, S. M.; ROTHKOPF, M. H.; BARNETT, A. (eds.). **Handbooks in OR & MS.** [S.l]: Elsevier, 1994. cap.6 , p.1-22. ISBN 0444892044.

TAKEDA, R. A. **Uma contribuição para avaliar o desempenho de sistemas de transporte emergencial de saúde.** 2000. Tese (Doutorado em Engenharia de Transportes) – Universidade de São Paulo (USP), São Carlos, 2000.

TAKEDA, R. A.; WIDMER, J. A.; MORABITO, R. Uma proposta alternativa para avaliação do desempenho de sistemas de transporte emergencial de saúde brasileiros. **Transportes**, v. 9, n. 2, p. 9-27, 2000.

TAKEDA, R.; WIDMER, J. A.; MORABITO, R., Aplicação do modelo hipercubo de filas para avaliar a descentralização de ambulâncias em um sistema urbano de atendimento médico de urgência. **Pesquisa Operacional**, v. 24, n. 1, p. 39-72, 2004.

TAKEDA, R.; WIDMER, J.; MORABITO, R. Analysis of ambulance decentralization of an urban emergency medical service using the hypercube queuing model. **Computers and Operations Research**, v. 34, n. 3, p. 727-741, 2007.

TANAKA, K.; SHIOYAMA, T. A new aggregation-disaggregation algorithm. **European Journal of Operational Research**, v. 83, p. 655-669, 1995.

TIJMS, H. C. **Stochastic models: an algorithmic approach.** New York: John Wiley, 1994.

VARGA, A., The OMNET++ discrete event simulation system. In: European SIMULATION MULTICONFERENCE, 15., 2001, Praga, República Tcheca, **Proceedings…** Praga: [s.n], p. 319-324.

WHITEHOUSE, G. E.; WECHSLER, B. L. **Operations research:** a survey. New York: John Wiley & Sons, 1976.

APÊNDICE A

UMA BIBLIOTECA ORIENTADA A OBJETOS PARA O MODELO HIPERCUBO DE FILAS E SUAS EXTENSÕES

Neste apêndice, a modelagem de uma biblioteca orientada a objetos para o modelo Hipercubo de Filas e suas extensões, e alguns detalhes da implementação desta biblioteca em C++ e Java são apresentados. Esta biblioteca foi utilizada para a solução dos problemas testes cujos resultados foram apresentados neste trabalho.

A.1 Considerações iniciais

A carência de soluções de software extensíveis e de domínio público do modelo Hipercubo de Filas dificulta o desenvolvimento de pesquisas com o modelo e suas extensões, e obriga os interessados a desenvolverem soluções próprias.

A primeira implementação do modelo, escrita na linguagem PL/I, foi feita por Larson (1975b,c). Esta implementação não é apropriada para a execução de grandes conjuntos de casos de teste com o modelo, pois envolve a configuração, em arquivos, de uma grande quantidade de informações e exibe as medidas de desempenho em formatos de difícil tratamento.

Em Gau e Larson (1988, p.27) foi desenvolvida uma solução em Fortran para a implementação de uma extensão do modelo proposta pelos autores. Em Oliveira (2003, p.63), foi utilizado o MS-Excel em conjunto com a extensão Solver para que o modelo fosse resolvido e as medidas de interesse fossem calculadas. Em Mendonça (1999, p.67), foi utilizada a ferramenta Maple e em Iannoni (2005, p.142) e Takeda (2000, p.105), um programa implementado em Pascal.

Outros exemplos de solução de software do modelo são o “Desktop Hypercube” (SACKS, 2003), uma extensão para o Sistema de Informação

Geográfica - SIG ArcView, que permite a modelagem e avaliação da performance de sistemas através do Hipercubo, mas que apresenta restrições quanto ao uso, reservado especialmente para departamentos de polícia, mediante solicitação formal aos desenvolvedores, e a ferramenta OMNet++ (VARGA, 2001), que implementa a representação do espaço de estados de um hipercubo, mas que é baseada em simulação e não calcula diretamente as medidas de desempenho do sistema modelado.

Neste contexto, o desenvolvimento de uma solução para o modelo Hipercubo e suas extensões mostrou-se necessário para a execução dos testes apresentados neste trabalho. Espera-se que a solução apresentada possa ser utilizada por outros pesquisadores em trabalhos futuros com o modelo.

A seguir, são apresentados detalhes sobre a modelagem conceitual da solução desenvolvida e sobre a implementação desta solução em Java e C++.

