

Métodos Computacionais para Modelos Markovianos com Recompensa



UFRJ

Tese submetida para a obtenção do título de

Doutor em Ciências em Engenharia de Sistemas e Computação

ao Programa de Pós-Graduação de Engenharia de Sistemas e Computação

da COPPE/UFRJ

por

Ana Paula Couto da Silva

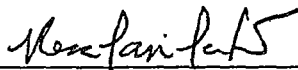
Outubro 2006

MÉTODOS COMPUTACIONAIS PARA MODELOS MARKOVIANOS COM
RECOMPENSA


Ana Paula Couto da Silva

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO
DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR
EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

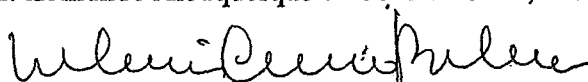
Aprovada por:




Prof.ª Rosa Maria Meri Leão, Dr.



Prof. Edmundo Albuquerque de Souza e Silva, Ph.D.



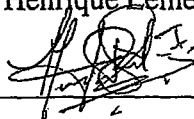
Prof. Valmir Carneiro Barbosa, Ph.D.



Prof.ª Morganna Carmem Diniz, Dr.



Prof. Paulo Henrique Lemelle Fernandes, Dr.



Prof. Gerardo Rubino, Dr.

RIO DE JANEIRO, RJ - BRASIL

OUTUBRO DE 2006

SILVA, ANA PAULA COUTO DA

Métodos Computacionais para Modelos
Markovianos com Recompensa [Rio de Janeiro]
2006

xx, 196 p. 29,7 cm (COPPE/UFRJ, D.Sc.,
Engenharia de Sistemas e Computação, 2006)

Tese - Universidade Federal do Rio de
Janeiro, COPPE

1. Modelos Markovianos com Recompensas
2. Modelos de Fluido
3. Modelos com Grande Espaço de Estados
4. Modelos de Markov Oculto
5. Avaliação Subjetiva de QoS

I. COPPE/UFRJ II. Título (Série)

A Deus em primeiro lugar.

Aos meus familiares e amigos com muito carinho.

Agradecimentos

São inúmeras as pessoas que contribuíram direta ou indiretamente durante esta longa caminhada. Serei eternamente grata a todas. Muito Obrigada!

Primeiramente a Deus, que por sua imensa graça e misericórdia nunca nos desampara. “*Sem Mim (Jesus), nada podeis fazer. (João 15:5(b))*”.

A minha família, pelo amor e apoio incondicionais. Tudo que sou hoje eu devo aos meus pais, Ana Maria e Paulo Araújo, que sempre me incentivaram na realização dos meus sonhos, principalmente na minha vida profissional. A minha irmã, Ana Cristina e ao meu cunhado, Fábio. Estas são as pessoas que mais me incentivaram e me apoiaram desde o momento que eu optei pela carreira acadêmica. A minha vó, minhas tias, tios, primas e primos. Não tenho palavras para expressar a minha gratidão.

Aos meus amigos, sendo que cada um teve uma participação especial nesta caminhada. Primeiramente, aos amigos do LAND. Ao Flávio, pelo companheirismo e ajuda em todos os momentos, desde o primeiro *ls* no Linux. Ao Guto, a Fabianne e ao GD pelas diversas *caronas* do Fundão até o metrô, sempre acompanhadas de uma ótima sessão de *conversa jogada fora*. Ao Fernando, pelas discussões sobre as dúvidas que *atormentavam* a minha cabeça, principalmente as que se referiam ao problema da recompensa em diversos estados. Ao Edson, que foi o braço direito nos testes de avaliação subjetiva de qualidade realizadas aqui no LAND, bem como no seu interesse constante no meu trabalho de tese. Não posso deixar de citar também Daniel Sadoc e Kleber, pelas diversas discussões em torno das disciplinas que cursamos juntos. Obrigada também ao Allyson, Bernardo, Carolzinha, Bene, Hugo e David, Isabela, Diana e ao Ed, pela amizade e atenção.

A Carol, a *mãezona* de todos do laboratório! Obrigada pelas palavras de carinho,

incentivo e apoio nos momentos mais difíceis. E não poderia deixar de agradecer pelos inúmeros *cafezinhos* que me mantiveram acordada!

A Regiane, Guilherme, Sandro, Estevan, Max e Rodrigo, meus amigos da época da graduação, na UFJF. Estes foram os primeiros a me incentivar quando eu decidi ingressar em um curso de mestrado e doutorado. Queria agradecer especialmente a Regiane e ao Gui, meus melhores amigos, que estiveram junto comigo durante toda esta caminhada, me apoiando nos momentos de altos e baixos, típicos desta jornada.

Ao Bruno, que teve um papel muito especial na minha vida e no meu coração nos primeiros anos do doutorado. A sua mãe, Neyde, e as suas irmãs, Flávia e a Aline, que me acolheram como um membro da família, principalmente nestes últimos meses que antecederam a defesa desta tese. Não poderia deixar de falar do Lucas, que com as descobertas típicas de uma criança de dois aninhos alegraram o final de cada dia árduo de trabalho.

A duas pessoas extremamente especiais que entraram na minha vida de uma maneira totalmente inesperada. *A Martín qui a été une personne spéciale pendant mon séjour en France. Il m'a beaucoup aidée, tant pendant mes moments de tristesse, comme pendant ceux de joie. Merci pour les agréables journées de dimanche, avec un film, du gâteau, du chocolat e du café. Je ne t'oublierai jamais.* E a Aline, que se transformou em uma grande amiga, me aceitando como *co-locataire* em seu apartamento em Rennes. Muito obrigada por me escutar e me apoiar nos momentos de saudade e desânimo. Uma pena que você tenha mudado para Rennes somente nos últimos três meses da minha estadia na cidade. No entanto, este curto espaço de tempo foi o suficiente para que hoje eu te considere uma grande amiga!

Je voudrais bien remercier au professeur Gerardo Rubino, qui m'a accueilli comme une vraie élève a lui pendant mon séjour en France. Merci pour l'attention et pour le travaille que nous avons fait ensemble. Vous m'avez beaucoup appris. Merci pour tout.

Aos meus orientadores, Rosa Leão e Edmundo de Souza e Silva. Muito obrigada por acreditarem no meu potencial, mesmo quando eu mesma não acreditava. Obrigada por me darem a oportunidade de fazer parte deste grande grupo de pesquisa, onde as

pessoas compartilham do conhecimento acumulado. Obrigada por me aceitarem como aluna, apesar de ter vindo de *terras estrangeiras* :) e desconhecidas até então. Tudo que eu sei hoje sobre o que é ser uma pesquisadora me foi ensinado por vocês. Obrigada pelos momentos de troca de idéias, que não foram poucos. Serei sempre uma aluna de vocês!

A todos os professores presentes nesta banca de tese. Aos professores que lecionaram as matérias que eu cursei durante o mestrado e o doutorado. Muito obrigada.

Aos funcionários do Programa de Engenharia de Sistema e Computação. Pesquisa também é composta por diversos problemas burocráticos, que são resolvidos pelas secretárias e auxiliares do corpo de funcionários.

Em especial, ao governo brasileiro, por me dar a oportunidade de cursar o mestrado e o doutorado em uma das maiores instituições de pesquisa deste país, recebendo bolsas do CNPq e da CAPES. Tenho a plena consciência de que, a partir da defesa da minha tese, farei parte de uma pequena parte privilegiada desta nação. Espero poder retribuir todo este investimento, despertando o interesse de futuros alunos para a pesquisa e a docência.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

MÉTODOS COMPUTACIONAIS PARA MODELOS MARKOVIANOS COM RECOMPENSA

Ana Paula Couto da Silva

Outubro/2006

Orientadores: Rosa Maria Meri Leão

Edmundo Albuquerque de Souza e Silva

Programa: Engenharia de Sistemas e Computação

A modelagem de sistemas de computação e comunicação é essencial para o entendimento e a predição do comportamento de sistemas. Através dos modelos, diversas perguntas sobre o comportamento destes sistemas podem ser respondidas, tanto relacionadas a confiabilidade quanto ao desempenho.

O objetivo principal desta tese é propor métodos computacionais para a solução de processos estocásticos Markovianos com recompensa. Iremos abordar o problema da modelagem de filas de fluido, através da utilização de processos do tipo *Quasi-Birth-Death*. Adicionalmente, propomos uma equação de fluido onde a cada estado do modelo é atribuído um conjunto de recompensas de taxa que ocorrem segundo uma distribuição de probabilidades.

Como terceira contribuição desta tese apresentaremos um método aproximado, com controle de erro, para o cálculo do valor esperado da recompensa acumulada para sistemas com estados absorventes. O método proposto é atrativo para o caso onde a cardinalidade do espaço de estados do modelo é da ordem de milhões.

Finalizando, apresentaremos uma metodologia para avaliação da Qualidade de Serviço percebida por um usuário de uma aplicação interativa de voz utilizando a *Internet*, definida como *Pseudo-Subjective Quality Assessment*. O trabalho é baseado na teoria desenvolvida recentemente para *streaming* de áudio e foi estendida nesta tese para a aplicação interativa de voz sobre IP (*Internet Protocol*).

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

COMPUTATIONAL METHODS FOR MARKOV REWARD MODELS

Ana Paula Couto da Silva

October/2006

Advisors: Rosa Maria Meri Leão

Edmundo Albuquerque de Souza e Silva

Department: Computer and System Engineering

The importance of modeling computer and communication systems has been increasing during the last years and several questions concerned to the behavior of these systems can be answered using a set of models.

The main contribution of this thesis is to propose some computational methods for solving Reward Markov Models. We will present a method for solving fluid queuing models, using *quasi birth-death* process. Furthermore, we will present a differential equation that models a fluid queuing system with several rate rewards associated to the same model state, together with an efficient solution for this equation.

As another contribution, we will present an approximated method, with bounds, used for calculating the mean cumulated reward up to absorption. This approach is specially attractive to the models with millions of states.

Finally, we will present the utilization of the *Pseudo-Subjective Quality Assessment* methodology for the interactive voice over IP applications (VoIP). Previous results of using this same methodology for the streaming audio encouraged us to extend its utilization to the interactive VoIP application.

Sumário

Resumo	vii
Abstract	viii
1 Introdução	1
1.1 Objetivos da Tese	4
2 Fundamentação Teórica	8
2.1 Introdução	8
2.2 Processos Markovianos	8
2.3 Cadeias de Markov de Tempo Discreto (DTMCs)	9
2.4 Cadeias de Markov de Tempo Contínuo (CTMCs)	10
2.5 Processos Markovianos com Recompensa	11
2.6 Uniformização	13
2.7 Complemento Estocástico	15
2.8 Cadeias de Markov com Estruturas Repetidas	17
2.8.1 Processo <i>Quasi-Birth-Death</i>	18
2.9 Modelos de Fluido	20
2.10 Cadeias de Markov Ocultas (HMM's)	22
3 Método de Solução para Modelos de Filas de Fluido	25

3.1	Introdução	25
3.2	Trabalhos Relacionados	29
3.2.1	Anick <i>et al</i> [5]	29
3.2.2	Ahn e Ramaswami ([3] e [4])	32
3.2.3	Algoritmo <i>Folding</i> [142]	36
3.3	Método Proposto para uma única Fila de Fluido	38
3.4	Método Proposto para Duas Filas de Fluido	45
3.5	Custo Computacional	50
3.6	Comparação entre os Métodos	54
3.7	Exemplos	56
3.7.1	Modelo de uma Fonte de Vídeo	57
3.7.2	Modelo de Tráfego Agregado	57
3.7.3	Modelos com Duas Filas de Fluido	61
3.8	Sumário das Contribuições	64
4	Valor Esperado da Recompensa Acumulada para Modelos com Estados Absorventes	65
4.1	Introdução	65
4.2	Trabalhos Relacionados	68
4.2.1	Método de Courtois e Semal [17]	69
4.2.2	Métodos de Muntz <i>et al</i> [104] e de Mahevas <i>et Rubino</i> [91]	71
4.2.3	Métodos de Bobbio <i>et Trivedi</i> [10] e de Pourret [116]	73
4.3	Definições Preliminares do Método Proposto	74
4.3.1	Notação	74
4.3.2	Modelo	75
4.3.3	Exemplo Base	78

4.3.4	Classificação dos Estados	79
4.3.5	Uniformização	80
4.3.6	Decompondo a Recompensa Acumulada	81
4.3.7	Complemento Estocástico	82
4.4	O Método de Aproximação Proposto	84
4.4.1	Limite Inferior	84
4.4.2	Limite Superior	87
	Limite Superior para $\varrho^{\mathcal{Y},\mathcal{S}}$	87
	Limite Superior para $\varrho^{\mathcal{Y},\mathcal{F}}$	93
	Definição de Limites para a Matriz Exata \tilde{U}	95
4.5	Redefinindo o Conjunto de Estados Lentos	96
4.6	Visão Geral do Método Proposto e Custo Computacional	97
4.7	Exemplos	99
4.7.1	Sistema de Filas $M/M/1/H$	100
4.7.2	Sistema com C componentes	102
4.7.3	Sistema com vários componentes idênticos	105
4.7.4	Sistema em <i>Grid</i>	108
4.7.5	Sistema de Banco de Dados	109
4.8	Sumário das Contribuições	111
5	Medidas de Interesse para Modelos com Várias Recompensas Associadas a cada Estado	113
5.1	Introdução	113
5.2	Descrição dos Modelos de Markov oculto e expandido	115
5.3	Uma Equação Diferencial para o Modelo de Markov oculto	119

5.4	Algoritmo Iterativo para Obtenção da Solução Aproximada da Equação	
5.26	125
5.4.1	Notação	126
5.4.2	O Algoritmo	126
5.4.3	Solução Inicial	126
5.4.4	Definindo um valor aproximado para o vetor $w^l(y)$	128
5.4.5	Definindo um valor aproximado para o vetor $w(y)$	129
5.4.6	Visão Geral do Algoritmo Iterativo Proposto e o Custo Computacional	132
5.5	Exemplos	135
5.5.1	Classe 1: Dois estados ocultos	137
5.5.2	Classe 2: Variando a carga do sistema	140
5.5.3	Classe 3: HMM baseada em [37]	145
5.6	Sumário das Contribuições	148
6	Avaliando a Qualidade de Serviço para Aplicações Interativas de Voz na Internet	150
6.1	Introdução	150
6.2	Trabalhos Relacionados	153
6.3	A Metodologia PSQA para Avaliação de Qualidade	155
6.3.1	Rede Neural Aleatória (<i>Random Neural Network</i>)	161
6.3.2	Estudo dos Parâmetros que são Variáveis Aleatórias	163
6.4	Descrição dos Experimentos Realizados	166
6.4.1	Ferramenta Interativa de Voz - VivaVoz	166
6.4.2	<i>Speex</i>	168
6.4.3	Metodologia e Ambiente dos Experimentos	168

6.5	Resultados Obtidos	171
6.5.1	Desempenho da Metodologia PSQA	171
6.5.2	Estudo da Influência dos Parâmetros na Avaliação de Qualidade (MOS)	174
	Percentual de Perda, Retardo e esquema de FEC	174
	Percentual de Perda, Tamanho Médio da Rajada de Perda	177
	<i>Jitter</i>	177
	Taxa de Envio de <i>Bits</i>	178
6.6	Sumário das Contribuições	180
7	Conclusões e Trabalhos Futuros	182
	Referências Bibliográficas	185

Lista de Figuras

2.1	Comparação entre IR(t) e CR(t).	12
2.2	Diagrama de estados de uma fila $M/M/1$	18
2.3	Exemplo de um reservatório de fluido modulado por um processo de Markov com três estados $\{0, 1, 2\}$	21
2.4	Cassino Desonesto.	23
3.1	Matriz \mathbf{P}^* do processo \mathcal{Y}^* com as respectivas submatrizes. Os blocos com o mesmo índice representam submatrizes idênticas.	40
3.2	Diagrama do particionamento do estados do processo \mathcal{Y}^*	41
3.3	Matriz \mathbf{P}^* e a representação do segundo nível de particionamento.	42
3.4	Etapa (a): Eliminação do subconjunto \mathcal{S}_l^+	42
3.5	Etapa (b): Eliminação do subconjunto \mathcal{S}_{l+1}^-	43
3.6	Diagrama ilustrando a eliminação dos blocos em cada passo do algoritmo.	43
3.7	Diagrama para um modelo com duas filas de fluido.	45
3.8	Modelo agregado para duas fontes do tipo <i>on-off</i> . A taxa ϕ_i , com $i = \{1, 2\}$, é a taxa de transição entre os estados <i>on</i> e o <i>off</i> . A taxa β_i , com $i = \{1, 2\}$, é a taxa de transição entre os estados <i>off</i> e o <i>on</i>	46
3.9	Diagrama com a representação da cadeia para o exemplo com duas fontes <i>on-off</i> e duas filas de fluido.	49
3.10	Visão Geral dos métodos para solução de modelos de fluido.	50
3.11	Sistema onde $B/\gamma = 10000$	54

3.12	Sistema onde $B/\gamma = 200000$	54
3.13	Sistema onde $B/\gamma = 100$	55
3.14	Distribuição complementar da recompensa acumulada - modelo de uma fonte de vídeo.	58
3.15	Distribuição complementar da recompensa acumulada - Modelo do tráfego agregado do canal de saída do Departamento de Sistemas e Computação. .	59
3.16	Distribuição complementar da recompensa acumulada - Modelo do tráfego agregado de um canal <i>Internet</i> [112].	60
3.17	Estudo comparativo entre o paradigma de modelagem de fluido, o modelo proposto, o modelo de chegada e saída em <i>batch</i> de pacotes e o modelo real de pacotes.	61
3.18	Distribuição Complementar do conteúdo de cada uma das filas - modelo com duas filas. O tamanho para cada uma das filas é igual 100.	62
3.19	Distribuição Complementar do conteúdo de cada uma das filas - modelo com duas filas. O tamanho para cada uma das filas é igual 1000.	63
3.20	Distribuição Complementar do conteúdo de cada uma das filas - modelo com duas filas. O tamanho para cada uma das filas é igual a 5000.	63
3.21	Distribuição Complementar do conteúdo de cada uma das filas - modelo com duas filas. O tamanho para cada uma das filas é igual 50000.	63
4.1	Cadeia de Markov para um exemplo simples que será usado como base para a apresentação do método proposto.	79
4.2	Cadeia de Markov discreta \mathcal{Z}	89
4.3	Representação gráfica da medida ν_i	94
4.4	Nova representação para o conjunto de estados lentos. A redefinição faz com que um número fixo de estados, originalmente rápidos, seja considerado para que a medida aproximada se torne mais próxima da medida real ECRA. Assim, todo o método é aplicado ao novo conjunto \mathcal{S}'	97
4.5	Diagrama com a visão geral do método proposto.	98
4.6	Distribuição <i>Coxian</i>	100

4.7	Representação do sistema com diversos componentes.	103
4.8	Cadeia de Markov que modela a evolução de cada um dos componentes do sistema.	106
4.9	Cadeia de Markov para o sistema em <i>grid</i>	108
5.1	(a) Modelo considerado neste capítulo, onde a cada recompensa r_i , associamos uma probabilidade de ocorrência da mesma. (b) Fluido de segunda ordem.	114
5.2	Representação gráfica do modelo de Markov expandido e o modelo de Markov oculto, para um pequeno exemplo com dois estados ocultos e emissão de recompensas $\{r_1, r_2\}$	118
5.3	HMM utilizado para o primeiro conjunto de exemplos. O processo possui dois estados ocultos e três recompensas distintas.	137
5.4	Comportamento do erro absoluto quando a diferença entre os valores de recompensas atribuídas a um estado oculto varia dentro de um determinado intervalo.	138
5.5	Distribuição da recompensa acumulada em uma fila de tamanho $B = 1000$ e $r_2 \approx r_3$	139
5.6	Distribuição da recompensa acumulada em uma fila de tamanho $B = 1000$ e $r_2 \ll r_3$	140
5.7	HMM utilizado para o segundo conjunto de exemplos. O processo possui dois estados ocultos três recompensas distintas.	141
5.8	Resultados obtidos pelo algoritmo iterativo para um modelo de Markov oculto com dois estados ocultos e carga no sistema de $\rho = 0.99$ e $B = 500$	141
5.9	Resultados obtidos pelo algoritmo iterativo para um HMM com dois estados ocultos e carga no sistema de $\rho = 0.7$ e $B = 500$	142
5.10	Resultados obtidos pelo algoritmo iterativo para um HMM com dois estados ocultos, $B = 1000$ e carga no sistema de $\rho = 0.7$	142
5.11	Resultados obtidos pelo algoritmo iterativo para um HMM com dois estados ocultos, $B = 1000$ e carga no sistema de $\rho = 0.5$	143

5.12	Modelo de Markov oculto utilizado para o segundo conjunto de exemplos. Neste, temos três estados ocultos, com a possibilidade de emissão de nove recompensas distintas.	144
5.13	Resultados obtidos pelo algoritmo iterativo para um HMM com três estados ocultos, $B = 100$ e carga no sistema de $\rho = 0.99$	144
5.14	Resultados obtidos pelo algoritmo iterativo para um HMM com três estados ocultos, $B = 100$ e carga no sistema de $\rho = 0.9$	145
5.15	Resultados obtidos pelo algoritmo iterativo para um HMM com três estados ocultos, $B = 100$ e carga no sistema de $\rho = 0.8$	146
5.16	Processo HMM em estudo.	146
5.17	Distribuição da recompensa acumulada em uma fila de tamanho $B = 1000$, com valores de recompensas referentes ao tráfego real coletado em um canal do Departamento de Engenharia de Sistemas e Computação/COPPE/UFRJ, e previamente mostrado em [37].	147
6.1	Representação gráfica das três etapas principais da metodologia PSQA.	157
6.2	Visão global da metodologia PSQA aplicada a avaliação de aplicações interativas de VoIP.	161
6.3	Componentes de uma rede neural aleatória. Este esquema representa uma parte de uma rede do tipo <i>feed-forward</i> com três camadas.	162
6.4	Modelo de <i>Gilbert</i> . Quando o processo está no estado 0, a transmissão é sem erros. No estado 1, erros ocorrem com probabilidade d	164
6.5	Modelo simplificado de <i>Gilbert</i> . Quando o processo está no estado 0, a transmissão é sem erros. No estado 1, todos os pacotes são perdidos.	165
6.6	Esquema de redundância do tipo 2:6.	167
6.7	Esquema 1:2::3:6 utilizado durante os testes subjetivos realizados. O retângulo em preto representa a redundância do esquema 1:2, enquanto o listrado representa a redundância do esquema 3:6. Os arcos acima dos pacotes representam o esquema 1:2, e os abaixo, o esquema 3:6.	167
6.8	MOS real, obtido a partir dos testes subjetivos e MOS estimado usando duas arquiteturas diferentes para a rede neural.	173

6.9	MOS em função do retardo para as arquiteturas estudadas.	173
6.10	Qualidade percebida pelo usuário como função do retardo, para algumas taxas de perda (sem esquema de FEC). O valor do jitter é de 10% do retardo.	175
6.11	Qualidade percebida pelo usuário como função do retardo, para algumas taxas de perda (com esquema de FEC - 1:2::3:6). O valor do jitter é de 10% do retardo.	175
6.12	Qualidade percebida pelo usuário em função da taxa de perda, para alguns valores de retardo (sem esquema de FEC).	176
6.13	Qualidade percebida pelo usuário em função do percentual de perda e do retardo, para cenários sem FEC.	176
6.14	Qualidade percebida pelo usuário em função do percentual de perda, para três diferentes esquemas de FEC [47], retardo igual a 200ms, jitter = 0.1*retardo e taxa <i>bits</i> igual a 14.2.	177
6.15	Qualidade percebida pelo usuário em função do percentual de perda e de valores para o MLBS, sem esquema de FEC.	178
6.16	Qualidade percebida pelo usuário em função do retardo e do <i>jitter</i> , para percentual de perda de 0%.	179
6.17	Qualidade percebida pelo usuário em função do retardo e do <i>jitter</i> , para percentual de perda de 5%.	179
6.18	Qualidade percebida em função da taxa de <i>bits</i> , para diversos valores de percentual de perda, com utilização de esquema de FEC (1:2:3:6). O retardo considerado foi de 125 ms.	180
6.19	Qualidade percebida em função da taxa de <i>bits</i> , para diversos valores de percentual de perda, sem esquema de FEC.	180

Lista de Tabelas

4.1	Descrição dos processos, matrizes e vetores utilizados durante este capítulo com seus respectivos significados.	76
4.2	Descrição dos processos, matrizes e vetores utilizados durante este capítulo com seus respectivos significados.	77
4.3	Resultados para o sistema $M/M/1/10$, com: $\gamma = 100$, $\phi = 0.001$, $\omega = 1$, $\delta = 1$ e $c = 0.99999$	101
4.4	Resultados para o sistema $M/M/1/10$, com: $\gamma = 10$, $\phi = 0.001$, $\omega = 1$, $\delta = 1$ e $c = 0.99999$	102
4.5	Resultados para o sistema com $C = 1000$ componentes e $\gamma = 1$	103
4.6	Redução do espaço de estados para o sistema com $C = 1000$ componentes.	103
4.7	Resultados para o sistema com $C = 5000$ componentes, com $\gamma = 1$, $K = 1$ e $\phi = 1e-4$	104
4.8	Redução do espaço de estados para o sistema com $C = 5000$ componentes.	104
4.9	Resultados para o sistema com $C = 10000$ componentes, com tempo de reparo com distribuição <i>Coxian</i> , $\gamma = 1$, $K = 1$ e $\phi = 1e-5$	105
4.10	Redução do espaço de estados para o sistema com $C = 10000$ componentes.	105
4.11	Resultados para um sistema com vários componentes idênticos e independentes, usando as simetrias da cadeia de Markov (<i>strong lumpability</i>). Os valores para os parâmetros são $\phi c 1e-6$, $\phi(1 - c) = 1e-7$ e $\gamma = 1$	106
4.12	Redução do espaço de estados para o sistema com vários componentes idênticos e independentes, usando as simetrias da cadeia de Markov (<i>strong lumpability</i>).	107

4.13	Resultados para o sistema com vários componentes idênticos, sem utilizarmos as simetrias da cadeia de Markov. Os valores exatos para o MTTF são obtidos explorando a propriedade <i>strong lumpability</i> . Os valores para os parâmetros são: $\phi c = 1e-6$, $\phi(1 - c) = 1e-7$ e $\gamma = 1$, para todos os casos.	107
4.14	Redução do espaço de estados para o sistema com vários componentes idênticos e independentes, sem utilizar as simetrias da cadeia de Markov (<i>strong lumpability</i>).	107
4.15	Resultados para o sistema com diversos componentes idênticos, sem utilizar as simetrias da cadeia de Markov.	108
4.16	Resultados para o sistema em <i>grid</i> com $C = 5000$ processadores.	109
4.17	Resultados para o sistema em <i>grid</i> com $C = 5000$ processadores, utilizando a redefinição \mathcal{S}'	109
4.18	Resultados para o sistema de banco de dados.	110
4.19	Redução do espaço de estados para o sistema de banco de dados composto por CPUs e discos rígidos.	110
4.20	Resultados para o sistema de banco de dados, considerando o comportamento das unidades de memória.	111
4.21	Redução do espaço de estados para o sistema de banco de dados composto por CPUs, discos rígidos e unidades de memória.	111
5.1	Descrição das matrizes e vetores utilizados na apresentação do algoritmo iterativo proposto.	127
6.1	Influência do retardo em uma direção na interatividade da comunicação segundo a recomendação G.114.	158
6.2	Descrição dos valores de MOS e seus respectivos significados.	160
6.3	Parâmetros de rede e de codificação utilizados durante os testes subjetivos.	169

Capítulo 1

Introdução

Há mais de 30 anos, a modelagem e análise de sistemas de computação e comunicação tem se consolidado como uma área essencial para o entendimento e a predição do comportamento de sistemas reais. O aumento da dependência em relação a esses sistemas por parte da sociedade é diretamente proporcional a necessidade de eficiência e confiabilidade dos mesmos. Deste modo, respostas precisas devem ser fornecidas a determinadas questões, como por exemplo: “Quanto tempo o sistema em questão estará em funcionamento, sem que haja uma interrupção?”; “Qual é a probabilidade do sistema operar acima de um certo nível de eficiência, durante um período de observação t ?”; entre diversas outras questões que objetivam prever o comportamento dos sistemas modernos.

Para responder a questões relacionadas a eficiência e/ou confiabilidade, modelos são desenvolvidos. Um modelo é uma abstração do sistema em estudo, tendo, no entanto, as características necessárias para capturar com precisão o comportamento deste, dentro de uma margem de tolerância. Os modelos servem para, por exemplo, estudar o comportamento de sistemas quando parâmetros são variados, para prever a resposta destes quando as cargas variam, e, obviamente são a única forma de estudá-los quando um protótipo ainda não existe.

Em muitos casos, incluindo os abordados nesta tese, modelos estocásticos são escolhidos para analisar o desempenho de um sistema real. O comportamento de um modelo estocástico é probabilístico por natureza: por exemplo, o tempo de serviço de um roteador em uma rede de dados não é conhecido exatamente, sendo atribuído a este parâmetro uma distribuição de probabilidades.

Uma classe particular de modelos estocásticos é a classe de modelos *Markovianos*.

Neste tipo de modelagem, proposto pelo matemático russo A.A. Markov em 1906, o sistema em estudo é representado por uma cadeia, definida como “*uma seqüência infinita de variáveis $x_1, x_2, \dots, x_k, x_{k+1}, \dots$ conectadas de tal maneira que x_{k+1} , para qualquer k , é independente de x_1, x_2, \dots, x_k , para o caso em que x_k é conhecido*” (a referência [6] cita a versão original do trabalho de A. A. Markov [93] além de apresentar a definição para processos Markovianos dada por A.A. Markov). Em outras palavras, a evolução do processo Markoviano em um tempo futuro, condicionado no presente e no passado, somente depende do seu valor no presente.

Os modelos *Markovianos*, com espaço de estados discretos, podem ser divididos em dois principais grupos: os modelos de tempo contínuo e os modelos de tempo discreto. No primeiro caso, o estado do sistema pode mudar em qualquer instante de tempo enquanto que no segundo caso, as mudanças somente são realizadas em instantes discretos de tempo. No Capítulo 2 apresentaremos uma propriedade importante encontrada na literatura onde um processo de Markov contínuo pode ser reescrito na sua versão discreta equivalente, possibilitando a definição de medidas de interesse de uma forma mais simplificada e eficiente.

Um modelo permite que uma série de medidas de interesse sejam calculadas e que o sistema real em estudo seja avaliado. Na literatura existem diversos paradigmas para definição das medidas de interesse que devem ser calculadas pelos métodos de solução propostos. Nesta tese estamos interessados em particular nas medidas de interesse que podem ser definidas através de recompensas atribuídas a estados ou transições entre estados. Modelos de Markov utilizando este paradigma são referenciados na literatura como *Modelos Markovianos com Recompensas* (ver, por exemplo, [30],[100] e referências nestes artigos).

O conceito de recompensas é amplo o suficiente para se adequar ao cálculo de diversas medidas de interesse na área de desempenho e confiabilidade de sistemas. Conseqüentemente, é vasta a literatura sobre a utilização do paradigma de recompensas em modelos Markovianos para o cálculo destas medidas ([30],[29],[88],[100],[27],[106],[37],[38]). Como exemplo de medidas de interesse, podemos citar a vazão, tempo de resposta em sistemas de filas, a disponibilidade de um sistema, entre outras. A recompensa atribuída a um processo Markoviano pode se classificar em duas categorias distintas: recompensas de taxa, que indicam uma recompensa ganha por unidade de tempo em que o sistema permanece no estado ao qual a recompensa é associada, e recompensas de impulso, onde as recompensas são atribuídas à transições que ocorrem entre os estados do modelo.

A partir da definição das medidas de interesse e do paradigma a ser utilizado, devemos escolher o tipo de análise a ser empregada, sendo esta relacionada com o tempo de observação do sistema em questão. Desta forma, as medidas de interesse podem ser classificadas em dois tipos: medidas em estado estacionário e medidas em estado transiente.

No cálculo das medidas em estado estacionário, o tempo de observação t do modelo é muito grande quando comparado com o tempo de ocorrência dos eventos, ou seja $t \rightarrow \infty$. Diferentemente, no cálculo de medidas em estado transiente, o tempo de observação não é muito grande. Estas medidas fornecem o comportamento do sistema em um período relativamente curto de observação, supondo uma determinada distribuição inicial de probabilidades.

Uma vez definidas as medidas de interesses a serem calculadas pelo modelo e o período de observação t do sistema real em estudo, um método de solução apropriado deve ser usado. Dado a enorme diferença entre as características dos sistemas, um determinado método de solução pode ser extremamente eficiente para uma classe de modelos, não tendo o mesmo desempenho para uma outra classe específica. Podemos medir a eficiência, por exemplo, através do número total de operações realizadas ou do espaço de memória exigido para obtenção dos resultados.

Desta forma, para a solução de modelos Markovianos devemos analisar as vantagens e desvantagens quando escolhemos um determinado método de solução em particular. Em alguns casos, podemos utilizar métodos iterativos para obtenção de medidas de interesse em estado estacionário, como os métodos de Gauss-Seidel, Jacobi ou SOR. No entanto, temos o problema da definição do critério de parada e como declarar que a solução encontrada é a solução real do problema. Adicionalmente, podemos efetuar uma mudança no modelo de tal maneira que este se torne mais facilmente tratável (por exemplo, truncando o espaço de estados). Obviamente é importante estudar os erros introduzidos na solução quando simplificações são feitas.

A ampla área de pesquisa de modelagem de sistemas de computação e comunicação, composta por diversas questões ainda em aberto (relacionadas com a eficiência da modelagem e a definição do melhor método de solução para uma classe específica de modelos), bem como o aumento da complexidade dos sistemas a serem representados e o surgimento de diversas outras medidas de interesse são motivações para que novos tipos de métodos computacionais continuem a ser pesquisados e propostos.

1.1 Objetivos da Tese

O objetivo principal desta tese é propor novos métodos de solução para classes de modelos Markovianos onde recompensas são empregadas. Para tal, dividimos este trabalho em quatro partes: a primeira parte focaliza a classe de modelos estocásticos de fluido, para os casos onde o tempo de observação tende ao infinito; a segunda parte estuda um método de solução aproximado, com limites do erro, para a classe de modelos Markovianos com espaço de estados da ordem de milhões de estados e que possuem estados absorventes; a terceira parte destina-se ao caso da análise, em estado estacionário, para modelos ocultos de Markov com diversas recompensas onde taxas de recompensas são atribuídas a cada estado do modelo de acordo com uma dada distribuição de probabilidade; e, por último, apresentamos uma metodologia para avaliar a Qualidade de Serviço percebida pelo usuário de uma aplicação de voz interativa, utilizando a *Internet*. A escolha desta última aplicação deve-se ao fato de que parte dos modelos teóricos estudados podem ser aplicados a modelos de taxa de erros em aplicação de voz sobre *Internet Protocol* (IP) que, por sua vez, podem ser baseados em cadeias ocultas de Markov ([48] e [50]). A seguir apresentaremos brevemente cada uma destas partes. A discussão detalhada de cada uma das contribuições desta tese é apresentada no início de cada capítulo.

O capítulo 2 apresenta, brevemente, o conjunto principal de conceitos que será a base de todas as contribuições desta tese. Dentre estes conceitos, podemos ressaltar a definição formal de processos Markovianos, bem como as definições relacionadas com os modelos que utilizam o paradigma de recompensas para representação das medidas de interesse a serem calculadas.

Modelos estocásticos de fluido têm sido amplamente utilizados na literatura como ferramentas para a análise de diversos sistemas de computação e comunicação. Em particular, quando as taxas dos eventos do sistema em investigação variam em muitas ordens de grandeza, o uso de modelos de fluido podem resultar em considerável redução do custo computacional quando comparado com os modelos onde todos os eventos são explicitamente representados. Uma importante área de aplicação é a modelagem de filas.

Uma das principais medidas de interesse para esta primeira classe de modelos Markovianos é a distribuição de probabilidades complementar da ocupação de uma fila de fluido, ou em termos de recompensa, da recompensa acumulada, CR , quando o tempo de observação t tende ao infinito ($P[CR > x] = \lim_{t \rightarrow \infty} P[CR(t) > x]$). Métodos tradicionais encontrados na literatura utilizam a teoria de autovalores e autovetores para obtenção destes resultados. A limitação deste paradigma, principalmente no caso de fi-

las com capacidade finita, é que os autovalores utilizados para obtenção da solução são positivos e negativos e erros numéricos podem gerar soluções instáveis, resultando em probabilidades negativas ou até mesmo em valores que podem crescer sem limites.

O principal objetivo do Capítulo 3 é apresentar uma nova abordagem para a solução dos modelos descritos no parágrafo anterior, através de um processo conhecido na literatura como *Quasi-Birth-Death* (QBD). A principal vantagem desta nova abordagem é evitar os problemas de precisão e eficiência que acontecem na solução numérica de equações diferenciais, no caso de filas com capacidade infinita, e problemas de determinação de condições de contorno que ocorrem nos casos das filas finitas. O método proposto neste capítulo possui duas contribuições principais. A primeira contribuição está direcionada a modelagem proposta, onde transformamos uma cadeia de Markov com recompensas de taxa em tempo contínuo em uma cadeia equivalente em tempo discreto e com recompensas de impulso. A segunda contribuição é a apresentação de um algoritmo de eliminação mais eficiente para a solução do processo QBD resultante.

Medidas do tempo médio até a ocorrência de uma falha são muito importantes na análise de confiabilidade. Porém, quando a cardinalidade do espaço de estados é grande (milhões de estados), métodos tradicionais de solução são extremamente custosos em termos computacionais e de armazenamento, sendo muitas das vezes proibitivos. Obter essas medidas não é uma tarefa trivial, principalmente se a matriz de taxas de transição do modelo não possui estruturas especiais. Podemos citar como técnicas mais conhecidas para resolver problemas de alta cardinalidade do espaço de estados: a agregação de estados ([10],[130],[132]), a teoria de complemento estocástico (caso especial do conceito mais geral conhecido como *Perron complementation*, apresentado em [16]) e a técnica de *path-based*, que permite calcular as medidas de interesse usando somente uma descrição local do sistema, sem a exploração de todos os estados do modelo ([13],[118],[116]). Todos estes métodos apresentam um resultado aproximado para a medida de interesse. Podemos também citar a teoria de autômatos estocásticos juntamente com a técnica de álgebra tensorial, onde é possível obter solução de medidas de interesse sem gerar todo o espaço de estados ([46] e referências), dado que somente uma parte do espaço de estados é utilizada.

No Capítulo 4 apresentamos um método aproximado, com controle de erro, para o cálculo do valor esperado da recompensa acumulada para sistemas com estados absorventes e com cardinalidade da ordem de milhões de estados. Entende-se por um estado absorvente aquele em que, considerando um processo estocástico de tempo discreto, a probabilidade de retorno, em um passo, a um determinado estado i , dado que o processo

está em i , é igual a 1. Isto significa que quando o processo entra em um estado classificado como absorvente, este não pode mais evoluir para qualquer outro estado do sistema.

Como um exemplo para a aplicação do método proposto no Capítulo 4, seja um sistema com diversos componentes diferentes. Se, para este sistema, associamos a cada estado uma perda de capital devido ao não funcionamento de um componente, podemos calcular o montante de capital perdido até que o sistema se torne não operacional.

A classe de modelos Markovianos conhecidos como Modelos de Markov ocultos ou *Hidden Markov Models* (HMM), cujo a aplicação original está relacionada a processamento de sinal de voz [119], tem se tornado uma poderosa ferramenta de modelagem para sistemas de comunicação e computação ([41],[37],[138],[50]). A principal característica desta classe de modelos é que, para cada estado oculto do sistema, diferentes símbolos são emitidos com uma dada distribuição de probabilidades. Estes símbolos podem ser vistos como recompensas de taxa que estão sendo emitidas em cada estado do modelo oculto, segundo uma distribuição de probabilidades. Neste caso, os métodos tradicionais encontrados na literatura para determinação de medidas de interesse para modelos com uma única recompensa emitida em cada estado não são computacionalmente eficientes, sendo, em alguns casos não aplicáveis ([31],[5],[30],[100],[106],[128],[3],[4],[22],[19],[21]). O objetivo do Capítulo 5 é apresentar um método de solução, em estado estacionário, para a obtenção da distribuição de probabilidades acumulada, para a variável aleatória que representa a recompensa acumulada no sistema.

A última contribuição apresentada nesta tese refere-se a aplicações interativas de voz sobre IP (*Internet Protocol*). Como a *Internet* atual não garante nenhuma qualidade ao usuário final, é importante que a Qualidade de Serviço (QoS) provida pela rede seja monitorada constantemente, de forma que as aplicações possam implementar maneiras de manter a QoS em um nível mínimo aceitável. No entanto, é essencial que sejam determinados quais são os parâmetros que influenciam a QoS, bem como entender como o efeito combinado destes parâmetros afetam a QoS percebida pelo usuário final das aplicações em tempo real. Como os modelos de recompensa são aplicáveis a modelos relacionados a QoS de aplicações de voz sobre IP ([49],[50],[94]), abordamos essa área de forma a entender o comportamento de aplicações reais e em como melhor parametrizar estes modelos.

No Capítulo 6 iremos estudar aplicações de voz sobre IP (VoIP), com interatividade. Mostraremos que a qualidade da conversa, percebida pelo usuário final, depende de uma combinação complexa de vários parâmetros. Estendemos a utilização da metodologia

1.1 Objetivos da Tese

apresentada em [101] ao contexto de conversas interativas e mapeamos os valores de um conjunto de parâmetros em uma única nota que é próxima ao que um observador humano atribuiria a uma sessão de conversação interativa, caso este tivesse participado de um teste subjetivo.

Finalizando, o Capítulo 7 apresenta as considerações finais sobre a tese e as propostas de alguns temas para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

2.1 Introdução

O objetivo deste capítulo é apresentar, resumidamente, os principais fundamentos teóricos que foram utilizados no desenvolvimento desta tese. A estrutura do capítulo é indicada a seguir. Na Seção 2.2 apresentamos a definição formal de processos Markovianos, uma vez que todos os sistemas a serem estudados nesta tese serão representados através desta categoria de processo estocástico. As Seções 2.3 e 2.4 são destinadas a apresentar as duas principais classificações, quando consideramos o intervalo entre observações, de um processo Markoviano: discreto e contínuo, respectivamente. A Seção 2.5 apresenta o paradigma de recompensas para obtenção de medidas de interesse que representam o comportamento do sistema em estudo. A técnica de uniformização, que mapeia um processo Markoviano contínuo em um equivalente na sua versão discreta, é descrito na Seção 2.6. A teoria de complemento estocástico é apresentada na Seção 2.7. Apresentar o processo Markoviano do tipo *Quasi-Birth-Death* é o objetivo da Seção 2.8. A formalização de modelos de fluido é apresentada na seção 2.9. Finalizando, a Seção 2.10 apresenta as principais definições dos denominados modelos ocultos de Markov (*Hidden Markov Models*).