A.2 Modelagem de uma biblioteca orientada a objetos

A solução para o modelo Hipercubo de Filas e suas extensões foi desenvolvida na forma de uma biblioteca orientada a objetos. O paradigma orientado a objetos foi escolhido principalmente por apresentar características que facilitam o reuso, a extensão e a gestão da complexidade. Um software orientado a objetos é organizado a partir de uma coleção de objetos distintos, que incorporam estruturas de dados e comportamento (BLAHA; RUMBAUGH, 2006).

Esta coleção de objetos é classificada e especificada por um conjunto de classes, abstrações do problema que definem quais dados e qual comportamento os objetos por elas especificados devem apresentar. As classes, por sua vez, podem ter seu comportamento especificado por um conjunto de interfaces e podem ser organizadas conceitualmente e estruturalmente em pacotes.

A modelagem da biblioteca foi realizada através de diagramas de pacotes (pacotes e suas classes) e classes (classes, interfaces e seus relacionamentos) da Linguagem de Modelagem Unificada (*Unified Modeling Language* - UML), um padrão para modelagem de sistemas orientados a objetos (BOOCH et al., 2000).

A.2.1 Requisitos da biblioteca orientada a objetos

Foram definidos como requisitos funcionais da biblioteca: permitir a solução de um modelo Hipercubo de Filas qualquer, através dos métodos iterativos de Gauss-Seidel e Gauss-Jacobi, e calcular todas as medidas de desempenho definidas por Larson e Odoni (1981). Como requisito não-funcional, foi definida a possibilidade de extensão da biblioteca para incorporar extensões propostas na literatura, como: co-localização e múltiplo despacho de servidores, inclusão de atividades não relacionadas às solicitações de serviço, entre outras, sem que sejam necessárias alterações significativas em sua estrutura básica.

A.2.2 Estrutura da biblioteca

A biblioteca foi organizada em 9 pacotes (Figura A.1). O pacote *hypercube* é o principal e agrupa além de 8 sub-pacotes, a classe central para definição do modelo. Os sub-pacotes são: *customer* (organiza classes relacionadas às solicitações de serviço), *server* (organiza classes relacionadas às realizações de serviço), *dispatch* (organiza classes relacionadas às preferências de despacho dos servidores aos clientes), *solution*, *solution.approximate*, *solution.exact* e *solution.peformancemeasures* (organiza classes relacionadas aos métodos de solução do modelo) e *extensions* (organiza classes que estendem o modelo Hipercubo original) .

A classe central do modelo, *Hypercube* (Figura A.2), permite a definição de um modelo Hipercubo qualquer. Esta classe tem como atributo o tamanho máximo da fila do sistema e se relaciona com as interfaces *Customer* e *Server* que

definem o comportamento comum às classes que implementam solicitações e realizações de serviço, respectivamente.

Para que os clientes e servidores de um modelo possam ser recuperados, foram definidas as classes *CustomerIterator* e *ServerIterator*, respectivamente.

Estas interfaces foram definidas para permitir que extensões do modelo referentes às solicitações e realizações de serviço, como, por exemplo, a dependência das taxas de serviço dos servidores em relação ao tipo de cliente que originou a chamada (HALPERN, 1977; JARVIS, 1975; JARVIS, 1985), pudessem ser implementadas.