2.2 Processos Markovianos

Um processo Markoviano é um processo estocástico que possui uma dependência limitada em relação ao seu passado [133]. Seja o processo estocástico $\mathcal{X} = \{X(t) : t \geq 0\}$.

2.3 Cadeias de Markov de Tempo Discreto (DTMCs)

Os valores que $X(t)$ pode assumir no tempo representam os estados do processo, pertencentes ao conjunto denominado \mathcal{S} . Para que um processo estocástico seja classificado como um processo Markoviano, este deve satisfazer a seguinte propriedade:

$$\begin{aligned} P[X(n) = i_n \mid X(0) = i_0, X(1) = i_1 \cdots, X(n-1) = i_{n-1}] \\ = P[X(n) = i_n \mid X(n-1) = i_{n-1}], \end{aligned} \quad (2.1)$$

conhecida na literatura como propriedade da falta de memória (*memoryless property*)[133].

Consideremos $t = n$ como o tempo presente de observação. Segundo a definição dada pela Equação 2.1, a evolução do processo Markoviano em um tempo futuro, condicionado no presente e no passado, somente depende do seu valor no presente.

2.3 Cadeias de Markov de Tempo Discreto (DTMCs)

Considere um processo Markoviano, com espaço de estados discretos, conforme definição dada pela Equação 2.1 e, sem perda de generalidade, que o espaço de estados \mathcal{S} seja composto por números inteiros não-negativos. A cadeia de Markov de tempo discreto (DTMC) que caracteriza este processo mostra a evolução entre os estados de \mathcal{S} nos instantes de tempo $t = \{0, 1, 2, \dots\}$.

Seja a probabilidade de transição do estado $i \in \mathcal{S}$ para o estado $j \in \mathcal{S}$, no instante $t = n$, definida como $P[X(n) = j \mid X(n-1) = i]$. A matriz de transição de probabilidades \mathbf{P} da cadeia DTMC é representada por:

$$\mathbf{P} = \begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,j} & \cdots \\ p_{1,0} & p_{1,1} & \cdots & p_{1,j} & \cdots \\ \vdots & \vdots & \cdots & \vdots & \cdots \\ p_{i,0} & p_{i,1} & \cdots & p_{i,j} & \cdots \\ \vdots & \vdots & \cdots & \vdots & \cdots \end{bmatrix}. \quad (2.2)$$

Cada linha da matriz \mathbf{P} representa a probabilidade da cadeia transicionar de um estado i para um estado j , em um único passo. Esta matriz pode ser classificada, quanto a sua dimensão, como finita caso o espaço de estados \mathcal{S} seja finito, ou infinita, caso contrário.

Partindo da matriz de probabilidades de transição em um passo, podemos definir a probabilidade de uma DTMC atingir o estado $j \in \mathcal{S}$, no n -ésimo passo, começando em um estado qualquer $i \in \mathcal{S}$ no passo $n = 0$, através da equação *Chapman-Kolmogorov*

2.4 Cadeias de Markov de Tempo Contínuo (CTMCs)

[133]:

$$p_{i,j}(n) = \sum_{k \in S} p_{i,k}(m) p_{k,j}(n-m), 0 \leq m \leq n. \quad (2.3)$$

Seja $\pi(n) = (\pi_0(n), \pi_1(n), \dots)$ o vetor de probabilidades cujo o elemento $\pi_i(n)$ representa a probabilidade da DTMC estar no estado i no passo n . Como $\pi(n) = \pi(0)\mathbf{P}(n)$, $n = \{1, 2, \dots\}$, então a probabilidade de estar no estado i no passo n é simplesmente a soma das probabilidades de todos os caminhos amostrais a partir de i até j em n passos multiplicado pela probabilidade de iniciar este caminho a partir do estado i .

Conforme apresentado em [133], para uma cadeia discreta finita, aperiódica¹ e irredutível², a distribuição de probabilidades em estado estacionário é única, não dependente da distribuição inicial e pode ser obtida através da resolução do seguinte sistema de equações lineares:

$$\pi = \pi\mathbf{P} \quad (2.4)$$

com a seguinte condição de normalização:

$$\pi\mathbf{1}^T = 1 \quad (2.5)$$

onde o vetor $\mathbf{1}$ é um vetor linha com todos elementos iguais a 1.

2.4 Cadeias de Markov de Tempo Contínuo (CTMCs)

Em uma cadeia de Markov de tempo contínuo, o processo transiciona de um estado para o outro após um tempo aleatório exponencialmente distribuído. Da mesma forma que foi construída uma matriz de probabilidades de transição para uma DTMC, podemos construir uma matriz geradora infinitesimal \mathbf{Q} com as taxas de mudança entre os estados do processo com a seguinte estrutura:

$$\mathbf{Q} = \begin{bmatrix} -q_0 & q_{0,1} & \cdots & q_{0,j} & \cdots \\ q_{1,0} & -q_1 & \cdots & q_{1,j} & \cdots \\ \vdots & \vdots & \cdots & \vdots & \cdots \\ q_{i,0} & q_{i,1} & \cdots & -q_i & \cdots \\ \vdots & \vdots & \cdots & \vdots & \cdots \end{bmatrix} \quad (2.6)$$

¹Define-se por período de um estado i , representado por d_i o maior divisor comum de um conjunto de valores positivos n , tal que $p_{ii}(n) > 0$. Um estado é classificado como aperiódico, se $d_i = 1$.

²Define-se por cadeias do tipo irredutíveis aquelas onde todos os estados podem ser alcançados por qualquer outro estado em um número finito de passos.

2.5 Processos Markovianos com Recompensa

com $q_i = -\sum_{\forall j \neq i} q_{ij}$.

Analogamente ao caso discreto, o vetor de probabilidades em estado estacionário é obtido através da solução do seguinte sistema de equações lineares:

$$\pi \mathbf{Q} = 0 \quad (2.7)$$

com a seguinte condição de normalização:

$$\pi \mathbf{1}^T = 1. \quad (2.8)$$

Conforme recapitulado em [28], uma cadeia contínua é análoga a uma cadeia discreta, ou seja, a partir de um processo contínuo, podemos obter um processo discreto similar. Esta propriedade é extremamente importante, dado que utilizar cadeias de Markov discretas no tempo por muitas vezes simplificam a solução do problema. Na seção 2.6, mostraremos como estabelecer a relação entre os processos Markovianos discreto e contínuo.

2.5 Processos Markovianos com Recompensa

O conceito de recompensas é amplo o suficiente para se adequar ao cálculo de diversas medidas de interesse na área de desempenho e confiabilidade de sistemas, conseqüentemente é vasta a literatura sobre a utilização do paradigma de recompensas em modelos Markovianos para o cálculo destas medidas ([30],[29],[88],[100],[27],[106],[37],[38]). Como exemplo de medidas de interesse, podemos citar a vazão de informação em um sistema, disponibilidade do sistema, tamanho médio de filas, entre outras.

A recompensa atribuída a um processo Markoviano pode se classificar em duas categorias distintas:

1. **Recompensas de Taxa** - A cada estado da cadeia é atribuída uma recompensa, sendo que o acúmulo desta recompensa é uma função do tempo de permanência no estado em questão;
2. **Recompensas de Impulso** - A recompensa está associada a transições entre os estados da cadeia.

Para melhor definirmos modelos Markovianos com recompensas de taxa, consideremos uma cadeia de Markov homogênea de tempo contínuo $\mathcal{X} = \{X(t), t \geq 0\}$, com

2.5 Processos Markovianos com Recompensa

espaço de estados finito $\mathcal{S} = \{i; i = 1, \dots, M\}$. A cada estado $i \in \mathcal{S}$ está associada uma recompensa de taxa de um conjunto $\mathcal{R} = \{r_i; i = 1, \dots, M\}$, onde r_i representa a recompensa ganha por unidade de tempo em que o sistema se encontra no estado i . Consideremos, também, a variável aleatória $IR(t) = r_i$ que representa a recompensa instantânea no tempo t , se $X(t) = i$.

Como um exemplo de medida de interesse a ser estudada, podemos citar a medida de recompensa de taxa acumulada, definida da seguinte forma. Seja $[0, t]$ o período de observação de um sistema em particular. A recompensa acumulada, neste intervalo, é definida como:

$$CR(t) = \int_0^t IR(\tau) d\tau.$$

Se dividirmos a recompensa acumulada pelo período de observação, obtemos a média da recompensa acumulada:

$$ACR(t) = \frac{CR(t)}{t}.$$

Consideremos, como exemplificação, modelos de confiabilidade onde podemos associar a recompensa $r_i = 1$ para todos os estados i onde o sistema está operacional e $r_i = 0$ para os estados onde o sistema não está operacional. Neste caso, a recompensa acumulada $CR(t)$, entre $[0, t]$, é o tempo total em que o sistema está operacional neste intervalo de observação.

A diferença entre os conceitos de recompensa acumulada e de recompensa instantânea pode ser observada na Figura 2.1. A recompensa acumulada é uma função crescente, se a recompensa ganha em um intervalo de tempo é positiva, ou decrescente, caso a recompensa em um intervalo de tempo seja negativa. Em contrapartida, a recompensa instantânea é constante dentro do intervalo de tempo considerado³.

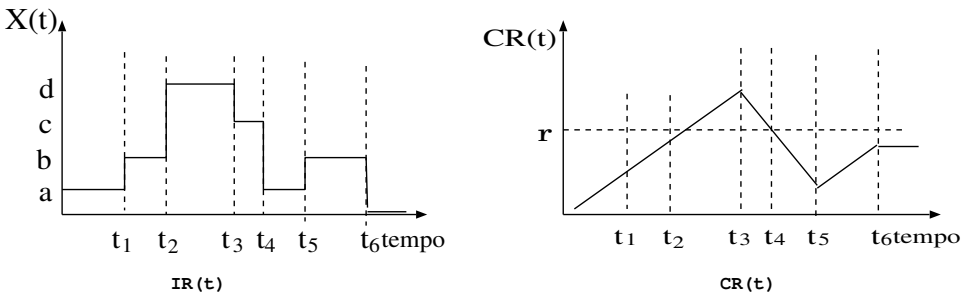


Figura 2.1: Comparação entre $IR(t)$ e $CR(t)$.

³A recompensa de taxa pode ser uma função do tempo de permanência em um estado. No entanto, nesta tese a recompensa de taxa instantânea é constante em todo o tempo de permanência em um estado.

No caso de recompensas de impulso, consideremos o mesmo processo Markoviano \mathcal{X} descrito anteriormente. A este processo, podemos atribuir K recompensas de impulso distintas $\{\rho_1, \rho_2, \rho_3, \dots, \rho_K\}$, onde ρ_i é uma recompensa dada ao processo quando uma transição particular entre estados ocorre. A recompensa atribuída a uma transição entre os estados i e j é definida por $\rho_{\hat{c}(i,j)}$, onde $\hat{c}(i, j)$ é o índice da recompensa associada a transição (i, j) . Seja $N(t)$ o número de transições de \mathcal{X} que ocorrem durante um intervalo de tempo $(0, t)$, e seja σ_n a n -ésima transição. A recompensa média de impulso acumulada durante o intervalo $(0, t)$ é dada por:

$$ACI(t) = \frac{1}{t} \sum_{n=0}^{N(t)} \rho_{\hat{c}(\sigma_n)}.$$

2.6 Uniformização

A técnica de uniformização é amplamente utilizada na análise de modelos de desempenho e confiabilidade. Esta técnica foi proposta por Jensen em 1953 [75], e na literatura podemos encontrar diversos métodos de solução que se baseiam nesta teoria ([27] e [28]).

O principal objetivo desta técnica é transformar uma cadeia de Markov de tempo contínuo em uma cadeia de Markov análoga de tempo discreto, o que facilita o cálculo de diversas medidas de interesse, tanto para medidas em tempo finito quanto para tempo “infinito” de observação⁴. A utilização da matriz \mathbf{P} definida como a matriz de probabilidades de transição em substituição a matriz geradora infinitesimal \mathbf{Q} evita problemas de erros de arredondamento, provenientes da manipulação dos elementos negativos presentes na diagonal principal de \mathbf{Q} , além de uma série de outras vantagens [28].

A seguir, fazemos uma breve apresentação da técnica e a sua aplicação direta ao cálculo da distribuição de probabilidades para um tempo finito de observação⁵. Veja [64] para uma discussão mais detalhada sobre este método e sua utilização na área de análise de desempenho de sistemas de comunicação e computação.

Para a análise da técnica de uniformização, consideremos uma cadeia de Markov \mathcal{X} com espaço de estados finito \mathcal{S} , composta por M estados. O tempo de permanência em cada estado da cadeia é exponencialmente distribuído e a probabilidade de transição entre estados é igual a q_{ij}/q_i , onde q_{ij} é a taxa de transição entre os estados (i, j) e q_i é a taxa de saída do estado i . Seja Λ um parâmetro cujo o valor, em módulo, é maior ou igual à

⁴Referenciada na literatura como Estado Estacionário de observação.

⁵Referenciada na literatura como Estado Transiente de observação.

2.6 Uniformização

maior taxa de saída de um estado (isto é, maior ou igual ao módulo do maior elemento da diagonal principal de \mathbf{Q}). O valor Λ sempre pode ser calculado no caso em que o espaço de estados é finito.

Desta forma, podemos definir um processo \mathcal{X}' com as seguintes características:

1. \mathcal{X}' possui o mesmo espaço de estados de \mathcal{X} ;
2. O tempo de residência em qualquer estado da cadeia antes de uma transição é exponencialmente distribuído com taxa Λ . Portanto, o número de transições em $(0, t)$ é dado por uma distribuição de *Poisson* com a mesma taxa Λ ;
3. A probabilidade de \mathcal{X}' transicionar de um estado i para um estado j ($i \neq j$) é igual a q_{ij}/Λ . Todavia, pode existir a transição para o mesmo estado, com probabilidade $(1 - \sum_{\forall j; j \neq i} q_{ij}/\Lambda)$.

Observando o processo \mathcal{X}' , podemos notar que a probabilidade de transição entre os estados, o tempo de permanência em cada estado da cadeia e o espaço de estados são equivalentes aos do processo \mathcal{X} (maiores detalhes em [39] e [27]), o que nos possibilita afirmar que os processos \mathcal{X} e \mathcal{X}' são equivalentes.

Seja $\mathbf{P} = \mathbf{I} + \mathbf{Q}/\Lambda$, a matriz que fornece as transições de probabilidade entre os estados do processo \mathcal{X}' . Assim, o processo \mathcal{X} é equivalente à cadeia de Markov de tempo discreto subordinado a um processo de *Poisson* com taxa Λ .

Consideremos uma cadeia de Markov de tempo discreto $\mathcal{Z} = \{Z_n : n = 0, 1, \dots\}$ com espaço de estados finito \mathcal{S} e matriz de transições de probabilidade \mathbf{P} , e seja $\mathcal{N} = \{N(t) : t \geq 0\}$ o processo de *Poisson* com taxa Λ independente de \mathcal{X} . A transição de tempos de \mathcal{X}' ocorre de acordo com o processo de *Poisson*, e as transições entre estados ocorrem de acordo com a cadeia discreta \mathcal{Z} e são dadas pela matriz \mathbf{P} . Assim, $\mathcal{X}(t) = \mathcal{X}'(t) = \mathcal{Z}_{N(t)}$.

Como ilustração para a aplicação da técnica de uniformização, seja $\boldsymbol{\pi}(t) = (\pi_1(t), \pi_2(t), \dots, \pi_M(t))$ o vetor de probabilidades em um tempo finito de observação, onde $\pi_i(t)$ é a probabilidade do processo se encontrar no estado i no tempo t . Descondicionando no número de transições no período $[0, t]$, temos:

$$\boldsymbol{\pi}(t) = \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \boldsymbol{\pi}(0) \mathbf{P}^{(n)} \quad (2.9)$$

onde $\boldsymbol{\pi}(0)$ é o vetor de probabilidades iniciais e $\mathbf{P}^{(n)}$ corresponde ao n -ésimo passo da matriz de probabilidades de transição em \mathcal{Z} .

2.7 Complemento Estocástico

Para avaliar numericamente a Equação 2.9, podemos truncar a série infinita para um valor de N , e o erro resultante do truncamento pode ser estimado a partir dos termos resultantes da distribuição de *Poisson*. Temos, portanto:

$$\pi(t) = \sum_{n=0}^N e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \pi(0) \mathbf{P}^{(n)} + \epsilon(N), \quad (2.10)$$

onde

$$\epsilon(N) = 1 - \sum_{n=0}^N e^{-\Lambda t} \frac{(\Lambda t)^n}{n!}. \quad (2.11)$$

Finalizando, a utilização da técnica de uniformização é extremamente atrativa tanto computacionalmente quanto em termos das interpretações probabilísticas que podem ser formalizadas em diversos sistemas, após a sua aplicação.

2.7 Complemento Estocástico

O conceito de complemento estocástico ocorre em diversas aplicações que envolvem cadeias de Markov finitas [95], sendo amplamente utilizado pelos diversos algoritmos de eliminação que determinam a distribuição de probabilidades em estado estacionário ([139],[19],[142], entre outros).

Podemos descrever o complemento estocástico da seguinte forma. Seja a matriz estocástica \mathbf{P} de dimensão $m \times m$, particionada da seguinte forma:

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \cdots & \mathbf{P}_{1k} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \cdots & \mathbf{P}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{k1} & \mathbf{P}_{k2} & \cdots & \mathbf{P}_{kk} \end{pmatrix}$$

onde \mathbf{P}_{ij} representa a matriz de probabilidades de transição entre os estados pertencentes ao bloco i e os estados pertencentes ao bloco j .

Assuma que todas as matrizes diagonais sejam quadradas. Se a matriz \mathbf{P}_i representa uma submatriz de \mathbf{P} obtida através da retirada da i -ésima linha e da i -ésima coluna de blocos da matriz particionada \mathbf{P} , cada submatriz $\mathbf{I} - \mathbf{P}_i$ é uma matriz não-singular. Desta forma, a inversa $(\mathbf{I} - \mathbf{P}_i)$ existe.

Seguindo a definição de complemento estocástico encontrada em [95], para um dado índice i , seja \mathbf{P}_i uma matriz obtida a partir da matriz \mathbf{P} após a retirada da i -ésima linha e

2.7 Complemento Estocástico

da i -ésima coluna de blocos da matriz particionada \mathbf{P} , e seja \mathbf{P}_{i*} e \mathbf{P}_{*i} definidos como:

$$\mathbf{P}_{i*} = (\mathbf{P}_{i,1} \ \mathbf{P}_{i,2} \ \cdots \ \mathbf{P}_{i,i-1} \ \mathbf{P}_{i,i+1} \ \cdots \ \mathbf{P}_{i,k})$$

e

$$\mathbf{P}_{*i} = (\mathbf{P}_{1,i} \ \mathbf{P}_{2,i} \ \cdots \ \mathbf{P}_{i-1,i} \ \mathbf{P}_{i+1,i} \ \cdots \ \mathbf{P}_{k,i})^T.$$

Ou seja, \mathbf{P}_{i*} é a i -ésima linha de blocos com \mathbf{P}_{ii} removido, e \mathbf{P}_{*i} é a i -ésima coluna de blocos com \mathbf{P}_{ii} removido. O complemento estocástico de \mathbf{P}_{ii} em \mathbf{P} é definido como sendo a matriz:

$$\mathbf{S}_{ii} = \mathbf{P}_{ii} + \mathbf{P}_{i*}(\mathbf{I} - \mathbf{P}_i)^{-1}\mathbf{P}_{*i}. \quad (2.12)$$

Por exemplo, o complemento estocástico de \mathbf{P}_{22} em :

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{P}_{23} \\ \mathbf{P}_{31} & \mathbf{P}_{32} & \mathbf{P}_{33} \end{pmatrix}$$

é dado por:

$$\mathbf{S}_{22} = \mathbf{P}_{22} + (\mathbf{P}_{21} \ \mathbf{P}_{23})(\mathbf{I} - \mathbf{P}_2)^{-1}(\mathbf{P}_{12} \ \mathbf{P}_{32})^T,$$

com $(\mathbf{I} - \mathbf{P}_2)$ igual a:

$$(\mathbf{I} - \mathbf{P}_2) = \begin{pmatrix} \mathbf{I} - \mathbf{P}_{11} & -\mathbf{P}_{13} \\ -\mathbf{P}_{31} & \mathbf{I} - \mathbf{P}_{33} \end{pmatrix}.$$

O complemento estocástico pode ser facilmente encontrado quando características especiais ocorrem na matriz \mathbf{P} . Em [39] alguns casos especiais são apresentados. Identificar estes casos é de extrema importância para a redução do custo computacional dos métodos de solução baseados em eliminação de variáveis.

Finalizando, apresentamos, brevemente, a interpretação probabilística para o conceito de complemento estocástico. Seja a seguinte matriz \mathbf{P} , particionada em dois subconjuntos \mathcal{A} e \mathcal{B} :

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{\mathcal{A}\mathcal{A}} & \mathbf{P}_{\mathcal{A}\mathcal{B}} \\ \mathbf{P}_{\mathcal{B}\mathcal{A}} & \mathbf{P}_{\mathcal{B}\mathcal{B}} \end{pmatrix}$$

e

$$\mathbf{S}_{\mathcal{A}\mathcal{A}} = \mathbf{P}_{\mathcal{A}\mathcal{A}} + \mathbf{P}_{\mathcal{A}\mathcal{B}}(\mathbf{I} - \mathbf{P}_{\mathcal{B}\mathcal{B}})^{-1}\mathbf{P}_{\mathcal{B}\mathcal{A}},$$

sendo $\mathbf{S}_{\mathcal{A}\mathcal{A}}$ o complemento estocástico da matriz \mathbf{P} em relação ao subconjunto \mathcal{A} .

Consideremos um novo processo, chamado *reduzido* [95], definido pelas observações ao processo original, onde todos os estados estão representados, somente quando este

2.8 Cadeias de Markov com Estruturas Repetidas

último está em um estado pertencente ao subconjunto \mathcal{A} . No novo processo, transições entre estados pertencentes ao subconjunto \mathcal{B} são descartadas. Neste caso, um caminho entre dois estados \mathcal{A}_k e \mathcal{A}_j pertencentes a \mathcal{A} no processo reduzido, corresponde a um caminho direto entre estes dois estados no processo original ou então, um caminho que passa pelo subconjunto \mathcal{B} . Por exemplo, considerando $k = 1$ e $j = 7$, se:

$$\mathcal{A}_1 \rightarrow \mathcal{B}_2 \rightarrow \mathcal{A}_3 \rightarrow \mathcal{B}_4 \rightarrow \mathcal{B}_5 \rightarrow \mathcal{A}_6 \rightarrow \mathcal{A}_7$$

é um caminho entre \mathcal{A}_1 e \mathcal{A}_7 na cadeia original, então:

$$\mathcal{A}_1 \rightarrow \mathcal{A}_3 \rightarrow \mathcal{A}_6 \rightarrow \mathcal{A}_7$$

é o caminho correspondente na cadeia reduzida, onde visitas ao subconjunto \mathcal{B} são desconsideradas.

Resumindo, as probabilidades relacionadas com os estados em \mathcal{B} devem ser redistribuídas entre os estados pertencentes ao processo *reduzido*, de tal forma que a matriz resultante seja estocástica⁶. Esta redistribuição considera os caminhos diretos entre os estados pertencentes a \mathcal{A} e os caminhos que passam pelo conjunto de estados pertencentes a \mathcal{B} . O tempo de permanência no subconjunto \mathcal{B} é desconsiderado pelo novo processo *reduzido*.

2.8 Cadeias de Markov com Estruturas Repetidas

Analisar a estrutura da matriz de probabilidades \mathbf{P} referente a Cadeia de Markov é de extrema importância para o desenvolvimento de métodos de solução que resolvam o sistema de equações da Equação 2.4 com maior eficiência computacional. Em muitos casos, a matriz apresenta uma estrutura especial, por exemplo, com repetição de linhas, colunas ou submatrizes, ideal para desenvolvimento de algoritmos que explorem esta característica. Diversos trabalhos são encontrados na literatura cujo o enfoque é determinar estas estruturas e propor algoritmos que as utilizem com o intuito de reduzir o custo computacional para a obtenção de diversas medidas de interesse ([107],[108],[109],[111],[86],[87],[85],[52],[61],[121]).

O exemplo mais simples de uma matriz \mathbf{P} com estrutura com repetição ocorre no sistema de filas $M/M/1$, onde o intervalo entre chegadas e saídas no/do sistema são exponencialmente distribuídos. Neste caso, como não existem restrições no tamanho da

⁶Uma matriz é dita estocástica quando todas as linhas da matriz somam 1.

2.8 Cadeias de Markov com Estruturas Repetidas

população que chega ao sistema ou na capacidade de armazenamento da fila, o espaço de estados é infinito, e, portanto, a matriz \mathbf{P} também possui uma dimensão infinita.

A Figura 2.2 apresentada o diagrama de estados desta cadeia. Uma fila $M/M/1$ também é conhecida como um processo estocástico do tipo *Birth-Death*.

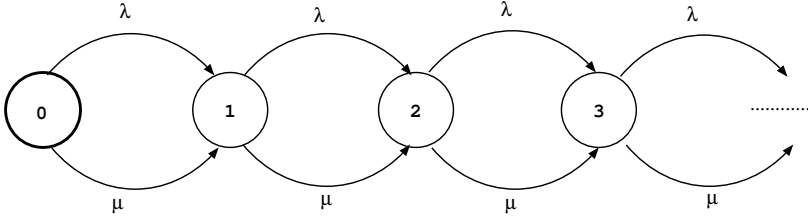


Figura 2.2: Diagrama de estados de uma fila $M/M/1$.

Na CTMC apresentada na Figura 2.2, cada estado da cadeia representa o número de indivíduos no sistema, sendo $\mathcal{S} = \{0, 1, 2, \dots\}$. Se a taxa média de chegada no sistema é λ e a taxa média de saída é μ , o vetor distribuição de probabilidades em estado estacionário π é dado por: $\pi_i = (1 - \lambda/\mu)(\lambda/\mu)^i$, $i \geq 0$, onde π_i é a probabilidade estacionária do estado i [81]. Se os estados forem ordenados em ordem crescente de índice i , temos que o gerador infinitesimal da fila $M/M/1$ é:

$$\mathbf{Q}_{M/M/1} = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & 0 & \dots \\ \mu & -(\mu + \lambda) & \lambda & 0 & 0 & \dots \\ 0 & \mu & -(\mu + \lambda) & \lambda & 0 & \dots \\ 0 & 0 & \mu & -(\mu + \lambda) & \lambda & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \quad (2.13)$$

Quando a capacidade da fila é limitada, por exemplo em M , a matriz apresentada se modifica somente no $(M+1)$ -ésimo estado, mantendo estrutura semelhante a apresentada na matriz 2.13.

Entre os diversos tipos de processos Markovianos que possuem matrizes com estruturas repetidas (maiores detalhes podem ser encontrados em [87]), destacamos os processos classificados como *Quasi-Birth-Death* [87], que serão utilizados nesta tese. A seguir, fazemos um resumo desta categoria de processos.

2.8.1 Processo *Quasi-Birth-Death*

Para exemplificar este tipo de processo, considere uma fila onde a chegada de clientes ocorre de acordo com um processo de *Poisson*, com taxa λ . O serviço se divide em dois

2.8 Cadeias de Markov com Estruturas Repetidas

estágios exponencialmente distribuídos: o primeiro com taxa μ_1 e o segundo com taxa μ_2 . Este tipo de distribuição é chamada de hipoexponencial com duas fases, e a fila descrita é do tipo $M/Hypo(2)/1$. Cada estado é definido por (i, s) , onde i , neste exemplo, é o número de clientes na fila e s é o estágio do serviço em que o cliente se encontra. Seja o conjunto S ordenado da seguinte maneira: $S = \{(0, 0), (0, 1), (0, 2), (1, 1), (1, 2), (2, 1), \dots\}$. Assim, temos a seguinte matriz geradora infinitesimal do processo:

$$\mathbf{Q}_{M/Hypo(2)/1} = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & -\alpha_1 & \mu_1 & \lambda & 0 & 0 & 0 & \dots \\ \mu_2 & 0 & -\alpha_2 & 0 & \lambda & 0 & 0 & \dots \\ 0 & 0 & 0 & -\alpha_1 & \mu_1 & \lambda & \dots & \dots \\ 0 & \mu_2 & 0 & 0 & -\alpha_2 & \mu_1 & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & -\alpha_1 & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \quad (2.14)$$

onde $\alpha_j = \lambda + \mu_j$, $j = \{1, 2\}$. Podemos agrupar as entradas da matriz representada em 2.14 de acordo com o número total de clientes no sistema, o que faz com que a estrutura da Matriz 2.14 se aproxime da estrutura mostrada em 2.13.

Segundo este agrupamento, o vetor de probabilidades π pode ser particionado em subvetores $\pi^{(i)} = (\pi_{(i,1)}, \pi_{(i,2)})$, para $i > 0$, e $\pi^{(0)} = (\pi_{(0,0)}, \pi_{(0,1)}, \pi_{(0,2)})$. Baseado nesta partição, podemos definir os subconjuntos $\mathcal{S}^{(i)}$ para $i > 0$ e $\mathcal{S}^{(0)}$ no conjunto \mathcal{S} . No caso da matriz infinita, $\mathcal{S}^{(0)}$ é definido como subconjunto de borda (*boundary portion*) e os demais subconjuntos como conjuntos repetidos (*repeating portion*).

Considerando estes subconjuntos, sejam as seguintes submatrizes do processo:

$$\begin{aligned} \mathbf{A}_0 &= \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}, & \mathbf{A}_1 &= \begin{pmatrix} -\alpha_1 & \mu_1 \\ 0 & -\alpha_2 \end{pmatrix}, \\ \mathbf{A}_2 &= \begin{pmatrix} 0 & 0 \\ \mu_2 & 0 \end{pmatrix}, & \mathbf{B}_0 &= \begin{pmatrix} -\lambda & -\lambda & 0 \\ 0 & -\alpha_1 & \mu_1 \\ \mu_2 & 0 & -\alpha_2 \end{pmatrix}, \\ \mathbf{C}_1 &= \begin{pmatrix} 0 & 0 \\ \lambda & 0 \\ 0 & \lambda \end{pmatrix}, & \mathbf{B}_1 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & \mu_2 & 0 \end{pmatrix}. \end{aligned}$$

Utilizando o conjunto de matrizes $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_0, \mathbf{C}_1, \mathbf{B}_1$, podemos reescrever a Ma-

triz 2.14 na estrutura de blocos que se repetem:

$$Q_{QBD} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{C}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{B}_1 & \mathbf{A}_1 & \mathbf{A}_0 & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad (2.15)$$

onde $\mathbf{0}$ é uma matriz nula com dimensão apropriada. A matriz A_0 representa as mudanças entre os níveis i e $i + 1$; a matriz A_1 representa mudanças entre as fases do serviço e a matriz A_2 representa as mudanças entre os níveis i e $i - 1$.

2.9 Modelos de Fluido

A pesquisa em modelos de fluido tem recebido grande atenção na literatura, desde que o modelo de pacotes é pouco escalável para representar as redes de dados atuais. A ineficiência se deve, principalmente, ao crescimento significativo na taxa de envio e serviço de informações.

A título de ilustração, consideremos a Figura 2.3 que mostra um reservatório de água que esvazia segundo uma taxa constante de $c = 1$. A entrada de água no reservatório é controlada por uma cadeia de Markov de tempo contínuo com três estados: $\{0,1,2\}$. Quando o processo está no estado 0, a taxa de entrada é $d_0 = 0$, quando está no estado 1, a taxa é $d_1 = 1$ e quando está no estado 2, a taxa é $d_2 = 2$. Assim, a variação no líquido do reservatório é de: $r_0 = d_0 - c = -1$, no estado 0 e neste caso o conteúdo do reservatório diminui com taxa constante 1; $r_1 = d_1 - c = 0$, no estado 1 e neste caso o conteúdo do reservatório permanece constante e o conteúdo do reservatório aumenta com taxa 1 quando o estado é o 2. Se a taxa de entrada é igual a -1 por um tempo longo, o reservatório pode ficar vazio, permanecendo no nível zero. Se o reservatório é infinito e a taxa de entrada é igual a 1 por um longo tempo, o conteúdo aumenta sem limites. Se o reservatório possui capacidade finita e a taxa de entrada permanece igual a 1 por um período longo, o conteúdo do reservatório vai transbordar; neste caso, o nível do reservatório permanece no seu máximo valor.

Formalmente, consideraremos uma fila de fluido modulada por um processo de Markov bidimensional $\{(X(t), CR(t)) : t \in \mathfrak{R}\}$, onde:

- $X(t)$ assume um valor pertencente ao conjunto finito \mathcal{S} e é denominado *fase*. Este

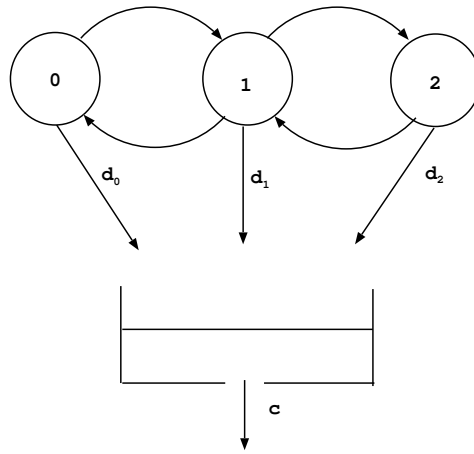


Figura 2.3: Exemplo de um reservatório de fluido modulado por um processo de Markov com três estados $\{0, 1, 2\}$.

valor é o estado, no tempo t , do processo de Markov que modula o padrão de evolução do conteúdo da fila; e,

- $CR(t) \in \mathfrak{R}$ é denominado *nível* e representa o conteúdo da fila em um instante de tempo t .

A modulação do conteúdo na fila pode ser descrita da seguinte forma: durante os intervalos de tempo em que $X(t)$ é constante e igual a i , o nível $CR(t)$ varia linearmente a uma taxa r_i . No caso da fila com capacidade infinita, $CR(t)$ pode assumir qualquer valor real não negativo; no caso da fila com capacidade finita, $CR(t)$ poderá assumir valores reais até o nível máximo B , previamente determinado. Quando $CR(t) = 0$ e a taxa r_i no tempo t é negativa, o nível permanece zero.

O uso de modelos de fluido pode resultar em vantagens computacionais tanto para o caso onde o modelo é simulado quanto para o caso em que este é resolvido analiticamente. No caso da simulação, os eventos que representam as chegadas discretas de pacotes não são explicitamente considerados, e sim, somente os eventos que representam as mudanças na taxa de chegada do fluido. O conteúdo total no dispositivo de armazenamento é obtido através de equações. Esta modelagem aproxima o efeito da chegada de pacotes através de “blocos de fluido” que preenchem o dispositivo de armazenamento a uma taxa constante de *bits*. Como consequência desta modelagem, a diminuição do custo computacional é grande quando a taxa de chegada de pacotes é muitas ordens de grandeza maior que as taxas de mudanças destes valores. Similarmente, quando resultados analíticos são utilizados, as mudanças nas taxas de fluido são modeladas, por exemplo, por uma cadeia de Markov, enquanto o fluido acumulado no dispositivo de armazenamento é representado

por um conjunto adicional de equações.

Como um exemplo, consideremos uma fonte *on-off* onde em *on* os pacotes chegam a uma fila finita segundo um processo de *Poisson* e nenhum pacote chega quando o estado é o *off*. Adicionalmente, assumamos que a fila possua capacidade para armazenar centenas de pacotes e que estes são servidos a uma taxa de 200 Mbps. A taxa de chegada de pacotes em *on* é de 1 Gbps e as mudanças de *on* para *off* ocorrem na ordem de centenas de milissegundos. Então, 15 minutos de tempo de simulação de um sistema simples de fila reduz o número de eventos a serem simulados de 10^8 para 10^3 , quando é usado um modelo de fluido. Da mesma forma, a solução analítica do sistema de fluido pode reduzir de 2 a 3–4 ordens de magnitude o custo computacional em termos de armazenamento e número total de multiplicações, respectivamente.

Liu *et al* [89] investigam o custo computacional requerido pela simulação de fluido e estabelecem uma comparação com a simulação de pacotes. As referências ([67],[25]) são exemplos de simulações do tipo híbridas, que utilizam pacotes e fluido.

Métodos analíticos para solução de modelos de fluido têm sido estudados por mais de 25 anos ([82],[5],[98],[45],[3],[4]). Na literatura podemos encontrar diversos métodos de solução para modelos Markovianos com recompensa ([26],[106],[27],[64]) que podem ser aplicados para modelos de fila de fluido, com adicional cuidado no tratamento das condições de contorno, no momento onde o dispositivo de armazenamento atinge o nível zero ou a sua capacidade máxima [32].

2.10 Cadeias de Markov Ocultas (HMM's)

Uma cadeia de Markov oculta é composta por dois processos estocásticos acoplados, onde o primeiro é uma cadeia de Markov e o segundo é um processo de observação cuja a distribuição, em qualquer instante de tempo t , é totalmente determinada pelo estado atual da cadeia. Maiores detalhes sobre este processo e as aplicações relacionadas podem ser encontrados em [119].

Por exemplo, vamos considerar um cassino desonesto que usa dois tipos de dados para um determinado jogo. Um dado é normal, ou seja, a probabilidade de sair qualquer uma de suas 6 faces é $1/6$. O outro dado é viciado de modo que a face com 6 saia 50% das vezes. Um jogador não sabe que o cassino está usando mais de um dado e vai apenas ver os resultados dos dados, ou, em outras palavras, a seqüência de observações deste

2.10 Cadeias de Markov Ocultas (HMM's)

experimento. Como o jogador não vê a troca dos dados, ele pode apenas inferir sobre o processo, cujo estado indica o tipo de dado usado, através da seqüência de observações. O processo de troca de dados é oculto para o jogador. A Figura 2.4 mostra o diagrama de estados deste modelo de Markov Oculto.

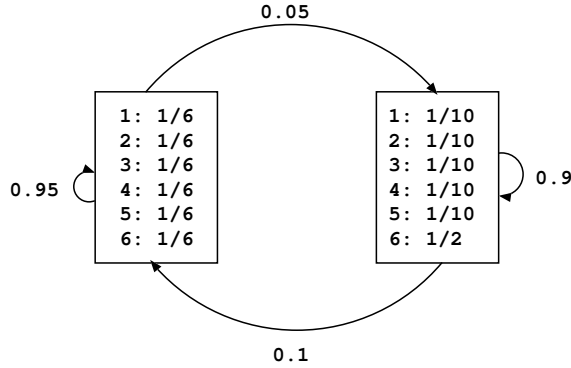


Figura 2.4: Cassino Desonesto.

Formalizando a definição de uma HMM, seja $\{X_t\}$ o n -ésimo estado da cadeia de Markov. A distribuição de probabilidades inicial é dada por um vetor n -dimensional π , com $\pi_i = P[X_1 = i]$. As probabilidades de transição entre os estados são controladas por uma matriz $\mathbf{P} = \{p_{ij}\}$, de ordem $n \times n$, onde $p_{ij} = P[X_t = j | X_{t-1} = i]$. O processo de observação $\{Y_t\}$ possui M símbolos e é governado pela matriz $\mathbf{\Gamma} = \{\gamma_{ij}\}$, de ordem $n \times m$, onde, $\gamma_{ij} = P[Y_t = j | X_t = i]$. Para cada HMM o conjunto de parâmetros relacionado é definido pela tripla $\lambda = (\pi, \mathbf{P}, \mathbf{\Gamma})$.

Descrevendo cada um dos elementos de uma HMM supracitados, temos:

1. N : número de estados do modelo

Apesar de, em geral, os estados serem ocultos, em muitas aplicações podemos associar um sentido físico a cada estado do processo \mathcal{X} . No exemplo do cassino desonesto, o analista pode criar um modelo onde cada estado representa um dado diferente com diferentes probabilidades de emissão de cada face. Os estados serão denotados por $\mathcal{S} = \{1, 2, \dots, N\}$, onde N é o número total de estados, e o estado no instante de tempo t como $\{X_t\}$.

2. M : número de símbolos distintos do modelo

Também conhecido como tamanho do alfabeto. Os símbolos são as saídas físicas do sistema sendo modelado. No exemplo do cassino desonesto, os símbolos são as faces dos dados. Os símbolos serão denotados por $\mathcal{V} = \{v_1, v_2, \dots, v_M\}$.

3. **P: matriz de probabilidade de transição**

A matriz **P** indica a probabilidade do sistema sair de um estado para outro em um passo. Ainda no exemplo do cassino, a matriz de probabilidade de transição indica a probabilidade do cassino trocar do dado normal para o dado viciado e vice-versa. A probabilidade de transição do estado i para o estado j em um passo é $p_{ij} = P[X_{t+1} = j | X_t = i]$, $1 \leq i, j \leq N$.

4. **Γ : matriz de probabilidade de emissão**

A matriz **Γ** indica a probabilidade de emissão do símbolo v_j , dado que o sistema está no estado i , onde cada elemento γ_{ij} é igual a $P[Y_t = v_j | X_t = i]$.

5. **π : vetor distribuição do estado inicial**

O vetor **π** define a probabilidade do sistema começar no estado i , onde cada elemento significa $\pi_i = P[X_1 = i]$, $1 \leq i \leq N$.

O primeiro problema ao se criar um novo modelo HMM está relacionado em como especificar um espaço de estados tanto para o processo de observação $\{Y(t)\}$ quanto para o processo oculto $\{X(t)\}$. A partir destas definições, tanto o conjunto de parâmetros, como as probabilidades de transição entre os estados do processo oculto, podem ser obtidos através de algoritmos [119].

Capítulo 3

Método de Solução para Modelos de Filas de Fluido

3.1 Introdução

Modelos estocásticos de fluido têm sido amplamente utilizados na literatura como ferramentas para a análise de sistemas de computação e comunicação. Em particular, quando as taxas dos eventos do sistema em investigação variam em muitas ordens de grandeza, o uso de modelos de fluido podem resultar em considerável redução do custo computacional quando comparado com os modelos onde todos os eventos são explicitamente representados. A redução do custo computacional está relacionada com a diminuição do total de multiplicações, no caso de modelos analíticos, e na diminuição dos eventos a serem escalonados, no caso de modelos simulados.

A característica citada no parágrafo anterior é encontrada, por exemplo, em modelos de análise de *dependability* [64]¹, onde os eventos que representam mudanças estruturais no sistema (falhas e reparos) ocorrem com taxas muito mais baixas do que as taxas dos eventos de chegada e serviço de clientes. Consideremos um modelo de filas de um canal de comunicação de redes de computadores. Os intervalos entre eventos associados com a chegada e saída de pacotes na fila podem ser muito menores do que os intervalos que representam mudanças nas taxas de chegada de pacotes.

O uso de modelos de fluido para modelar sistemas de comunicação resulta em vantagens computacionais tanto para o caso onde o modelo é simulado quanto para o caso em

¹Modelos que unem medidas de análise de desempenho e confiabilidade.

3.1 Introdução

que este é resolvido analiticamente. No caso da simulação, os eventos que representam as chegadas de pacotes não são explicitamente considerados, e sim, somente os eventos que representam as mudanças na taxa de chegada do fluido. O conteúdo total no dispositivo de armazenamento é obtido através de equações. Como consequência desta modelagem, a diminuição do custo computacional pode ser grande quando a taxa de chegada de pacotes é muitas ordens de grandeza maior que as taxas de mudanças destes valores. Similarmente, quando resultados analíticos são utilizados, as mudanças nas taxas de fluido são modeladas, por exemplo, por uma cadeia de Markov, enquanto o fluido acumulado no dispositivo de armazenamento é representado por um conjunto adicional de equações.

Um fluido pode ser representado como uma recompensa de taxa acumulada ou perda por unidade de tempo, dependendo do estado do sistema. Por exemplo, em um sistema de fila, a chegada e serviço de pacotes podem ser representados por uma recompensa associada aos estados do modelo. Em outras palavras, se o estado i indica que a taxa de chegada de pacotes é igual a λ_i , então a recompensa atribuída ao estado i é $r_i = \lambda_i - \mu$, onde μ é a capacidade de serviço no sistema.

Modelos de fluido têm atraído a atenção de muitos pesquisadores durante décadas e muitos métodos têm sido propostos para analisar estes modelos ([82], [5],[98],[45],[9],[128], [27],[88],[32]). Consideremos a medida de distribuição de probabilidades, em estado estacionário, do conteúdo de uma fila de fluido. Métodos que utilizam análise espectral são os mais comuns ([82],[5],[98],[45]), onde a medida citada anteriormente é descrita através de um conjunto de equações diferenciais cuja solução é uma combinação linear de exponenciais dos autovalores e autovetores de uma matriz. A limitação deste paradigma, principalmente no caso de filas com capacidade finita, é que os autovalores utilizados para obtenção da solução são positivos e negativos e erros numéricos podem gerar soluções instáveis, resultando em probabilidades negativas ou até mesmo em valores que podem crescer sem limites. Adicionalmente, o método exige o cálculo de autovalores e autovetores de uma matriz cujo tamanho é igual a da matriz de transição de estados, representando as mudanças de taxas. Portanto, o custo computacional pode ser proibitivo quando a dimensão da matriz de transição é grande.

Trabalhos recentes nesta área de pesquisa têm apresentado novas metodologias para a obtenção de medidas de desempenho em filas de fluido. Nabli [105] estendeu o trabalho apresentado em [106] com o objetivo de obter medidas em estado estacionário. A distribuição acumulada é calculada usando equações recorrentes.

Com uma abordagem bem diferente das tradicionais, alguns trabalhos na literatura

e que foram desenvolvidos paralelamente ao método proposto neste capítulo, transformam filas de fluido com capacidades finita e infinita em um processo estocástico equivalente ao modelo conhecido na literatura como *Quasi-Birth-Death* (QBD). Formalmente, um processo QBD infinito, em tempo discreto, pode ser descrito da seguinte maneira. Consideremos uma DTMC $\mathcal{X} = \{X(t); t \geq 0\}$ com espaço de estados bidimensional $\{(j, i) : j \geq 0, 1 \leq i \leq m\}$, com a seguinte partição definida como $\cup_{j \geq 0} l(j)$, onde $l(j) = \{(j, 1), (j, 2), \dots, (j, m)\}$ para $j \geq 0$. A primeira coordenada, j , é definida como *nível* ou conteúdo da fila, e a segunda, i , é chamada *fase* do estado (j, i) . Uma cadeia de Markov é definida como QBD se transições em um passo são restritas a estados do mesmo *nível* (j) ou entre dois *níveis* adjacentes: é possível mudanças em um passo entre (j, i) para (j', i') , somente se $j' = (j), (j + 1)$ ou $(j - 1)$, com $j \geq 1$, para o último caso. Para a instância finita de um processo QBD, a coordenada j passa a ser limitada superiormente por um valor $m \in \mathbb{N}$.

Ahn e Ramaswami [3] mostraram a relação entre modelo de fluido e processo QBD e no trabalho de Soares e Latouche em [23], os autores mostraram uma interpretação probabilística para o processo apresentado por Ahn e Ramaswami. Trabalhos recentes de Ahn e Ramaswami [4] e Soares e Latouche [22], transformam uma fila finita de fluido em um modelo QBD finito, estendendo os trabalhos apresentados em [3] e [23], respectivamente. Conforme veremos neste capítulo, as metodologias utilizadas pelos trabalhos supracitados e pelo trabalho descrito neste capítulo são diferentes no que diz respeito a própria modelagem e ao respectivo custo computacional dos métodos para calcular a distribuição complementar da ocupação da fila, apesar de representarem o comportamento do modelo de fluido através de um processo QBD.

Como principal vantagem do uso de processos QBD, podemos citar que este tipo de processo evita os problemas de precisão que acontecem na solução numérica de equações diferenciais, no caso de filas com capacidade infinita, e problemas de determinação de condições de contorno que ocorrem nos casos das filas finitas (a determinação destas condições de contorno é uma etapa difícil no cálculo dos resultados, tanto por instabilidade numérica como também pela dificuldade na identificação dos estados que obedecem a estas condições). Além dos pontos citados, certos tipos de solução de equações diferenciais são instáveis, causando consideráveis dificuldades técnicas [120].

Para obter a distribuição complementar em estado estacionário de um processo QBD finito, podemos citar o método de solução conhecido como *Folding* [142], que se caracteriza por ser um algoritmo de eliminação do tipo par-ímpar (*odd-even*), que refere-se a posição na matriz do bloco eliminado a cada passo do algoritmo. Em Latouche e

3.1 Introdução

Ramaswami [87] dois outros métodos são apresentados, sendo que o primeiro representa a distribuição através da combinação linear de duas matrizes na sua forma geométrica e o segundo requer que o processo QBD possua matrizes abaixo da diagonal com *rank* igual a 1 (como exemplo, podemos citar os processos M/PH/1/k e PH/M/1/k que satisfazem esta característica). Em Meo *et al* [99] os autores apresentam um procedimento eficiente de eliminação que é particularmente atrativo para as matrizes onde os blocos abaixo da diagonal principal possuem uma estrutura simples, de fácil inversão.

O maior objetivo deste capítulo é estudar e apresentar métodos eficientes para a solução de modelos de fluido, pela transformação do sistema em um processo QBD. Sejam CR a recompensa acumulada em uma fila, ou similarmente a quantidade de conteúdo, e x uma quantidade previamente escolhida. Iremos propor um método, baseado na utilização de processos QBD, que calcula a distribuição de probabilidades complementar da recompensa acumulada na fila, ou seja, $P[CR > x] = \lim_{t \rightarrow \infty} P[CR(t) > x]$.

Apresentaremos um método aproximado que pode ser dividido em duas contribuições principais. A primeira é a forma pela qual o modelo de fluido é transformado em um sistema QBD. Desta forma, a cadeia de Markov com recompensas de taxa em tempo contínuo é mapeada em uma cadeia equivalente em tempo discreto e com recompensas de impulso. Neste modelo discreto, “blocos de fluido” são acumulados pelo processo a cada mudança de estado da fonte de dados. O modelo resultante pode ser resolvido usando métodos para recompensas de impulso ou pode ser pensado como um processo QBD com estrutura especial.

Adicionalmente a metodologia de modelagem proposta, a segunda contribuição deste capítulo é um algoritmo de eliminação mais eficiente para o processo QBD resultante. O algoritmo possui custo computacional inferior quando comparado ao custo computacional do algoritmo de eliminação *Folding*. Este método de eliminação pode ser utilizado em qualquer modelo, independente deste estar representando ou não um modelo original de fluido. Para tal, basta que a matriz de probabilidades do processo em estudo possua uma estrutura especial, que descreveremos no decorrer deste capítulo.

Resultados preliminares da teoria desenvolvida neste capítulo foram apresentados em [19]. Neste capítulo ampliamos o escopo da utilização do método para modelos com duas filas de fluido. Adicionalmente, apresentamos resultados quando o método proposto é aplicado para avaliar a distribuição complementar, em estado estacionário, da ocupação de filas de fluido alimentadas por fontes de dados parametrizadas por traces de tráfego real coletados nos roteadores do Departamento de Engenharia de Sistemas e

Computação/COPPE/UFRJ [37] e em um *backbone* Internet [112]. Para analisarmos o desempenho do método, realizamos um conjunto de testes variando os diversos parâmetros do modelo. A aproximação proposta possui um comportamento homogêneo, conforme será verificado posteriormente, através do conjunto de exemplos. Como uma nova contribuição, aplicamos o método proposto para o caso da fila de fluido com capacidade infinita.

A estrutura deste capítulo é mostrada a seguir. Na Seção 3.2 apresentamos os principais trabalhos encontrados na literatura que propõem o cálculo da medida anteriormente citada através da utilização de processos QBD e através da abordagem “tradicional”, com o cálculo dos autovalores e dos autovetores. A Seção 3.3 é destinada a descrever o método proposto para os sistemas com uma única fila de fluido. A extensão para os sistemas com duas filas é apresentada em 3.4. O custo computacional do método está descrito na Seção 3.5. Comparações entre os métodos abordados na Seção 3.2 e o método apresentado neste capítulo são discutidas na Seção 3.6. Exemplos para ilustrar o método proposto são descritos na Seção 3.7. Apresentar o sumário com as principais contribuições deste capítulo é o objetivo da Seção 3.8.

3.2 Trabalhos Relacionados

Conforme descrito na Introdução, a literatura de fluido é vasta e diferentes abordagens para o cálculo da distribuição de probabilidades da ocupação de uma fila podem ser encontradas ([82],[5],[98],[45],[9],[128],[27],[88]).

Nesta seção iremos restringir a apresentação dos métodos cujas as idéias mais se aproximam do método proposto neste capítulo. Descreveremos o método tradicional proposto por Anick *et al* [5] e a abordagem da utilização do processo QBD proposta por Ahn e Ramaswami ([3] e [4]). Finalizando, apresentaremos o método de eliminação *Folding* proposto em [142].

3.2.1 Anick *et al* [5]

Um dos primeiros trabalhos na literatura destinado à solução da distribuição da ocupação de uma fila de fluido em estado estacionário foi desenvolvido por Anick *et al* [5]. O método proposto utiliza expansão espectral e será descrito a seguir.

3.2 Trabalhos Relacionados

Seja uma fila que recebe informação de N fontes que são independentes e que de maneira assíncrona alternam entre os estados *on* e *off*. Os períodos de *on* e de *off* de cada fonte são exponencialmente distribuídos. Sem perda de generalidade, a unidade de tempo escolhida é a média de tempo no estado *on*; a média de tempo do período em *off* é dada por $1/\lambda$ e a unidade de informação escolhida é a quantidade de informação gerada pela fonte no estado *on*. Assim, a fonte transmite uma unidade de informação por unidade de tempo em *on*. Quando i fontes estão em *on* simultaneamente, a taxa instantânea recebida pela fila é de i . A quantidade de informação que excede a taxa de serviço μ é armazenada na fila. Quando a fila não está vazia, a taxa instantânea de mudança no conteúdo da fila é de $i - \mu$. Uma vez vazia, a fila continua neste nível enquanto $i \leq \mu$. O modelo matemático para este sistema é descrito a seguir.

Seja i o total de fontes ativas (em *on*) no instante t . Dois eventos podem ocorrer durante o próximo intervalo Δt : uma nova fonte pode se tornar ativa ou uma fonte pode se tornar inativa. Desde que os períodos de *on* e *off* sejam exponencialmente distribuídos, as probabilidades de ocorrências dos eventos são, respectivamente, $(N - i)\lambda\Delta t$ e $i\Delta t$. A ocorrência de dois eventos no intervalo tem a probabilidade igual a $o(\Delta t^2)$. A probabilidade de não ocorrer mudanças é de $1 - \{(N - i)\lambda + i\}\Delta t + o(\Delta t^2)$.

Seja $P_i(t, x)$, $0 \leq i \leq N$, $t \geq 0$, $x \geq 0$, a probabilidade de i fontes estarem ativas em t e o conteúdo na fila não exceder x . Assim definimos:

$$\begin{aligned} P_i(t + \Delta t, x) = & \{N - (i - 1)\}\lambda\Delta t P_{i-1}(t, x) + \\ & (i + 1)\Delta t P_{i+1}(t, x) + [1 - \{(N - i)\lambda + i\}\Delta t] \\ & P_i\{t, x - (i - \mu)\Delta t\} + O(\Delta t^2). \end{aligned} \quad (3.1)$$

No limite, $\Delta t \rightarrow 0$:

$$\frac{\partial P_i}{\partial t} + (i - \mu)\frac{\partial P_i}{\partial x} = (N - i + 1)\lambda P_{i-1} - \{(N - i)\lambda + i\}P_i + (i + 1)P_{i+1} \quad (3.2)$$

Como o objetivo é calcular as probabilidades em equilíbrio, considere $w_i(x)$ a probabilidade em estado estacionário de i fontes estarem ativas e o conteúdo da fila não exceder x .

Como $\partial P_i / \partial t = 0$, para $t \rightarrow \infty$ e $i \in [0, N]$

$$(i - \mu)\frac{dw_i(x)}{dx} = (N - i + 1)\lambda w_{i-1}(x) - \{(N - i)\lambda + i\}w_i(x) + (i + 1)w_{i+1}(x). \quad (3.3)$$

Em notação matricial:

$$\mathbf{R}\frac{d\mathbf{w}(x)}{dx} = \mathbf{Q}\mathbf{w}(x), \quad x \geq 0, \quad (3.4)$$

3.2 Trabalhos Relacionados

onde \mathbf{R} é uma matriz diagonal com os elementos $\{-\mu, 1 - \mu, 2 - \mu, \dots, N - \mu\}$ e a matriz \mathbf{Q} é definida por:

$$\mathbf{Q} = \begin{bmatrix} -N\lambda & 1 & 0 & \cdots & 0 & 0 \\ N\lambda & -\{(N-1)\lambda + 1\} & 2 & \cdots & 0 & 0 \\ 0 & (N-1)\lambda & -\{(N-1)\lambda + 2\} & 3 & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & \vdots & \vdots \\ \vdots & \cdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \lambda & -N \end{bmatrix}.$$

Para obtenção da medida $w_i(x)$, os autovalores e autovetores a direita da matriz $\mathbf{R}^{-1}\mathbf{Q}$ devem ser calculados, o que pode resultar em problemas numéricos. Como a solução da equação 3.4 é uma combinação linear dos autovetores e dos autovalores da matriz $\mathbf{R}^{-1}\mathbf{Q}$, devemos definir uma solução em particular, através das condições de contorno inerentes ao problema em questão. Para tal, é necessário a solução de um sistema de equações lineares, com o total de constantes de integração a_i igual ao total de fases da fonte sendo modelada.

Assim, $w_i(x)$, que é a probabilidade de i fontes estarem ativas e o nível de fluido ser menor ou igual a x , é dada, em forma vetorial, pela seguinte equação:

$$\mathbf{w}(x) = \sum_{i=0}^N a_i \phi_i e^{z_i x}, \quad (3.5)$$

onde ϕ_i é o autovetor relacionado com o autovalor z_i .

Passemos a definição das condições de contorno para a Equação 3.3. Para tal, dois casos devem ser abordados:

1. Fila com capacidade infinita: neste caso, todos os autovalores positivos devem ser desconsiderados, para assegurar a convergência da Equação 3.5. Os autores em [5] provam que um dos autovalores da matriz $\mathbf{R}^{-1}\mathbf{Q}$ é o autovalor zero. Desta forma, como temos a matriz geradora infinitesimal, o autovetor associado a este autovalor é o vetor que representa a distribuição de probabilidades $\boldsymbol{\pi}$, em estado estacionário, da fonte de dados. Desta forma a Equação 3.5, se reduz a seguinte equação:

$$\mathbf{w}(x) = \boldsymbol{\pi} + \sum_{z_i < 0} a_i \phi_i e^{z_i x}. \quad (3.6)$$

Em [5], demonstra-se que o número total de autovalores negativos é igual ao total de fases da fonte onde a recompensa é positiva (*filling phases*). Este número é igual

ao total de fatores a_i a serem calculados e a seguinte condição de contorno pode ser definida:

$$w_i(0) = 0, \forall r_i > 0,$$

sendo esta explicada pelo fato de que, no caso de recompensa positiva, o conteúdo da fila sempre será maior que zero.

2. Fila com capacidade finita: para este caso, tanto os autovalores positivos quanto os negativos devem ser calculados. Da mesma forma, um dos autovetores da matriz $\mathbf{R}^{-1}\mathbf{Q}$ é o vetor $\boldsymbol{\pi}$, associado ao autovalor 0. Os coeficientes a_i são estabelecidos através das seguintes condições de contorno, considerando que o tamanho máximo da fila é igual a B^- :

$$w_i(0) = 0, \forall r_i > 0, \quad w_i(B^-) = \pi_i, \forall r_i < 0,$$

onde a primeira equação é idêntica a equação para o caso infinito, e a segunda se aplica dado que para que o tamanho da fila seja menor que B^- basta o processo estar em um estado onde a recompensa é negativa (*emptying phases*).

Passemos a análise do caso onde a fila possui ocupação total B . Para as fases da fonte onde $r_i < 0$, temos que:

$$w_i(B^-) = P[CR \leq B^-, \mathcal{X} = i] = w_i(B) = P[CR \leq B, \mathcal{X} = i] = \pi_i; \quad \forall r_i < 0.$$

Para o caso dos estados com recompensas positivas, temos que:

$$P[CR = B] = \pi_i - w_i(B^-); \quad \forall r_i > 0.$$

Em termos computacionais, é necessário encontrar, no pior caso que está relacionado com a fila com capacidade finita, os $N + 1$ autovalores z_i da matriz $\mathbf{R}^{-1}\mathbf{Q}$, os correspondentes autovetores ϕ_i e os $N + 1$ coeficientes a_i . A instabilidade numérica também atinge o cálculo dos coeficientes a_i , tornando este método não muito atrativo para matrizes com dimensão da ordem de dezenas, matrizes que não possuem estrutura especial ou matrizes mal-condicionadas [5].

3.2.2 Ahn e Ramaswami ([3] e [4])

Antes de apresentarmos o método proposto em ([3] e [4]), seja a seguinte definição formal de modelo de fluido a ser utilizada neste capítulo. Consideremos uma cadeia

de Markov em tempo contínuo (CTMC) $\mathcal{X} = \{X(t), t \geq 0\}$ com espaço de estados finito $\mathcal{S} = \{1, \dots, M\}$ e matriz geradora infinitesimal \mathbf{Q} , onde $q_i = \sum_{\forall j; j \neq i} q_{ij}$ é a taxa de saída do estado i . Seja $\boldsymbol{\pi}$ o vetor com a distribuição de probabilidades em estado estacionário, resultante da solução do sistema linear $\boldsymbol{\pi}\mathbf{Q} = 0$. A recompensa de taxa², ou semelhantemente o fluido, r_i é associada ao estado i . A recompensa r_i é acumulada por unidade de tempo em que a cadeia permanece no estado i . Seja $CR(t)$ a recompensa de taxa acumulada durante o intervalo $[0, t]$. Logo:

$$CR(t) = \int_0^t r_{X(\tau)} d\tau.$$

Seja $0 \leq CR(t) \leq B$. Estamos interessados na medida $\mathbf{w}(x) = P[CR > x] = \lim_{t \rightarrow \infty} P[CR(t) > x]$. No caso de fila infinita, $CR(t)$ não possui limite superior, ou seja, $B \rightarrow \infty$.

O método encontrado na literatura que mais se aproxima do método proposto neste capítulo é o método desenvolvido por Ahn e Ramaswami [3]. Naquele trabalho, os autores apresentam o cálculo da distribuição de probabilidades em estado estacionário para uma fila de fluido com capacidade infinita, através de um modelo QBD infinito. Em [4], os autores usam o mesmo paradigma apresentado em [3] para obtenção da mesma medida para o caso da fila com capacidade finita.

Como o caso da fila com capacidade finita é baseada no caso da fila com capacidade infinita, apresentaremos as idéias principais do trabalho para filas com capacidade infinita e ao final da seção estenderemos para o caso da fila com capacidade finita.

Consideremos que o processo \mathcal{X} , com matriz geradora infinitesimal \mathbf{Q} , seja uniformizado, seguindo a teoria apresentada no Capítulo 2. Consideremos, adicionalmente, dois processos de *Poisson*, \mathcal{M} e \mathcal{N} , independentes, com taxa λ , e seja $\{\tau_k : k \geq 0\}$ os instantes de tempo que representam épocas de chegada da superposição dos processos \mathcal{M} e \mathcal{N} . Para o caso infinito, é possível obter o resultado da medida de interesse através de um modelo equivalente ao original, denominado modelo de fluido homogêneo, em que as recompensas de taxa possuem valores iguais a 1 e -1 e matriz geradora infinitesimal $\mathbf{Q}' = \mathbf{R}^{-1}\mathbf{Q}$, onde a matriz \mathbf{R} é uma matriz diagonal com os valores absolutos das recompensas originais atribuídas ao processo que modela a fonte de dados. O conjunto \mathcal{S} que descreve os estados da fonte que modulam o fluido na fila é particionado em subconjuntos de acordo com a recompensa atribuída a cada estado: os subconjuntos \mathcal{S}^+ , \mathcal{S}^- e \mathcal{S}^0

²Conforme citado anteriormente, um modelo de fluido pode ser representado por um modelo Markoviano com recompensas de taxa.

incluem todos os estados onde são encontradas recompensas positivas, negativas e nulas, respectivamente.

O comportamento de CR , que define o estado da fila de fluido, é descrito a seguir. CR aumenta a uma taxa constante igual a 1 quando o processo \mathcal{X} está em \mathcal{S}^+ ; diminui a uma taxa constante igual a -1 quando o processo está em \mathcal{S}^- ou permanece constante quando o processo \mathcal{X} está em \mathcal{S}^0 . Seja $\mathcal{Q} = \{Q(t) : t \geq 0\}$ um processo que representa uma fila tal que $Q(t)$ é incrementado ou decrementado em τ_k , dependendo do estado onde o processo uniformizado \mathcal{X} se encontra no instante k .

O principal ponto do método proposto por Ahn e Ramaswami está relacionado com a distribuição de probabilidades de dois processos: o original, que representa o modelo de fluido e o que representa \mathcal{Q} nos pontos embutidos τ_k .

A seguir, apresentamos uma nova interpretação para a metodologia de [3]. Consideremos o processo \mathcal{Q} . Como o processo está uniformizado, a cada transição (que ocorre com uma taxa de uniformização λ), devemos considerar que, se o processo está no subconjunto \mathcal{S}^+ , um pacote chega a fila e este tem um tamanho com distribuição exponencial e média igual a uma unidade. Este fato é verdadeiro, dado que o processo original foi modificado de tal maneira que todas as recompensas positivas são iguais a 1. Adicionalmente, enquanto o processo permanece em \mathcal{S}^+ , a recompensa acumulada, ou trabalho no sistema, aumenta com uma variável aleatória que tem distribuição exponencial com média $1/\lambda$, independente da fase em que o processo se encontra.

Consideremos, agora, que o processo entre em um estado pertencente ao subconjunto \mathcal{S}^- . Neste caso, dois eventos exponenciais competem entre si: um pacote pode ser servido com taxa exponencial λ ou uma transição no processo \mathcal{X} pode ocorrer (o tempo entre a ocorrência de transições é exponencialmente distribuído, com taxa λ devido ao processo de uniformização). Se o evento “serviço de pacotes” ocorre primeiro, então a fila é decrementada. Todavia, se o evento de “transição de uma fase da fonte” ocorre, o tamanho da fila permanece constante. Podemos notar que, quando o processo volta ao estado em \mathcal{S}^- a “quantidade” do pacote a ser servido é o restante do tamanho do pacote que teve o serviço iniciado na visita anterior a um estado pertencente ao subconjunto \mathcal{S}^- . Todavia, como o tamanho do pacote tem distribuição exponencial, o restante do serviço a ser executado é também exponencialmente distribuído com taxa λ . Desta forma, o evento de serviço é exponencialmente distribuído, não importando a quantidade de serviço realizada anteriormente. Assim, como temos dois processos exponencialmente distribuídos com a mesma taxa e competindo entre si, podemos dizer que o tamanho da fila é decrementada com

probabilidade 0.5, após uma visita a um estado pertencente a \mathcal{S}^- .

Resumindo, um processo QBD é construído a partir do modelo original, após uniformização, introduzindo uma fila que é aumentada de tamanho a cada unidade de tempo que o processo deixa um estado em \mathcal{S}^+ . O crescimento da fila acontece com probabilidade 1, dado que a construção do QBD acopla a saída de um estado pertencente ao subconjunto \mathcal{S}^+ a chegada de um pacote na fila. Após uma visita a um estado em \mathcal{S}^- , a fila é decrementada com probabilidade 0.5. Nos pontos de transição τ_k , o processo uniformizado e o processo original são estocasticamente idênticos. O processo embutido é resolvido usando a teoria de filas QBD com capacidade infinita [87].

O processo QBD utilizado fornece resultados para descrever o trabalho, ou a quantidade de fluido no sistema, nos pontos embutidos τ_k . Por exemplo, se o processo QBD se encontra em um estado cujo o nível da fila é igual a 2, isto significa que a quantidade de fluido na fila é exponencialmente distribuída com taxa 2λ . Em [3] é mostrado que:

$$\mathbf{w}_{\mathcal{S}^+}(x) = -\boldsymbol{\pi}_+ \mathbf{K} e^{\mathbf{K}x}, \quad (3.7)$$

$$\mathbf{w}_{\mathcal{S}^-}(x) = \mathbf{w}_{\mathcal{S}^+}(x) \boldsymbol{\Psi}, \quad (3.8)$$

onde o vetor $\mathbf{w}_{\mathcal{S}^+}(x)$ ($\mathbf{w}_{\mathcal{S}^-}(x)$) inclui todas as entradas $\mathbf{w}(x)$ correspondentes aos estados associados com as recompensas positivas (negativas). As matrizes \mathbf{K} e $\boldsymbol{\Psi}$ são obtidas da matriz \mathbf{R} , conhecida como *rate matrix*, que é a mínima solução não-negativa da seguinte equação:

$$\mathbf{R} = \mathbf{P}_0 + \mathbf{R}\mathbf{P}_1 + \mathbf{R}^2\mathbf{P}_2, \quad (3.9)$$

sendo as matrizes \mathbf{P}_i obtidas a partir da matriz \mathbf{P} , resultante do procedimento de uniformização do processo \mathcal{X} . O vetor $\mathbf{w}_{\mathcal{S}^0}(x)$ é calculado a partir dos vetores $\mathbf{w}_{\mathcal{S}^+}(x)$ e $\mathbf{w}_{\mathcal{S}^-}(x)$.

Cada elemento na matriz \mathbf{R} representa o número esperado de visitas ao nível (n), começando a partir do nível ($n - 1$), antes do primeiro retorno ao nível ($n - 1$). Como utilizamos para a matriz de recompensas a mesma letra representativa (\mathbf{R}), iremos explicitamente definir a que matriz nos referimos, quando necessário.

Em [23], Soares e Latouche apresentaram uma interpretação probabilística para o conjunto de matrizes utilizadas em [3]. Os autores mostraram que a matriz $\boldsymbol{\Psi}$ é a matriz que representa a probabilidade de alcançar, pela primeira vez, o nível zero da fila representada no processo QBD.

Em [4], os autores estendem a metodologia apresentada em [3] para analisar modelos de fluido com fila finita, com tamanho máximo B . Eles representam o limite determinístico B através da soma de uma seqüência de variáveis aleatórias exponenciais que

convergem, no limite, para o tamanho B . A modelagem se baseia nos mesmos processos \mathcal{N} e \mathcal{M} citados anteriormente e várias instâncias destes processos são consideradas, fornecendo a granularidade necessária para que, no limite, a soma definida anteriormente, convirja para B . A fila resultante do processo de discretização é analisada através da combinação linear de matrizes geométricas [62].

Os métodos apresentados para filas com capacidade infinita [3] e com capacidade finita [4] não podem ser facilmente estendidos para um sistema com duas filas de fluido. A definição da matriz $\mathbf{Q}' = \mathbf{R}^{-1}\mathbf{Q}$ baseia-se no fato da existência de uma única recompensa atribuída a cada estado. No entanto, nos casos com duas filas, podemos ter estados em que uma fila recebe uma recompensa de valor r_1 diferente do valor de recompensa, r_2 , associado a outra fila, inviabilizando a aplicação direta dos métodos apresentados em [3] e [4].

3.2.3 Algoritmo *Folding* [142]

Na literatura podem ser encontrados vários métodos de solução que exploram a estrutura especial do processo QBD objetivando a diminuição do custo computacional para a obtenção da distribuição de probabilidades do processo em estado estacionário.

Para o caso de filas infinitas, a distribuição em estado estacionário do processo QBD pode ser escrita na forma *matrix-geometric*, onde o fator geométrico é obtido através da equação 3.9 [108]. Para um processo QBD finito, a solução é escrita em uma forma mais complexa, definida como *matrix-geometric* modificada [62]. A vantagem desta abordagem é que o custo computacional é independente do total de níveis da fila, e, mais importante, que a distribuição, em estado estacionário, de um determinado nível somente depende da distribuição de probabilidade no nível zero e da matriz *rate matrix* \mathbf{R}^3 do processo original e uma outra *rate matrix* $\hat{\mathbf{R}}$ para o processo reverso (maiores detalhes em [4] e [87]). Em contrapartida, o custo computacional está atrelado ao valor do máximo autovalor da matriz \mathbf{R} e da matriz $\hat{\mathbf{R}}$, cujo valor absoluto não é conhecido previamente⁴.

Exceto para alguns casos especiais [108], a solução para um QBD finito pode ser drasticamente simplificada, através de métodos de eliminação. O algoritmo *Folding* proposto

³Esta característica é utilizada para o caso de fila finita apresentada em [4].

⁴Em [110], Neuts mostra que a convergência dos métodos iterativos para o cálculo da matriz \mathbf{R} está relacionada com o comportamento da cauda da distribuição de probabilidades do processo QBD. Em [87], os autores mostram que o máximo autovalor da matriz \mathbf{R} está relacionado com o comportamento dinâmico do processo QBD.

em [142], reduz a complexidade computacional da ordem linear em função do total de níveis da fila para a ordem logarítmica em função do mesmo parâmetro. O algoritmo *Folding* pode ser dividido em duas fases e um passo intermediário, na seguinte ordem: primeiramente a fase de redução, em segundo um passo intermediário para a determinação da distribuição de probabilidades dos estados pertencentes a submatriz de borda, e, finalmente, a fase de expansão.

Na fase de redução, a matriz \mathbf{Q} , originalmente com K níveis, é reduzida a uma matriz de um único nível, através de sucessivos passos de redução. A cada passo uma matriz “filha” é produzida, com o tamanho igual a metade do tamanho da matriz “mãe” de acordo com a seguinte equação:

$$\mathbf{Q}_t^* = \mathbf{Q}_t + \mathbf{Q}_{ts}(-\mathbf{Q}_s)^{-1}\mathbf{Q}_{st},$$

onde as submatrizes \mathbf{Q}_t , \mathbf{Q}_{ts} , \mathbf{Q}_s , \mathbf{Q}_{st} são definidas através da seguinte partição da matriz geradora infinitesimal \mathbf{Q} :

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_s & \mathbf{Q}_{st} \\ \mathbf{Q}_{ts} & \mathbf{Q}_t \end{pmatrix}.$$

Uma propriedade do processo de redução é que através de uma permutação da matriz “mãe”, a matriz “filha” preserva a mesma estrutura QBD. Ao final do procedimento, a matriz original \mathbf{Q} possui somente um bloco de estados, e a distribuição de probabilidades pode ser calculada utilizando qualquer método de solução tradicional exata, como o GTH [139], ou iterativa, como o SOR, Gauss-Seidel ou Jacobi [132]. O método proposto neste capítulo também preserva a estrutura QBD em cada passo da eliminação, sem a necessidade de permutações auxiliares.

A fase de expansão é o processo reverso da fase de redução. Esta fase segue exatamente a ordem reversa na qual a seqüência de matrizes “filhas” foram geradas. A observação chave a ser feita é que a solução de cada matriz “filha” produz a metade das soluções da matriz “mãe”. A outra metade da solução é calculada da seguinte maneira:

$$\pi_s = -\pi_t \mathbf{Q}_{ts} \mathbf{Q}_s^{-1}.$$

A cada passo do algoritmo, o total de estados cuja a distribuição de probabilidades torna-se conhecida dobra de tamanho.

3.3 Método Proposto para uma única Fila de Fluido

Nesta seção apresentaremos um novo método para obtenção da distribuição complementar de probabilidades, em estado estacionário, da quantidade de fluido armazenado em uma fila finita. Em linhas gerais, o processo conjunto da fonte e da fila de fluido é modelado através de um processo Markoviano com recompensas de impulso. O modelo de fluido proposto é um processo do tipo QBD finito, onde a matriz de probabilidades \mathbf{P} possui uma estrutura especial. Um novo algoritmo de eliminação foi elaborado para resolver a matriz com estrutura especial obtida na modelagem proposta.

Consideremos uma cadeia de Markov homogênea de tempo contínuo (CTMC) $\mathcal{X} = \{X(t), t \geq 0\}$ com espaço de estados finito $\mathcal{S} = \{1, \dots, M\}$ e gerador infinitesimal \mathbf{Q} , onde $q_i = -\sum_{j; j \neq i} q_{ij}$ é a taxa de saída do estado i . A cada estado i está associada uma recompensa de taxa (ou fluido) $r_i = \lambda_i - \mu$ (sendo λ_i a taxa de chegada de informação no estado i e μ a taxa de serviço do sistema), onde esta recompensa é acumulada por cada unidade de tempo passada no estado i . Seja $CR(t)$ a recompensa de taxa acumulada durante o intervalo de tempo $[0, t]$. A variável aleatória $CR(t)$ está limitada entre 0 e o tamanho total da fila de fluido B . A medida de interesse a ser calculada é $P[CR > x] = \lim_{t \rightarrow \infty} P[CR(t) > x]$.

Para obtenção do modelo aproximado, alguns parâmetros devem ser definidos. Seja γ um parâmetro que satisfaça a seguinte relação:

$$\gamma \leq \gamma_{max} = \frac{|r_l|}{q_l} \text{ tal que } l = \operatorname{argmin}_i \left\{ \frac{|r_i|}{q_i} \right\}. \quad (3.10)$$

Para cada estado i do modelo, a taxa de saída passa a ser definida por $\Lambda_i = |r_i/\gamma|$. Conforme podemos notar, $\Lambda_i \geq q_i$. Iremos transformar o modelo de fluido definido no parágrafo anterior, ou equivalentemente o modelo de Markov com recompensa de taxa, em uma cadeia de Markov de tempo discreto com estrutura especial.

Seja \mathcal{Y} uma CTMC com o mesmo espaço de estados do processo \mathcal{X} definida da seguinte maneira. A cada estado i de \mathcal{Y} associamos um evento ϱ_i que ocorre com taxa Λ_i ao final de um intervalo exponencialmente distribuído. Seja p_{ij} a probabilidade de transição entre os estados i e j quando o evento ϱ_i ocorre. Definimos p_{ij} igual a q_{ij}/Λ_i . Apesar dos processos \mathcal{X} e \mathcal{Y} serem diferentes, a distribuição de probabilidades em estado estacionário do processo \mathcal{X} pode ser obtida através da solução do processo \mathcal{Y} .

Seja δ_i a variável aleatória que representa a quantidade de fluido atribuída ao processo \mathcal{Y} a cada visita ao estado i . Se $CR(t)$ é ilimitada, isto é, se não existe limite para o total

3.3 Método Proposto para uma única Fila de Fluido

de recompensa acumulada pelo sistema, $|E[\delta_i]| = \gamma$, independente do estado i .

A aproximação proposta assume que o total de recompensa atribuída ao processo, quando ocorre uma transição entre os estados i e j no processo \mathcal{Y} , é igual ao valor médio da recompensa acumulada no processo original. Em outras palavras, dada a transformação do processo \mathcal{X} em \mathcal{Y} , o módulo do valor esperado do fluido recebido a cada visita a um estado i é igual a γ , independente do estado i . A recompensa a ser acumulada pode ser positiva ou negativa, dependendo do estado sendo visitado. No entanto, o valor absoluto é independente do estado, devido a escolha dos parâmetros γ e Λ_i . A idéia principal da aproximação é assumir que o processo ganha uma recompensa de impulso de módulo igual a γ a cada visita a um dos estados. O parâmetro γ é chamado de “bloco de fluido”, ou *batch fluid*.

A partir do processo \mathcal{Y} , podemos construir uma cadeia de Markov discreta (DTMC) \mathcal{Y}^* . O espaço de estados \mathcal{S}^* de \mathcal{Y}^* é tal que $\mathcal{S}^* = \{(i, l) : 1 \leq i \leq M, 0 \leq l \leq \lfloor B/\gamma \rfloor\}$ e para cada estado do processo \mathcal{Y} existem $\lfloor B/\gamma \rfloor + 1$ estados em \mathcal{Y}^* .

A matriz de probabilidades $\mathbf{P}^* = [p_{ij,kl}^*]$ é obtida da seguinte maneira: se $r_i > 0$, então $p_{ij,kl}^* = p_{ij}$, para $l = \min\{k + 1, B\}$; se $r_i < 0$, então $p_{ij,kl}^* = p_{ij}$, para $l = \max\{k - 1, 0\}$; e 0 caso contrário. A partir dos valores de $p_{ij,kl}^*$ é fácil ver que, a cada transição de \mathcal{Y} , o processo \mathcal{Y}^* aumenta ou diminui a recompensa acumulada do valor do *batch fluid*. A matriz \mathbf{P}^* obtida apresenta estrutura semelhante a de um processo QBD conforme apresentada na Figura 3.1.

A matriz \mathbf{P}^* relacionada ao processo QBD possui uma estrutura especial. As submatrizes na diagonal são zeradas, exceto a primeira e a última. Esta característica ocorre devido a modelagem proposta: atrelada a mudança de fases, existe sempre o evento aumentar/diminuir o conteúdo da fila de fluido. O aumento ocorre quando a fase da fonte muda a partir de um estado com recompensa positiva, e a diminuição ocorre nos casos complementares (conforme será descrito posteriormente, juntamente com o algoritmo proposto, basearemos a apresentação do método proposto para modelos com recompensas diferentes de zero, sendo o caso igual a zero de fácil extensão).

Com base na estrutura particular da matriz \mathbf{P} , iremos propor um algoritmo computacionalmente mais vantajoso, para a classe de modelos com esta estrutura em particular, que outras propostas encontradas na literatura ([142],[139],[99]) para o cálculo da solução $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$. Ressaltamos que este algoritmo pode ser usado para obter a solução, em estado estacionário, de uma cadeia de Markov, cuja a matriz de probabilidades \mathbf{P} tenha a estrutura da Figura 3.2. Portanto, o algoritmo proposto pode ser usado para resolver outros

3.3 Método Proposto para uma única Fila de Fluido

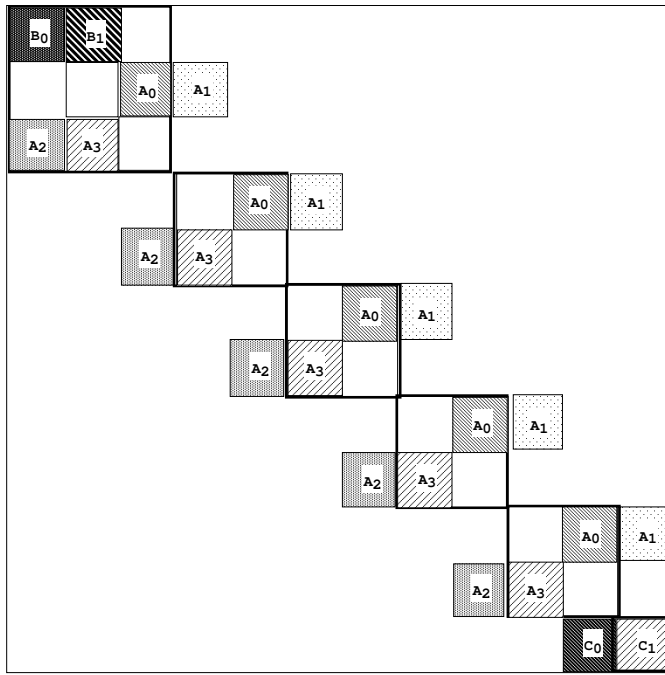


Figura 3.1: Matriz P^* do processo \mathcal{Y}^* com as respectivas submatrizes. Os blocos com o mesmo índice representam submatrizes idênticas.

problemas, além de fazer parte da solução de filas de fluido aqui proposta. Relembramos que solucionar a equação $\pi = \pi P$, onde P possui estrutura especial, é o objetivo de vários trabalhos (ver uma série de referências em [87]).

Conforme descrito na Seção 3.2.3, o algoritmo *Folding* [142] foi proposto para obter soluções, em estado estacionário, de modelos QBD com espaço de estados finito. Em linhas gerais, este algoritmo realiza uma permutação denominada *par-ímpar* na matriz de probabilidades original. A matriz resultante desta permutação possui dois blocos diagonais de tamanhos idênticos. A cada passo de redução, um dos blocos é eliminado. A matriz resultante passa a ter a metade da dimensão da matriz original, sendo que a estrutura QBD permanece inalterada. O algoritmo prossegue eliminando a metade dos estados a cada passo da redução. Ao final, as probabilidades são calculadas através das equações de balanceamento de fluxo. A ordem de complexidade do custo computacional deste algoritmo é de $O(\lceil \log_2 B \rceil M^3)$, sendo o parâmetro B o total de níveis na fila e M o total de fases da fonte modelada.

O algoritmo proposto neste trabalho, denominaremos *Algoritmo Bloco Diagonal Zero* dado que pela modelagem proposta os sub-blocos diagonais são zerados. Assim, como o algoritmo *Folding*, é um método de eliminação de blocos [87] com custo computacional logarítmico obtido pela eliminação simultânea de conjuntos de blocos, mas que leva em

3.3 Método Proposto para uma única Fila de Fluido

consideração a estrutura particular do modelo obtido. Primeiramente, o espaço de estados do processo \mathcal{Y}^* é particionado em dois subconjuntos \mathcal{S}_l^+ e \mathcal{S}_l^- tal que o estado $(i, l) \in \mathcal{S}^+$, se e somente se, $r_{(i,l)} > 0$ e $(i, l) \in \mathcal{S}^-$, se e somente se, $r_{(i,l)} < 0$. Assumiremos que todas as recompensas de taxa são diferentes de zero. No entanto, a extensão para modelos com recompensas nulas ($r_i = 0$) é simples. Para tal, basta agruparmos estes casos no conjunto de estados que serão eliminados na etapa (b) do algoritmo de eliminação, ou seja, podemos definir subconjuntos \mathcal{S}_l^+ , com $r_{(i,l)} > 0$ ou \mathcal{S}_l^- com $r_{(i,l)} < 0$.