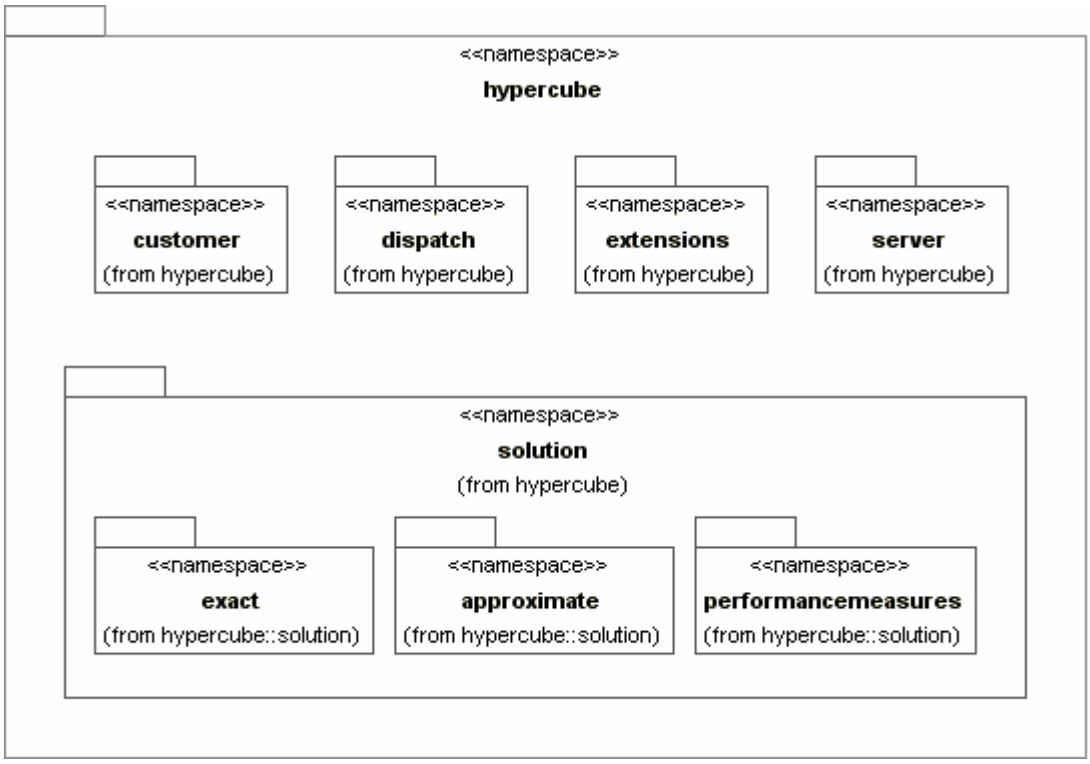


Figura A.1 – Pacotes da biblioteca.

A biblioteca prevê duas classes que implementam estas interfaces de acordo com o modelo original, *GeographicalAtom* e *DefaultServer*. Os átomos geográficos (*GeographicalAtom*) e servidores (*DefaultServer*) possuem uma

identificação única e um valor de taxa de chegada e serviço, respectivamente (hipóteses 1,2 e 4).

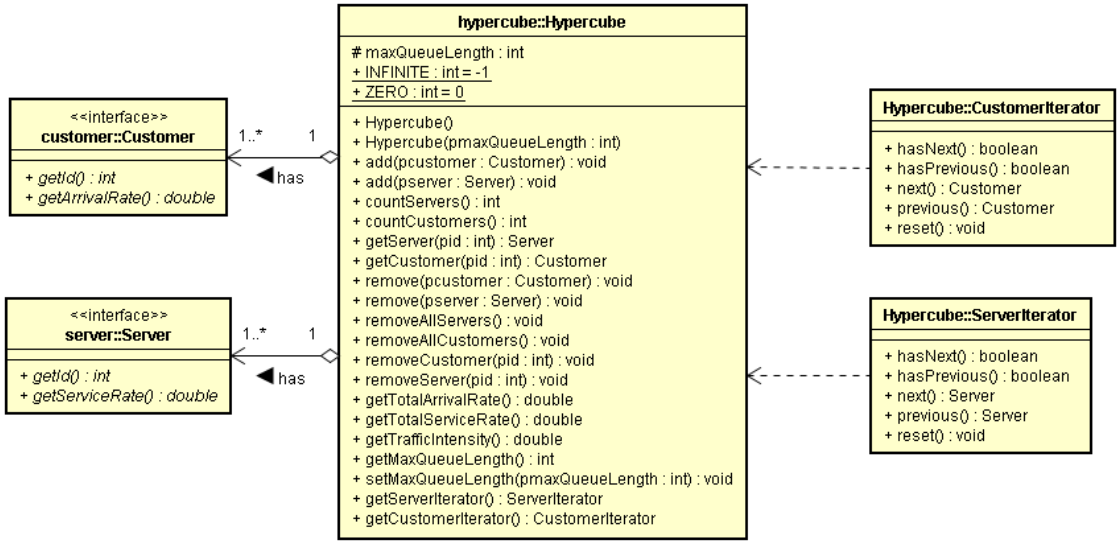


Figura A.2 – Classe central do modelo e principais classes relacionadas.

Para a modelagem do tempo de viagem entre átomos geográficos (hipótese 3) e da localização dos servidores (hipótese 5) foram definidas uma auto-associação na classe *GeographicalAtom* e uma associação entre essa classe e a classe *DefaultServer*. Para garantir localizações válidas para os servidores, foi definido o método *validateLocations()* na classe *DefaultServer* que valida as localizações verificando se a soma delas é igual a unidade.