A Figura 3.2 mostra o particionamento dos estados. As matrizes \mathbf{A}_i , com $i = \{0, 1, 2, 3\}$, representam as probabilidades de transição entre os subconjuntos, \mathcal{S}_l^+ e \mathcal{S}_l^- .

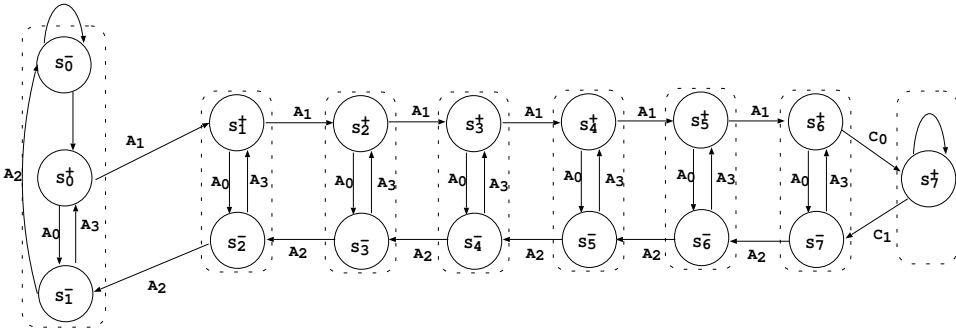


Figura 3.2: Diagrama do particionamento dos estados do processo \mathcal{Y}^* .

Definiremos uma nova partição que engloba o particionamento anterior. Sejam os seguintes blocos de estados $\mathcal{L}_0 = \mathcal{S}_0^- \cup \mathcal{S}_0^+ \cup \mathcal{S}_1^-$, $\mathcal{L}_l = \mathcal{S}_l^+ \cup \mathcal{S}_{l+1}^-$ para $l > 0$, $\mathcal{L}_L = \mathcal{S}_L^+$ onde $L = \lfloor B/\gamma \rfloor$. A Figura 3.3 mostra a matriz \mathbf{P}^* particionada nos sub-blocos $\mathcal{S}_l^{(+/-)}$ e nos blocos de estados \mathcal{L}_l .

A estrutura da matriz apresentada na Figura 3.1 claramente pode ser resolvida através do algoritmo *Folding* [142]. Para tal, basta redefinirmos \mathcal{L}_l como $\mathcal{L}_l = \mathcal{S}_l^+ \cup \mathcal{S}_l^-$, para $l \geq 0$. Neste caso, a cada passo da redução do algoritmo *Folding*, uma nova matriz diagonal deve ser invertida, com dimensão igual ao número total de fases da fonte, como consequência do preenchimento das matrizes diagonais que originalmente eram nulas. Organizar a matriz conforme mostrado na Figura 3.1, faz com que o custo computacional diminua, principalmente devido a presença de submatrizes diagonais nulas. Esta característica faz com que apenas uma matriz com a metade da dimensão da matriz invertida pelo algoritmo *Folding* seja invertida a cada passo quando o algoritmo proposto é utilizado.

O algoritmo proposto pode ser descrito da seguinte maneira: a cada passo todos os blocos \mathcal{L}_l , com l ímpar são eliminados (com exceção dos blocos limites). Cada passo

3.3 Método Proposto para uma única Fila de Fluido

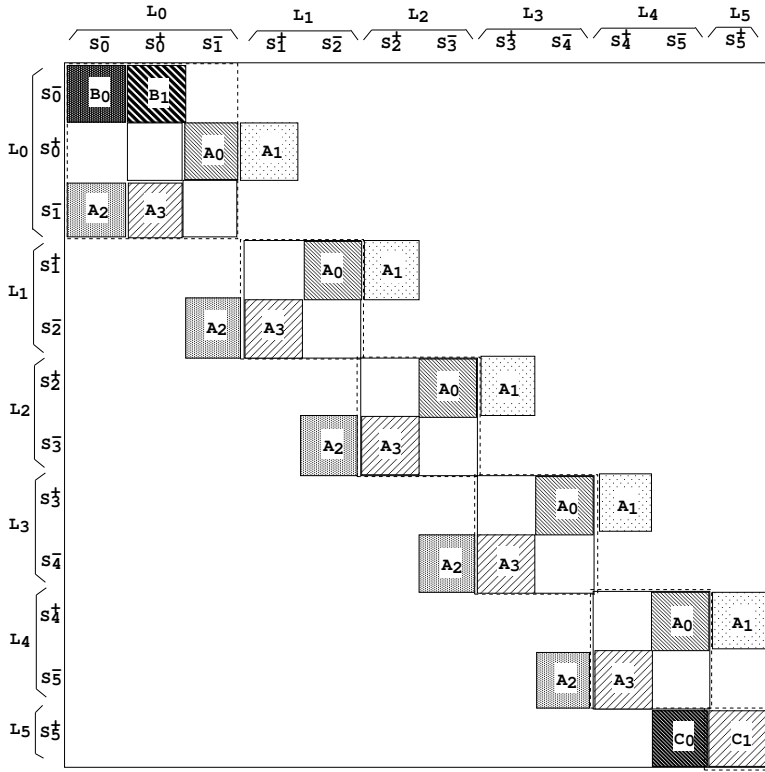


Figura 3.3: Matriz P^* e a representação do segundo nível de particionamento.

se divide em duas etapas que manipulam os subconjuntos S_l^+ e S_{l+1}^- separadamente. Na etapa (a) o subconjunto S_l^+ é eliminado, sem que haja necessidade de calcular inversas (a matriz em evidência é nula). Na etapa (b) o subconjunto S_{l+1}^- é eliminado e o cálculo de uma inversa é realizado. Esta inversão é necessária dado ao preenchimento da matriz, decorrente da redistribuição das probabilidades de entrada e saída do sub-bloco S_l^+ . As Figuras 3.4 e 3.5 apresentam as etapas (a) e (b) de um passo do algoritmo proposto. A matriz resultante após uma passo da eliminação preserva a estrutura QBD. Todo procedimento é repetido até a eliminação de todos os blocos exceto o primeiro e o último.

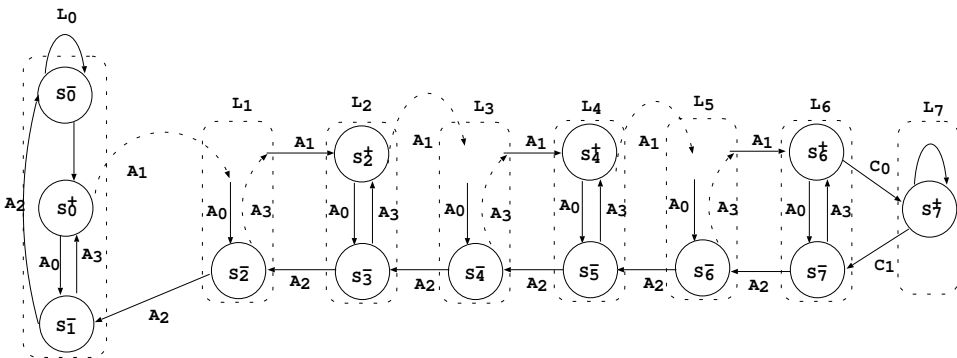


Figura 3.4: Etapa (a): Eliminação do subconjunto S_l^+ .

3.3 Método Proposto para uma única Fila de Fluido

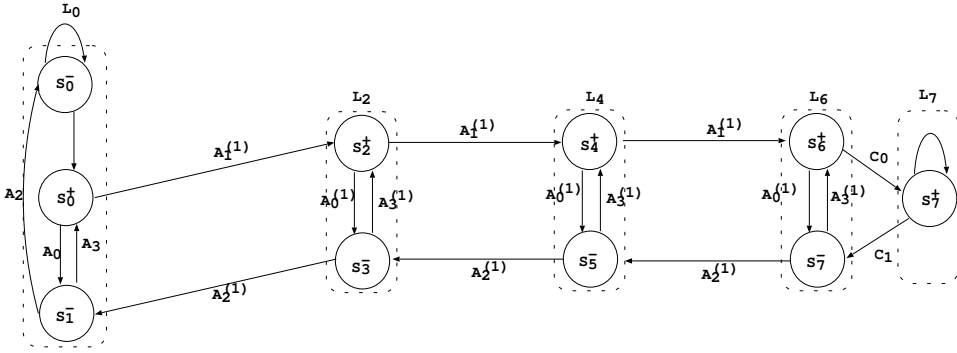


Figura 3.5: Etapa (b): Eliminação do subconjunto \mathcal{S}_{l+1}^- .

Combinando as etapas (a) e (b) teremos no final de cada etapa i , $i \leq \log_2(B/\gamma)$, as seguintes matrizes modificadas:

$$\begin{aligned} \mathbf{A}_0^{(i)} &= \mathbf{A}_0^{(i-1)} + \mathbf{A}_1^{(i-1)} \mathbf{A}_0^{(i-1)} (\mathbf{I} - \mathbf{A}_3^{(i-1)} \mathbf{A}_0^{(i-1)})^{-1} \mathbf{A}_2^{(i-1)} \\ \mathbf{A}_1^{(i)} &= (\mathbf{A}_1^{(i-1)})^2 + \mathbf{A}_1^{(i-1)} \mathbf{A}_0^{(i-1)} (\mathbf{I} - \mathbf{A}_3^{(i-1)} \mathbf{A}_0^{(i-1)})^{-1} \mathbf{A}_3^{(i-1)} \mathbf{A}_1^{(i-1)} \\ \mathbf{A}_2^{(i)} &= \mathbf{A}_2^{(i-1)} (\mathbf{I} - \mathbf{A}_3^{(i-1)} \mathbf{A}_0^{(i-1)})^{-1} \mathbf{A}_2^{(i-1)} \\ \mathbf{A}_3^{(i)} &= \mathbf{A}_3^{(i-1)} + \mathbf{A}_2^{(i-1)} (\mathbf{I} - \mathbf{A}_3^{(i-1)} \mathbf{A}_0^{(i-1)})^{-1} \mathbf{A}_3^{(i-1)} \mathbf{A}_1^{(i-1)} \end{aligned}$$

Consideremos, um caso em que o total de partições da matriz \mathbf{P}^* seja igual a 16. O diagrama ilustrando os blocos eliminados em cada passo do algoritmo pode ser visualizado na Figura 3.6, sendo que cada passo é formado pelas etapas (a) e (b). Podemos observar que os blocos limites $\mathcal{L}_0, \mathcal{L}_{15}$ permanecem durante o processo de eliminação, sendo utilizados no passo final para obtenção das probabilidades relacionadas aos estados pertencentes a estes blocos.

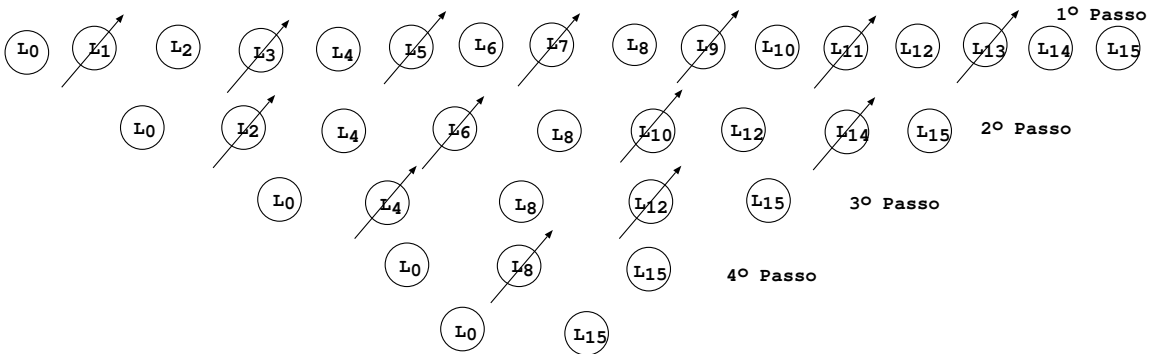


Figura 3.6: Diagrama ilustrando a eliminação dos blocos em cada passo do algoritmo.

O algoritmo pode ser resumido da seguinte maneira:

ALGORITMO BLOCO DIAGONAL ZERO ()

1 $i \leftarrow 1$

3.3 Método Proposto para uma única Fila de Fluido

```

2   $\mathbf{P}^{(0)} \leftarrow \mathbf{P}$ 
3  Calcule a matriz  $\mathbf{P}^{(i)}$  usando a matriz  $\mathbf{P}^{(i-1)}$ . Cada matriz  $\mathbf{P}^{(i)}$  é definida a partir das seguintes equações
4   $\mathbf{A}_0^{(i)} = \mathbf{A}_0^{(i-1)} + \mathbf{A}_1^{(i-1)} \mathbf{A}_0^{(i-1)} (\mathbf{I} - \mathbf{A}_3^{(i-1)} \mathbf{A}_0^{(i-1)})^{-1} \mathbf{A}_2^{(i-1)}$ 
5   $\mathbf{A}_1^{(i)} = (\mathbf{A}_1^{(i-1)})^2 + \mathbf{A}_1^{(i-1)} \mathbf{A}_0^{(i-1)} (\mathbf{I} - \mathbf{A}_3^{(i-1)} \mathbf{A}_0^{(i-1)})^{-1} \mathbf{A}_3^{(i-1)} \mathbf{A}_1^{(i-1)}$ 
6   $\mathbf{A}_2^{(i)} = \mathbf{A}_2^{(i-1)} (\mathbf{I} - \mathbf{A}_3^{(i-1)} \mathbf{A}_0^{(i-1)})^{-1} \mathbf{A}_2^{(i-1)}$ 
7   $\mathbf{A}_3^{(i)} = \mathbf{A}_3^{(i-1)} + \mathbf{A}_2^{(i-1)} (\mathbf{I} - \mathbf{A}_3^{(i-1)} \mathbf{A}_0^{(i-1)})^{-1} \mathbf{A}_3^{(i-1)} \mathbf{A}_1^{(i-1)}$ 
8   $i \leftarrow i + 1$ 
9  if  $i \leq \lfloor \log_2(B/\gamma) \rfloor$  then
10   return 2
11 endif
12 Calcule a distribuição de  $\mathcal{L}_0$  e  $\mathcal{L}_i$  com  $i = \lfloor \log_2(B/\gamma) \rfloor$ 
13  $\pi_{\mathcal{L}_i}^- = (\pi_{\mathcal{L}_0}^+ A_1^i + \pi_{\mathcal{L}_{B/\gamma}} A_2^i) (A_3^i A_2^i)^{-1}$ 
14  $\pi_{\mathcal{L}_i}^+ = (\pi_{\mathcal{L}_0}^+ A_1^{(i-1)} + \pi_{\mathcal{L}_i}^- A_3^{(i-1)}) (A_1^{(i-1)} A_0^{(i-1)})^{-1}$ 
15 onde  $i = \lceil \log_2(B/\gamma) \rceil$  e  $\mathcal{L}_i$  é o último bloco eliminado.
16 for  $i \leftarrow \lfloor \log_2(B/\gamma) \rfloor$  to  $i = 1$  do
17   Calcule as probabilidades para os subvetores de cada um dos blocos eliminados na iteração  $i$ .
18   if  $(l + 2^{i-1} = B/\gamma)$  then  $\pi_{\mathcal{L}_l}^- = (\pi_{\mathcal{L}_{l-2^{i-1}}}^+ A_1^i + \pi_{\mathcal{L}_{B/\gamma}}^+ A_2^i) (A_3^i A_2^i)^{-1}$ 
19   else  $\pi_{\mathcal{L}_l}^- = (\pi_{\mathcal{L}_{l-2^{i-1}}}^+ A_1^i + \pi_{\mathcal{L}_{l+2^{i-1}}}^- A_2^i) (A_3^i A_2^i)^{-1}$ 
20   endif
21    $\pi_{\mathcal{L}_l}^+ = (\pi_{\mathcal{L}_{l-2^{i-1}}}^+ A_1^{(i-1)} + \pi_{\mathcal{L}_l}^- A_3^{(i-1)}) (A_1^{(i-1)} A_0^{(i-1)})^{-1}$ 
22    $i \leftarrow i - 1$ 
23 endfor
24 Normalize o vetor  $\pi$ .
25 for  $k \leftarrow 1$  to  $k = \lfloor \log_2(B/\gamma) \rfloor$  do
26   for  $j \leftarrow 1$  to  $j = M$  do
27      $\theta_j^k = \frac{\pi_j^k (1/\Lambda_j)}{\sum_{l=1}^M (\sum_{i=1}^{\lfloor \log_2(B/\gamma) \rfloor} \pi_i^i) 1/\Lambda_l}$ 
28   endfor
29 endfor

```

O algoritmo apresentado usa somente os blocos ímpares e somente uma matriz deve ser invertida a cada etapa (b). Diferentemente do algoritmo *Folding*, o algoritmo proposto não requer uma inversão extra quando a matriz reduzida tem um número ímpar de blocos [87]. Adicionalmente, como o algoritmo de *Algoritmo Bloco Diagonal Zero* manipula os sub-blocos \mathcal{S}_l^+ e \mathcal{S}_{l+1}^- , é necessária a inversão de uma matriz de ordem que é igual, no máximo, a metade da matriz invertida a cada passo do algoritmo *Folding*.

3.4 Método Proposto para Duas Filas de Fluido

Nesta seção apresentaremos o cálculo da distribuição complementar para um modelo de fluido com duas filas e duas fontes, baseado na mesma metodologia utilizada para o modelo de uma única fila.

Consideremos um sistema de fluido composto por duas filas de fluido e um único servidor com tempo de serviço exponencialmente distribuído com taxa μ . A disciplina de atendimento considerada é a disciplina *Generalized Processor Sharing* (GPS), onde a capacidade do canal é dividida entre os fluxos, segundo um conjunto de parâmetros de particionamento, previamente definidos [115]. A disciplina GPS é conservadora de trabalho e opera de forma que a banda alocada para uma parte ociosa seja distribuída para as demais na proporção de seus parâmetros de particionamento. Cada uma das filas, no conjunto $k = \{1, 2\}$, recebe uma parte da capacidade total de serviço, definida por $\alpha^{(k)}\mu$. A Figura 3.7 mostra, graficamente, o sistema previamente descrito.

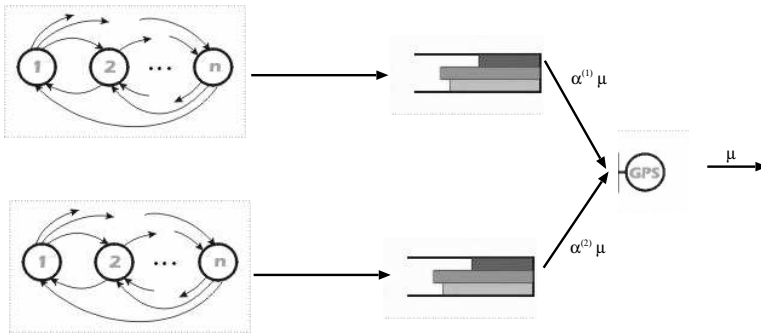


Figura 3.7: Diagrama para um modelo com duas filas de fluido.

Utilizando a mesma notação para o modelo com uma fila de fluido, seja $\mathcal{X} = \{X(t), t \geq 0\}$ uma CTMC com espaço de estados finito $\mathcal{S} = \{1, \dots, M\}$ e matriz geradora infinitesimal \mathbf{Q} , onde $q_i = -\sum_{j \neq i} q_{ij}$ é a taxa de saída do estado i . Como mencionado anteriormente, consideraremos duas fontes diferentes e independentes de fluido. Estas fontes podem ser representadas através de um processo que agrega o comportamento das duas fontes.

Cada estado da CTMC que modela a fonte indica se uma delas encontra-se em *on* ou *off*. É válido ressaltar que qualquer outro tipo de fonte poderia ser utilizada, aumentando somente a complexidade para a determinação do processo agregado das fontes. A Figura 3.8, mostra o processo \mathcal{X} para o caso específico de duas fontes do tipo *on-off*.

O primeiro passo do procedimento é calcular o parâmetro γ da mesma forma que

3.4 Método Proposto para Duas Filas de Fluido

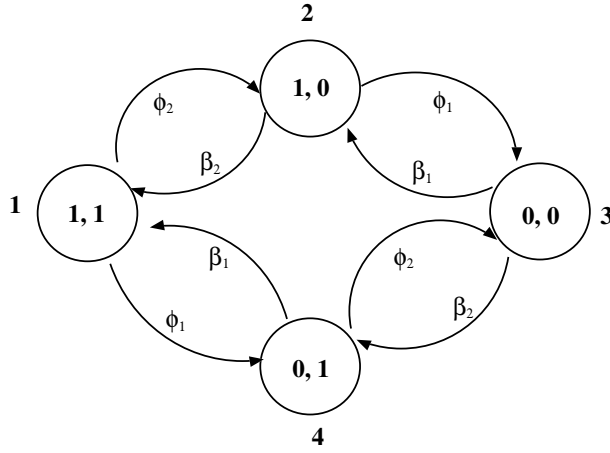


Figura 3.8: Modelo agregado para duas fontes do tipo *on-off*. A taxa ϕ_i , com $i = \{1, 2\}$, é a taxa de transição entre os estados *on* e o *off*. A taxa β_i , com $i = \{1, 2\}$, é a taxa de transição entre os estados *off* e o *on*.

calculamos anteriormente para o caso de uma única fila de fluido. A cada estado do processo \mathcal{X} , temos uma recompensa $r_{(i)}^{(k)}$. O valor de γ satisfaz a seguinte relação:

$$\gamma \leq \gamma_{\max} = \frac{|r_l|}{q_l} \text{ tal que: } l = \operatorname{argmin}_{i,k} \left\{ \frac{|r_{(i)}^{(k)}|}{q_i} \right\}. \quad (3.11)$$

Com o parâmetro γ definido, o próximo passo é determinar a recompensa em cada estado da DTMC anteriormente a construção do processo \mathcal{Y}^* . Neste caso, diferentemente do caso para uma única fila, teremos que considerar o tamanho das filas para a determinação das recompensas a serem atribuídas, com o objetivo de redistribuir toda a capacidade μ não utilizada. Note que, para o caso de duas filas GPS, a quantidade de fluido atribuída a cada fila depende não apenas da fase da fonte, mas também do tamanho de cada uma das filas.

Sejam $r_{(i,l)}^{(k)}$ a quantidade de fluido atribuída ou retirada da fila k , $k = \{1, 2\}$, onde a fase da fonte é representada por i e o tamanho da fila é igual a l , a capacidade alocada para a fila 1 igual a $\alpha\mu$, a capacidade alocada para a fila 2 igual a $(1 - \alpha)\mu$ e $\lambda_i^{(k)}$ a taxa de chegada de fluido na fila k . Quando o conteúdo de uma das filas atinge o valor igual a zero, a capacidade alocada a esta última será repassada a outra fila. Desta forma, os seguintes casos devem ser considerados para a determinação do valor de cada recompensa $r_{(i,l)}^{(k)}$ (para os casos em que a recompensa atribuída é maior que zero, omitimos as demais combinações de eventos, dado que a capacidade de serviço de uma fila não será repassada a outra fila, pois sempre haverá fluido a ser servido):

3.4 Método Proposto para Duas Filas de Fluido

- $\lambda_i^{(1)} - \alpha\mu < 0$ e $\lambda_i^{(2)} - (1 - \alpha)\mu < 0$
 1. $l_{(1)} = 0$ e $l_{(2)} = 0$
 - $r_{(i,0)}^{(1)} = 0$
 - $r_{(i,0)}^{(2)} = 0$
 2. $l_{(1)} > 0$ e $l_{(2)} = 0$
 - $r_{(i,l)}^{(1)} = \lambda_i^{(1)} - \mu$
 - $r_{(i,0)}^{(2)} = 0$
 3. $l_{(1)} = 0$ e $l_{(2)} > 0$
 - $r_{(i,0)}^{(1)} = 0$
 - $r_{(i,l)}^{(2)} = \lambda_i^{(2)} - \mu$
 4. $l_{(1)} > 0$ e $l_{(2)} > 0$
 - $r_{(i,l)}^{(1)} = \lambda_i^{(1)} - \alpha\mu$
 - $r_{(i,l)}^{(2)} = \lambda_i^{(2)} - (1 - \alpha)\mu$
- $\lambda_i^{(1)} - \alpha\mu > 0$ e $\lambda_i^{(2)} - (1 - \alpha)\mu > 0$
 1. $l_{(1)} > 0$ e $l_{(2)} > 0$
 - $r_{(i,l)}^{(1)} = \lambda_i^{(1)} - \alpha\mu$
 - $r_{(i,l)}^{(2)} = \lambda_i^{(2)} - (1 - \alpha)\mu$
- $\lambda_i^{(1)} - \alpha\mu < 0$ e $\lambda_i^{(2)} - (1 - \alpha)\mu > 0$
 1. $l_{(1)} = 0$ e $l_{(2)} > 0$
 - $r_{(i,0)}^{(1)} = 0$
 - $r_{(i,l)}^{(2)} = \lambda_i^{(2)} - \mu$
 2. $l_{(1)} > 0$ e $l_{(2)} > 0$
 - $r_{(i,l)}^{(1)} = \lambda_i^{(1)} - \alpha\mu$
 - $r_{(i,l)}^{(2)} = \lambda_i^{(2)} - (1 - \alpha)\mu$
- $\lambda_i^{(1)} - \alpha\mu > 0$ e $\lambda_i^{(2)} - (1 - \alpha)\mu < 0$
 1. $l_{(1)} > 0$ e $l_{(2)} = 0$
 - $r_{(i,l)}^{(1)} = \lambda_i^{(1)} - \mu$
 - $r_{(i,0)}^{(2)} = 0$

3.4 Método Proposto para Duas Filas de Fluido

2. $l_{(1)} > 0$ e $l_{(2)} > 0$
 - $r_{(i,l)}^{(1)} = \lambda_i^{(1)} - \alpha\mu$
 - $r_{(i,l)}^{(2)} = \lambda_i^{(2)} - (1 - \alpha)\mu$

Ao definirmos o parâmetro γ , escolheremos uma fila *pivot* e procederemos de forma idêntica à modelagem de proposta para os sistemas com uma fila. O comportamento para a outra fila será descrito a seguir. Assumiremos que a fila considerada como *pivot* é a fila 1.

O próximo passo é a construção do processo DTMC \mathcal{Y}^* . O espaço de estados \mathcal{S}^* do processo \mathcal{Y}^* é tal que $\mathcal{S}^* = \{(i, l_{(1)}, l_{(2)}) : 1 \leq i \leq M, 0 \leq l_{(1)} \leq \lfloor B^{(1)}/\gamma \rfloor, 0 \leq l_{(2)} \leq \lfloor B^{(2)}/\eta \rfloor\}$, onde o valor η tem o papel semelhante a γ e representa um *batch* de fluido que será acrescido/decrementado na fila 2. Para minimizar erros de arredondamento, η é escolhido como múltiplo do parâmetro γ . Como veremos a seguir a quantidade de fluido destinada a fila 2 é escrita em função de γ . Temos que $\Lambda_i = |r_{(i,l)}^{(1)}/\gamma|$ para cada um dos estados em \mathcal{Y}^* . Definiremos como $\psi_{(i,l)}^{(2)}$, o valor da recompensa destinada a fila 2, sendo que este valor dependerá tanto do parâmetro γ , como das taxas de recompensas $r_{(i,l)}^{(k)}$ determinadas *a priori*, conforme abaixo:

$$\psi_{(i,l)}^{(2)} = |r_{(i,l)}^{(2)}| \frac{\gamma}{|r_{(i,l)}^{(1)}|}, \text{ para cada estado } (i, l). \quad (3.12)$$

O parâmetro $\psi_{(i,l)}^{(2)}$, relacionado a fila 2, será determinado a partir da Equação 3.12. Note que, devido a escolha de γ , estamos assegurando que o tempo passado em cada estado i do processo \mathcal{Y}^* aumentará/diminuirá a quantidade de fluido da fila *pivot* 1 pelo valor γ .

O próximo passo é a construção da matriz de probabilidades \mathbf{P} . Esta é definida da seguinte forma: se $r_{(i,l)}^{(1)} > 0$, $r_{(i,l)}^{(2)} > 0$, então $p_{ij,kl,mn}^* = p_{ij}$, $l = \max\{k + 1, B^{(1)}\}$ e $n = \max\{m + 1, B^{(2)}\}$; se $r_{(i,l)}^{(1)} < 0$, $r_{(i,l)}^{(2)} < 0$, então $p_{ij,kl,mn}^* = p_{ij}$, $l = \min\{k - 1, 0\}$ e $n = \min\{m - 1, 0\}$ e 0 caso contrário.

O algoritmo de eliminação descrito na Seção 3.3 pode ser utilizado para obter a distribuição de probabilidades para o processo \mathcal{Y}^* . A partição $\mathcal{L}_{l^{(k)}}$ é a união dos subconjuntos $\mathcal{S}_{l^{(k)}}^+$ e $\mathcal{S}_{l+1^{(k)}}^-$, onde k é a fila *pivot*. A variação no tamanho da outra fila do sistema ocorre no intervalo 0 e o seu tamanho máximo. Para a fila *pivot* o aumento/diminuição do tamanho pode somente ocorrer entre níveis vizinhos. Para a fila complementar, esta variação pode ser maior e depende do parâmetro $\psi_{(i,l)}^{(2)}$.

3.4 Método Proposto para Duas Filas de Fluido

Finalizando, a seguinte relação pode ser estabelecida entre η e $\psi_{(i,l)}^{(2)}$. Como diversos valores reais podem ser atribuídos a variação da fila 2 (determinados pelo parâmetro $\psi_{(i,l)}^{(2)}$), o parâmetro η é definido para discretizar o ganho da fila 2, visando minimizar os possíveis erros de arredondamento. Podemos, por exemplo, atribuir o valor inteiro mais próximo ao valor que ocorre com maior frequência. Outra alternativa é a escolha do mínimo divisor comum, principalmente em casos que este é diferente de 1.

A Figura 3.9 representa os subconjuntos $S_i^{+/-}$ para um exemplo de duas fontes do tipo *on-off*, onde a fila 1 é a fila escolhida como *pivot*. O tamanho total da fila 1 é igual a M e da fila 2 é igual a K . Conforme podemos observar, a estrutura final é a mesma estrutura de um processo QBD, onde as mudanças ocorrem somente entre blocos vizinhos.

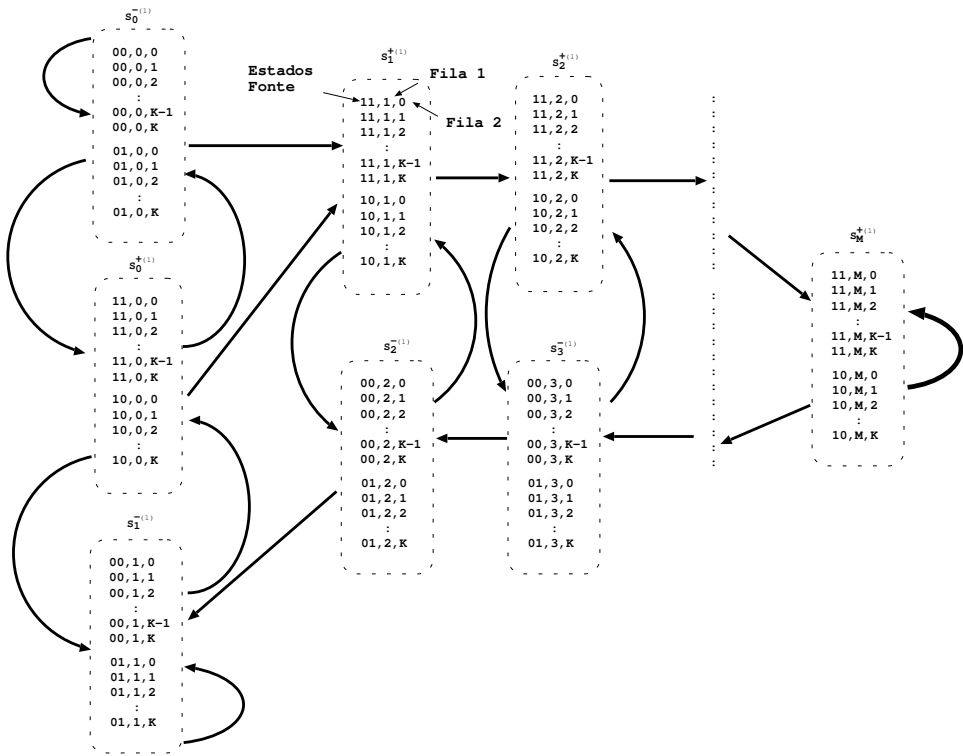


Figura 3.9: Diagrama com a representação da cadeia para o exemplo com duas fontes *on-off* e duas filas de fluido.

A escolha da fila *pivot* é de livre arbítrio do analisador do sistema, com a única restrição da garantia do valor do parâmetro γ .

3.5 Custo Computacional

Esta seção destina-se a descrever o custo computacional do método proposto e a comparação com os métodos descritos na literatura. Para esta comparação, três diferentes abordagens podem ser consideradas: (1) o custo computacional quando o modelo de pacotes é diretamente utilizado, mesmo para os casos onde a modelagem de fluido pode ser aplicada; (2) quando utilizamos o método proposto (neste caso é também feita a comparação com a eliminação de blocos com o algoritmo *Folding* no último passo do método); (3) quando utilizamos os métodos propostos em [3] e [4]. Para definição dos custos computacionais, iremos denotar como M o total de fases da fonte e B o tamanho da fila a ser modelada.

Primeiramente, apresentamos o diagrama da Figura 3.10 para uma visão geral dos métodos que podem ser utilizados para a solução de modelos de fluido.

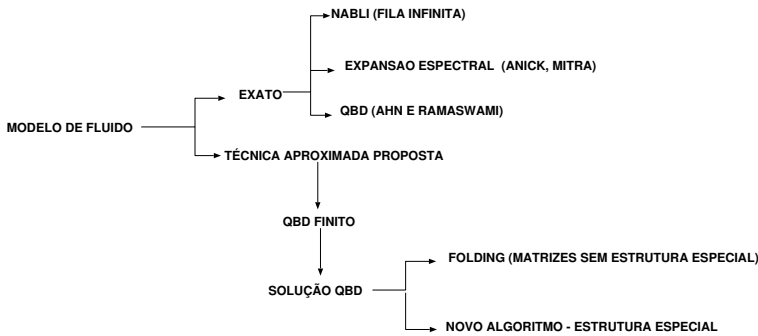


Figura 3.10: Visão Geral dos métodos para solução de modelos de fluido.

Considere o custo computacional quando a abordagem de pacotes é utilizada. Na grande maioria dos casos, estamos modelando filas onde o espaço de armazenamento que pode chegar a ordem de dezenas de milhares de pacotes e as fontes possuem várias fases. Assim, os algoritmos de eliminação considerarão a representação total da fila de dados, resultando em alto custo computacional. No caso do uso direto do algoritmo *Folding*, este custo será da ordem de $O(\lceil \log_2 B \rceil M^3)$, onde M é o total de fases da fonte e B o tamanho da fila.

Consideremos, também, a representação da fila através de modelos de fluido [5]. Neste caso, o custo computacional se restringe ao cálculo de autovalores e autovetores de uma matriz da ordem $M \times M$. No entanto, a instabilidade numérica dos sistemas a serem resolvidos e do próprio cálculo dos autovalores e autovetores, incentivaram propostas de modelagem alternativas [22].

Passemos a abordagem da representação dos modelos de fluido através de processos do tipo QBD. Primeiramente, consideremos o caso para fila com capacidade finita, apresentado em [4]. O custo computacional deste método se divide nos seguintes passos principais:

1. Cálculo da *rate matrix* \mathbf{R} , relacionada com o processo original e o respectivo cálculo da *rate matrix* $\hat{\mathbf{R}}$, relacionada com o processo reverso. Estas matrizes são necessárias dado que a distribuição de probabilidades dos estados do processo QBD são escritas através da forma *matrix-geometric* modificada [62]. O método iterativo mais eficiente para o cálculo destas matrizes, denominado *logarithmic reduction* ou *cyclic reduction*, encontra-se descrito em [86], onde em cada iteração a inversão de uma matriz de ordem $M \times M$ é efetuada, bem como, 8 multiplicações de matrizes, resultando em um total de $10M^3$ multiplicações (para inversa, podemos utilizar o algoritmo de eliminação LU, com o custo de $2M^3$ multiplicações).

Seja I_R o total de iterações para a convergência do algoritmo *logarithmic reduction*. Assim, o custo total para a definição da matriz \mathbf{R} é de $10I_R M^3$. Equivalentemente, temos que $I_{\hat{\mathbf{R}}}$ representa o total de iterações para obter a matriz $\hat{\mathbf{R}}$, com custo computacional de $10I_{\hat{\mathbf{R}}} M^3$ multiplicações;

2. Cálculo de quatro exponenciais de matrizes. Para estes cálculos, iremos considerar métodos de decomposição de matrizes, que são indicados para os problemas que envolvem matrizes com ordem elevada. Para estes métodos, o custo computacional, no melhor caso é de $10M^3$ multiplicações [103]. Para o custo das exponenciais de matrizes, o parâmetro M refere-se ao total dos estados onde as recompensas são positivas (2 exponenciais) e negativas (exponenciais complementares). Iremos considerar o total de fases negativas igual ao total de fases positivas. Assim, temos que, o custo para o cálculo das exponenciais de matrizes é de $5M^3$;
3. Para a solução final, é necessário o cálculo de 3 inversas de matrizes e 21 multiplicações entre matrizes, com ordem de $M/2$ e custo total de $27M^3/8$ multiplicações.

Desta forma, a equação do custo para o método apresentado em [3] é de:

$$T_F = 10(I_{\mathbf{R}} + I_{\hat{\mathbf{R}}})M^3 + 8.375M^3. \quad (3.13)$$

Para o caso infinito apresentado em [3], os principais passos e os respectivos custos computacionais são listados a seguir:

3.5 Custo Computacional

1. Cálculo da matriz \mathbf{R} . Similarmente ao caso de capacidade finita, o custo computacional é de $10I_{\mathbf{R}}M^3$;
2. Cálculo de uma exponencial de matriz, com ordem de $M/2$, resultando no total de multiplicações de $10M^3/8$;
3. Finalizando, multiplicações entre matrizes, da ordem de $M/2$ devem ser realizadas, resultando em $2M^3$ multiplicações.

Assim, a equação do custo para o método apresentado em [4] é de:

$$T_I = 10I_{\mathbf{R}}M^3 + 3.25M^3. \quad (3.14)$$

Consideremos a abordagem proposta neste capítulo. O custo computacional está restrito ao custo do *Algoritmo Bloco Diagonal Zero*, onde os blocos ímpares são eliminados a cada passo do algoritmo e uma única inversão de matriz é calculada. Como consideramos os subconjuntos \mathcal{S}_i^+ e \mathcal{S}_i^- , todas as operações matriciais são feitas com matrizes cuja a ordem é, no pior caso, de $M/2$. Este caso corresponde aos tipos de fonte onde o total de recompensas positivas é igual ao total de fases da fonte onde a recompensa é negativa. Na fase de redução do algoritmo, são feitas 11 multiplicações de matrizes e uma inversão de matriz. Na etapa de definição dos vetores π , são realizadas 6 multiplicações entre matriz vetor e uma multiplicação entre matrizes e uma inversa de uma matriz. A equação 3.15 representa o custo do algoritmo de eliminação proposto:

$$T_A = 2M^3 \log_2 \left(\frac{B}{\gamma} \right) + 1.5M^2 \left(\frac{B}{\gamma} \right). \quad (3.15)$$

Com o objetivo de comparações entre o método proposto e os método para a capacidade finita apresentado em [4], seja P_A , o total de passos necessários para o cálculo da medida de interesse através do método proposto neste capítulo (note que o número de passos é finito e pode ser calculado em função de B e γ). A seguinte relação pode ser estabelecida:

$$T_A \leq T_F \text{ se } P_A \leq 5(I_{\mathbf{R}} + I_{\hat{\mathbf{R}}}).$$

Podemos concluir que o método proposto será atrativo, em termos de custo computacional, para os casos onde P_A é menor que 5 vezes a soma do número de iterações para obter as matrizes \mathbf{R} e $\hat{\mathbf{R}}$. Seja, por exemplo, o melhor caso para obtenção das matrizes \mathbf{R} e $\hat{\mathbf{R}}$, com o total de uma iteração para obtenção de cada uma delas. Neste caso, o método proposto sempre será computacionalmente melhor, se o total de passos do algoritmo de eliminação for menor que $\log_2(B/\gamma) = 10$. Isto significa que se representarmos

3.5 Custo Computacional

um número inferior a 1024 níveis, o custo computacional do método proposto é menor. Conforme poderemos observar na seção de exemplos, valores pequenos de (B/γ) são utilizados para modelar a fila (no máximo 500), resultando em boa aproximação do valor exato da medida. Adicionalmente, em [86], para um exemplo prático de modelagem de sistema de pacotes de dados, o total de iterações varia entre 4 e 30, somente para a matriz \mathbf{R} .

Finalizando, iremos comparar a eliminação de blocos proposta onde é levado em consideração a estrutura especial de QBD (Figura 3.2) com o algoritmo *Folding*. Enfatizamos que na nossa eliminação, não existe a necessidade de um inversão extra quando a matriz reduzida possui um número ímpar de blocos [142]. A equação 3.16 mostra os custos para o algoritmo em [142], onde M representa o total de fases da fonte e B/γ o tamanho máximo da fila modelada.

$$5.33\alpha M^3 \log_2 \left(\frac{B}{\gamma} \right) + 2M^2 \left(\frac{B}{\gamma} \right) \quad (3.16)$$

com $1 \leq \alpha < 2$, dependendo do total B/γ de níveis na fila.

As Figuras 3.11, 3.12 e 3.13 exemplificam a comparação do custo computacional em função do parâmetro M para três valores de B/γ : $B/\gamma = 10000$, $B/\gamma = 200000$ e $B/\gamma = 100$, respectivamente. Conforme podemos notar, o algoritmo de eliminação proposto apresenta um custo computacional menor quando comparado com o algoritmo *Folding*. Este custo pode diminuir dependendo da estrutura das matrizes \mathbf{A}_0 e \mathbf{A}_3 , dado que estas são invertidas na seguinte operação entre matrizes: $(\mathbf{I} - \mathbf{A}_3^{(i-1)} \mathbf{A}_0^{(i-1)})^{-1}$. Em alguns casos, o custo computacional do algoritmo *Folding* é, aproximadamente, três vezes maior que o custo computacional do nosso método de eliminação. Adicionalmente, o custo do método proposto pode reduzir significativamente, quando o total de fases positivas e negativas são assimétricas. A ordenação da matriz \mathbf{P} pode ser estabelecida de tal forma que tenhamos que inverter a matriz referente ao menor conjunto de estados a cada etapa (b) do algoritmo proposto.