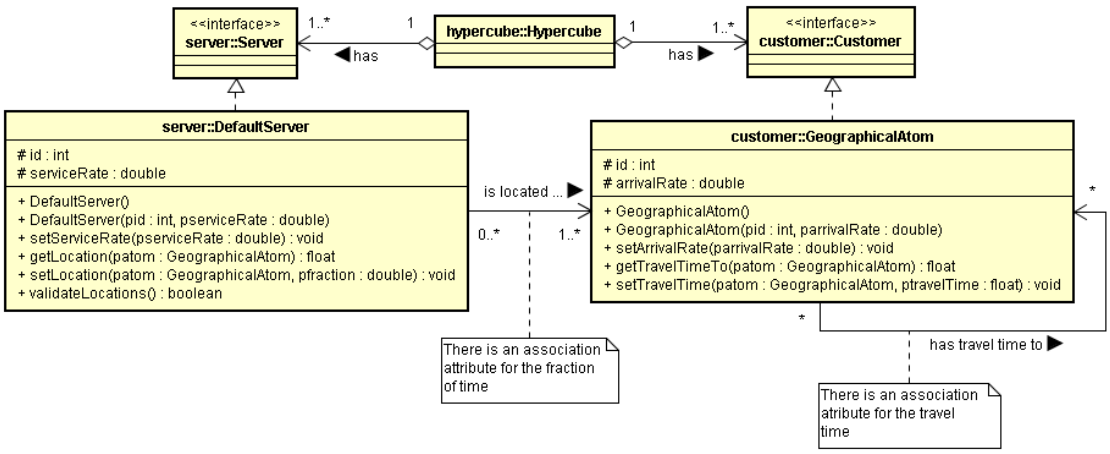


Figura A.3 – Classes relacionadas a solicitação e realização de serviço de acordo com o modelo Hipercubo conforme originalmente definido.

O despacho de servidores (hipóteses 6 e 7) para o atendimento a solicitações de serviço é representado pela interface *Preference*. Esta interface define o comportamento das classes que implementam preferências de despacho. A classe *DefaultPreference* representa o despacho de acordo com o modelo original, isto é, para uma chamada, apenas um servidor pode ser despachado (Figura A.4). As preferências de um determinado cliente podem ser recuperadas através da classe *PreferenceIterator*.

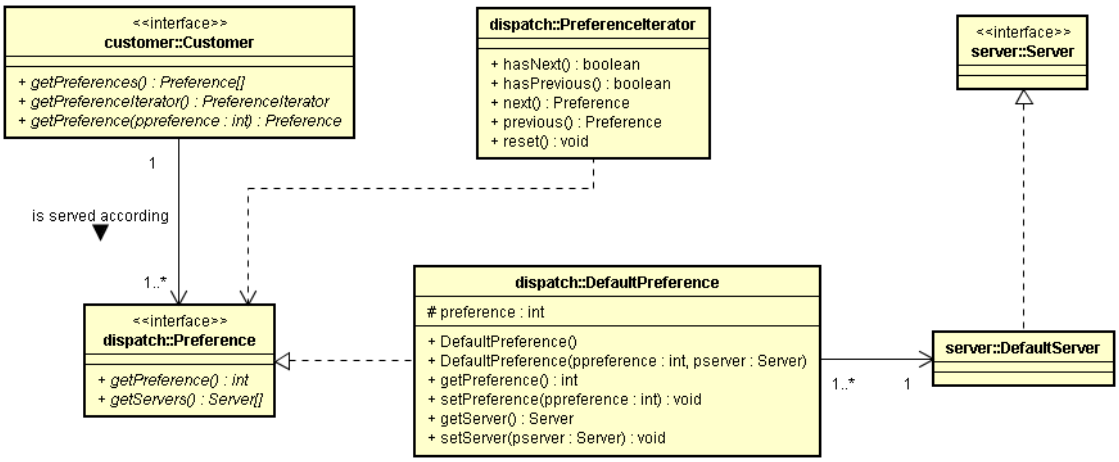


Figura A.4 – Classes relacionadas ao despacho dos servidores para atendimento aos clientes.

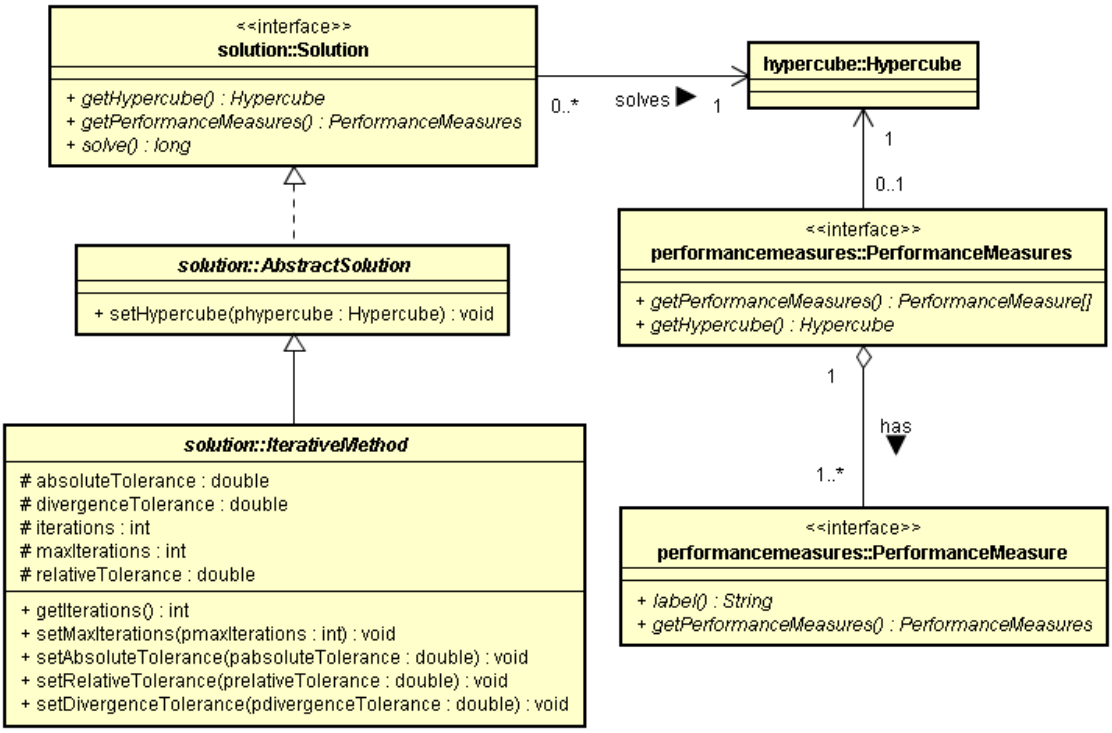


Figura A.5 – Classes relacionadas aos métodos de solução do modelo.

Para a solução do modelo, foram definidas a interface *Solution*, que especifica o comportamento comum a métodos de solução, e a classe abstrata *AbstractSolution*, que implementa parte deste comportamento. Esta interface apresenta métodos para a solução do modelo e para a recuperação das medidas de desempenho associadas a essa solução, representadas por classes que implementam as interfaces *PerformanceMeasures* (que representa todo o conjunto de medidas de desempenho do modelo) e *PerformanceMeasure* (que representa um tipo específico de medida de desempenho) (Figura A.5).

Para o modelo original, foram previstas as classes *BasicPerformanceMeasures*, *DispatchPerformanceMeasures* e *TravelTimePerformanceMeasures* que calculam as medidas gerais do sistema e medidas relacionadas à frações de despacho e tempos de viagem,

respectivamente. Estas medidas são todas agrupadas na classe *HypercubePerformanceMeasures*.