Finalizando esta seção, devemos ressaltar que o custo do algoritmo de eliminação descrito nesta seção refere-se ao caso de uma fila de fluido. Para o caso de duas filas, a cada passo do algoritmo, matrizes da ordem de $M/2 \times B^{(2)}/\eta$ deverão ser manipuladas. Com isso a Equação 3.15 fica:

$$T_A = 2M^3 \left(\frac{B^{(2)}}{\eta} \right)^3 \log_2 \left(\frac{B^{(1)}}{\gamma} \right) + 1.5M^2 \left(\frac{B^{(2)}}{\eta} \right)^2 \left(\frac{B^{(1)}}{\gamma} \right). \quad (3.17)$$

3.6 Comparação entre os Métodos

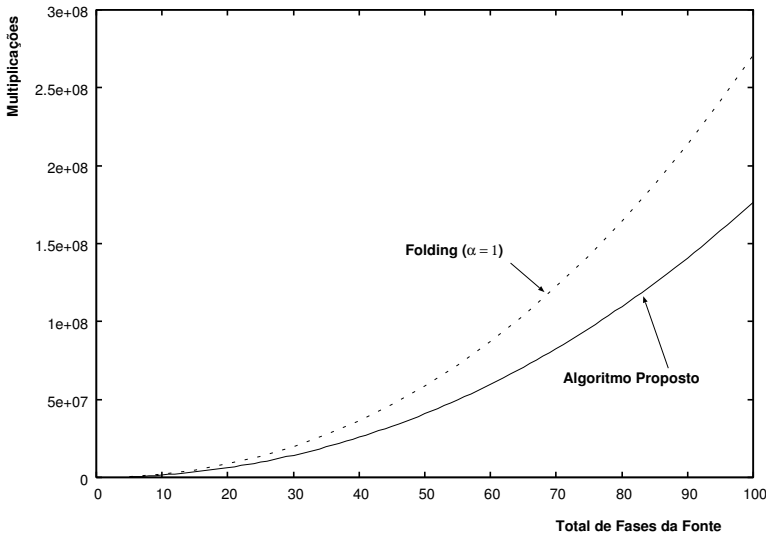


Figura 3.11: Sistema onde $B/\gamma = 10000$.

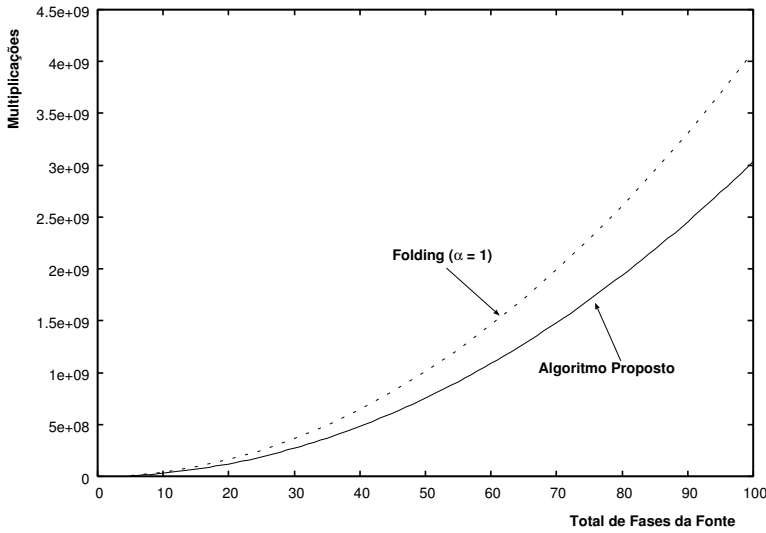


Figura 3.12: Sistema onde $B/\gamma = 200000$.

3.6 Comparação entre os Métodos

Apesar dos métodos propostos neste capítulo, em [3] e [4] e em [22] utilizarem o paradigma de processos QBD para obterem a distribuição de probabilidades, várias características diferentes entre os métodos podem ser apontadas.

Primeiramente, seja a comparação entre a modelagem tradicional de pacotes e a modelagem de fluido aproximada apresentada neste capítulo. Conforme mencionado anteriormente, a modelagem de pacotes resulta em alto custo computacional, tanto para a simulação (dado o processamento de um grande número de eventos) quanto para soluções

3.6 Comparação entre os Métodos

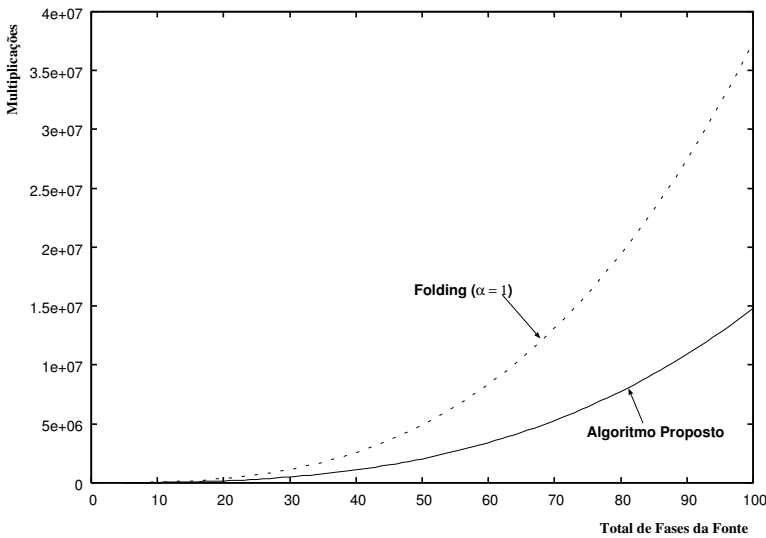


Figura 3.13: Sistema onde $B/\gamma = 100$.

analíticas (dado que filas com grande capacidade devem ser resolvidas). Em contrapartida, o método proposto neste capítulo, representa a chegada de fluido e a capacidade da fila pertencente ao sistema em estudo através de “blocos de fluido”. Tecnicamente, propomos um nível de discretização do fluido de tal forma que podemos obter resultados aproximados com baixo custo computacional. Diferentemente do modelo de pacotes em *batches*, que resulta em grande variância no conteúdo da fila, o modelo de “bloco de fluido” aproxima o modelo original de pacotes, dado que este pode ser representado através de um modelo de fluido tradicional [5], com grande precisão, conforme mostraremos na Seção 3.7.

Os métodos apresentados em [3] e [4] usam o processo QBD como um passo intermediário para a modelagem: a distribuição do processo QBD é usada para estudar o processo de *trabalho* associado com o modelo de fluido. A partir do processo que descreve o *trabalho* no sistema, a distribuição do modelo de fluido é calculada. No método aqui proposto, o processo QBD é utilizado diretamente para a obtenção do resultado aproximado para o modelo de fluido.

Como uma outra característica a ser ressaltada, o método proposto uniformiza cada estado da cadeia com uma taxa de uniformização diferente, e, conseqüentemente, temos a transformação do modelo original de fluido em um processo discreto QBD. Como utilizamos a recompensa original do modelo para cada estado, diferentemente dos autores em [3],[4] e [22] que utilizam a idéia de fluido homogêneo com o objetivo de simplificar as soluções matemáticas, a distribuição aproximada é obtida diretamente, sem a necessidade da tradução de volta para o modelo de fluido original.

A maneira como o processo QBD é definido, é uma outra diferença entre os métodos. No método proposto neste capítulo, a chegada de “blocos de fluido” é ligada a mudanças de fases na fonte assim como as saídas destes “blocos de fluido”. No processo QBD apresentado em [3] e [4], somente as chegadas na fila estão ligadas as mudanças nas fases da fonte. O serviço na fila é totalmente independente das mudanças do comportamento da fonte de dados.

Apesar do método proposto fornecer uma resposta aproximada, a modelagem é bem simples e os resultados são bem precisos, com um pequeno custo computacional, conforme mostraremos na seção seguinte.

3.7 Exemplos

Nesta seção apresentaremos alguns exemplos visando o estudo do desempenho do método proposto, tanto em termos de custo computacional, como em termos de precisão.

O primeiro exemplo representa um modelo de fonte do tipo histograma [131], com oito estados diferentes e uma fila única de fluido. Este exemplo simples é considerado pois podemos estabelecer uma solução exata para a distribuição de probabilidades utilizando a equação tradicional de fluidos, apresentada em [5]. Para este exemplo pequeno, a solução exata utilizando os autovalores e autovetores não é influenciada por erros numéricos.

O segundo conjunto de exemplos mostra a utilização do método para uma fonte de dados, parametrizada através de *traces* com tráfego real coletados no Departamento de Engenharia de Sistemas e Computação/COPPE/UFRJ e de um *backbone Internet* [112], cujo resultados preliminares com o objetivo de dimensionar redes de dados foram apresentados em [37]. Com este exemplo mostramos a aplicabilidade do método proposto para sistemas reais com centenas de estados. Concluindo, o terceiro exemplo ilustra o uso do método proposto quando duas fontes *on-off* enviam informações para duas filas de fluido servidas com a política GPS.

O método apresentado é uma aproximação para a medida real e, portanto, a sua acurácia será avaliada através de diversos exemplos, onde os parâmetros foram variados dentro de um conjunto de valores. As medidas utilizadas para a precisão do método são as seguintes: (1) erro relativo máximo de $\hat{P}[CR > x]$, onde $\hat{P}[CR > x]$ é o valor da aproximação calculada pelo método proposto.; e (2) erro relativo para a probabilidade da fila estar cheia (capacidade total). O erro relativo é calculado a partir da seguinte equação:

$(P[CR > x] - \hat{P}[CR > x])/P[CR > x]$. No primeiro exemplo, $P[CR > x]$ é o valor exato obtido através da solução apresentada em [5]. No segundo conjunto de exemplos, usamos o método apresentado em [88] para obter a solução $P[CR > x]$. A solução em [88] é estabelecida para o estudo em estado transiente. Para aplicar neste contexto, realizamos o cálculo para diversos tempos de observação, até que o sistema estivesse em equilíbrio (estado estacionário). No quarto conjunto de exemplos, a simulação de fluido foi utilizada ([25],[34],[36]).

Os custos computacionais foram calculados da seguinte forma. Comparamos o custo computacional do nosso algoritmo de eliminação com o custo do algoritmo *Folding* aplicado tanto para a matriz resultante após a transformação do modelo de fluido em modelo com recompensas de impulso.

3.7.1 Modelo de uma Fonte de Vídeo

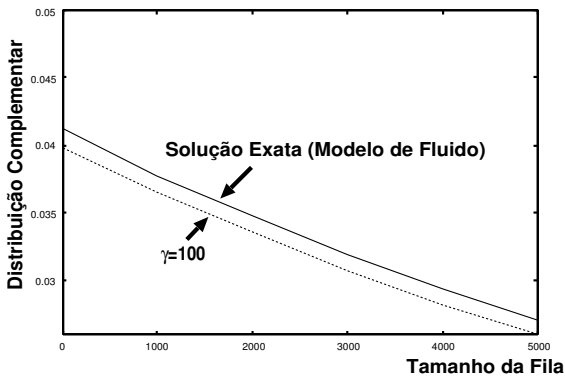
Neste primeiro exemplo, consideramos um modelo de histograma [131] com oito estados parametrizados a partir de um sequência de vídeo Asterix, codificada em MPEG. Variamos os tamanhos da fila dentre quatro valores entre 1000 e 50000, bem como os valores do parâmetro γ . A Figura 3.14 apresenta a distribuição complementar da recompensa acumulada para diversos valores de tamanho da fila. O valor máximo do erro relativo $P[CR > x]$ e o erro relativo da probabilidade da fila estar cheia variam entre $2e-2$ e $1e-1$ para todos os cenários analisados. Os resultados, conforme esperado, mostram que a precisão do método aumenta com a diminuição do parâmetro γ . A diminuição do valor de γ significa um *batch* de fluido menor, resultando em aumento da acurácia dos resultados.

Consideremos o ganho computacional quando utilizamos a modelagem de fluido, em substituição ao paradigma tradicional de pacotes. Para os cenários mostrados, o ganho computacional em número total de multiplicações, aplicando o método proposto, é de, no máximo, aproximadamente 2.5 vezes menor do que quando comparamos com a modelagem proposta e a utilização do algoritmo *Folding*.

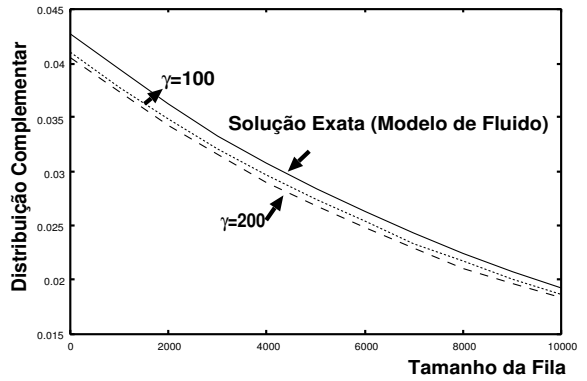
3.7.2 Modelo de Tráfego Agregado

O segundo conjunto de exemplos é baseado em um modelo de tráfego agregado que representa o comportamento do tráfego coletado no Departamento de Sistemas e

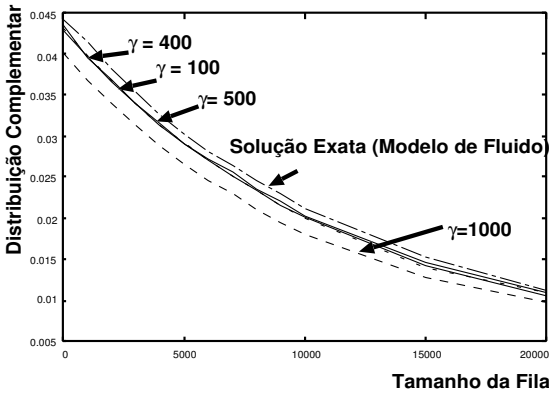
3.7 Exemplos



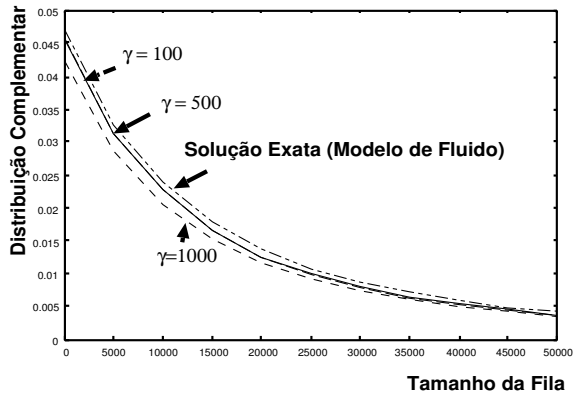
(a) Distribuição Complementar - Fila = 5000



(b) Distribuição Complementar - Fila = 10000



(c) Distribuição Complementar - Fila = 20000



(d) Distribuição Complementar - Fila = 50000

Figura 3.14: Distribuição complementar da recompensa acumulada - modelo de uma fonte de vídeo.

Computação/COPPE/UFRJ [37] e de um *backbone* na Internet [112].

Como primeiro conjunto de exemplos, parametrizamos o modelo que representa o canal de saída do Departamento através das estatísticas coletadas e consideramos diversos valores para a capacidade do canal. No primeiro passo da modelagem, obtemos as probabilidades de transição entre os estados do modelo de tráfego e os valores para a taxa de entrada, λ_i , para cada estado i do modelo de fluido. A cada estado do modelo, atribuímos uma recompensa de taxa igual a $\lambda_i - \mu_j$, com $\mu_j = \{1100Mbps, 1200Mbps, 1300Mbps, 1500Mbps\}$.

A Figura 3.15 mostra a distribuição complementar da recompensa acumulada para uma fila de tamanho igual a 1000, considerando diversos valores de γ . Como no exemplo anterior, o erro relativo da probabilidade da fila estar cheia varia entre $2e-2$ e $1e-1$, sendo que o erro relativo máximo variou entre $9e-2$ e $2e-1$. Para a resposta exata, utilizamos o método proposto em [88] considerando diversos tempos de observação até atestarmos que a medida tenha atingido o equilíbrio.

3.7 Exemplos

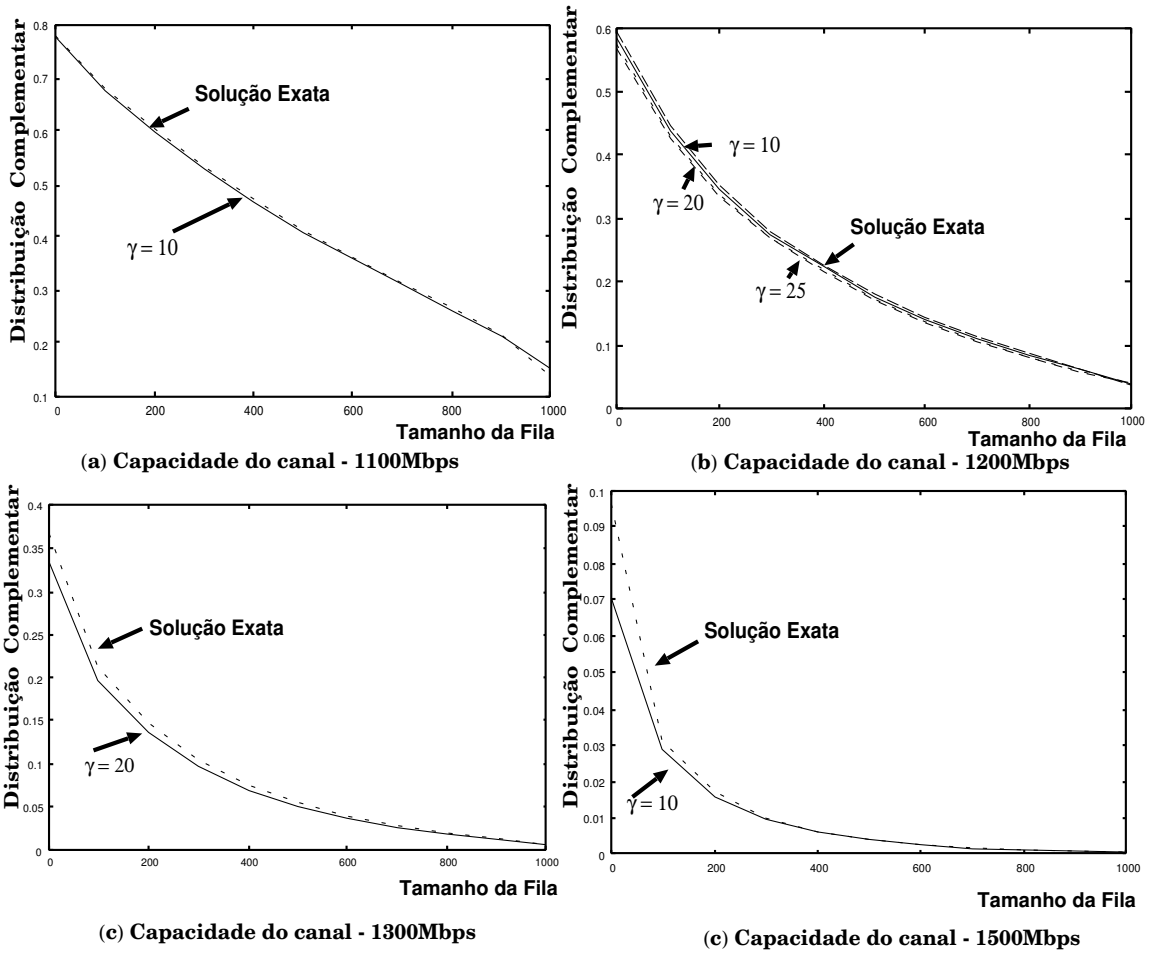


Figura 3.15: Distribuição complementar da recompensa acumulada - Modelo do tráfego agregado do canal de saída do Departamento de Sistemas e Computação.

O segundo conjunto de modelos refere-se a modelagem de um tráfego em um *backbone Internet* disponível em [112]. Consideramos um valor fixo para a capacidade μ e variamos o tamanho da fila e o parâmetro γ . O erro relativo máximo da medida $P[CR > X]$ varia entre $2e-2$ e $1e-1$ e o erro relativo da probabilidade da fila estar cheia varia entre $8e-4$ e $3e-3$. Os resultados são mostrados na Figura 3.16. Para os cenários considerados, a diminuição do custo computacional em número de multiplicações quando o método proposto é utilizado ao invés de resolvermos o mesmo modelo proposto e o algoritmo *Folding* é de, no máximo, aproximadamente 3 vezes.

O custo computacional para a solução do sistema em estudo através do paradigma de pacotes pode ser ordens de grandeza maior que quando resolvemos o mesmo sistema utilizando a modelagem de fluido. O custo computacional pode ser reduzido se considerarmos que os pacotes podem ser servidos em *batches*, similarmente a servir um *batch* de fluido no método que apresentamos neste capítulo. Uma questão que podemos nos per-

3.7 Exemplos

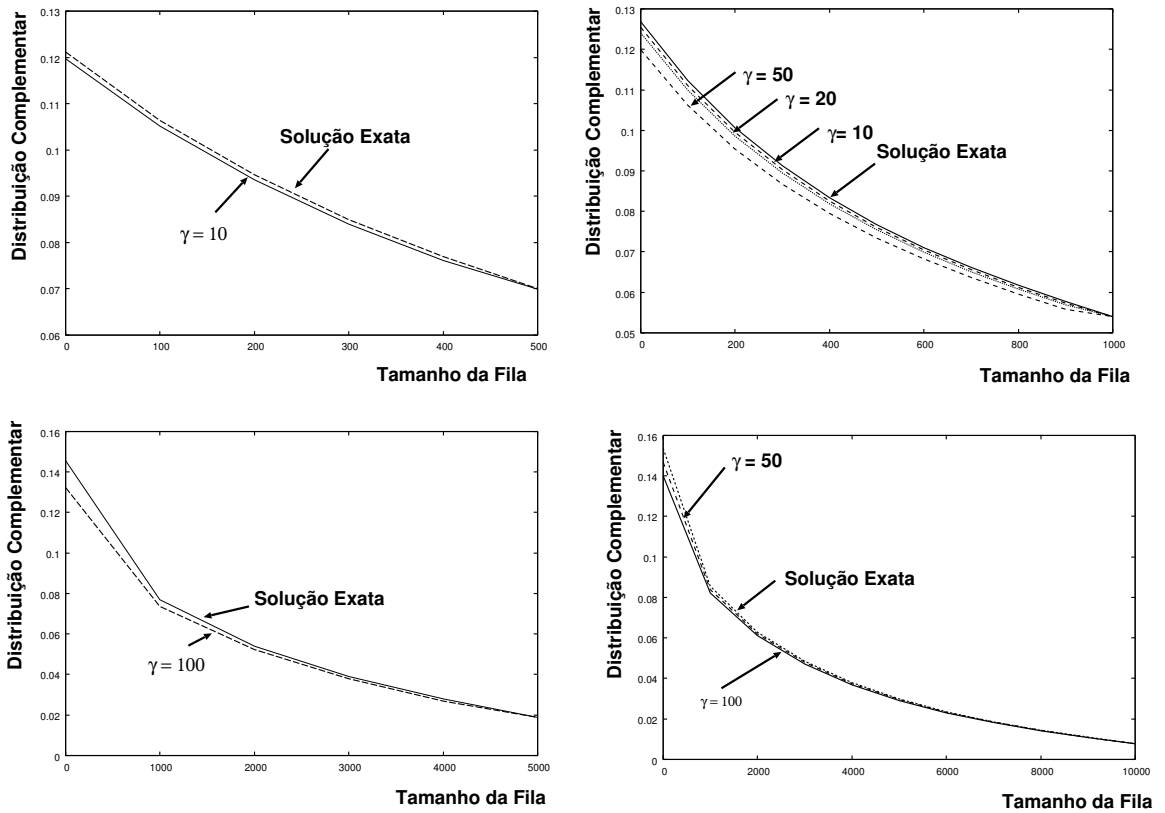


Figura 3.16: Distribuição complementar da recompensa acumulada - Modelo do tráfego agregado de um canal *Internet* [112].

guntar é a seguinte: “Qual é a exatidão dos resultados obtidos com o modelo de pacotes do tipo *batch* quando comparado com o resultado do método proposto?” Para fazermos esta comparação, utilizamos um dos modelos apresentados anteriormente e comparamos com o paradigma de modelagem que consiste em descrever a chegada e serviço de pacotes através de *batches*. No exemplo de *batch* de pacotes, todas as taxas dos eventos são reescaladas de tal maneira que a fila possua 50 níveis, cada um correspondendo a *batches* de 200 pacotes. A Figura 3.17 apresenta os resultados obtidos. O exemplo mostra que para este modelo, utilizar a modelagem de fluido fornece bons resultados quando o modelo real de pacotes é calculado, com a vantagem de diminuição no custo computacional. Todavia, podemos observar que o modelo em *batch* de pacotes resulta em uma aproximação muito ruim para a medida da distribuição complementar do tamanho da fila. Assim, a aplicação do método proposto, ou seja a união entre a modelagem proposta e o algoritmo de *Algoritmo Bloco Diagonal Zero* fornece resultados precisos, com custo computacional baixo.

Finalizando o conjunto de exemplos destinados a uma fila de fluido, podemos observar que em todos os exemplos com uma única fila de fluido, os resultados aproximados estão

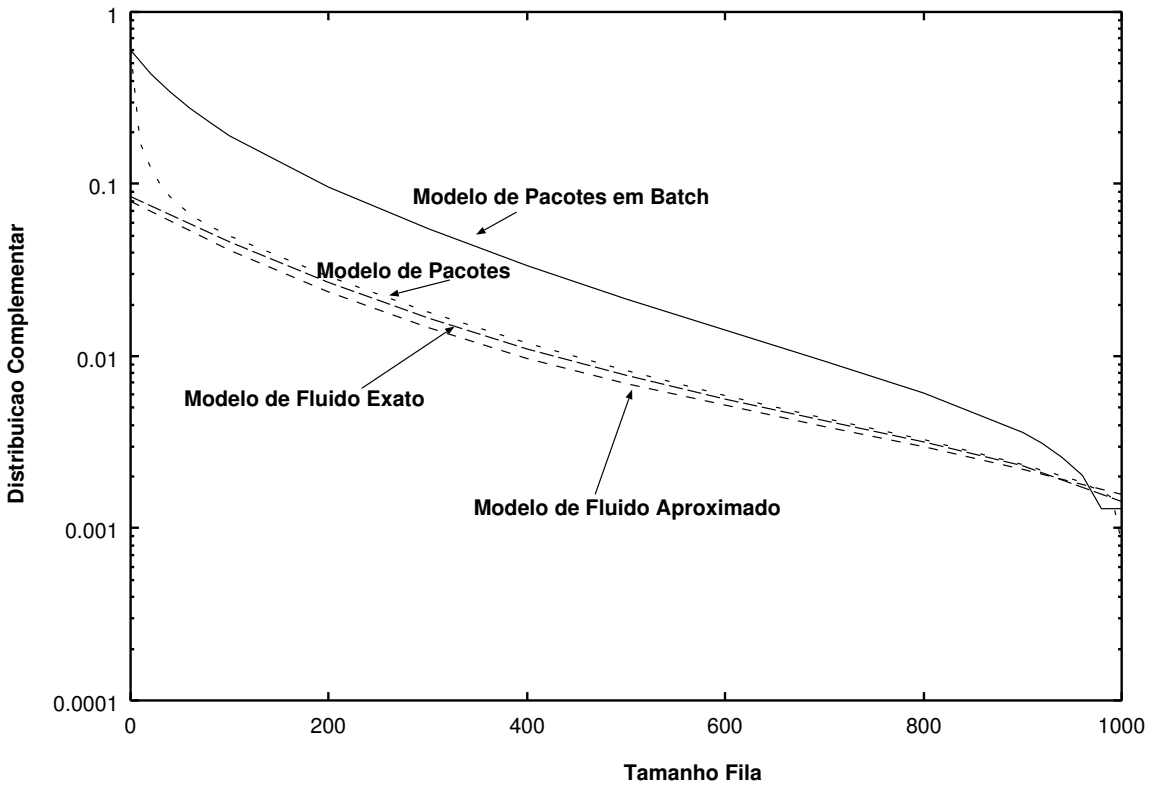


Figura 3.17: Estudo comparativo entre o paradigma de modelagem de fluido, o modelo proposto, o modelo de chegada e saída em *batch* de pacotes e o modelo real de pacotes.

abaixo do valor exato da medida.

3.7.3 Modelos com Duas Filas de Fluido

O objetivo deste exemplo é analisar o método proposto quando temos o caso de duas filas de fluido. Para tal, consideramos duas fontes *on-off* com tempo exponencialmente distribuído no estado *on* e no estado *off* com parâmetros $\phi^{(k)}$ e $\beta^{(k)}$, respectivamente, onde $k = \{1, 2\}$. Enquanto a k -ésima fonte está no estado *on* (*off*), o fluido do tipo k entra na fila com uma taxa $\lambda_i^{(k)}$ (0). Os valores para os parâmetros descritos anteriormente são os seguintes: $\beta^{(1)} = 3$, $\beta^{(2)} = 4$, $\phi^{(1)} = 2$, $\phi^{(2)} = 3$, $B^{(1)} = 100$, $B^{(2)} = 100$, $\alpha = 0.5$, $\lambda_i^{(1)} = \lambda_i^{(2)} = 1000$, $\gamma = 5$ e $\mu = 500$.

Para estimar a acurácia do método, simulações utilizando o simulador de fluido da ferramenta *Tangram-II* foram realizadas ([25],[34],[36]). Para todas as simulações realizadas, o nível de confiança definido foi de 99%, e a variância na ordem de $1e-6$. A Figura 3.18 mostra a distribuição complementar para ambas as filas. Podemos observar que a acurácia dos resultados para a fila 1 é melhor que para fila 2. Este fato pode ser

3.7 Exemplos

facilmente explicado pois descrevemos o aumento/diminuição do tamanho da segunda fila em função da primeira, resultando em erros de arredondamento. Os erros relativos máximo foram de $1e-1$ e de $2.6e-1$ para as filas 1 e 2 respectivamente. Para o caso da fila cheia, o erro relativo foi de $5e-2$ para a fila 1 e $1e-1$ e para a fila 2. Para este caso, onde o tamanho das filas é muito pequeno, o erro de arredondamento inerente a Equação 3.12, diminui muito a precisão do método proposto. Adicionalmente, o comportamento do resultado para fila 2 pode ser explicado pelo fato de que os níveis observados não são múltiplos do parâmetro η escolhido para a fila 2.

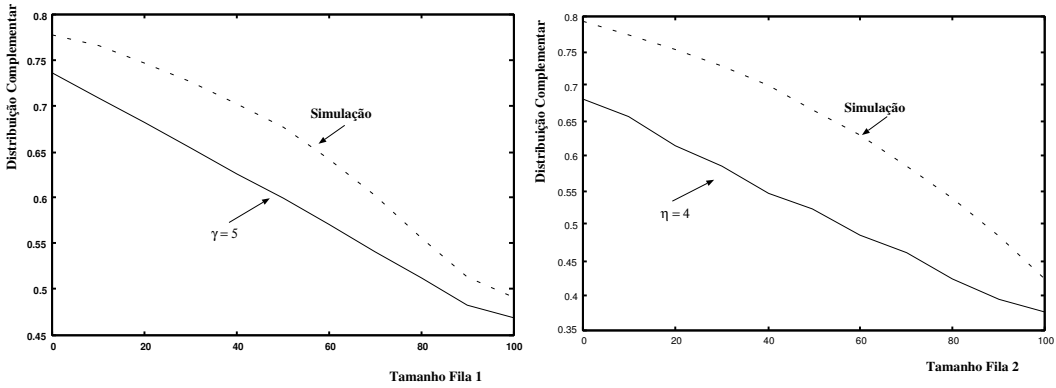


Figura 3.18: Distribuição Complementar do conteúdo de cada uma das filas - modelo com duas filas. O tamanho para cada uma das filas é igual 100.

Assumamos que o mesmo modelo de fila com os mesmos parâmetros seja outra vez utilizado mudando o tamanho das duas filas para 1000. A Figura 3.19 mostra os resultados do método proposto. Para este caso, onde as filas possuem um tamanho maior, os resultados são mais precisos, com erro relativo máximo de $1e-2$ e aproximadamente $1.3e-1$ para as filas 1 e 2, respectivamente. Para o caso de fila cheia, temos que o erros relativos foram de $4e-2$ e $1e-1$, para as filas 1 e 2, respectivamente.

Para o caso de tamanhos de fila iguais a 5000, os resultados são mostrados na Figura 3.20. A relação entre a diminuição dos parâmetros γ e η e a melhoria da precisão também se verifica para o caso de duas filas, conforme esperado.

Consideremos um sistema de filas com as seguintes taxas: $\beta^{(1)} = 3$, $\beta^{(2)} = 4$, $\phi^{(1)} = 2$, $\phi^{(2)} = 3$, $\alpha = 0.5$, $\lambda_i^{(1)} = \lambda_i^{(2)} = 7000$, $\mu = 6000$. O tamanho de cada uma das filas é de $B = 50000$. Os resultados para este sistema são mostrados na Figura 3.21. Como esperado, a precisão aumenta com o aumento da capacidade da fila. Os erros relativos para a probabilidade de fila cheia são iguais a $1e-1$ para ambas as filas.

Sobre os resultados obtidos para a aproximação de duas filas de fluido, podemos esta-

3.7 Exemplos

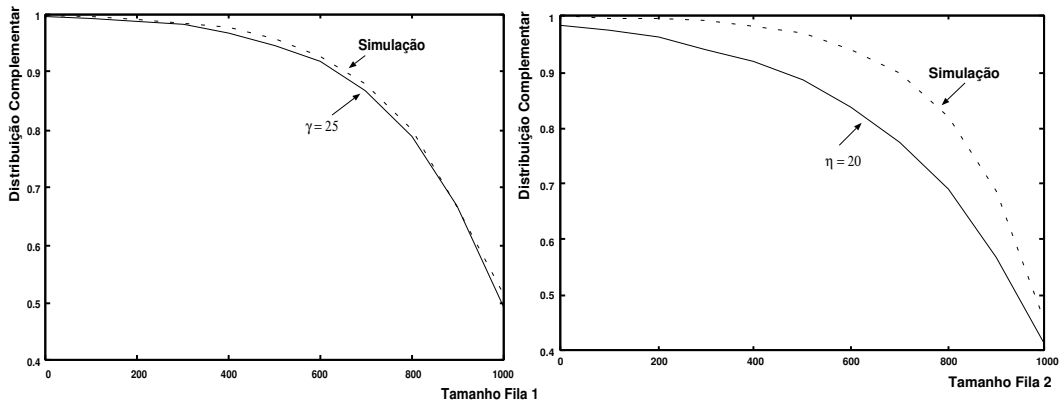


Figura 3.19: Distribuição Complementar do conteúdo de cada uma das filas - modelo com duas filas. O tamanho para cada uma das filas é igual 1000.

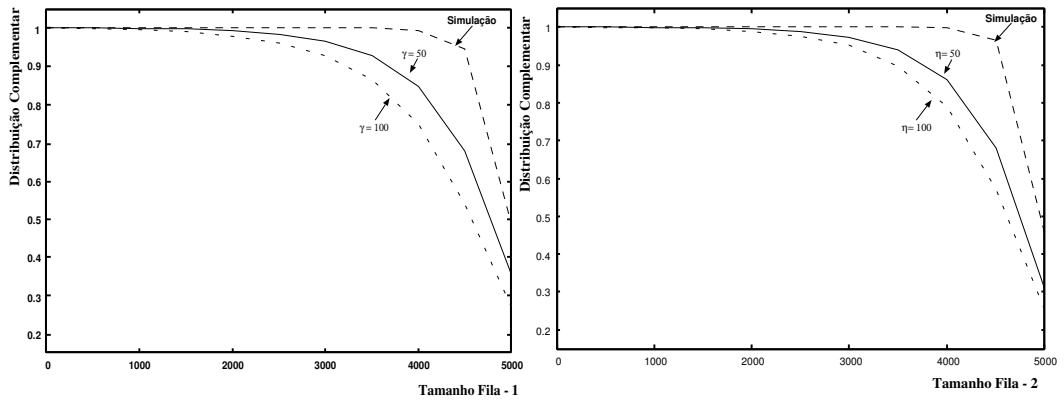


Figura 3.20: Distribuição Complementar do conteúdo de cada uma das filas - modelo com duas filas. O tamanho para cada uma das filas é igual a 5000.

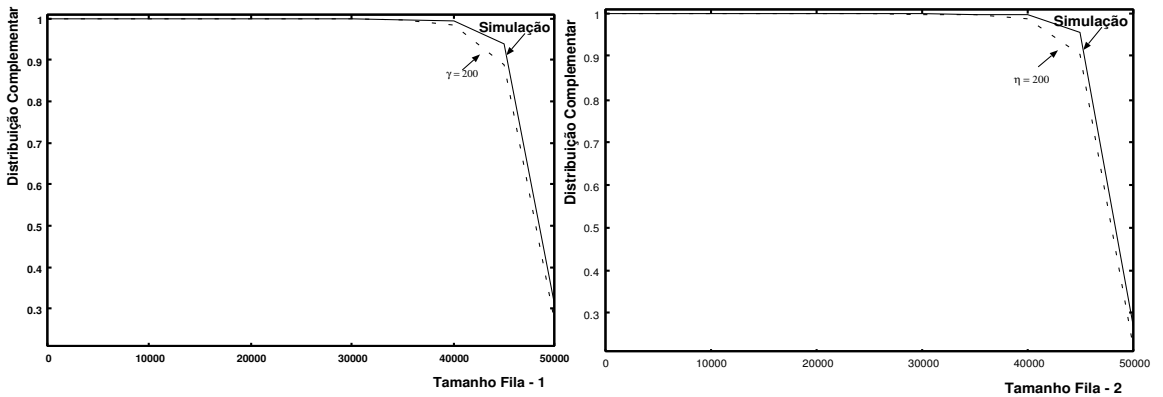


Figura 3.21: Distribuição Complementar do conteúdo de cada uma das filas - modelo com duas filas. O tamanho para cada uma das filas é igual 50000.

belecer os seguintes comentários:

1. Da mesma forma que para os modelos com uma única fila de fluido, a aproximação do método proposto sempre está abaixo do valor real da medida;
2. O aumento e diminuição da fila 2 são escritos em função do comportamento da fila *pivot*. Neste caso, o resultado aproximado, principalmente da fila 2, será influenciado por erros de arredondamento (Equação 3.12).

3.8 Sumário das Contribuições

A modelagem de fluido tem sido amplamente utilizada na literatura para estudos de sistemas de comunicação, reduzindo o custo computacional do paradigma tradicional de modelagem de pacotes. Uma das principais medidas a ser calculada é a distribuição de probabilidades complementar do conteúdo de uma fila de dados, em estado estacionário, definida por: $P[CR > x] = \lim_{t \rightarrow \infty} P[CR(t) > x]$. Métodos tradicionais utilizam análise espectral para obter esta medida. Neste capítulo propomos um novo método para solução de modelos de fluido. Podemos citar como contribuições principais:

1. Propomos a modelagem de fluido através de recompensas de impulso. Como resultado desta modelagem, obtemos diretamente um processo QBD a partir do modelo original de fluido. Este novo modelo nos fornece um resultado aproximado para a medida de interesse $P[CR > x]$. A acurácia da aproximação está relacionada com o parâmetro γ utilizado. A modelagem é baseada em processos QBD, também utilizada pelos métodos desenvolvidos paralelamente com o trabalho aqui apresentado e discutidos em ([3],[4],[22]);
2. Propomos um novo algoritmo de eliminação, definido como *Algoritmo Bloco Diagonal Zero*, cujo custo computacional é menor que o custo dos métodos similares encontrados na literatura, para matrizes de transição de probabilidades com uma determinada estrutura especial. Vale ressaltar que este algoritmo pode ser utilizado para solução de qualquer modelo, bastando que este tenha a matriz \mathbf{P} com estrutura semelhante a matriz apresentada na Figura 3.1;
3. O método proposto foi estendido para sistemas com duas filas de fluido finitas;
4. Para o estudo da acurácia do método proposto, um conjunto de exemplos foi feito, considerando taxas reais de envio de informações em redes de dados reais.

Capítulo 4

Valor Esperado da Recompensa Acumulada para Modelos com Estados Absorventes

4.1 Introdução

O cálculo de medidas de desempenho e confiabilidade de modelos que representam sistemas complexos é um tópico extensivamente estudado na literatura. Soluções analíticas e numéricas para estas medidas têm sido uma intensa área de pesquisa ([64],[44],[30],[96],[1]) e um grande número de ferramentas tem sido implementado com objetivo tanto de facilitar a especificação dos sistemas quanto de disponibilizar métodos de solução para obtenção de medidas de interesse ([35],[24],[125],[64]). Todavia, o aumento da complexidade dos sistemas a serem representados e o surgimento de diversas outras medidas de interesse fazem com que novos tipos de métodos continuem a ser pesquisados e propostos.

Como no Capítulo anterior, usaremos modelos Markovianos para representarmos o sistema real a ser estudado. Apesar da teoria matemática de processos Markovianos possuir diversos resultados bem fundamentados, de fácil interpretação probabilística e manipulação algébrica, podemos citar duas principais desvantagens da sua aplicabilidade. Sistemas complexos, possuem, em geral, valor elevado da cardinalidade do espaço de estados. Não é incomum modelos com milhões de estados, o que dificulta a utilização dos métodos de solução clássicos encontrados na literatura tais como GTH, Jacobi, Gauss-Seidel, SOR, etc ([90],[39]). O segundo problema denomina-se *stiffness*, sendo este rela-

cionado com a grande diferença dos valores numéricos das taxas de eventos que podem ocorrer no sistema sendo estudado. Esta característica pode gerar instabilidade numérica quando métodos tradicionais são aplicados [10].

O principal objetivo em modelar um sistema é a possibilidade de definir e calcular medidas de interesse que permitam avaliar o comportamento deste sob condições pré-definidas. Em alguns casos é possível obter uma solução analítica em forma fechada. Nos casos em que o modelo de estudo não possui solução fechada, métodos numéricos podem ser tentados. Para os modelos a serem estudados neste capítulo, que se caracterizam pelo número total de estados na ordem de milhões, os métodos numéricos tradicionais, que consideram todo o espaço de estados para obtenção da medida de interesse, possuem alto custo computacional em tempo e memória e se tornam rapidamente não aplicáveis.

Uma maneira de lidar com este problema é utilizar métodos que reduzem o espaço de estados a ser considerado para obter a medida de interesse desejada. Podemos citar como técnicas mais conhecidas a agregação de estados ([10],[130],[132]), a teoria de complemento estocástico (caso especial do conceito mais geral conhecido como *Perron complementation*, apresentado em [16]), a técnica de *path-based*, que permite calcular as medidas de interesse usando somente uma descrição local do sistema, sem a exploração de todos os estados do modelo ([13],[118],[116],[40]). Todos estes métodos apresentam um resultado aproximado para a medida de interesse, dado que somente uma parte do espaço de estados é utilizada. Em [46] (e referências), o autor apresenta métodos numéricos que visam acelerar métodos iterativos de solução de modelos de Markov com um grande espaço de estados, sendo que as matrizes dos modelos considerados são representadas em seu formato tensorial. O autor apresenta algoritmos mais eficientes para efetuar multiplicações entre vetores de probabilidades e a matriz em formato tensorial, sendo esses algoritmos aplicados em métodos tradicionais de solução de sistemas lineares para obter o vetor de probabilidades em estado estacionário da cadeia de Markov.

Podemos dividir os métodos de solução em dois grupos: 1. métodos que fornecem somente um resultado aproximado sem nenhum controle de erro para a aproximação obtida; 2. métodos que fornecem, juntamente com o resultado aproximado, limites inferior e/ou superior para a medida estudada.

Neste capítulo apresentaremos um método aproximado ([20] e [21]), com controle de erro, para o cálculo do valor esperado da recompensa acumulada para sistemas com estados absorventes, por brevidade de notação, definido como ECRA. O método proposto estabelece um intervalo no qual o valor exato da medida de interesse se encontra e, para

efetuarmos uma comparação com o valor exato, calculamos o ponto médio deste intervalo.

Entende-se por um estado absorvente aquele em que, considerando um processo estocástico de tempo discreto, a probabilidade de retorno, em um passo, a um determinado estado i , dado que o processo está em i , é igual a 1 [133]. Isto significa que quando o processo entra em um estado classificado como absorvente, este não pode mais evoluir para qualquer outro estado do sistema.

Como exemplo clássico de modelos com estados absorventes podemos citar sistemas compostos por um conjunto finito de processadores, N , onde o sistema permanece operacional enquanto no mínimo 1 processador está em funcionamento. Quando o último processador em funcionamento falha, o sistema torna-se não operacional, sem possibilidade de recuperação. Podemos exemplificar como medida de interesse a ser estudada neste sistema, o tempo médio até que o evento de falha drástica ocorra. Esta medida é conhecida na literatura como *Mean Time to Failure* (MTTF), ou *Tempo Médio até a Falha do Sistema* ([65],[15] e referências nestes trabalhos).

Como um outro exemplo para a aplicação do método proposto, seja um sistema com diversos componentes diferentes. Se, para este sistema associamos a cada estado uma perda de capital devido ao não funcionamento de um componente, podemos calcular o montante de capital perdido até que o sistema se torne não operacional.

O método que será apresentado neste capítulo possibilita tanto a redução do espaço de estados, quanto a definição do erro da aproximação. Técnicas de definição de limites são utilizadas quando o modelo possui, simultaneamente, grande cardinalidade do espaço de estados e a característica de *stiffness*. A segunda característica auxilia a contornar os problemas decorrentes da primeira. Quando o modelo apresenta a característica de *stiffness*, o processo estocástico permanece a maior parte do tempo em uma pequena parte do espaço de estados. Então, é natural aproximar a medida de interesse considerando o conjunto de estados onde o processo permanece a maior parte do tempo, resultando na diminuição dos estados a serem utilizados pelo método numérico [104].

O método foi desenvolvido durante o estágio de doutoramento na equipe ARMOR/IRISA/FRANÇA, e combina a idéia de complemento estocástico com a técnica *path-based* para reduzir o espaço de estados do modelo a ser resolvido. Para a aplicação do método, apresentamos uma classificação do espaço de estados que se baseia no trabalho apresentado em Bobbio e Trivedi [10]. O método de aproximação, denominado aproximação lenta-rápida, baseia-se parcialmente nas idéias apresentadas em Pourret [116], no que diz respeito a definição do limite superior de permanência em um conjunto de es-

tados, que definiremos posteriormente como *rápidos*. Esses dois trabalhos são destinados ao cálculo da distribuição de probabilidades em estado transiente.

A estrutura deste capítulo é descrita a seguir. A Seção 4.2 apresenta os trabalhos encontrados na literatura que se direcionam a calcular medidas aproximadas para modelos Markovianos com grande espaço de estados, cujas abordagens são próximas as abordagens apresentadas neste capítulo. Além destes métodos, apresentamos também os métodos que serviram de base para o trabalho proposto. Na Seção 4.3 são descritos os conceitos básicos para a aplicação do método, como, por exemplo, a classificação dos estados entre lentos e rápidos. O método proposto é descrito na Seção 4.4. A Seção 4.5 apresenta uma redefinição do conjunto de estados lentos que aumenta a precisão do método proposto, sem impactar significativamente o custo computacional. A Seção 4.6 é destinada a discussão do custo computacional do método. Um grande número de exemplos são mostrados na seção 4.7. Finalizando, a Seção 4.8 apresenta o sumário das contribuições deste capítulo.

4.2 Trabalhos Relacionados

Nesta seção apresentamos os trabalhos encontrados na literatura que mais se aproximam do paradigma do método proposto. Os métodos que serão descritos se dividem em três grupos:

1. O método proposto por Courtois e Semal [17], que calcula o vetor de distribuição de probabilidades em estado estacionário. Este método é o método mais clássico encontrado na literatura e que foi base para os demais métodos desenvolvidos posteriormente para o cálculo aproximado da distribuição em estado estacionário. Abordaremos brevemente este método dado que modelos com estados absorventes podem ser transformados em modelos irreduzíveis;
2. Os métodos propostos em [104] e [91] que calculam medidas de disponibilidade e confiabilidade em modelos irreduzíveis;
3. Os métodos apresentados em [10] e [116] e que foram base para algumas das definições do método proposto neste trabalho.

4.2.1 Método de Courtois e Semal [17]

Nesta seção apresentamos o método aproximado proposto por Courtois e Semal [17] para o cálculo da distribuição, em estado estacionário, de probabilidades dos estados de uma cadeia de Markov \mathcal{G} , quando o espaço de estados do sistema a ser estudado é da ordem de milhões de estados.

Este método é apresentado dado que medidas de interesse do tipo ECRA podem ser reescritas utilizando distribuição de probabilidades em estado estacionário. Para tal, devemos transformar a cadeia original com estados absorventes, em uma cadeia irredutível.

Consideremos, por exemplo, a medida de MTTF. O cálculo da medida de MTTF em uma cadeia com estados absorventes é equivalente ao cálculo da distribuição em estado estacionário para uma cadeia irredutível. Se modificarmos a cadeia \mathcal{Y} , com um único estado absorvente a , conectando a a todos os outros estados i da cadeia e associando a probabilidade de transição entre o estado a e cada estado i igual a distribuição de probabilidades inicial $Pr(\mathcal{Y}(0) = i)$, então, a partir da distribuição da nova cadeia, calculamos a medida MTTF. Denotemos o MTTF por $E(T)$. Se π_a é a fração de tempo que o processo \mathcal{Y} passou no estado absorvente a , quando consideramos que o tempo t de observação tende ao infinito e se λ é a taxa de saída deste estado, temos a seguinte relação entre o MTTF e a probabilidade em estado estacionário π_a [39]:

$$\pi_a = \frac{1}{1 + \lambda E(T)}. \quad (4.1)$$

Seja a seguinte descrição para o método de Courtois e Semal [17], com uma notação que se restringirá a esta seção.

Seja \mathcal{G} uma cadeia de Markov em tempo contínuo, e \mathbf{G} a sua matriz geradora infinitesimal. Para a apresentação do método de [17], particionemos a matriz $\mathbf{Q} = \mathbf{G}/\Lambda + \mathbf{I}$, após a uniformização do processo \mathcal{G} , da seguinte forma:

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_{00} & \mathbf{Q}_{01} & \mathbf{Q}_{02} & \cdots & \mathbf{Q}_{0N} \\ \mathbf{Q}_{10} & \mathbf{Q}_{11} & \mathbf{Q}_{12} & \cdots & \mathbf{Q}_{1N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{Q}_{N0} & \mathbf{Q}_{N1} & \mathbf{Q}_{N2} & \cdots & \mathbf{Q}_{NN} \end{pmatrix}.$$

Cada submatriz $(\mathbf{Q}_{00}, \mathbf{Q}_{11}, \dots, \mathbf{Q}_{NN})$ corresponde a um subconjunto particular de estados. Estes subconjuntos de estados formam uma partição \mathcal{P} do espaço de estados. Seja \mathcal{F}_i o subconjunto de estados da partição \mathcal{P} correspondente a submatriz \mathbf{Q}_{ii} . Consideremos

um exemplo simples, onde a partição possui cardinalidade 2:

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_{00} & \mathbf{Q}_{01} \\ \mathbf{Q}_{10} & \mathbf{Q}_{11} \end{pmatrix}.$$

Para prosseguirmos na apresentação do método, consideremos o contexto de modelagem de sistemas para estudo de medidas de disponibilidade (*availability*). Neste contexto, suponhamos que \mathcal{F}_0 corresponda aos estados mais “populares”, ou seja, estados com poucos componentes falhos. O subconjunto \mathcal{F}_1 consiste dos demais estados do modelo. Os estados no subconjunto \mathcal{F}_0 são os estados mais prováveis. No entanto, somente uma pequena fração de estados do modelo pertencem a este subconjunto. Para o melhor desempenho dos métodos aproximados, em número de total de multiplicações, \mathcal{F}_0 deve possuir poucas centenas de estados, enquanto que \mathcal{F}_1 possui a ordem de milhares. A idéia é que as submatrizes \mathbf{Q}_{01} e \mathbf{Q}_{10} não sejam geradas.

Uma questão a ser respondida é a seguinte: “o que podemos dizer sobre a distribuição de probabilidades em estado estacionário para os estados pertencentes ao subconjunto \mathcal{F}_0 dada somente a matriz \mathbf{Q}_{00} ?”. A submatriz \mathbf{Q}_{00} não é uma matriz estocástica, pois a soma de cada linha da matriz pode ser menor que 1. A seguir, apresentamos alguns resultados encontrados em Courtois e Semal [17] que fornecem uma resposta a esta pergunta.

Teorema 4.1 (Courtois e Semal [17]) : *Seja \mathbf{L} uma matriz da ordem $n \times n$ com $\mathbf{L} \geq 0$ tal que a soma dos elementos de cada linha é menor ou igual a 1. Seja $\beta(\mathbf{L}) = \{\mathbf{B} | \mathbf{B} \text{ é uma matriz estocástica irredutível da ordem } n \times n \text{ e } \mathbf{B} \geq \mathbf{L}\}$. Seja \mathbf{L}_i , $0 \leq i \leq n - 1$ uma matriz estocástica igual a \mathbf{L} exceto pela i -ésima coluna. \mathbf{L}_i é a matriz \mathbf{L} com elementos na i -ésima linha aumentados da quantia necessária para transformar a matriz \mathbf{L}_i em uma matriz estocástica. Seja \mathbf{z}_i o vetor, em estados estacionário, dos elementos pertencentes a \mathbf{L}_i . Seja $\mathcal{V}_{\mathbf{L}} = \{\mathbf{v} | \mathbf{v} \text{ é o vetor de probabilidades em estado estacionário para alguma matriz } \mathbf{B} \in \beta(\mathbf{L})\}$. Seja $\mathcal{Z}_{\mathbf{L}} = \{\mathbf{v} | \exists \beta_i, 0 \leq i \leq n - 1 \text{ tal que } \sum \beta_i = 1, \mathbf{v} = \sum_{i=0}^{n-1} \beta_i \mathbf{z}_i\}$. Então, $\mathcal{V}_{\mathbf{L}} = \mathcal{Z}_{\mathbf{L}}$.*

Com respeito a matriz \mathbf{Q}_{00} que é a principal submatriz do processo estocástico com grande espaço de estados, o Teorema 1 possui uma interpretação probabilística [17]. A matriz \mathbf{Q}_{00} tem o mesmo papel da matriz \mathbf{L} do teorema enunciado. Consideremos que o sistema comece em um estado pertencente a \mathcal{F}_0 e que a cada transição do processo original feita para um estado não pertencente a \mathcal{F}_0 esta é redirecionada a um estado $i \in \mathcal{F}_0$. Para cada escolha de um estado de retorno i , existe uma matriz correspondente \mathbf{L}_i formada pelo aumento dos valores dos elementos na i -ésima linha de \mathbf{Q}_{00} . O vetor em estado estacionário de \mathbf{L}_i (i.e, \mathbf{z}_i) é a probabilidade condicional dado que cada transição de retorno

a \mathcal{F}_0 é feita pelo estado i . Adicionalmente, o teorema prova que a probabilidade condicional para os estados em \mathcal{F}_0 (denotado por \mathbf{v}_0) é a solução (autovetor) de um elemento do conjunto $\beta(\mathbf{Q}_{00})$. Mais importante, dado os vetores \mathbf{z}_i , $1 \leq i \leq n$, então \mathbf{v}_0 , o vetor de probabilidades condicional, é a combinação linear de \mathbf{z}_i .

Seja o seguinte teorema, também encontrado em [17]:

Teorema 4.2 (Courtois e Semal [17]) : *(Agregação Exata) Seja \mathbf{Q} particionada como definida anteriormente. Consideremos uma matriz estocástica, $N \times N$, \mathbf{Q}_{Ag} , tal que:*

$$\mathbf{Q}_{Ag}[I, J] = \mathbf{v}_I \mathbf{Q}_{IJ} \mathbf{1}^T.$$

Informalmente, $\mathbf{Q}_{Ag}[IJ]$ é a probabilidade em estado estacionário para transicionar de um estado do agregado J (i.e \mathcal{F}_j) dado que o sistema está em algum estado do agregado I . Se $X = (X_1, X_2, \dots, X_N)$ é o vetor em estado estacionário para \mathbf{Q}_{Ag} , então, X_i , $1 \leq i \leq N$ é a probabilidade em estado estacionário do processo se encontrar em um estado pertencente a \mathcal{F}_i no modelo original \mathbf{Q} .

Para aplicar o Teorema 2 e obter a distribuição de probabilidades em estado estacionário, e, conseqüentemente a medida de MTTF, por exemplo, é necessário, primeiramente, obter todas as probabilidades condicionais. Estes cálculos se tornam proibitivos para a classe de modelos abordada neste capítulo, onde o espaço de estados é extremamente grande.

4.2.2 Métodos de Muntz *et al* [104] e de Mahevas *et Rubino* [91]

Semelhantemente ao método proposto neste capítulo, os trabalhos de Muntz *et al* [104] e de Mahevas e Rubino [91] calculam o valor esperado da recompensa acumulada através de recompensas atribuídas a cada estado da cadeia de Markov que representa o sistema em estudo. Esses trabalhos serão brevemente apresentados nesta seção.

Esses dois métodos encontrados na literatura são utilizados para obter a medida de interesse em cadeias irredutíveis, ou seja, sem estados absorventes. Os autores em [104] mostram como calcular os limites inferior e superior para a medida de disponibilidade assintótica, evitando calcular diversos submodelos com o objetivo de estabelecer as probabilidades condicionais como no trabalho apresentado por Courtois e Semal [17]. A idéia principal é particionar o espaço de estados em diversas classes, em uma ordem específica C_0, C_1, \dots tal que quanto maior o índice da classe, menor é a probabilidade

do sistema visitá-la. Em outras palavras, a cadeia permanecerá a maior parte do tempo nas primeiras classes. Um exemplo natural é agrupar em uma mesma classe C_m todos os estados do processo que representam a evolução de um sistema sujeito a falhas e reparos e que possuem exatamente m componentes em funcionamento. Assim, todos os estados pertencentes a uma classe k são agregados em um único estado, o que resulta em um modelo com um novo espaço de estados que é muito menor que o espaço de estados do modelo original. As medidas são estimadas a partir deste novo processo com espaço de estados reduzido.

Uma das condições necessárias para a aplicabilidade do método em [104] é que o evento de reparo possa ocorrer em qualquer estado de qualquer uma das classes C_i . Assim, os autores em [91] propõem modificações no método de [104], possibilitando tratar os sistemas que não obedecem a esta condição.

Os métodos descritos nesta seção se assemelham ao método proposto neste capítulo por representar o sistema em estudo através de modelos Markovianos com recompensa e por definir esta medida com subconjuntos do espaço de estados original do modelo, obtendo as medidas de interesse através da resolução de sistemas lineares bem menores em dimensão. Tanto em [104] quanto em [91], a estrutura do modelo a ser resolvido é classificada como *left-to-right*, com transições lentas da esquerda para direita e transições rápidas da direita para esquerda, ou seja, dada as classes $\{C_0, C_1, C_2, \dots, C_N\}$, as transições lentas ocorrem, de um estado i para um estado $i + 1$, por exemplo entre os estados C_0 e C_1 , e as transições rápidas ocorrem entre um estado $i + 1$ e i , por exemplo entre C_1 e C_0 .

Para obter a medida valor esperado da recompensa acumulada até a absorção utilizando os métodos [91] e [104], temos que, primeiramente, modificar a estrutura da cadeia de Markov que modela o sistema em estudo. Semelhantemente a cadeia utilizada pelos métodos supracitados, a cadeia resolvida pelo método proposto de aproximação também possui transições lentas da esquerda para a direita e transições rápidas da direita para esquerda, exceto que, existe um estado na extrema direita da cadeia de Markov que é absorvente, que denominaremos por a . Para simplificar a argumentação, vamos considerar que somente exista um estado inicial, onde o sistema está totalmente operacional (que é a situação mais comum), denominado estado 0. A partir da transformação da cadeia original com estados absorventes para uma cadeia irreduzível, podemos calcular, por exemplo, a medida de MTTF, conforme mostrado na Seção 4.2.1, após a adição de uma transição entre o estado a e o estado 0.

Podemos observar na Equação 4.1 que somente é necessário a definição de π_a para

obter o MTTF. Dois pontos principais impossibilitam a aplicação direta dos métodos apresentados em [91] e [104]:

1. Estes métodos são precisos no cálculo da distribuição de probabilidades para estados que se encontram mais a esquerda da cadeia de Markov, dado que o processo sendo modelado se concentra a maior parte do tempo nestes estados. No entanto, para o caso do MTTF, por exemplo, é necessário informações precisas sobre um único estado, à extrema direita da cadeia de Markov;
2. Todos os teoremas provados para a aplicação dos métodos apresentados em [91] e [104] baseiam-se na estrutura *upper hesseberg* da cadeia de Markov irreduzível utilizada. Na aplicação com estados absorventes, existe a transição entre os estados a e 0, destruindo a estrutura necessária a aplicação dos métodos propostos em [91] e [104]. Assim, um estudo mais aprofundado destas técnicas passa a ser necessário, seja por meio da extensão dos teoremas relacionados aos métodos supracitados, ou através de novos teoremas que possibilitam a aplicação desses para cadeias com estados absorventes.

O método proposto neste capítulo baseia-se na própria definição da medida de interesse para obtenção do resultado aproximado e dos limites superior e inferior deste resultado, em contrapartida da aplicação da definição de distribuição de probabilidades utilizada pelos demais métodos.

4.2.3 Métodos de Bobbio *et Trivedi* [10] e de Pourret [116]

Como dito anteriormente, o método proposto neste capítulo utiliza o conceito da divisão do espaço de estados em lentos e rápidos, baseando-se na metodologia apresentada por Bobbio e Trivedi em [10] e utiliza as idéias de limite superior de permanência no conjunto de estados rápidos mostradas anteriormente por Pourret [116]. Os métodos apresentados em [10] e [116] são destinados a solução da distribuição de probabilidades em regime transiente.

Em [10], os autores apresentam um algoritmo para converter uma cadeia de Markov *stiff* em uma cadeia *nonstiff* com um espaço de estados muito menor do que o do modelo original. Após a classificação dos estados em lentos e rápidos, o algoritmo apresentado prossegue classificando os estados rápidos em recorrentes ou transientes. Relembramos que um estado i é classificado como recorrente, se e somente se, o processo inicialmente

4.3 Definições Preliminares do Método Proposto

está em i , ele retorna a este mesmo estado com probabilidade 1 e que um estado i é classificado como transiente se e somente se existe uma probabilidade positiva do processo não retornar a este estado i .

No método, cada um dos subconjuntos de estados rápidos recorrentes é substituído por um único estado lento, enquanto o estado rápido transiente é substituído por um *switch* probabilístico, dado que o tempo de permanência nesses estados se aproxima de zero quando a taxa de saída tende ao infinito. Desta forma, a passagem por esses estados reduz-se somente a escolha de um novo estado com certa probabilidade, não sendo significativo o tempo de permanência. Após este processo de redução, a cadeia de Markov reduzida e *nonstiff* é analisada por métodos tradicionais encontrados na literatura.

Diferentemente de [10], o autor em [116] apresenta um método para obtenção da distribuição de probabilidades em estado transiente, que não utiliza a idéia de agregação dos estados da cadeia. Naquele trabalho, os estados do processo também são classificados em lentos e rápidos. O resultado final possui controle de erro, o que não ocorre em [10], onde para alguns dos subconjuntos de estados a precisão da aproximação depende do grau de agregação entre os estados pertencentes ao subconjunto classificado como rápido e os demais estados do processo.

O resultado mais relevante do trabalho apresentado em [116] para o trabalho apresentado neste capítulo, refere-se a maneira como o tempo de permanência no conjunto de estados rápidos é limitado superiormente. Maiores detalhes serão descritos na Seção 4.4.2.

4.3 Definições Preliminares do Método Proposto

Nesta seção descreveremos as definições básicas, o modelo a ser estudado, a medida de interesse a ser calculada e a classificação de estados utilizada. Na Seção 4.3.3, apresentaremos um pequeno modelo que será utilizado para ilustrar cada etapa do método proposto.

4.3.1 Notação

A notação em negrito é utilizada para matrizes, em letra maiúscula, como por exemplo \mathbf{Q} e para vetores, em letra minúscula, como por exemplo $\boldsymbol{\varrho}$. O elemento específico de uma

4.3 Definições Preliminares do Método Proposto

matriz \mathbf{Q} será denotado por $Q(i, j)$ e de um vetor \boldsymbol{q} por q_i , onde i e j indexam um estado dentro do espaço de estados do modelo.

Para representarmos um subconjunto de elementos de um determinado vetor, utilizaremos a letra minúscula e em negrito, com índice superior em maiúscula. Neste índice, teremos o processo e o conjunto de estados a serem considerados. Por exemplo, temos $\boldsymbol{q}^{\mathcal{Y}, \mathcal{S}}$, relacionado ao processo \mathcal{Y} e ao subconjunto \mathcal{S} . Se somente uma letra está presente no índice, esta estará relacionada com o processo em questão. A notação utilizada neste capítulo está listada nas Tabelas 4.1 e 4.2.

4.3.2 Modelo

O sistema a ser estudado é representado por uma cadeia de Markov homogênea de tempo contínuo $\mathcal{X} = \{X(t), t \geq 0\}$, com espaço de estados finito $\mathcal{M} = \{1, \dots, M, M + 1\}$ e gerador infinitesimal \mathbf{Q} , sendo cada elemento desta matriz representado por $Q(i, j)$. A taxa de saída do i -ésimo estado é igual a $q_i = -\sum_{j \neq i} Q(i, j)$.

Em relação as características dos estados que descrevem o sistema estudado, o estado $M + 1$ é considerado absorvente: no momento em que o processo estocástico visita este estado, podemos dizer que ele “morre”, ou seja, nenhum evento será disparado e, portanto, não existe mais a possibilidade de se transicionar para outro estado da cadeia \mathcal{X} (Em alguns casos, este estado pode representar a agregação de vários estados.). Os demais estados da cadeia são denominados transientes e representados pelo conjunto $\Omega = \{1, 2, \dots, M\}$, com cardinalidade $|\Omega|$.

Supõe-se que a partir de qualquer estado $i \leq M$, exista um caminho para o estado $M + 1$, definido pela seqüência de estados $(i, i_1, i_2, \dots, i_k, M + 1)$, com $Q(i, i_1)Q(i_1, i_2) \dots Q(i_k, M + 1) > 0$. Seguindo esta construção, a cadeia de Markov \mathcal{X} será absorvida após um tempo aleatório T , definido como:

$$T = \inf\{t \geq 0 : X(t) = M + 1\} \quad (4.2)$$

com $\Pr(T < \infty) = 1$, ou seja, com probabilidade 1 o sistema alcançará o estado absorvente em um tempo $T < \infty$.

Associada a cada estado $i \leq M$, existe uma recompensa r_i , que como definida no capítulo precedente, é um número real interpretado como a taxa com que o sistema gera benefícios ou acumula custos. Por exemplo, se o estado i possui uma recompensa $r_i = 5$ e o processo permanece 10 unidades de tempo neste estado, a recompensa atribuída ao

4.3 Definições Preliminares do Método Proposto

Tabela 4.1: Descrição dos processos, matrizes e vetores utilizados durante este capítulo com seus respectivos significados.

Termo	Descrição
\mathcal{X}	Cadeia de Markov em tempo contínuo
Q	Matriz geradora infinitesimal da cadeia \mathcal{X}
Ω	Conjunto dos estados transientes da cadeia \mathcal{X}
R_∞	Recompensa acumulada por \mathcal{X} até a absorção
ECRA	Valor esperado da recompensa acumulada até absorção
\mathbf{r}	Vetor de recompensas atribuídas a cada estado da cadeia \mathcal{X}
$\boldsymbol{\rho}$	Vetor do valor esperado da recompensa acumulada condicionado no estado inicial i (cadeia \mathcal{X})
$\boldsymbol{\alpha}$	Vetor da distribuição inicial de probabilidades da cadeia \mathcal{X}
\mathcal{F}	Conjunto de estados classificados como rápidos
\mathcal{S}	Conjunto de estados classificados como lentos
\mathcal{Y}	Cadeia de Markov em tempo discreto associada a \mathcal{X}
U	Matriz de probabilidades associada a cadeia \mathcal{Y}
$R_\infty^{\mathcal{Y}}$	Recompensa acumulada pela cadeia \mathcal{Y} até a absorção
$\boldsymbol{\rho}^{\mathcal{Y}}$	Vetor do valor esperado da recompensa acumulada condicionado no estado inicial i (cadeia \mathcal{Y})
$R_\infty^{\mathcal{Y},\mathcal{S}}$	Recompensa acumulada pela cadeia \mathcal{Y} , restrito ao conjunto \mathcal{S} , até a absorção
$R_\infty^{\mathcal{Y},\mathcal{F}}$	Recompensa acumulada pela cadeia \mathcal{Y} , restrito ao conjunto \mathcal{F} , até a absorção
$\boldsymbol{\rho}^{\mathcal{Y},\mathcal{S}}$	Vetor do valor esperado da recompensa acumulada condicionado no estado inicial i (cadeia \mathcal{Y} e conjunto \mathcal{S})
$\boldsymbol{\rho}^{\mathcal{Y},\mathcal{F}}$	Vetor do valor esperado da recompensa acumulada condicionado no estado inicial i (cadeia \mathcal{Y} e conjunto \mathcal{F})
$\tilde{\mathcal{Y}}$	Cadeia de Markov reduzida a partir de \mathcal{Y} , eliminando o conjunto \mathcal{F}

4.3 Definições Preliminares do Método Proposto

Tabela 4.2: Descrição dos processos, matrizes e vetores utilizados durante este capítulo com seus respectivos significados.

Termo	Descrição
\tilde{U}	Submatriz do complemento estocástico em relação a $\mathcal{S} \cup \{M + 1\}$
$\tilde{\varrho}$	Vetor limite inferior para ϱ^y
U'	Limite inferior para a matriz \tilde{U}
ϱ'	Vetor limite inferior para $\tilde{\varrho}$
U''	Matriz diferença entre a matriz \tilde{U} e U'
σ	Limite superior para o tempo de permanência no conjunto \mathcal{F}
V	Matriz com o número médio de visitas considerando os estados em \mathcal{F}
σB	Limite superior para a matriz U''
φ	Vetor limite superior para $\varrho^{y, \mathcal{S}}$
ν	Vetor número médio de visitas ao conjunto \mathcal{F} , considerando os estados em \mathcal{S}
r^*	Recompensa máxima associada a qualquer estado em \mathcal{F}
$u_{\mathcal{F}}$	Vetor com as probabilidades de transicionar, de um estado em \mathcal{S} para o conjunto \mathcal{F}
τ	Vetor limite superior para o número médio de visitas a \mathcal{F} , a partir dos estados em \mathcal{S}

4.3 Definições Preliminares do Método Proposto

processo será igual a 50. Temos que a recompensa acumulada até absorção é definida como a seguinte variável aleatória:

$$R_\infty = \int_0^\infty r_{X_t} dt.$$

Um caso particular para medida R_∞ ocorre quando para todo $i \leq M$, temos $r_i = 1$. Neste caso, $R_\infty = T$ (definido pela Equação 4.2). Mais especificamente, quando o processo \mathcal{X} representa o estado de um sistema sujeito a falhas e reparos e se os estados $1, \dots, M$ representam o sistema em funcionamento e $M + 1$ representa o sistema falho, $E(R_\infty) = E(T)$ é a medida de confiabilidade MTTF. O objetivo do método proposto é o cálculo do valor esperado da variável aleatória R_∞ , chamada valor esperado da recompensa acumulada até a absorção, ou seja, $E(R_\infty)$.

Denotemos por \mathbf{r} o vetor coluna indexado em Ω cuja a i -ésima entrada é igual a r_i e por $\varrho_i = E_i(R_\infty)$, onde $E_i(\cdot)$ denota o operador valor esperado condicionado que o estado inicial do sistema é o estado $i \in \Omega$. $\boldsymbol{\varrho}$ é o vetor indexado em Ω cuja a i -ésima entrada é ϱ_i . Desta forma:

$$ECRA_i = \varrho_i$$

e, considerando o vetor $\boldsymbol{\alpha}$ o vetor linha com as probabilidades iniciais, temos que:

$$ECRA = \boldsymbol{\alpha}\boldsymbol{\varrho}. \tag{4.3}$$

4.3.3 Exemplo Base

Nesta seção apresentamos um exemplo simplificado que será utilizado para introduzir os conceitos relativos ao método de aproximação proposto.

A cadeia de Markov em tempo contínuo \mathcal{X} de um sistema hipotético, é mostrada na Figura 4.1. As taxas do modelo e seus respectivos valores são: $\alpha = 1e-5, \beta = 1, \gamma = 1e-5, \phi = 1e-4, \delta = 10e-6$.

A recompensa atribuída r_i a cada estado transiente i é igual a 1. A evolução da cadeia é bem simples: transições ocorrem entre os estados transientes até o momento $T < \infty$ em que o processo é absorvido pelo estado a .

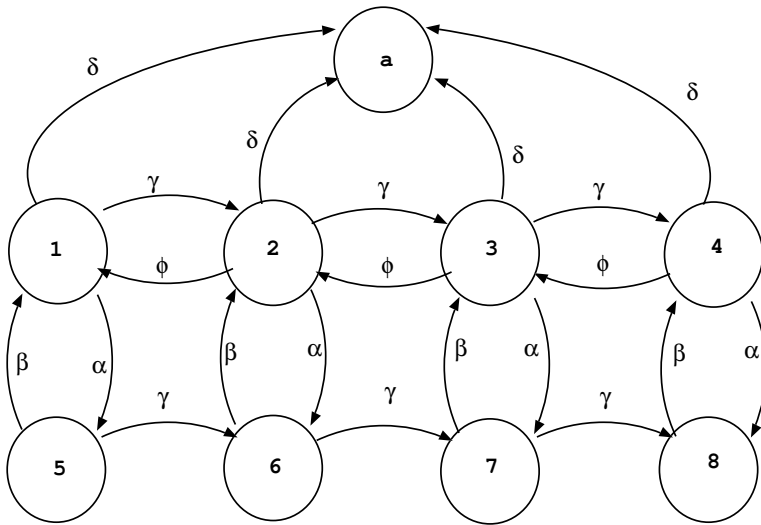


Figura 4.1: Cadeia de Markov para um exemplo simples que será usado como base para a apresentação do método proposto.

4.3.4 Classificação dos Estados

Suponhamos que as taxas de transição no sistema possam ser divididas em dois grupos de valores e seja um número positivo θ previamente definido. Uma transição (i, j) , entre o estado i e o estado j , é classificada como lenta se $0 < Q(i, j) < \theta$ e como rápida se $Q(i, j) \geq \theta > 0$.

Esta definição particiona o espaço de estados Ω em três subconjuntos:

1. \mathcal{F} - Conjunto de estados rápidos, composto pelos estados que possuem pelo menos uma taxa de saída classificada como rápida;
2. \mathcal{S} - Conjunto de estados lentos, composto pelos estados que não possuem nenhuma transição rápida. Nos sistemas estudados consideramos que a distribuição de probabilidades inicial está concentrada em \mathcal{S} ;
3. Conjunto de estados absorventes - Neste conjunto consideramos o estado $M + 1$, que, no modelo real, pode significar um único estado ou então, a agregação de diversos estados.

Consideremos que a matriz \mathbf{Q} , cujos os elementos representam as taxas de transições entre os estados do modelo, seja reordenada de tal maneira que os estados que pertençam ao conjunto \mathcal{S} sejam numerados de 1 até $|\mathcal{S}|$ e os estados pertencentes ao conjunto \mathcal{F} de

4.3 Definições Preliminares do Método Proposto

$|\mathcal{S}| + 1$ até $|\mathcal{S}| + |\mathcal{F}| = M$. Esta matriz, com todos os elementos, desconsiderando o estado absorvente $\{M + 1\}$, é particionada da seguinte maneira:

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_{SS} & \mathbf{Q}_{S\mathcal{F}} \\ \mathbf{Q}_{\mathcal{F}S} & \mathbf{Q}_{\mathcal{F}\mathcal{F}} \end{pmatrix}.$$

Para aplicação do método, assumimos que a cadeia \mathcal{X} não possua ciclos compostos somente por transições rápidas, ou, analogamente, por estados classificados como rápidos. Esta situação é muito comum no caso de modelos de confiabilidade, que é um dos principais tipos de modelo que consideramos neste capítulo, onde as transições rápidas estão associadas à transições de reparo. No entanto, em alguns modelos a escolha do parâmetro θ pode gerar ciclos de transições rápidas, o que invalida a condição para a aplicação do método. Esta condição para a aplicabilidade do método deve-se a maneira de como o tempo máximo no conjunto de estados \mathcal{F} é calculado. A seção 4.4.2 detalha o procedimento para a determinação desse limite. Em [10] os autores propõem um algoritmo para tratar os casos onde a cadeia do modelo possui ciclos de transições rápidas.

Exemplo Base: Seja o exemplo da Seção 4.3.3. O valor do parâmetro θ utilizado para definir os estados lentos e rápidos, é igual a 1. Neste caso, temos os seguintes conjuntos definidos:

- $\mathcal{S} = \{1, 2, 3, 4\}$;
- $\mathcal{F} = \{5, 6, 7, 8\}$;
- e o estado absorvente a .

4.3.5 Uniformização

Denotemos por \mathcal{Y} a cadeia de Markov de tempo discreto (DTMC), associada com o processo \mathcal{X} , obtida através da aplicação do método de uniformização do segundo processo, em relação a taxa de uniformização η , conforme definido no Capítulo 2. A matriz \mathbf{U} representa a matriz de probabilidades de \mathcal{Y} . Esta matriz é particionada seguindo a mesma partição proposta para a matriz \mathbf{Q} . Assim:

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_{SS} & \mathbf{U}_{S\mathcal{F}} \\ \mathbf{U}_{\mathcal{F}S} & \mathbf{U}_{\mathcal{F}\mathcal{F}} \end{pmatrix}$$

4.3 Definições Preliminares do Método Proposto

A medida ECRA definida previamente em relação ao processo \mathcal{X} pode ser calculada através da divisão da mesma medida definida em relação ao processo \mathcal{Y} pelo fator de uniformização η . Assim, se definimos:

$$R_{\infty}^{\mathcal{Y}} = \sum_{k=0}^{\infty} r_{\mathcal{Y}(k)},$$

então

$$\boldsymbol{\varrho}^{\mathcal{Y}} = \boldsymbol{\alpha}(\mathbf{I} - \mathbf{U})^{-1}\mathbf{r}$$

e

$$\text{ECRA} = \frac{1}{\eta}\boldsymbol{\alpha}(\mathbf{I} - \mathbf{U})^{-1}\mathbf{r}. \quad (4.4)$$

Utilizando a mesma notação definida previamente, $\boldsymbol{\varrho} = \boldsymbol{\varrho}^{\mathcal{Y}}/\eta$. Resumindo, podemos aplicar o método de aproximação proposto utilizando a cadeia \mathcal{Y} e obter a medida ECRA definida na cadeia \mathcal{X} contínua e que representa o sistema a ser estudado.

Observando a Equação 4.4, o cálculo da medida ECRA é um problema linear. A idéia principal do método apresentado neste capítulo é obter um valor aproximado para esta medida, através dos limites inferior e superior, que representam o controle de erro. Neste método, somente uma parte do espaço de estados é utilizado, e, conseqüentemente, um sistema linear com um número muito menor de equações que a cardinalidade de Ω é resolvido.

O método proposto é extremamente interessante para os casos onde $|\Omega| \gg 1$ e quando a cadeia de Markov \mathcal{X} possui a característica *stiffness*. A primeira característica influencia diretamente no custo computacional do método proposto. A segunda, influencia no quão distante estará o limite inferior do superior.

4.3.6 Decompondo a Recompensa Acumulada

O próximo passo para a aplicação do método proposto é a decomposição da recompensa acumulada. Esta decomposição é estabelecida para que os limites da medida ECRA possam ser definidos. Assim:

$$R_{\infty}^{\mathcal{Y},\mathcal{S}} = \sum_{k=0}^{\infty} r_{\mathcal{Y}(k)}1_{\mathcal{Y}(k) \in \mathcal{S}}, \quad R_{\infty}^{\mathcal{Y},\mathcal{F}} = \sum_{k=0}^{\infty} r_{\mathcal{Y}(k)}1_{\mathcal{Y}(k) \in \mathcal{F}},$$

onde $R_{\infty}^{\mathcal{Y},\mathcal{S}}$ é a recompensa acumulada, pelo processo \mathcal{Y} , nos estados lentos e $R_{\infty}^{\mathcal{Y},\mathcal{F}}$ é a recompensa acumulada, pelo mesmo processo, nos estados rápidos. O acúmulo de

4.3 Definições Preliminares do Método Proposto

recompensa ocorre até que o processo \mathcal{Y} seja “absorvido”. Temos que $R_\infty^{\mathcal{Y}} = R_\infty^{\mathcal{Y},\mathcal{S}} + R_\infty^{\mathcal{Y},\mathcal{F}}$ e, da mesma forma, $E(R_\infty^{\mathcal{Y}}) = E(R_\infty^{\mathcal{Y},\mathcal{S}}) + E(R_\infty^{\mathcal{Y},\mathcal{F}})$.

Consideremos os elementos $\varrho^{\mathcal{Y}i} = E(R_\infty^{\mathcal{Y}} | \mathcal{Y}(0) = i)$, $\varrho_i^{\mathcal{Y},\mathcal{S}} = E(R_\infty^{\mathcal{Y},\mathcal{S}} | Y(0) = i)$ e $\varrho_i^{\mathcal{Y},\mathcal{F}} = E(R_\infty^{\mathcal{Y},\mathcal{F}} | \mathcal{Y}(0) = i)$. A recompensa acumulada pelo processo \mathcal{Y} dado um estado inicial i , $\varrho_i^{\mathcal{Y}}$, é dado pela soma $\varrho_i^{\mathcal{Y},\mathcal{S}} + \varrho_i^{\mathcal{Y},\mathcal{F}}$. Conforme definido anteriormente, assumimos que os estados iniciais pertencem ao conjunto \mathcal{S} (situação comum em modelos de confiabilidade, por exemplo). A mesma definição, em forma vetorial, pode ser escrita, respectivamente, da seguinte forma: $\varrho^{\mathcal{Y}}$, $\varrho^{\mathcal{Y},\mathcal{S}}$ e $\varrho^{\mathcal{Y},\mathcal{F}}$. Finalizando, a seguinte relação também é verdadeira: $\varrho^{\mathcal{Y}} = \varrho^{\mathcal{Y},\mathcal{S}} + \varrho^{\mathcal{Y},\mathcal{F}}$.

4.3.7 Complemento Estocástico

Para os sistemas estudados neste capítulo, onde o número total de estados é muito grande para que os métodos tradicionais de solução sejam aplicados, é natural pensarmos em uma maneira de analisar a cadeia original utilizando duas ou mais cadeias menores. Os estados mais relevantes para a obtenção da medida desejada são preservados, redistribuindo as probabilidades anteriormente relacionadas com os estados que serão eliminados.

A teoria de complemento estocástico é amplamente utilizada na literatura, em diversas aplicações, sendo uma delas de auxiliar na redução do espaço de estados a ser utilizado por um método de solução ([95] e referências). No método proposto neste capítulo, o papel da teoria de complemento estocástico será na redução do espaço de estados Ω .

Seja $\tilde{\mathcal{Y}}$ a cadeia obtida após reduzirmos o processo \mathcal{Y} utilizando a técnica de complemento estocástico em relação ao subconjunto de estados $\mathcal{S} \cup \{M + 1\}$.

O processo $\tilde{\mathcal{Y}}$ decorrente do complemento estocástico, pode ser interpretado como sendo equivalente ao processo \mathcal{Y} quando este último está nos estados pertencentes a $\mathcal{S} \cup \{M + 1\}$, desconsiderando o tempo que o processo \mathcal{Y} passa nos estados pertencentes ao conjunto \mathcal{F} .

Mais detalhadamente, considere o caso em que no processo \mathcal{Y} ocorre uma transição de um estado $i \in \mathcal{S}$ para algum estado pertencente ao conjunto \mathcal{F} e que depois o processo \mathcal{Y} retorna ao conjunto de estados $\mathcal{S} \cup \{M + 1\}$ pelo estado k . No processo $\tilde{\mathcal{Y}}$, este tipo de transição é representada através de uma transição direta entre os estados i e k . A matriz

4.3 Definições Preliminares do Método Proposto

de probabilidades para este o processo $\tilde{\mathcal{Y}}$ é particionada da seguinte maneira:

$$\begin{pmatrix} \tilde{\mathbf{U}} & \tilde{\mathbf{u}} \\ 0 & 1 \end{pmatrix}$$

onde

$$\tilde{\mathbf{U}} = \mathbf{U}_{SS} + \mathbf{U}_{S\mathcal{F}}(\mathbf{I} - \mathbf{U}_{\mathcal{F}\mathcal{F}})^{-1}\mathbf{U}_{\mathcal{F}S} \quad (4.5)$$

e $\tilde{\mathbf{u}} = \mathbf{1} - \tilde{\mathbf{U}}\mathbf{1}$, e $\mathbf{1}$ é um vetor coluna, indexado no conjunto \mathcal{S} , com todas as entradas iguais a 1.