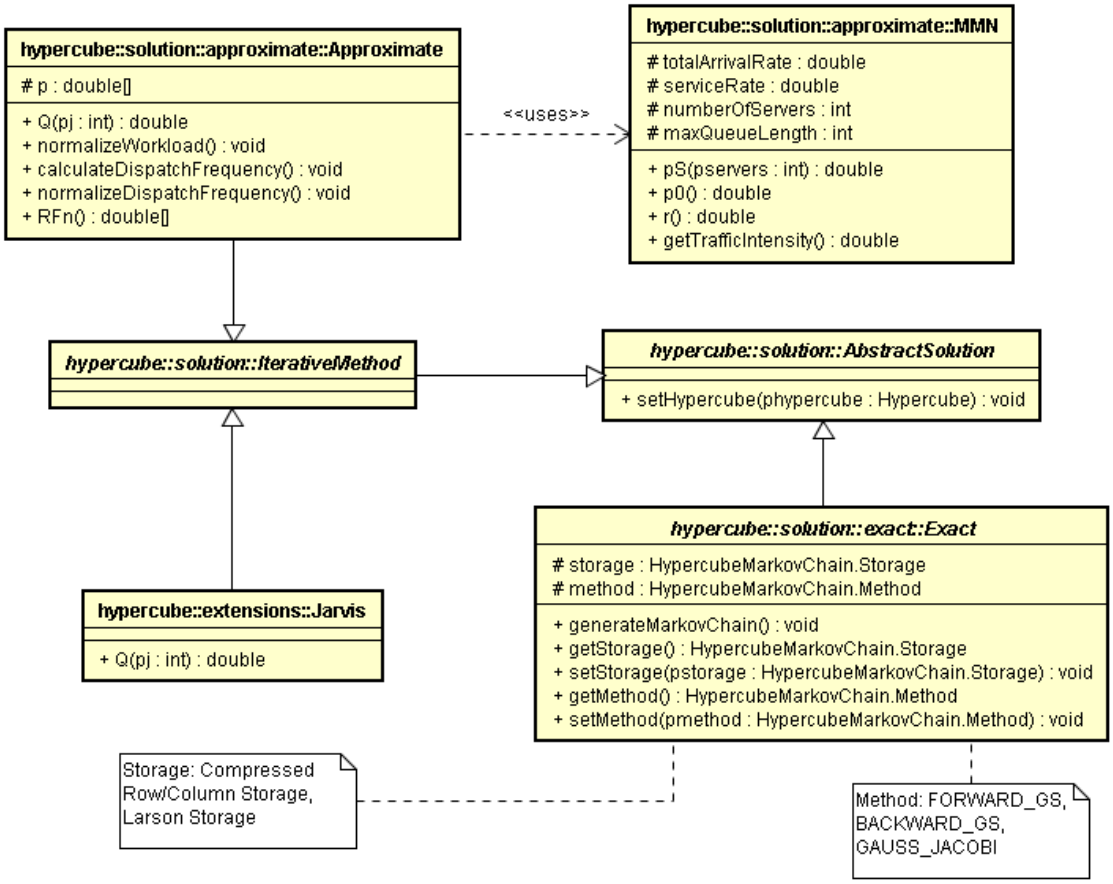


Figura A.6 – Classes relacionadas a solicitação e realização de serviço de acordo com o modelo Hipercubo conforme originalmente definido.

Foram previstos na biblioteca a execução dos métodos exato e aproximado de Larson e de Jarvis, representados pelas classes *Exact* e *Approximate* e *Jarvis*, respectivamente, que implementam a interface *Solution*. Como alguns destes métodos de solução possuem similaridades por serem iterativos, a classe *IterativeMethod* foi definida para implementar este comportamento comum. Como os métodos aproximados são baseados em um sistema de filas M/M/N, uma classe foi criada para representar tal sistema (*MMN*). Na classe *Exact* é possível definir o método de solução que será utilizado para o sistema de equações lineares (Forward e Backward Gauss-Seidel e Gauss-Jacobi) e na

classe *Aproximate* é possível definir o método que será utilizado para o cálculo das frequências de despacho dos servidores aos átomos geográficos (LARSON, 1975a; LARSON; ODONI, 1981).