Conforme definido anteriormente para o processo \mathcal{Y} , seja a recompensa média acumulada até a absorção, para o processo $\tilde{\mathcal{Y}}$, dada por $E(R_\infty^{\tilde{\mathcal{Y}}})$ e o vetor $\tilde{\boldsymbol{\rho}}$ indexado em \mathcal{S} cuja a i -ésima entrada é dada por $\tilde{\rho}_i = E(R_\infty^{\tilde{\mathcal{Y}}} | \tilde{\mathcal{Y}}(0) = i)$. Devido a definição de complemento estocástico, onde desconsideramos o tempo passado pelo processo \mathcal{Y} no subconjunto de estados \mathcal{F} , temos que:

$$\tilde{\boldsymbol{\rho}} = \boldsymbol{\rho}^{\mathcal{Y},\mathcal{S}}. \quad (4.6)$$

Esta relação será importante para a definição do limite inferior para a medida aproximada, conforme veremos nas seções seguintes.

Exemplo Base: Seguindo a aplicação dos conceitos do método proposto, as seguintes matrizes, após a uniformização utilizando o parâmetro de uniformização $\eta = 1.1$, são assim definidas:

$$\mathbf{U}_{SS} = \begin{pmatrix} 9.9998e-1 & 9.0909e-6 & 0 & 0 \\ 9.0909e-5 & 9.9989e-1 & 9.0909e-6 & 0 \\ 0 & 9.0909e-5 & 9.9989e-1 & 9.0909e-6 \\ 0 & 0 & 9.0909e-5 & 9.9989e-1 \end{pmatrix}$$

$$\mathbf{U}_{\mathcal{F}\mathcal{F}} = \begin{pmatrix} 9.0900e-2 & 9.0909e-6 & 0 & 0 \\ 0 & 9.090e-2 & 9.0909e-6 & 0 \\ 0 & 0 & 9.090e-2 & 9.0909e-6 \\ 0 & 0 & 0 & 9.0909e-2 \end{pmatrix}$$

$$\mathbf{U}_{S\mathcal{F}} = \begin{pmatrix} 9.0909e-6 & 0 & 0 & 0 \\ 0 & 9.090e-6 & 0 & 0 \\ 0 & 0 & 9.0909e-6 & 0 \\ 0 & 0 & 0 & 9.0909e-6 \end{pmatrix}$$

$$\mathbf{U}_{\mathcal{F}S} = \begin{pmatrix} 9.0909e-1 & 0 & 0 & 0 \\ 0 & 9.0909e-1 & 0 & 0 \\ 0 & 0 & 9.0909e-1 & 0 \\ 0 & 0 & 0 & 9.0909e-1 \end{pmatrix}$$

E a submatriz \tilde{U} :

$$\tilde{U} = \begin{pmatrix} 9.9999e-1 & 9.0909e-6 & 9.0907e-16 & 9.0907e-21 \\ 9.0909e-5 & 9.9999e-1 & 9.0909e-6 & 9.0908e-16 \\ 0 & 9.0909e-5 & 9.9999e-1 & 9.0909e-6 \\ 0 & 0 & 9.0909e-5 & 9.9999e-1 \end{pmatrix}$$

4.4 O Método de Aproximação Proposto

Nesta seção descrevemos o procedimento para obter os limites superior e inferior da medida de interesse ECRA. Primeiramente, estabelecemos um limite inferior da medida real $\varrho^{\mathcal{Y}}$ ($\varrho^{\mathcal{Y}}$ é igual a soma dos vetores $\varrho^{\mathcal{Y},S}$ e $\varrho^{\mathcal{Y},F}$), através do cálculo do vetor $\tilde{\varrho}$, para os casos em que conhecemos \tilde{U} e do vetor ϱ' para os casos em que a determinação desta matriz se torna inviável. A partir desta primeira etapa, um limite superior para $\varrho^{\mathcal{Y}}$ é obtido em duas partes: através da definição de um limite superior para $\varrho^{\mathcal{Y},S^1}$ e através de um limite superior para $\varrho^{\mathcal{Y},F}$.

Nas seções seguintes, todas as desigualdades entre vetores ou entre matrizes significam que a desigualdade apresentada aplica-se elemento a elemento. Por exemplo, se \mathbf{a} e \mathbf{b} são vetores de mesma dimensão, com o i -ésimo elemento dado por a_i e b_i respectivamente, então $\mathbf{a} \leq \mathbf{b} \iff a_i \leq b_i$ para todos os elementos i de cada um dos vetores considerados.

4.4.1 Limite Inferior

Conforme definido anteriormente, o vetor $\tilde{\varrho}$ é o limite inferior para o vetor $\varrho^{\mathcal{Y}}$, sendo que $\tilde{\varrho}$ contém o valor exato da medida para o processo uniformizado \mathcal{Y}^2 , considerando cada um dos estados como um estado inicial da cadeia de Markov discreta \mathcal{Y} . Este limite inferior está relacionado com o processo $\tilde{\mathcal{Y}}$, resultante do complemento estocástico aplicado ao processo discreto \mathcal{Y} .

Seja o seguinte sistema linear de equações, que define o limite inferior para $\varrho^{\mathcal{Y}}$:

$$(I - \tilde{U})\tilde{\varrho} = \mathbf{r}. \quad (4.7)$$

¹Este limite superior é necessário dado que o complemento estocástico que será utilizado pelo método é uma aproximação do complemento estocástico exato.

²Assim, temos diretamente o valor da medida ECRA, dada por $(\alpha\varrho^{\mathcal{Y}})/\eta$

4.4 O Método de Aproximação Proposto

A maior limitação inerente a Equação 4.7 é o fato da matriz \tilde{U} ser indeterminada para a grande maioria de modelos considerados neste capítulo. Esta indeterminação decorre do fato que o espaço de estados para os modelos considerados neste trabalho possui uma cardinalidade elevada (milhões de estados). Portanto, a determinação da matriz exata \tilde{U} implica em alto custo computacional em termos de geração da cadeia, de armazenamento das informações e de número total de multiplicações (operações entre matrizes e definição de uma inversa de ordem elevada).

A determinação do limite inferior está atrelada a dois casos distintos: o primeiro, quando a matriz \tilde{U} pode ser calculada com um custo computacional baixo; ou, na grande maioria dos casos, quando somente podemos definir uma matriz aproximada para a matriz \tilde{U} .

Consideremos o caso mais geral, onde o complemento estocástico exato não é conhecido. Para estes modelos, a seguinte técnica, denominada *path-based* ([13],[118],[116],[40]) para a determinação do complemento estocástico será utilizada. A técnica *path-based* consiste em explorar o grafo da cadeia de Markov que modela o sistema passo-a-passo, sem a geração de todos os caminhos e estados da cadeia. No caso específico do método proposto, exploramos um caminho que comece em um estado $i \in \mathcal{S}$ e termine em um estado final $j \in \mathcal{S}$, passando pelo conjunto de estados \mathcal{F} . A medida de interesse a ser determinada é a probabilidade de dado que o processo \mathcal{Y} deixou o conjunto \mathcal{S} pelo estado i este tenha retornado ao conjunto \mathcal{S} pelo estado j , considerando caminhos dentro do conjunto \mathcal{F} . O caminho a ser considerado dependerá do parâmetro N que será descrito a seguir.

Seja um inteiro $N \geq 2$. Para todos os estados i, j pertencentes a \mathcal{S} , construímos um conjunto denominado \mathcal{P}_N , que engloba todos os caminhos que possuem a seguinte forma: $\xi_{ij} = (i, f_1, f_2, \dots, f_{N-2}, j)$, onde os estados f_1, f_2, \dots, f_{N-2} pertencem a \mathcal{F} . Para cada um dos caminhos definidos, a probabilidade associada é a seguinte:

$$p(\xi_{ij}) = U(i, f_1)U(f_1, f_2) \dots U(f_{N-1}, j).$$

Então, a probabilidade do processo evoluir do estado inicial $i \in \mathcal{S}$ ao estado final $j \in \mathcal{S}$, considerando todos os caminhos no interior do conjunto \mathcal{F} com tamanho $N - 2$ é dada por:

$$\sum_{\xi_{ij} \in \mathcal{P}_N} p(\xi_{ij}) = (U_{\mathcal{S}\mathcal{F}} U_{\mathcal{F}\mathcal{F}}^{N-2} U_{\mathcal{F}\mathcal{S}})(ij).$$

Seja a matriz U' definida através da seguinte equação:

$$U' = U_{\mathcal{S}\mathcal{S}} + U_{\mathcal{S}\mathcal{F}} (I + U_{\mathcal{F}\mathcal{F}} + U_{\mathcal{F}\mathcal{F}}^2 + \dots + U_{\mathcal{F}\mathcal{F}}^{N-2}) U_{\mathcal{F}\mathcal{S}}. \quad (4.8)$$

4.4 O Método de Aproximação Proposto

Comparando as Equações 4.8 e 4.5, nota-se que os elementos na matriz \mathbf{U}' são menores ou iguais aos elementos da matriz $\tilde{\mathbf{U}}$, e se ϱ' é a solução para o sistema

$$(\mathbf{I} - \mathbf{U}')\varrho' = \mathbf{r}, \quad (4.9)$$

temos:

$$\varrho' \leq \tilde{\varrho} = \varrho^{\mathcal{Y}, \mathcal{S}} \leq \varrho^{\mathcal{Y}}.$$

e a matriz \mathbf{U}' é uma aproximação para a matriz exata $\tilde{\mathbf{U}}$.

O vetor ϱ' , solução para o sistema linear com $|\tilde{\Omega}|$ variáveis, é o limite inferior para $\tilde{\varrho}$ e por transitividade para o vetor $\varrho^{\mathcal{Y}}$. Para definir a matriz que representa o complemento estocástico aproximado, precisamos explorar o conjunto de estados rápidos cujo a distância é de, no máximo, $N - 2$ a partir do conjunto \mathcal{S} . A exploração destes estados da cadeia é feita através do tipo de busca em extensão³. Conforme iremos mostrar na seção 3.7, limites estreitos são definidos, mesmo considerando pequenos valores de N .

Exemplo Base: Para os valores calculados através do exemplo base, vamos considerar que inicialmente o sistema se encontra no estado $i = 1$ com probabilidade igual a 1. Por conseguinte, consideraremos somente o elemento $i = 1$ de todos os vetores apresentados até o momento.

Primeiramente, o valor exato da medida para a cadeia discretizada do sistema hipotético apresentado é de $\varrho_1^{\mathcal{Y}} = 1.0001e6$. Passemos ao caso onde não consideramos o espaço total de estados. Como dito anteriormente, o valor do limite inferior depende se é possível ou não a determinação da matriz exata $\tilde{\mathbf{U}}$. Para o caso onde conhecemos esta matriz, temos que $\tilde{\varrho}_1 = 9.9991e5$.

Para o caso onde podemos somente calcular a matriz aproximada \mathbf{U}' , devemos escolher o parâmetro N . A título de ilustração, consideremos três valores distintos:

1. $N = 3$. Para este valor de parâmetro, temos que a matriz \mathbf{U}' será definida da seguinte forma: $\mathbf{U}' = \mathbf{U}_{\mathcal{S}\mathcal{S}} + \mathbf{U}_{\mathcal{S}\mathcal{F}}(\mathbf{I} + \mathbf{U}_{\mathcal{F}\mathcal{F}})\mathbf{U}_{\mathcal{F}\mathcal{S}}$. Considerando $i = 1$ e $j = 1$ temos, por exemplo, que um caminho dentro do conjunto \mathcal{F} pode ser escrito como $\{1, 5, 5, 1\}$. Devemos considerar os caminhos que possuem somente uma transição dentro do conjunto \mathcal{F} . A medida resultante após utilizarmos este complemento estocástico aproximado é de $\varrho'_1 = 9.2366e5 < \tilde{\varrho}_1$.

³A busca em extensão é uma estratégia simples em que o nó raiz é expandido primeiro, em seguida todos os sucessores do nó raiz são expandidos, depois os sucessores desses nós e assim por diante.

4.4 O Método de Aproximação Proposto

2. $N = 4$. Para este valor de parâmetro, temos que a matriz \mathbf{U}' será definida da seguinte forma: $\mathbf{U}' = \mathbf{U}_{SS} + \mathbf{U}_{S\mathcal{F}}(\mathbf{I} + \mathbf{U}_{\mathcal{F}\mathcal{F}} + \mathbf{U}_{\mathcal{F}\mathcal{F}}^2)\mathbf{U}_{\mathcal{F}S}$. Considerando $i = 1$ e $j = 2$, por exemplo, devemos considerar os caminhos com no máximo duas transições dentro do conjunto \mathcal{F} . Um exemplo de um caminho com duas transições em \mathcal{F} pode ser escrito da seguinte maneira: $\{1, 5, 5, 6, 2\}$. A medida resultante após utilizarmos este complemento estocástico aproximado é de $\varrho'_1 = 9.9254e5 < \tilde{\varrho}_1$.
3. $N = 5$. Para este valor de parâmetro, temos que a matriz \mathbf{U}' será definida da seguinte forma: $\mathbf{U}' = \mathbf{U}_{SS} + \mathbf{U}_{S\mathcal{F}}(\mathbf{I} + \mathbf{U}_{\mathcal{F}\mathcal{F}} + \mathbf{U}_{\mathcal{F}\mathcal{F}}^2 + \mathbf{U}_{\mathcal{F}\mathcal{F}}^3)\mathbf{U}_{\mathcal{F}S}$. Se temos $i = 1$ e $j = 3$, por exemplo, temos que um caminho dentro do conjunto \mathcal{F} , com três transições é dado por $\{1, 5, 5, 6, 7, 3\}$. Para a construção da matriz \mathbf{U}' , devemos considerar caminhos com no máximo três transições dentro do conjunto \mathcal{F} . A medida resultante, após utilizarmos a matriz \mathbf{U}' , é igual a $\varrho'_1 = 9.9932e5 < \tilde{\varrho}_1$.

Este pequeno exemplo mostra, como esperado, que quanto mais caminhos exploramos dentro do conjunto \mathcal{F} , mais próximo estamos do valor do limite inferior quando o complemento estocástico exato é utilizado.

4.4.2 Limite Superior

Como segunda etapa do método proposto, devemos estabelecer um limite superior para a medida ECRA. A definição deste limite é dividida em dois passos:

1. cálculo de um limite superior para $\varrho^{\mathcal{Y},S}$, dado que somente é possível a definição da matriz aproximada \mathbf{U}' (o caso da matriz $\tilde{\mathbf{U}}$ será apresentado no final desta seção);
2. cálculo de um limite superior para $\varrho^{\mathcal{Y},\mathcal{F}}$.

Como $\varrho^{\mathcal{Y}} = \varrho^{\mathcal{Y},S} + \varrho^{\mathcal{Y},\mathcal{F}}$, teremos um limite superior para $\varrho^{\mathcal{Y}}$ se usarmos um limite superior para $\varrho^{\mathcal{Y},\mathcal{F}}$ e $\varrho^{\mathcal{Y},S}$, e, conseqüentemente, para a medida definida na cadeia contínua \mathcal{X} .

Limite Superior para $\varrho^{\mathcal{Y},S}$

Conforme mostrado na Seção 4.4.1, a matriz \mathbf{U}' é uma aproximação para a matriz exata $\tilde{\mathbf{U}}$, sendo esta última uma submatriz do complemento estocástico em relação ao

4.4 O Método de Aproximação Proposto

conjunto $\mathcal{S} \cup \{M+1\}$. Neste caso, se observamos cada elemento da matriz \mathbf{U}' estes serão menores ou iguais ao elemento correspondente da matriz $\tilde{\mathbf{U}}$. Baseado nesta característica, definimos a matriz $\mathbf{U}'' = \tilde{\mathbf{U}} - \mathbf{U}' \geq 0$, que tem a seguinte forma:

$$\begin{aligned} \mathbf{U}'' &= \mathbf{U}_{\mathcal{S}\mathcal{F}} (\mathbf{U}_{\mathcal{F}\mathcal{F}}^{N-1} + \mathbf{U}_{\mathcal{F}\mathcal{F}}^N + \cdots) \mathbf{U}_{\mathcal{F}\mathcal{S}} \\ &= \mathbf{U}_{\mathcal{S}\mathcal{F}} \mathbf{U}_{\mathcal{F}\mathcal{F}}^{N-1} (\mathbf{I} + \mathbf{U}_{\mathcal{F}\mathcal{F}} + \mathbf{U}_{\mathcal{F}\mathcal{F}}^2 + \mathbf{U}_{\mathcal{F}\mathcal{F}}^3 + \cdots) \mathbf{U}_{\mathcal{F}\mathcal{S}} \\ &= \mathbf{U}_{\mathcal{S}\mathcal{F}} \mathbf{U}_{\mathcal{F}\mathcal{F}}^{N-1} (\mathbf{I} - \mathbf{U}_{\mathcal{F}\mathcal{F}})^{-1} \mathbf{U}_{\mathcal{F}\mathcal{S}}. \end{aligned} \quad (4.10)$$

Cada elemento da matriz $(\mathbf{I} - \mathbf{U}_{\mathcal{F}\mathcal{F}})^{-1}$ é o número médio de visitas a um estado $j \in \mathcal{F}$, começando de um estado $i \in \mathcal{F}$, antes de deixar o conjunto \mathcal{F} , do processo \mathcal{Y} . Somente por questão de notação, escrevemos que:

$$\mathbf{V} = (\mathbf{I} - \mathbf{U}_{\mathcal{F}\mathcal{F}})^{-1}.$$

Para prosseguirmos na definição do limite superior de $\rho^{\mathcal{Y}, \mathcal{S}}$, devemos estabelecer um limite superior para a matriz \mathbf{U}'' . Para tal, devemos enunciar o seguinte lema, adaptado de [116]:

Lema 4.1 *Dado que não existem ciclos compostos por transições do tipo rápidas na cadeia de Markov \mathcal{X} , podemos particionar os estados pertencentes a \mathcal{F} em D classes denominadas $\mathcal{C}_1, \dots, \mathcal{C}_D$, onde \mathcal{C}_1 é a classe de estados diretamente conectados ao conjunto \mathcal{S} através de uma transição rápida, \mathcal{C}_2 é a classe dos estados diretamente conectados a \mathcal{C}_1 através de uma transição rápida, e assim sucessivamente. O inteiro D é, então, o número máximo de transições rápidas entre qualquer estado pertencente a \mathcal{F} e o conjunto \mathcal{S} . Assim, se definirmos:*

$$\lambda = \max_{f \in \mathcal{F}} \sum_{g \in \mathcal{F}, (f,g) \text{ lenta}} Q(f, g)$$

e

$$\mu = \min_{f \in \mathcal{F}} \sum_{g \in \mathcal{F}, (f,g) \text{ rápida}} Q(f, g),$$

e se assumirmos que $\lambda < \mu$, então, qualquer visita do processo \mathcal{Y} em \mathcal{F} tem duração menor, em média, que σ , onde:

$$\sigma = \frac{\eta}{\mu} \frac{1 - (1 - \psi)^D}{\psi(1 - \psi)^{D-1}}, \quad (4.11)$$

com $\psi = \lambda/\mu < 1$. Na classe de sistemas estudados neste capítulo esta característica é comum.

4.4 O Método de Aproximação Proposto

Prova: Seja o estado inicial do processo \mathcal{Y} um estado pertencente ao conjunto \mathcal{F} . Transformemos $\mathcal{S} \cup \{M + 1\}$ em um único estado denotado como estado 0. Denotemos por $\mathcal{I}(k)$ o índice da classe onde $\mathcal{Y}(k)$ se encontra no k -ésimo passo, isto é, $\mathcal{I}(k)$ é igual a d se e somente se o processo $\mathcal{Y}(k)$ se encontra na classe \mathcal{C}_d , convencionando que $\mathcal{I}(k) = 0$ se $\mathcal{Y}(k) = 0$.

Usaremos uma cadeia discreta auxiliar \mathcal{Z} , cuja a variável de estado associada está definida no conjunto de índices $\{0, 1, 2, \dots, D\}$. O estado 0 é um estado absorvente e os demais são estados transientes. A partir de qualquer $d > 0$, transições para $d - 1$ ocorrem com probabilidade $1 - \psi$ e para D com probabilidade ψ . A Figura 4.2 mostra a representação gráfica desta cadeia.

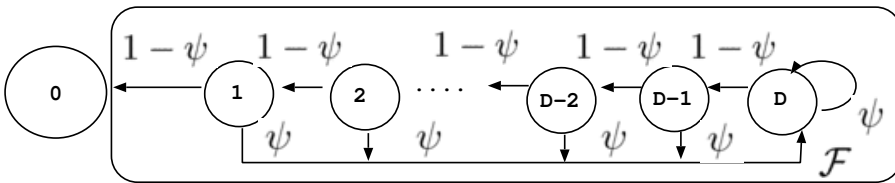


Figura 4.2: Cadeia de Markov discreta \mathcal{Z} .

A prova consiste em primeiramente mostrar que o processo \mathcal{Y} visita um número menor de classes do que o número de estados que o processo \mathcal{Z} visita. Esta afirmação é baseada no fato de que para qualquer $k \geq 0$, a variável aleatória $\mathcal{I}(k)$ é estocasticamente menor que $\mathcal{Z}(k)$ e pode ser provada através de uma recorrência em k .

Para $k = 0$, temos que $\mathcal{I}(k) \leq_{st} \mathcal{Z}(k)$, desde que, por construção, $\mathcal{I}(0) = \mathcal{Z}(0)$. A hipótese de recorrência é que $\mathcal{I}(k) \leq_{st} \mathcal{Z}(k)$. Temos que provar que, $\Pr(\mathcal{I}(k + 1) \geq l)$ é menor que $\Pr(\mathcal{Z}(k + 1) \geq l)$ para toda classe l . Ou seja, a probabilidade do processo \mathcal{I} estar em uma classe maior que l no $(k + 1)$ -ésimo passo é menor que a probabilidade deste mesmo evento no processo \mathcal{Z} .

Escrevendo:

$$\Pr(\mathcal{I}(k + 1) \geq l) = \sum_{d=0}^D p(l, d) \Pr(\mathcal{I}(k) = d),$$

onde $p(l, d) = \Pr(\mathcal{I}(k + 1) \geq l | \mathcal{I}(k) = d)$, e, da mesma forma,

$$\Pr(\mathcal{Z}(k + 1) \geq l) = \sum_{d=0}^D q(l, d) \Pr(\mathcal{Z}(k) = d),$$

onde $q(l, d) = \Pr(\mathcal{Z}(k + 1) \geq l | \mathcal{Z}(k) = d)$. Temos que $q(0, d) = q(1, d) = \dots = q(d - 1, d) = 1$ e que $q(d, d) = q(d + 1, d) = \dots = q(D, d) = \psi$.

4.4 O Método de Aproximação Proposto

Podemos afirmar que $p(l, d) \leq q(l, d) = 1$ para $l = 0, 1, \dots, d-1$. Para $l \geq d$,

$$p(l, d) = \sum_{f \in \mathcal{C}_d} \Pr(\mathcal{I}(k) \geq l \mid \mathcal{Y}(k) = f) \Pr(\mathcal{Y}(k) = f \mid \mathcal{Y}(k) \in \mathcal{C}_d).$$

A probabilidade condicional $\Pr(\mathcal{I}(k) \geq l \mid \mathcal{Y}(k) = f)$ pode ser escrita como:

$$\Pr(\mathcal{I}(k) \geq l \mid \mathcal{Y}(k) = f) = \frac{u_f}{u_f + v_f} = \frac{1}{1 + \frac{v_f}{u_f}},$$

onde:

$$u_f = \frac{1}{\eta} \sum_{g \in \mathcal{C}_l \cup \mathcal{C}_{l+1} \cup \dots} Q_{fg}, \quad v_f = \frac{1}{\eta} \sum_{g \in \mathcal{C}_{d-1} \cup \mathcal{C}_d \cup \dots \cup \mathcal{C}_{l-1}} Q_{fg}.$$

Podemos observar que u_f é a menor entre todas as probabilidades possíveis e v_f é a maior de todas, dado que u_f é definida através das taxas de transição do tipo lenta (dividida pela taxa de uniformização η) e v_f corresponde as taxas do tipo rápida. Então, pela definição de λ e μ ,

$$\frac{1}{1 + \frac{v_f}{u_f}} \leq \frac{1}{1 + \frac{\mu/\eta}{\lambda/\eta}} = \frac{\psi}{1 + \psi}.$$

Para $l \geq d$,

$$p(l, d) \leq \frac{\psi}{1 + \psi} \leq \psi = q(l, d).$$

Assim, para todos os casos, temos $p(l, d) \leq q(l, d)$.

Seja $q(l, d)$ escrito como uma função de d : $q(l, d) = g(d)$. Observe que $g(\cdot)$ é uma função crescente. Então:

$$\Pr(\mathcal{I}(k+1) \geq l) \leq \sum_{d_0}^D g(d) \Pr(\mathcal{I}(k) = d) = \mathbb{E}(g(\mathcal{I}(k))) \leq \mathbb{E}(g(\mathcal{Z}(k))) = \Pr(\mathcal{Z}(k+1) \geq l).$$

A última desigualdade é decorrente da hipótese de recorrência e do fato que $g(\cdot)$ é uma função crescente. Temos, então, provado que $\mathcal{I}(k) \leq_{\text{st}} \mathcal{Z}(k)$ para todo $k \geq 0$. Isto significa que o número de estados visitados por \mathcal{Z} antes da absorção é estocasticamente maior que o número de classes visitadas pelo processo \mathcal{Y} .

Como segundo passo da prova, devemos observar o tempo que o processo \mathcal{Y} permanece em cada uma das D classes. Provaremos aqui que, em média, este tempo é limitado superiormente por η/μ . Denotemos por: (1) π_f a probabilidade de iniciar a visita na classe \mathcal{C}_d pelo estado $f \in \mathcal{C}_d$; (2) $\boldsymbol{\pi}$ o vetor cuja a f -ésima entrada é π_f , indexado em \mathcal{C}_d .

Seja \mathbf{U}_d a submatriz da matriz \mathbf{U} formada pelos estados pertencentes ao conjunto de estados \mathcal{C}_d , também indexada na classe desses estados. Se S é o tempo que o processo \mathcal{Y}

4.4 O Método de Aproximação Proposto

passa em \mathcal{C}_d , quando o primeiro estado visitado é escolhido de acordo com a distribuição π , então temos $\Pr(S > k) = \pi^T \mathbf{U}_d^k \mathbf{1}$, onde π^T é o vetor transposto do vetor coluna π e $\mathbf{1}$ é o vetor coluna com todos os elementos iguais a 1. O valor esperado do tamanho de S é:

$$E(S) = \sum_{k \geq 0} \pi^T \mathbf{U}_d^k \mathbf{1} = \pi^T (I - \mathbf{U}_d)^{-1} \mathbf{1}.$$

Denotemos $\mathbf{U}_d \mathbf{1} = \mathbf{u}$. A entrada f do vetor \mathbf{u} é $u_f = 1 - \sum_{g \notin \mathcal{C}_d} \mathbf{U}_d(f, g)$. Para todo estado f em qualquer classe D ,

$$u_f \leq 1 - q, \quad q = \frac{\mu}{\eta} \quad (\leq 1)$$

que segue a definição de μ (e η). Então temos que todo elemento i do vetor \mathbf{u} é menor ou igual a $(1 - q)\mathbf{1}$. Da mesma forma, $\mathbf{U}_d^2 \mathbf{1} = \mathbf{U}_d \mathbf{u} \leq (1 - q)\mathbf{U}_d \mathbf{1} \leq (1 - q)^2 \mathbf{1}$. Recursivamente, é possível mostrar que $\mathbf{U}_d^k \mathbf{1} \leq (1 - q)^k \mathbf{1}$, $k \geq 0$. Assim:

$$E(S) = \sum_{k \geq 0} \pi^T \mathbf{U}_d^k \mathbf{1} \leq \sum_{k \geq 0} \pi^T (1 - q)^k \mathbf{1} = \sum_{k \geq 0} (1 - q)^k = \frac{1}{q} = \frac{\eta}{\mu}.$$

O passo final é provar que o número de visitas na cadeia \mathcal{Z} até a absorção, quando o estado inicial é D , é dado por:

$$\frac{1 - (1 - \psi)^D}{\psi(1 - \psi)^D}, \quad (4.12)$$

que poder ser definido recursivamente em D . Assim, concluímos a prova. _____ •

Assim, baseado no lema anterior, temos que a Equação 4.10 pode ser limitada superiormente por:

$$\mathbf{U}'' \leq \sigma \mathbf{U}_{S\mathcal{F}} \mathbf{U}_{\mathcal{F}\mathcal{F}}^{N-1} \mathbf{1} \mathbf{U}_{\mathcal{F}S}. \quad (4.13)$$

com $\mathbf{1}$ uma matriz onde todos os elementos são iguais a 1.

Por questão de notação, seja:

$$\mathbf{B} = \mathbf{U}_{S\mathcal{F}} \mathbf{U}_{\mathcal{F}\mathcal{F}}^{N-1} \mathbf{1} \mathbf{U}_{\mathcal{F}S}. \quad (4.14)$$

Logo, a Equação 4.13 pode ser reescrita como:

$$\mathbf{U}'' \leq \sigma \mathbf{B}. \quad (4.15)$$

Dado o Lema 1 e a definição do limite superior para a matriz \mathbf{U}'' descrita pela Equação 4.15, passemos a definição do limite superior para $\varrho^{\mathcal{Y}, S}$. Considerando $\varrho^{\mathcal{Y}, S} = \tilde{\varrho}$ (Equação

4.4 O Método de Aproximação Proposto

4.6), temos:

$$\begin{aligned}
 \varrho^{\mathcal{Y},\mathcal{S}} &= \tilde{\varrho} \\
 &= \sum_{n=0}^{\infty} \tilde{\mathbf{U}}^n \mathbf{r} \\
 &= \sum_{n=0}^{\infty} (\mathbf{U}' + \mathbf{U}'')^n \mathbf{r} \\
 &\leq \sum_{n=0}^{\infty} (\mathbf{U}' + \sigma \mathbf{B})^n \mathbf{r} = \varphi.
 \end{aligned}$$

O vetor φ é o limite superior para o termo $\varrho^{\mathcal{Y},\mathcal{S}}$ dado que para o seu cálculo utilizamos a matriz \mathbf{B} , que somada a matriz \mathbf{U}' , fornece um limite superior para a matriz exata $\tilde{\mathbf{U}}$.

A existência do vetor φ está condicionada na convergência da série $\sum_{n=0}^{\infty} (\mathbf{U}' + \sigma \mathbf{B})^n$. Esta convergência é garantida pela maneira como as matrizes envolvidas na soma são definidas: como \mathbf{U}' é um limite inferior para a matriz exata $\tilde{\mathbf{U}}$ e esta última é subestocástica, podemos concluir que a matriz \mathbf{U}' também será subestocástica.

Passemos a matriz $\sigma \mathbf{B}$ que é um limite superior da matriz \mathbf{U}'' . Para um N tendendo a infinito, a matriz $\mathbf{U}_{\mathcal{F}\mathcal{F}}$ tende a uma matriz nula (pois a matriz $\mathbf{U}_{\mathcal{F}\mathcal{F}}$, também é subestocástica). Assim, para N grande o suficiente, a matriz resultante da soma $(\mathbf{U}' + \sigma \mathbf{B})$ é subestocástica e o vetor φ existe. Este vetor é obtido através da solução do sistema linear:

$$(\mathbf{I} - \mathbf{U}' - \sigma \mathbf{B})\varphi = \mathbf{r}. \quad (4.16)$$

Caso o valor de N escolhido não seja grande o suficiente para garantir a convergência da série $\sum_{n=0}^{\infty} (\mathbf{U}' + \sigma \mathbf{B})^n$, um novo valor para N deve ser escolhido e uma nova matriz \mathbf{U}' deverá ser definida. No entanto, no amplo conjunto de exemplos apresentados na Seção 3.7 esta condição é satisfeita para pequenos valores de N . Isto se deve principalmente a característica da classe de modelos considerados neste capítulo, onde as probabilidades de transicionar entre os estados se concentra entre os estados pertencentes ao conjunto \mathcal{S} .

Exemplo Base: Primeiramente, apresentemos a matriz \mathbf{V} , onde cada elemento representa o número médio de visitas do processo \mathcal{Y} a um estado $j \in \mathcal{F}$ antes de deixar o conjunto \mathcal{F} , dado o estado inicial $i \in \mathcal{F}$. Temos, então:

$$\mathbf{V} = \begin{pmatrix} 1.0999e+0 & 1.0999e-5 & 1.0999e-10 & 1.0999e-15 \\ 0 & 1.0999e+0 & 1.0999e-5 & 1.0999e-10 \\ 0 & 0 & 1.0999e+0 & 1.0999e-5 \\ 0 & 0 & 0 & 1.1000e+0 \end{pmatrix}$$

4.4 O Método de Aproximação Proposto

Consideremos o caso onde o número máximo de transições dentro do conjunto \mathcal{F} é de 3. Assim, o parâmetro N considerado para a aplicação da técnica de *path-based* é $N = 5$. A matriz \mathbf{B} para este caso é a seguinte:

$$\mathbf{B} = \begin{pmatrix} 5.6424e-10 & 2.2572e-13 & 3.3861e-17 & 2.2577e-21 \\ 0 & 5.6424e-10 & 2.2572e-13 & 3.3863e-17 \\ 0 & 0 & 5.6424e-10 & 2.2575e-13 \\ 0 & 0 & 0 & 5.6447e-10 \end{pmatrix}$$

Podemos observar que os valores de probabilidades, mesmo para o caso onde somente caminhos com comprimento de no máximo 5 transições foram considerados, são extremamente pequenos. Mais uma vez, estes números se explicam em função das características dos modelos considerados neste capítulo.

O próximo passo para definir um limite superior para a matrix \mathbf{U}'' é o cálculo do fator σ . Utilizando a equação 4.11, temos que o valor para este parâmetro é de 4.4011. Se observarmos cada elemento da matriz \mathbf{V} mostrada neste exemplo base, veremos que σ é maior que o valor de todo elemento (i, j) da matriz \mathbf{V} .

Finalizando, apresentamos o resultado do limite superior para $\varrho^{\mathcal{Y}, \mathcal{S}}$. Este valor é dado pelo vetor φ , calculado pela equação 4.16. Como consideramos que a distribuição de probabilidades inicial está concentrada no estado $i = 1$, estamos interessados no elemento φ_1 . Assim podemos escrever: $\varphi_1 = 1.0020e6 > \varrho'_1 = 9.9932e5$, e um limite superior foi estabelecido.

Limite Superior para $\varrho^{\mathcal{Y}, \mathcal{F}}$

Para todo $i, j \in \mathcal{S}$, denotemos por $W(i, j)$ o número médio de visitas feitas pelo processo \mathcal{Y} ao estado j , começando pelo estado i . Seja, também, ν_i o número médio de visitas realizadas pelo processo \mathcal{Y} ao conjunto \mathcal{F} , considerando que $\mathcal{Y}(0) = i$. Temos que:

$$\nu_i = \sum_{j \in \mathcal{S}} W(i, j) U(j, \mathcal{F})$$

onde $U(j, \mathcal{F}) = \sum_{f \in \mathcal{F}} U(j, f)$. ν_i representa o número médio de visitas feitas ao conjunto \mathcal{F} , considerando todos os estados, a partir de um estado inicial i em \mathcal{S} . A Figura 4.3 mostra a representação gráfica da medida ν_i .

É possível calcularmos o número ν_i utilizando o processo reduzido $\tilde{\mathcal{Y}}$. A matriz \mathbf{W}

4.4 O Método de Aproximação Proposto

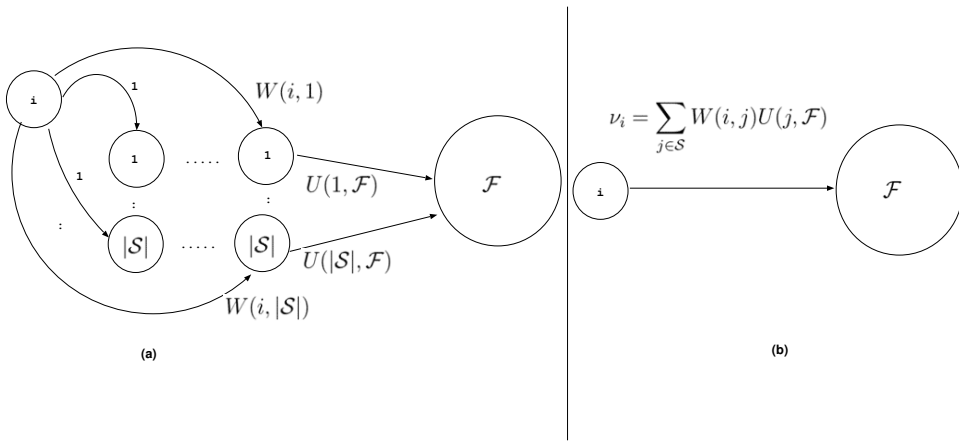


Figura 4.3: Representação gráfica da medida ν_i .

composta pelos elementos $W(i, j)$; $\forall i, j$, é definida pela equação:

$$\mathbf{W} = (\mathbf{I} - \tilde{\mathbf{U}})^{-1}.$$

A i -ésima entrada do vetor $\varrho^{\mathcal{Y}, \mathcal{F}}$ é limitada superiormente por:

$$\sigma r^* \nu_i$$

onde:

$$r^* = \max\{r_f, f \in \mathcal{F}\}.$$

A recompensa r^* representa a recompensa máxima atribuída a um estado pertencente ao conjunto \mathcal{F} .

Definindo os vetores $\mathbf{u}_{\mathcal{F}}$ e $\boldsymbol{\nu}$, indexados no conjunto \mathcal{S} , onde cada elemento i do vetor $u_{\mathcal{F}_i}$ é igual a $U(i, \mathcal{F})$, temos:

$$\boldsymbol{\nu} = \mathbf{W}\mathbf{u}_{\mathcal{F}}. \quad (4.17)$$

Para a determinação do limite superior procurado, podemos escrever a seguinte relação entre o vetor $\boldsymbol{\nu}$ e as demais matrizes definidas posteriormente:

$$\boldsymbol{\nu} = \mathbf{W}\mathbf{u}_{\mathcal{F}} = \sum_{n=0}^{\infty} \tilde{\mathbf{U}}^n \mathbf{u}_{\mathcal{F}} \leq \sum_{n=0}^{\infty} (\mathbf{U}' + \sigma \mathbf{B})^n \mathbf{u}_{\mathcal{F}} = \boldsymbol{\tau}.$$

O vetor $\boldsymbol{\tau}$ é a solução do sistema de equações lineares:

$$(\mathbf{I} - \mathbf{U}' - \sigma \mathbf{B})\boldsymbol{\tau} = \mathbf{u}_{\mathcal{F}}. \quad (4.18)$$

Concluindo, podemos escrever o limite superior para $\varrho^{\mathcal{Y}, \mathcal{F}}$ como:

$$\varrho^{\mathcal{Y}, \mathcal{F}} \leq \sigma r^* \boldsymbol{\tau}.$$

4.4 O Método de Aproximação Proposto

Resumidamente, o limite superior para o vetor $\varrho^{\mathcal{Y},\mathcal{F}}$ é estabelecido através do número máximo de visitas ao conjunto \mathcal{F} a partir de qualquer estado pertencente ao conjunto \mathcal{S} multiplicado pela recompensa máxima em \mathcal{F} . A cada visita ao conjunto \mathcal{F} , consideramos que o processo \mathcal{Y} permanece σ unidades e que cada unidade de tempo agrega ao processo a recompensa máxima r^* . Assim, como todos os valores envolvidos são limites superiores para os valores originais do sistema em estudo, estabelecemos um limite superior para $\varrho^{\mathcal{Y},\mathcal{F}}$.

Tendo definido os limites superiores para $\varrho^{\mathcal{Y},\mathcal{S}}$ e $\varrho^{\mathcal{Y},\mathcal{F}}$, podemos estabelecer o limite superior para $\varrho^{\mathcal{Y}}$. Finalmente podemos escrever que:

$$\varrho^{\mathcal{Y}} \leq \sigma r^* \tau + \varphi$$

onde o primeiro termo do lado direito é o limite superior para $\varrho^{\mathcal{Y},\mathcal{F}}$ e o segundo termo o limite superior para $\varrho^{\mathcal{Y},\mathcal{S}}$.

Exemplo Base: Para finalizar o método apresentado, devemos estabelecer o limite superior para cada um dos termos do vetor $\varrho^{\mathcal{Y},\mathcal{F}}$. Este limite é dado pelo vetor τ , após a solução do sistema linear dado pela Equação 4.18. Mais uma vez, consideraremos somente o primeiro elemento do vetor τ . Assim, temos que $\tau_1 = 9.1095$, e que $\sigma r^* \tau_1 = 40.092$.

Resumindo, temos o valor exato do primeiro elemento do vetor $\varrho_1^{\mathcal{Y}}$, e os limites inferior e superior para este valor. Os valores obtidos são os seguintes:

$$(9.9932e5) \leq (1.0001e6) \leq (1.0020e6) + (40.092) = (1.00204e6)$$

Definição de Limites para a Matriz Exata \tilde{U}

Em alguns casos é possível calcular o complemento estocástico exato, sem efetuar a inversão da matriz definida na Equação 4.5. Estes casos encontram-se listados abaixo:

1. modelos onde o espaço de estados possui uma cardinalidade suficientemente pequena (alguns milhares de estados). Neste caso, a matriz \tilde{U} pode ser calculada diretamente através da Equação 4.5;
2. modelos onde o espaço de estados possui uma cardinalidade na ordem de milhões de estados. Neste caso, a definição da matriz \tilde{U} é possível quando a matriz de transição possui estrutura especial: um único estado de retorno ao conjunto \mathcal{S} , um

4.5 Redefinindo o Conjunto de Estados Lentos

único estado de saída do conjunto \mathcal{F} ou quando todas as linhas diferentes de zero da matriz $U_{\mathcal{F}\mathcal{S}}$ são proporcionais.

Consideremos a segunda categoria de modelos e o caso especial onde existe somente um estado de retorno ao conjunto \mathcal{S} . Neste cenário, temos que existe um único estado $s \in \mathcal{S}$ tal que para algum estado $f \in \mathcal{F}$, $U(f, s) > 0$. Para todo $g \in \mathcal{F}$ e $z \in \mathcal{S}$, $z \neq s$, $U(g, z) = 0$. Conseqüentemente, a matrix $U_{\mathcal{F}\mathcal{S}}$ possui todas as colunas iguais a zero, exceto uma delas. Assim, $(\mathbf{I} - U_{\mathcal{F}\mathcal{F}})^{-1}U_{\mathcal{F}\mathcal{S}}$ pode ser calculada facilmente com pequeno custo computacional (para maiores detalhes veja [95] e [39]).