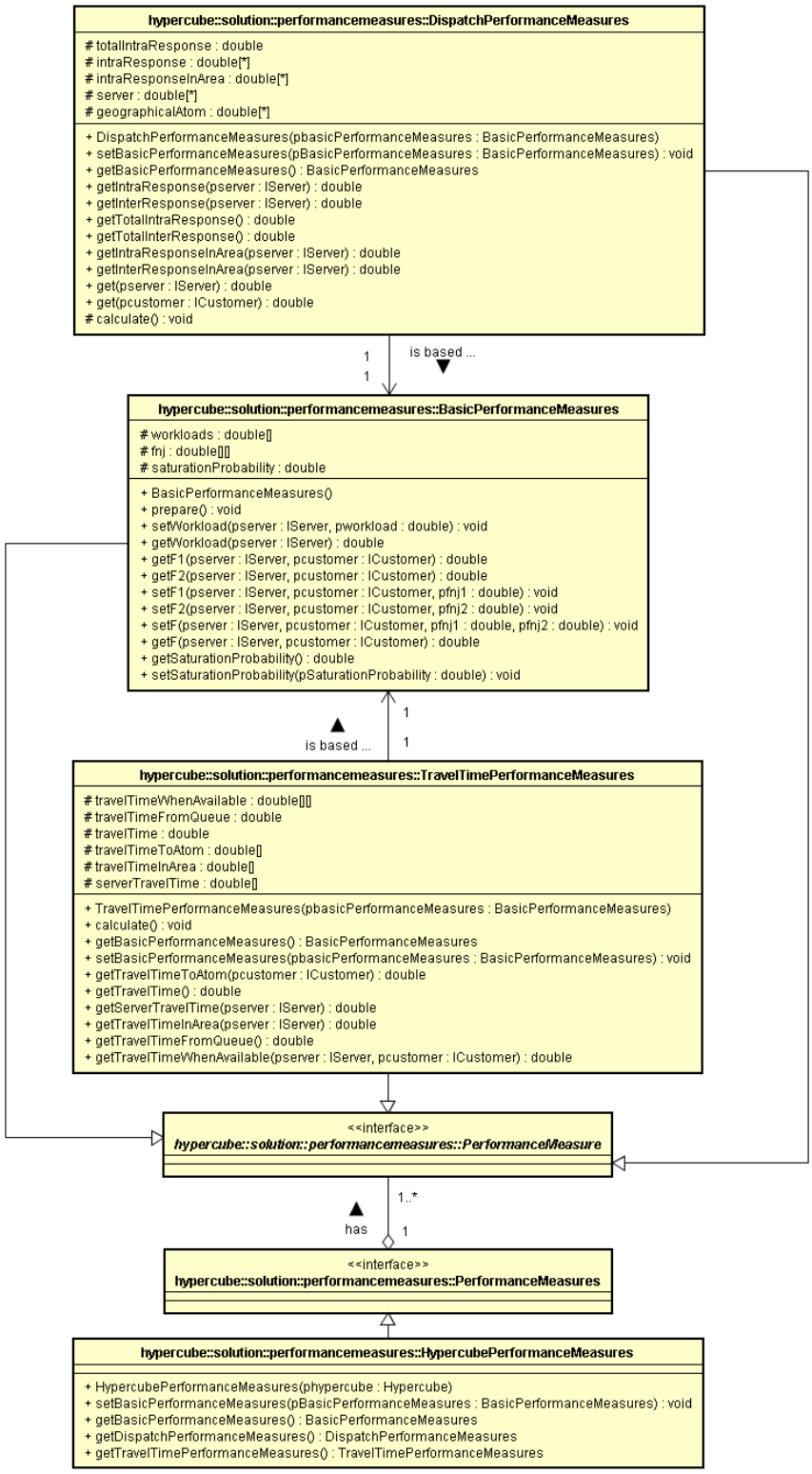


Figura A.7 – Classes relacionadas às medidas de desempenho.

As hipóteses 8 e 9 não foram consideradas na biblioteca porque estão relacionadas aos dados de entrada do modelo. Por motivos visuais, nos diagramas anteriores, foram representadas apenas as principais classes, atributos e métodos da biblioteca.

A.3 Implementação da biblioteca

A biblioteca descrita anteriormente foi implementada em Java e C++. Para a implementação do método exato foi necessária a modelagem e o desenvolvimento de outra biblioteca para a modelagem e solução de cadeias de Markov, já que as bibliotecas existentes disponíveis, como JMarkov () para o caso do Java e Modelos Estocásticos – MODESTO (ANDRADE et al., 1997) para o caso do C++, ou apresentam limitações quanto ao uso de memória ou a performance.

A seguir é apresentado um exemplo de uso da biblioteca implementada em Java para a solução do sistema exemplo apresentado no Capítulo 3.

Neste sistema existem 5 átomos geográficos e três servidores que possuem taxas de chegada e serviço especificadas nas Tabelas 3.1 e 3.2, respectivamente, que podem ser definidos a partir do seguinte trecho de código:

```
// cria o modelo Hipercubo.
Hypercube hipercubo = new Hypercube();
hipercubo.setMaxQueueLength(Hypercube.ZERO);

// define os átomos geográficos.
GeographicalAtom atomo1 = new GeographicalAtom(1, 0.5);
GeographicalAtom atomo2 = new GeographicalAtom(2, 0.4);
GeographicalAtom atomo3 = new GeographicalAtom(3, 0.2);
GeographicalAtom atomo4 = new GeographicalAtom(4, 1.0);
GeographicalAtom atomo5 = new GeographicalAtom(5, 2.0);
```

```
// adiciona os átomos geográficos ao modelo.
hipercubo.add(atomo1);
hipercubo.add(atomo2);
hipercubo.add(atomo3);
hipercubo.add(atomo4);
hipercubo.add(atomo5);

// define os servidores.
DefaultServer servidor1 = new DefaultServer(1, 1.0);
DefaultServer servidor2 = new DefaultServer(2, 2.0);
DefaultServer servidor3 = new DefaultServer(3, 2.0);

// adiciona os servidores ao modelo.
hipercubo.add(servidor1);
hipercubo.add(servidor2);
hipercubo.add(servidor3);
```

A tabela de preferências de despacho pode ser definida através do seguinte código:

```
// define as preferências de despacho.
atomo1.addPreference(new Preference(1, servidor2));
atomo1.addPreference(new Preference(2, servidor1));
atomo1.addPreference(new Preference(3, servidor3));

atomo2.addPreference(new Preference(1, servidor2));
atomo2.addPreference(new Preference(2, servidor1));
atomo2.addPreference(new Preference(3, servidor3));

atomo3.addPreference(new Preference(1, servidor1));
atomo3.addPreference(new Preference(2, servidor2));
```

```
atomo3.addPreference(new Preference(3, servidor3));

atomo4.addPreference(new Preference(1, servidor3));
atomo4.addPreference(new Preference(2, servidor2));
atomo4.addPreference(new Preference(3, servidor1));

atomo5.addPreference(new Preference(1, servidor1));
atomo5.addPreference(new Preference(2, servidor2));
atomo5.addPreference(new Preference(3, servidor3));.
```

Por sua vez, as distâncias entre os átomos geográficos e a localização dos servidores podem ser definidos através do seguinte código:

```
// define a localização dos servidores.
servidor1.setLocation(atomo1, 0.4);
servidor1.setLocation(atomo2, 0.3);
servidor1.setLocation(atomo3, 0.3);

servidor2.setLocation(atomo4, 0.5);
servidor2.setLocation(atomo5, 0.5);

servidor3.setLocation(atomo4, 1.0);

// define as distâncias entre os átomos geográficos.
atomo1.setTravelTimeTo(atomo2, 2);
atomo1.setTravelTimeTo(atomo3, 3);
atomo1.setTravelTimeTo(atomo4, 5);
atomo1.setTravelTimeTo(atomo5, 6);

atomo2.setTravelTimeTo(atomo1, 2);
atomo2.setTravelTimeTo(atomo3, 3);
atomo2.setTravelTimeTo(atomo4, 4);
```

```
atomo2.setTravelTimeTo(atomo5, 5);

atomo3.setTravelTimeTo(atomo1, 3);
atomo3.setTravelTimeTo(atomo2, 3);
atomo3.setTravelTimeTo(atomo4, 6);
atomo3.setTravelTimeTo(atomo5, 8);

atomo4.setTravelTimeTo(atomo1, 5);
atomo4.setTravelTimeTo(atomo2, 4);
atomo4.setTravelTimeTo(atomo3, 6);
atomo4.setTravelTimeTo(atomo5, 9);

atomo5.setTravelTimeTo(atomo1, 6);
atomo5.setTravelTimeTo(atomo2, 5);
atomo5.setTravelTimeTo(atomo3, 8);
atomo5.setTravelTimeTo(atomo4, 9) ;
```

Por fim, o modelo pode ser resolvido, por exemplo, através do método exato (utilizando o método de Gauss-Seidel) e as medidas de desempenho podem ser calculadas a partir do código seguinte:

```
// soluciona o modelo através do método exato (método de Gauss-Seidel).
Exact exato = new Exact();
exato.setMethod(HypercubeMarkovChain.Method.FORWARD_GS);
exato.setHypercube(hipercubo);
exato.solve();

// calcula as medidas de desempenho exatas.
HypercubePerformanceMeasures medidasDesempenho =
HypercubePerformanceMeasures) exato.getPerformanceMeasures();
BasicPerformanceMeasures mdBasicas =
medidasDesempenho.getBasicPerformanceMeasures();
```

```
DispatchPerformanceMeasures mdDespacho =
medidasDesempenho.getDispatchPerformanceMeasures();
TravelTimePerformanceMeasures mdTemposViagem =
medidasDesempenho.getTravelTimePerformanceMeasures();
```

Se, ao invés do método exato, o modelo fosse resolvido pelo método de Larson ou de Jarvis, o seguinte código poderia ser utilizado:

```
// soluciona o modelo através do método aproximado de Larson.
Approximate larson = new Approximate();
larson.setHypercube(hipercubo);
larson.solve();

// soluciona o modelo através do método aproximado de Larson.
JarvisApproximation jarvis = new JarvisApproximation();
jarvis.setHypercube(hipercubo);
jarvis.solve();
```

ÍNDICE POR ASSUNTO

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programa de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. São aceitos tanto programas fonte quanto executáveis.