A partir destes casos descritos, seja a matriz $\tilde{\mathbf{U}}$. O limite inferior para ϱ^y é simplesmente dado por $\tilde{\varrho}$, após resolver o seguinte sistema linear:

$$(\mathbf{I} - \tilde{\mathbf{U}})\tilde{\varrho} = \mathbf{r}. \quad (4.19)$$

O limite inferior $\tilde{\varrho}$ é melhor que o limite inferior dado por ϱ' . O limite superior é obtido através da soma dos termos $\sigma r^* \nu$ e $\tilde{\varrho}$. O vetor ν é obtido após resolver o seguinte sistema linear:

$$(\mathbf{I} - \tilde{\mathbf{U}})\nu = \mathbf{u}_{\mathcal{F}}. \quad (4.20)$$

E, novamente, este limite é melhor do que o valor obtido para o caso geral.

4.5 Redefinindo o Conjunto de Estados Lentos

Suponhamos que a cardinalidade de \mathcal{S} seja pequena. Nestes casos, podemos utilizar o mesmo método de limites inferior e superior apresentados, substituindo o conjunto \mathcal{S} por um conjunto $\mathcal{S}' = \mathcal{S} \cup \mathcal{F}'$, onde \mathcal{F}' é um subconjunto de \mathcal{F} . O conjunto \mathcal{F} passa a ser definido por $\mathcal{F} - \mathcal{F}'$. Esta nova definição para o conjunto \mathcal{S} acrescenta mais informação ao sistema reduzido que será resolvido, resultando em uma precisão melhor nos valores obtidos.

O número total de estados a ser adicionado ao conjunto original \mathcal{S} para obter uma melhor aproximação da medida ECRA é uma decisão tomada durante o processo de modelagem e que impacta diretamente no custo computacional da aplicação do método. Assim, podemos ter uma melhor aproximação acrescentando um número de estados que não aumente consideravelmente o custo computacional da aplicação do método proposto. A Figura 4.4 apresenta, graficamente, a redefinição utilizada.

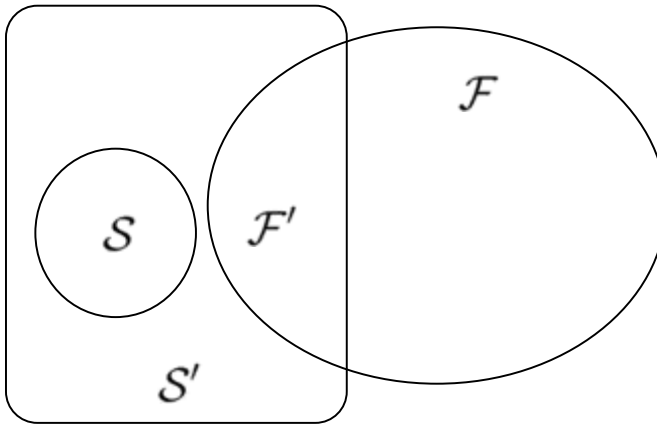


Figura 4.4: Nova representação para o conjunto de estados lentos. A redefinição faz com que um número fixo de estados, originalmente rápidos, seja considerado para que a medida aproximada se torne mais próxima da medida real ECRA. Assim, todo o método é aplicado ao novo conjunto S' .

4.6 Visão Geral do Método Proposto e Custo Computacional

Nesta seção apresentaremos o método proposto em forma algorítmica: um algoritmo para o caso geral, quando temos somente acesso a matriz aproximada U' e um segundo algoritmo quando temos acesso a matriz exata \tilde{U} .

Para o caso geral, temos o seguinte algoritmo:

MÉTODO ECRA - CASO GERAL ()

- 1 Escolha θ
- 2 Escolha η
- 3 Escolha N
- 4 Defina U' (equação (4.8))
- 5 Calcule ϱ' (sistema linear (4.9))
- 6 Calcule σ (equação (4.11))
- 7 Calcule B (equação (4.14))
- 8 Calcule φ (sistema linear (4.16))
- 9 Calcule τ (sistema linear (4.18))
- 10 Limites: $\varrho'/\eta \leq \varrho \leq (\sigma r^* \tau)/\eta + \varphi/\eta$

Para os casos onde o complemento estocástico é exato, temos o seguinte algoritmo simplificado:

MÉTODO ECRA – COMPLEMENTO ESTOCÁSTICO EXATO ()

4.6 Visão Geral do Método Proposto e Custo Computacional

- 1 Escolha θ
- 2 Escolha η
- 3 Calcule $\tilde{\mathbf{U}}$ (equação (4.5))
- 4 Calcule $\tilde{\varrho}$ (sistema linear (4.7))
- 5 Calcule σ (equação (4.11))
- 6 Calcule ν (sistema linear (4.20))
- 7 Limites: $\tilde{\varrho}/\eta \leq \varrho \leq \tilde{\varrho}/\eta + (\sigma r^* \nu)/\eta$

A Figura 4.5 mostra os principais passos dos dois algoritmos previamente descritos. Conforme podemos observar, a cadeia discreta reduzida proveniente do processo \mathcal{Y} depende se é possível, em termos de custo computacional, calcular a matriz $\tilde{\mathbf{U}}$. Ao final, temos os limites inferior e superior para a medida ECRA relacionada com a cadeia \mathcal{Y} . Após a divisão deste resultado pela taxa de uniformização η , temos a medida da cadeia contínua \mathcal{X} que representa o sistema em estudo.

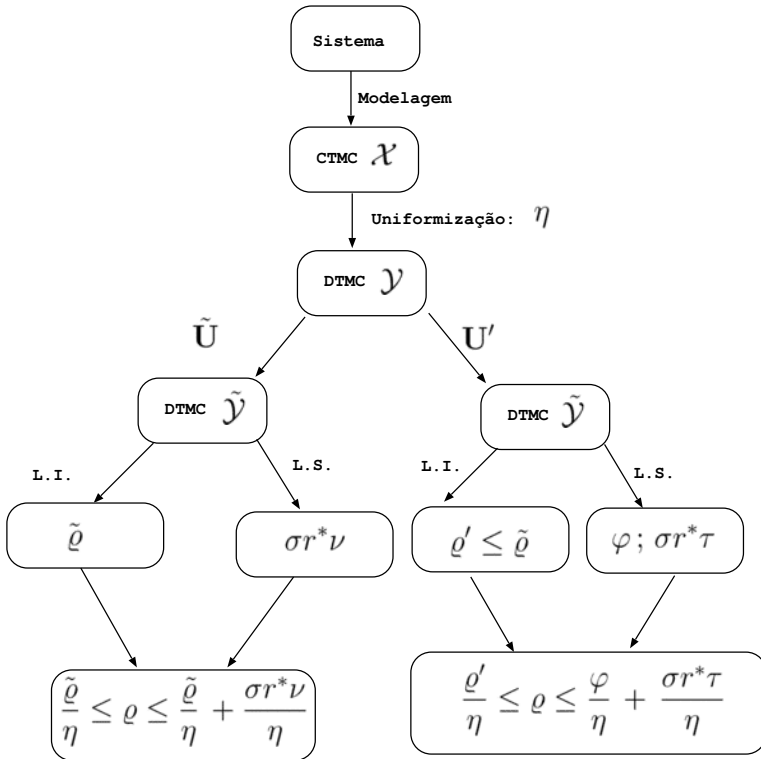


Figura 4.5: Diagrama com a visão geral do método proposto.

O custo computacional do método proposto, no caso geral, é dominado pela solução de três sistemas lineares, com o total de equações igual a cardinalidade do conjunto \mathcal{S} , muito menor que a cardinalidade do processo original \mathcal{Y} . Dado que dois sistemas lineares utilizam a mesma matriz $(\mathbf{I} - \mathbf{U}' - \sigma \mathbf{B})$, podemos obter uma redução em termos do número total de multiplicações.

No caso em que o complemento estocástico é exato, os limites calculados são mais precisos e o custo computacional é inferior. Para estes modelos, somente dois sistemas lineares devem ser calculados e não existe o custo relacionado a aplicação da teoria *path-based* utilizada para a definição da matriz U' .

4.7 Exemplos

Esta seção apresenta o estudo do desempenho do método proposto através de alguns exemplos. Uma pequena descrição de cada um deles e o objetivo na avaliação do método proposto são dados a seguir.

O primeiro exemplo apresenta um sistema de filas $M/M/1/H$ muito simples, onde o servidor está sujeito a falhas e reparos. Com este exemplo, compararemos o caso onde temos acesso ao complemento estocástico exato, dado que a matriz exata \tilde{U} é conhecida, com o caso onde somente temos acesso a matriz aproximada U' , determinada pela técnica de *path-based*.

A seguir, apresentamos um sistema com diversos componentes com reparos com histerese onde o reparo começa somente após o sistema ter H componentes fora de serviço (Este tipo de disciplina de reparo é conhecido como *deffered repairs*). Um estudo da precisão da aproximação, bem como dos limites para um conjunto de diferentes parâmetros e cardinalidades do espaço de estados é apresentado. Utilizamos também a distribuição do tipo *Coxian* para artificialmente aumentarmos o espaço de estados do modelo e ainda possibilitarmos a definição do valor exato da medida em estudo.

No terceiro exemplo temos um sistema com diversos componentes que se comportam de maneira idêntica. O principal objetivo deste exemplo é mostrar que mesmo em face a uma grande redução no espaço de estados, resultados com grande precisão são alcançados e limites estreitos estabelecidos. Uma exploração de valores do parâmetro N relacionado com a técnica de *path based* também é apresentada.

Em seqüência, apresentamos um exemplo baseado em [42] com o intuito de estudar o impacto da redefinição do conjunto de estados lentos \mathcal{S} no resultado do método proposto.

No último exemplo temos um modelo de banco de dados, onde estamos interessados em ampliar o estudo dos resultados providos pelo método proposto em medidas relacionadas com o desempenho econômico do sistema.

4.7 Exemplos

Em todos os exemplos, os resultados são apresentados em dois tipos de tabelas: a primeira fornece o conjunto de parâmetros que caracterizam o sistema modelado e os respectivos resultados numéricos: o resultado exato (ECRA), os limites inferior e superior (L.I. e L.S., respectivamente) e o erro relativo (E. R.) definido por $|(L.I. + L.S.)/2 - ECRA|/ECRA$. A segunda tabela fornece as informações relacionadas com o ganho computacional, que incluem a redução do tamanho do espaço de estados e os parâmetros relacionados com a técnica de *path-based*. Para os primeiros quatro exemplos, a medida ECRA é o MTTF, onde a recompensa em um estado é igual a 1, se o sistema está operacional no estado considerado e 0, caso contrário. O último exemplo apresenta o cálculo da medida ECRA para recompensas com valores diferentes de 0 e 1.

Para obtermos os valores exatos das medidas de interesse e, ao mesmo tempo, termos conjuntos de modelos com grande espaço de estados, selecionamos modelos que possuem a propriedade de *strong lumpability* [79], comumente usada para reduzir o espaço de estados de um modelo Markoviano. Para que um modelo Markoviano possua a propriedade de *strong lumpability* algumas condições devem estar presentes no modelo. Veja [79] e referências para estas condições. Estas propriedades se aplicam a todos os modelos apresentados neste capítulo.

Adicionalmente, na Seção 4.7.2, substituímos a variável aleatória exponencial por uma variável aleatória *Coxian*, aumentando significativamente o espaço de estados do processo. A Figura 4.6 mostra a representação gráfica desta distribuição, onde v_s representa a taxa de saída do s -ésimo estágio, e l_s a probabilidade do s -ésimo estágio ser o último estágio visitado.

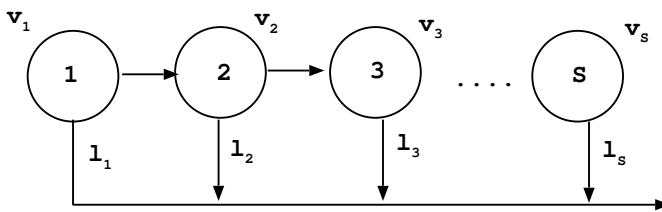


Figura 4.6: Distribuição *Coxian*.

4.7.1 Sistema de Filas $M/M/1/H$

Este primeiro exemplo é baseado em [10] e pode ser descrito da seguinte maneira. Considere uma fila do tipo $M/M/1/H$ onde H é a capacidade máxima do sistema. O servidor pode falhar (mesmo que o sistema esteja vazio) e esta falha é reparada com

4.7 Exemplos

Tabela 4.3: Resultados para o sistema $M/M/1/10$, com: $\gamma = 100$, $\phi = 0.001$, $\omega = 1$, $\delta = 1$ e $c = 0.99999$.

N	$L.I.$	$MTTF$	$L.S.$	$E.R.$
–	9.9999e7	1.0000e8	1.0001e8	4.5e-5
5	9.9903e7	1.0000e8	1.0097e8	4.3e-3
7	9.9999e7	1.0000e8	1.0001e8	4.5e-5

probabilidade c , definida em problemas de confiabilidade como *coverage factor*. Se a falha do servidor é do tipo que pode ser reparada, durante o procedimento de reparo novos clientes podem ser adicionados ao sistema, até que o número máximo seja alcançado. Se a falha não pode ser reparada, o processo que representa o sistema é absorvido.

Denotemos os estados da cadeia de Markov que representam o modelo pelo par de variáveis (k, h) , onde $h \leq H$ é o número de clientes no sistema e $k = \{0, 1\}$ é o número de servidores em funcionamento. O total de estados no modelo é de $2H + 3$. Sejam ω , δ , ϕ e γ , as taxas de chegada, de serviço, de falha e de reparo, respectivamente.

As taxas ω , δ e ϕ são classificadas como lentas e γ é a única taxa rápida presente no modelo. Isto significa que cada estado onde o servidor está em funcionamento é classificado como lento. Desta forma, temos que $|\mathcal{S}| = |\mathcal{F}| = H + 1$. Para o estudo do método proposto, vamos considerar uma pequena cadeia, com $H = 10$. Apesar do espaço de estados deste modelo ser pequeno para a necessidade de aplicar um método aproximado (23 estados), neste simples exemplo podemos obter o complemento estocástico exato e fazer a comparação tanto com o resultado exato, como também com o resultado aproximado quando a técnica *path-based* é utilizada para determinar a matriz U' .

A medida a ser calculada é o tempo médio até o sistema evoluir para o estado absorvente. As Tabelas 4.3 e 4.4 mostram os resultados. Para os casos onde o complemento estocástico é exato, o parâmetro N não é utilizado e a coluna correspondente a este parâmetro é preenchida com “-”.

Os resultados mostram que, como esperado, quando a diferença entre os valores das taxas diminuem, o erro relativo aumenta. Esta situação se torna mais perceptível quando um valor pequeno para o parâmetro N é utilizado para o cálculo do complemento estocástico aproximado. Para os modelos com característica *stiffness*, o erro relativo é menor e os limites mais estreitos. Estes resultados confirmam que o método proposto se torna mais atrativo para modelos onde a diferença das taxas de transição é grande. No

4.7 Exemplos

Tabela 4.4: Resultados para o sistema $M/M/1/10$, com: $\gamma = 10$, $\phi = 0.001$, $\omega = 1$, $\delta = 1$ e $c = 0.99999$.

N	$L.I.$	$MTTF$	$L.S.$	$E.R.$
–	9.9999e7	1.0001e8	1.0010e8	3.9e–4
5	9.9904e7	1.0001e8	1.0108e8	4.8e–3
7	9.9999e7	1.0001e8	1.0010e8	3.9e–4

entanto, mesmo para o caso contrário, o método apresenta bons resultados aproximados.

4.7.2 Sistema com C componentes

Consideremos um sistema com falhas e reparos (com histerese), onde os reparos somente são inicializados quando H componentes estão falhos. Uma vez iniciado, os reparos permanecem até que todos os componentes estejam recuperados (este tipo de reparo caracteriza o fenômeno de histerese). O número inicial de componentes em funcionamento no sistema é denotado por C . O tempo de falha de qualquer componente é exponencialmente distribuído com parâmetro ϕ . O sistema é considerado operacional se existem, pelo menos, $K - 1$ componentes em funcionamento. Para que esta situação seja possível, o sistema deverá ser formado por $C \geq H + K$ componentes.

Todos os estados que possuem uma taxa de saída igual a taxa de reparo são classificados como rápidos. Desta forma, temos que o número total de estados lentos é igual a H .

No primeiro conjunto de resultados o tempo de reparo é exponencialmente distribuído com parâmetro γ . Para um total de componentes $C = 1000$, apresentamos diversos resultados onde variamos os valores da taxa de falha e os valores dos parâmetros H e K .

O espaço de estados é composto pelos estados $i = C, C-1, \dots, C-H, \dots, K, \dots, 1, 0$ e pelos estados “espelhos” $(C-1)', (C-2)', \dots, (C-H+1)'$, representados devido ao fenômeno de *hysteresis*. Os reparos somente ocorrem entre os estados espelhos $[C', (C-H)']$. Podemos agregar os estados $K-1, \dots, 1, 0$ em um único estado absorvente $0'$, por exemplo. Este novo espaço de estados possui $C + H - K + 1$ estados, sendo um estado absorvente, H estados lentos e $C - K$ estados rápidos. A Figura 4.7 mostra a cadeia que modela o sistema descrito. O processo de reparo é iniciado após H componentes falhos no sistema. Supondo $C = 10$ e $H = 3$, o reparo se inicia quando

4.7 Exemplos

Tabela 4.5: Resultados para o sistema com $C = 1000$ componentes e $\gamma = 1$.

H	K	ϕ	L.I.	ECRA	L.S.	E.R.
3	1	$1e-4$	$2.5378e16$	$2.5466e16$	$2.6268e16$	$1.4e-2$
100	10	$1e-5$	$2.7186e16$	$2.7187e16$	$2.7189e16$	$1.8e-5$
100	10	$1e-4$	$2.5381e16$	$2.5471e16$	$2.5484e16$	$1.5e-3$

Tabela 4.6: Redução do espaço de estados para o sistema com $C = 1000$ componentes.

H	K	$ \Omega $	$ \tilde{\Omega} $	Taxa de Redução	N
3	1	1003	3	334	15
100	10	1100	100	11	106
100	10	1100	100	11	106

existem 7 componentes em funcionamento e temos três estados lentos $\{10, 9, 8\}$.

A Tabela 4.5 mostra os valores dos parâmetros utilizados e os resultados, onde a medida ECRA é a medida de MTTF. A Tabela 4.6 mostra o ganho computacional e os parâmetros relacionados com a técnica *path-based*. Para os casos analisados, o método proposto é extremamente eficiente mesmo quando um pequeno número de estados $|\tilde{\Omega}|$ é utilizado para o cálculo aproximado da medida, em contrapartida ao número original de estados $|\Omega|$. A coluna *Taxa de Redução* representa a divisão de $|\Omega|$ por $|\tilde{\Omega}|$.

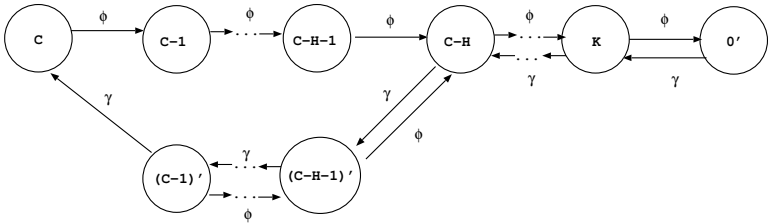


Figura 4.7: Representação do sistema com diversos componentes.

Consideremos o mesmo sistema agora com um total de $C = 5000$ componentes. O parâmetro a ser variado é o parâmetro H . A Tabela 4.7 apresenta os resultados e a Tabela 4.8 apresenta as informações adicionais sobre o total de estados utilizados para a obtenção da medida.

Conforme podemos observar, o valor do parâmetro N para todos os casos apresentados é pelo menos igual a H . Esta escolha é baseada na estrutura particular do modelo e na idéia de gerar pelo menos um caminho no interior do conjunto \mathcal{F} . Para este modelo, e-

4.7 Exemplos

Tabela 4.7: Resultados para o sistema com $C = 5000$ componentes, com $\gamma = 1$, $K = 1$ e $\phi = 1e-4$.

H	L.I.	ECRA	L.S..	E.R.
20	$2.5032e16$	$2.5471e16$	$2.7480e16$	$3.0e-2$
50	$2.5447e16$	$2.5472e16$	$2.5777e16$	$5.5e-3$
100	$2.5458e16$	$2.5472e16$	$2.5629e16$	$2.8e-3$
500	$2.5441e16$	$2.5472e16$	$2.5478e16$	$4.9e-4$

Tabela 4.8: Redução do espaço de estados para o sistema com $C = 5000$ componentes.

H	$ \Omega $	$ \tilde{\Omega} $	Taxa de Redução	N
20	5020	20	251	20
50	5050	50	101	60
100	5100	100	51	108
500	5500	500	11	516

xiste somente um estado de saída do conjunto \mathcal{S} para o conjunto \mathcal{F} e um estado de entrada no conjunto \mathcal{S} . Um número total de pelo menos H estados a partir do $N(H - 1)$ -ésimo componente em funcionamento devem ser gerados para que pelo menos tenhamos um caminho explorado no interior do conjunto \mathcal{F} . Neste exemplo, o fato de gerar pelo menos um caminho no conjunto \mathcal{F} não aumenta significativamente o custo computacional, e limites mais estreitos serão obtidos. É válido ressaltar que a idéia de gerar pelo menos um caminho no interior do conjunto \mathcal{F} não é uma condição necessária para a aplicação do método, dado que todos os limites são calculados baseando-se no fato de termos um limite inferior para o complemento estocástico exato.

O nosso objetivo agora é aumentar o número total de estados do modelo. Ao invés de considerarmos o tempo de reparo como uma variável aleatória exponencial de parâmetro γ , este tempo será modelado como uma variável *Coxian* com s estágios e média γ .

Seja o sistema com $C = 10000$ componentes, para o qual tenhamos o valor exato do MTTF. O sistema que possui distribuição *Coxian*, com $s = 10$ e $s = 100$, será utilizado para a aplicação do método de aproximação. A cardinalidade do espaço de estados aumentou em 10 e 100 vezes, respectivamente, quando comparado com o espaço de estados do sistema com o tempo de reparo exponencialmente distribuído. A Tabela 4.9 apresenta os resultados e a Tabela 4.10 a redução obtida usando o método proposto.

4.7 Exemplos

Tabela 4.9: Resultados para o sistema com $C = 10000$ componentes, com tempo de reparo com distribuição *Coxian*, $\gamma = 1$, $K = 1$ e $\phi = 1e-5$.

H	s	L.I.	ECRA	L.S.	E.R.
10	10	2.7170e16	2.7187e16	2.7227e16	4.2e-4
10	100	2.7186e16	2.7187e16	3.3415e16	1.5e-1

Tabela 4.10: Redução do espaço de estados para o sistema com $C = 10000$ componentes.

H	s	$ \Omega $	$ \tilde{\Omega} $	Taxa de Redução	N
10	10	100100	10	10010	20
10	100	1001000	10	100100	30

Os resultados apresentados na Tabela 4.9 merecem um comentário especial relacionado com os limites superiores calculados pelo método. Para o caso onde $s = 10$, o limite é mais próximo do valor exato do que para o caso onde $s = 100$. Este fato é decorrente da maneira como o tempo passado no conjunto \mathcal{F} é estimado. Um dos parâmetros a ser utilizado é D , a maior distância entre qualquer estado pertencente a \mathcal{F} e o conjunto \mathcal{S} . No sistema utilizado neste exemplo, o parâmetro D é a cardinalidade do conjunto \mathcal{F} . Todavia, mesmo para este caso extremo, a aproximação fornecida pelo método é satisfatória, com pequeno erro relativo, principalmente se considerarmos a alta taxa de redução do conjunto de estados e com um número pequeno de estados gerados.

4.7.3 Sistema com vários componentes idênticos

Este exemplo é baseado em [116], onde o sistema é composto por C componentes idênticos e independentes. A evolução de cada um destes componentes pode ser modelada pela cadeia de Markov mostrada na Figura 4.8. A taxa de falha é representada por ϕ , c é o *coverage factor* definido anteriormente, e γ a taxa de reparo. A medida a ser calculada é o MTTF. O total de componentes considerado é C . Estado 0 significa que o componente está em funcionamento; no estado 1, o componente não está operacional e não pode ser recuperado; No estado 2 o componente não está operacional e pode ser recuperado.

Consideramos inicialmente sistemas com um total de C componentes pertencentes ao conjunto de valores $\{10, 30, 40, 50\}$ e mostraremos como o método proposto se comporta quando o complemento estocástico exato é utilizado. Todos os sistemas foram modelados

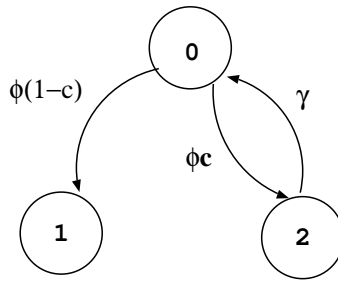


Figura 4.8: Cadeia de Markov que modela a evolução de cada um dos componentes do sistema.

Tabela 4.11: Resultados para um sistema com vários componentes idênticos e independentes, usando as simetrias da cadeia de Markov (*strong lumpability*). Os valores para os parâmetros são $\phi c 1e-6$, $\phi(1 - c) = 1e-7$ e $\gamma = 1$.

C	L.I.	ECRA	L.S.	E.R.
10	$2.9285e7$	$2.9285e7$	$2.9285e7$	0
30	$3.9932e7$	$3.9933e7$	$3.9933e7$	$1.2e-5$
40	$4.2765e7$	$4.2765e7$	$4.2767e7$	$2.3e-5$
50	$4.4959e7$	$4.4959e7$	$4.4961e7$	$2.2e-5$

usando a simetria presente na cadeia de Markov, que resulta na redução do espaço de estados (novamente a característica de *strong lumpability* foi aplicada). Um estado é classificado como rápido se pelo menos uma taxa de saída é maior ou igual a γ . As Tabelas 4.11 e 4.12 mostram os resultados e a taxa de redução do total de estados quando o método aproximado é utilizado, respectivamente.

Para aumentarmos o total de estados nos modelos que representam estes sistemas, iremos aplicar o método de aproximação sem explorar a propriedade de *strong lumpability*. Desta forma, poderemos explorar as características do método proposto em um grande espaço de estados. A Tabela 4.13 mostra os resultados e a Tabela 4.14 mostra a taxa de redução no total de estados utilizados e os valores do parâmetro N . Podemos observar que limites estreitos são obtidos mesmo quando temos uma alta taxa de redução no total de estados.

Como um último estudo de caso, consideremos um sistema com $C = 11$ componentes. Aplicaremos a técnica de aproximação para $N = \{3, 6\}$. A Tabela 4.15 mostra os resultados. O total de componentes utilizado é de $C = 11$. Os valores exatos para o MTTF são obtidos explorando a propriedade de *strong lumpability*. Os valores para os

4.7 Exemplos

Tabela 4.12: Redução do espaço de estados para o sistema com vários componentes idênticos e independentes, usando as simetrias da cadeia de Markov (*strong lumpability*).

C	$ \Omega $	$ \tilde{\Omega} $	Taxa de Redução
10	66	10	6
30	496	30	16
40	861	40	21
50	1326	50	26

Tabela 4.13: Resultados para o sistema com vários componentes idênticos, sem utilizarmos as simetrias da cadeia de Markov. Os valores exatos para o MTTF são obtidos explorando a propriedade *strong lumpability*. Os valores para os parâmetros são: $\phi c = 1e-6$, $\phi(1 - c) = 1e-7$ e $\gamma = 1$, para todos os casos.

C	L.I.	ECRA	L.S.	E.R.
11	3.0116e7	3.0177e7	3.0206e7	5.3e-4
12	3.0925e7	3.1026e7	3.1031e7	1.5e-3
18	3.4943e7	3.4951e7	3.4997e7	5.4e-4
20	3.5977e7	3.5977e7	3.5990e7	1.8e-4

Tabela 4.14: Redução do espaço de estados para o sistema com vários componentes idênticos e independentes, sem utilizar as simetrias da cadeia de Markov (*strong lumpability*).

C	$ \Omega $	$ \tilde{\Omega} $	Taxa de Redução	N
11	177147	2048	86	6
12	531441	4096	130	6
18	382742098	262144	1460	5
20	3486784401	1048576	3325	5

4.7 Exemplos

Tabela 4.15: Resultados para o sistema com diversos componentes idênticos, sem utilizar as simetrias da cadeia de Markov.

N	L.I.	ECRA	L.S.	E.R.
3	$3.0199e5$	$3.0202e5$	$3.0566e5$	$6.0e-3$
6	$3.0191e5$	$3.0202e5$	$3.0558e5$	$5.7e-3$

parâmetros são $\phi c = 1e-4$, $\phi(1 - c) = 1e-5$ e $\gamma = 1$, para todos os casos. O parâmetro variado foi N , tamanho máximo do caminho explorado em \mathcal{F} . Conforme esperado, o aumento no total de caminhos explorados no conjunto \mathcal{F} resulta em um erro relativo menor. Todavia, mesmo para o caso onde o tamanho máximo do caminho percorrido é $N = 3$ a aproximação alcançada é satisfatória.

4.7.4 Sistema em Grid

Consideremos o sistema apresentado em [42] utilizado para o estudo da sensibilidade que as medidas de confiabilidade e disponibilidade possuem em relação ao parâmetro c (*coverage factor*). O sistema aqui apresentado é um sistema em *grid*, com C processadores, onde o estado do modelo representa o total de processadores em funcionamento. O sistema está operacional se pelo menos um processador funciona. Cada processador falha com taxa $\phi = 1e-10$ é reparado com taxa $\gamma = 1$ e o sistema possui um total de $C = 5000$ componentes. As falhas são reparáveis com probabilidade c . Assim, para qualquer estado $i = \{1, \dots, C\}$ existe uma transição para um estado absorvente, com taxa $i\phi(1 - c)$. O diagrama de estados para este modelo é mostrado na Figura 4.9.

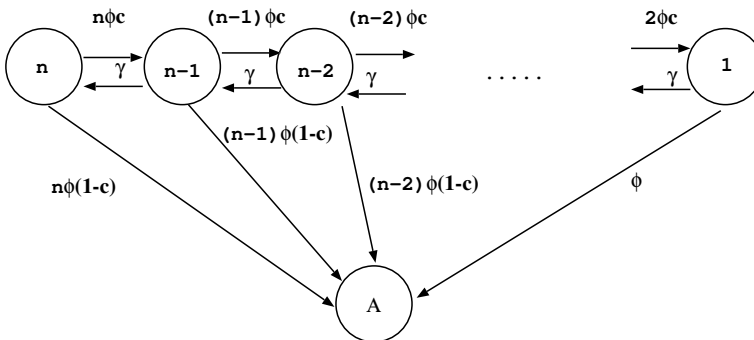


Figura 4.9: Cadeia de Markov para o sistema em *grid*.

Aplicando a definição do conjunto de estados lentos \mathcal{S} , conforme apresentado na

4.7 Exemplos

Tabela 4.16: Resultados para o sistema em *grid* com $C = 5000$ processadores.

L.I.	ECRA	L.S.	E.R.
$1.9984e11$	$2.0000e11$	$2.0034e11$	$4.5e-4$

Tabela 4.17: Resultados para o sistema em *grid* com $C = 5000$ processadores, utilizando a redefinição S' .

L.I.	ECRA	L.S.	E.R.
$1.9999e11$	$2.0000e11$	$2.0000e11$	$2.5e-5$

Seção 4.3.4, o sistema possui somente um estado lento (o estado onde todos os C processadores estão em funcionamento). Os resultados são mostrados na Tabela 4.16. Mesmo no caso em que somente um estado é utilizado para o cálculo aproximado da medida, o erro relativo é pequeno e o custo computacional é bem menor do que se considerássemos o conjunto de estados original.

Redefiniremos o conjunto de estados lentos \mathcal{S} , considerando alguns estados que pertencem a \mathcal{F} , ou seja, que originalmente são rápidos, como sendo “lentos”, com o objetivo de aumentar a precisão dos resultados obtidos. Neste caso, o complemento estocástico (e toda a análise subsequente) considerará o conjunto S' ao invés do conjunto \mathcal{S} , onde $\mathcal{S} \subset S'$. Esta redefinição é feita de uma maneira bem simples: adicionaremos 1000 estados ao conjunto lento \mathcal{S} original, resultando em $|S'| = 1001$. Vale ressaltar que o número de estados a serem adicionados possui uma relação direta com o custo computacional, ou seja, existe um compromisso entre custo computacional e precisão do resultado obtido.

A Tabela 4.17 mostra os novos resultados com o valor do erro relativo uma ordem de grandeza inferior ao caso onde $S = 1$. A redefinição para o conjunto de estados lentos, não possui critérios estabelecidos. Deve-se considerar o impacto no cálculo da medida, ou seja, dos sistemas lineares envolvidos.

4.7.5 Sistema de Banco de Dados

Como um último exemplo para o estudo do método proposto, iremos considerar um sistema de banco de dados composto por diversas CPUs, discos rígidos e unidades de memória. Cada componente possui o seguinte comportamento: falhas que podem ser reparadas ocorrem com taxa ϕc e as que não podem ser reparadas com taxa $\phi(1 - c)$. A

Tabela 4.18: Resultados para o sistema de banco de dados.

s	L.I.	ECRA	L.S.	E.R.
3	1.8139e6	1.8333e6	1.8354e6	4.7e-3
10	1.8150e6	1.8333e6	1.8870e6	9.6e-3

Tabela 4.19: Redução do espaço de estados para o sistema de banco de dados composto por CPUs e discos rígidos.

s	$ \Omega $	$ \tilde{\Omega} $	Taxa de Redução	N
3	390625	256	1525	7
10	429998169	256	1679680	7

taxa de reparo é de γ .

Primeiramente, suponhamos que o nosso interesse seja o comportamento das unidades de CPU e dos discos rígidos. Para este estudo, consideremos um sistema com 3 CPUs e 5 discos rígidos. A medida a ser calculada é o MTTF. Para aumentarmos o espaço original do modelo e ao mesmo tempo calcularmos a solução exata, aplicaremos o mesmo procedimento apresentado na Seção 4.3.4. Cada componente possui um tempo de reparo distribuído segundo uma distribuição *Coxian* com s estágios. As Tabelas 4.18 e 4.19 mostram os resultados e a taxa de redução do espaço de estados quando o método proposto é utilizado, respectivamente. Estamos interessados, primeiramente, no comportamento da CPU e do disco rígido. Os valores exatos para o MTTF são obtidos utilizando a propriedade de *strong lumpability*. Os valores para os parâmetros são: CPU: $\phi_c = 1e-5$, $\phi(1-c) = 1e-6$ e $\gamma = 1$; Disco: $\phi_c = 1e-3$, $\phi(1-c) = 1e-4$ e $\gamma = 5$. Para obter o resultado exato, modelamos o mesmo sistema utilizando a propriedade de *strong lumpability*.

Consideremos o comportamento da unidade de memória. Neste último estudo de caso, nosso sistema tem 4 CPUs, 5 discos rígidos e 3 unidades de memória. Suponhamos que a cada falha de um componente, sem direito a reparo, o nosso sistema tem uma perda econômica de R\$ 500,00 para cada CPU, R\$ 5000,00 para cada disco rígido e R\$ 1000,00 para cada memória. Para o caso onde a falha pode ser reparada, a perda econômica é de R\$ 100,00 para cada um dos três tipos de componentes. A medida de interesse a ser calculada é o total de capital perdido até que o sistema falhe totalmente (estado onde todos os componentes não estão em funcionamento). Este cenário pode representar um *site web* comercial, onde a perda de desempenho afeta o capital total acu-

4.8 Sumário das Contribuições

Tabela 4.20: Resultados para o sistema de banco de dados, considerando o comportamento das unidades de memória.

L.I.	ECRA	L.S.	E.R.
$-4.4740e10$	$-4.4741e10$	$-4.4753e10$	$1.2e-4$

Tabela 4.21: Redução do espaço de estados para o sistema de banco de dados composto por CPUs, discos rígidos e unidades de memória.

$ \Omega $	$ \tilde{\Omega} $	Taxa de Redução	N
531441	4096	129	6

mulado. A Tabela 4.20 mostra os resultados para este sistema. O valor exato para o valor esperado da perda econômica é obtido explorando a propriedade de *strong lumpability*. Os valores para os parâmetros são: CPU: $\phi c = 2e-6$, $\phi(1 - c) = 1e-5$; Disco: $\phi c = 1e-5$, $\phi(1 - c) = 1e-5$; Memória: $\phi c = 1e-5$, $\phi(1 - c) = 1e-4$ e $\gamma = 1$ para todos os componentes. Os valores negativos são explicados devido a natureza da medida e significam que o sistema está perdendo capital. A Tabela 4.21 mostra a taxa de redução no total de estados do modelo.

4.8 Sumário das Contribuições

Como principais contribuições deste capítulo, podemos citar os seguintes pontos:

1. O aumento da complexidade dos sistemas a serem representados e o surgimento de diversas outras medidas de interesse fazem com que novos tipos de métodos continuem a ser pesquisados e propostos. Partindo desta motivação, apresentamos um método aproximado que calcula a medida denominada valor esperado da recompensa acumulada até a absorção, considerando a classe de modelos Markovianos onde o espaço de estado é da ordem de milhões de estados. Os processos aqui abordados não são irredutíveis, diferentemente dos trabalhos encontrados anteriormente na literatura. O método apresentado é extremamente atrativo para os casos onde o processo que modela o sistema em estudo permanece a maior parte do tempo em um conjunto pequeno de estados;
2. O método apresentado possui controle de erro, apresentando limites inferior e su-

perior. Um amplo conjunto de exemplos foi apresentado, mostrando a eficiência do método proposto. Para todos os exemplos, observamos que os sistemas de equações a serem resolvidos possuíam uma dimensão muito menor do que o espaço de estados original do modelo.

3. O conjunto de medidas de interesse que podem ser obtidas por este método é bem amplo, variando de medidas importantes em confiabilidade, como o *Mean Time to Failure*, a medidas em análise de desempenho, como tamanho médio de uma fila até que o roteador entre em estado de congestionamento, etc.

Capítulo 5

Medidas de Interesse para Modelos com Várias Recompensas Associadas a cada Estado

5.1 Introdução

Conforme mostrado nos capítulos anteriores, modelos Markovianos com recompensas são uma ferramenta poderosa para obtenção de inúmeras medidas de interesse, como medidas de *dependability* e desempenho (Capítulo 4), ou medidas relacionadas a modelos de fila de fluido (Capítulo 3).

A característica comum em grande parte dos métodos encontrados na literatura para obtenção destas medidas de interesse é que cada estado da cadeia de Markov que representa o sistema a ser estudado possui uma *ÚNICA* taxa de recompensa associada a ele ([31],[5],[30],[100],[106],[128],[3],[4],[22],[19],[21]). Um conjunto restrito de trabalhos considera a variância da taxa associada a um estado. Estes são os modelos de *fluido de segunda ordem* ([78],[97]). Nestes modelos são considerados o primeiro e segundo momentos (média e variância) da distribuição do fluido atribuído ao processo, ou, equivalentemente, da recompensa atribuída por unidade de tempo em um estado. Estes modelos são úteis, por exemplo, quando os modelos de fluido de primeira ordem (que consideram somente a média), não conseguem capturar a variabilidade da chegada de informações em uma fila. Este comportamento difere do comportamento dos modelos que iremos tratar neste capítulo. Nos modelos aqui apresentados, várias recompensas de taxas são atribuídas a um estado. A cada entrada em um estado o processo acumula ou perde uma

5.1 Introdução

determinada recompensa segundo uma determinada probabilidade. A Figura 5.1 mostra a diferença entre os modelos de fluido de segunda ordem e o modelo que será considerado neste capítulo.

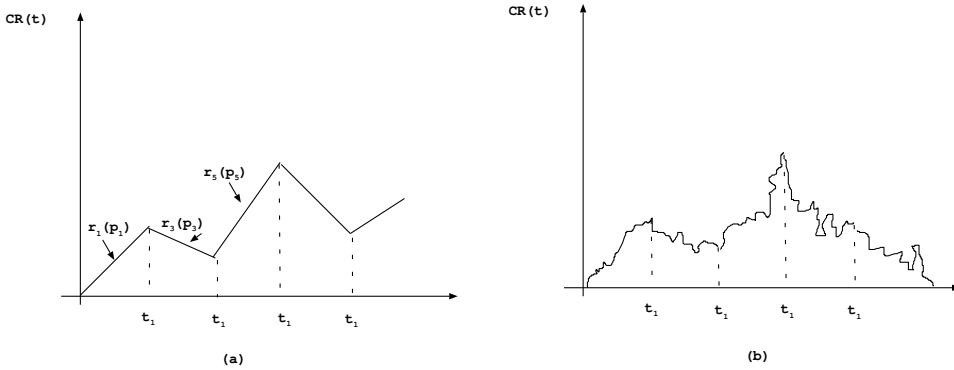


Figura 5.1: (a) Modelo considerado neste capítulo, onde a cada recompensa r_i , associamos uma probabilidade de ocorrência da mesma. (b) Fluido de segunda ordem.

A motivação para os modelos considerados neste capítulo são as cadeias de Markov ocultas (*Hidden Markov Models - HMM*), também referenciados como *Markov sources* ou funções probabilísticas de uma cadeia de Markov. A teoria básica para esta classe de modelos foi publicada em uma série de artigos clássicos por Baum *et al* nas décadas de 60 e 70, sendo utilizada em aplicações para processamento de sinal de voz (referências destes trabalhos podem ser encontradas em [119]). Todavia, a utilização destes modelos para aplicações em diversas outras áreas, incluindo redes de computadores, tem aumentado nos últimos anos ([41],[37],[138],[50],[126]).

Como exemplo de motivação do estudo de métodos para obter medidas de interesse para modelos com várias recompensas em cada estado consideremos o dimensionamento de uma rede de dados [37]. O tráfego de um canal da *Internet* pode ser modelado a partir de um modelo de Markov oculto. Neste caso, a recompensa de taxa é a taxa de chegada de pacotes no canal. Devido a diferença de algumas ordens de grandeza nas taxas de chegada e taxas de transição entre os estados, a abordagem de fluido é computacionalmente mais atrativa que a modelagem de pacotes.

Uma das medidas de interesse necessária para o dimensionamento da rede é a distribuição de probabilidade do conteúdo da fila de um roteador. Como um determinado estado pode ter um conjunto de recompensas atribuído a ele, os métodos tradicionais citados no começo desta seção não podem ser diretamente aplicados para esta medida. Desta forma, um modelo Markoviano equivalente foi utilizado no trabalho apresentado em [37] e as medidas de interesse foram calculadas utilizando os métodos tradicionais, onde uma

única taxa de recompensa é atribuída a cada estado. Chamaremos este modelo de modelo Markoviano expandido. Em muitos casos, o aumento considerável no tamanho da matriz geradora referente ao modelo Markoviano expandido inviabiliza a aplicação de alguns métodos, seja por limitações decorrentes do alto custo computacional ou por problemas de precisão numérica.

O maior objetivo deste capítulo é propor um método aproximado, mas computacionalmente eficiente para a obtenção da distribuição da recompensa acumulada de um modelo de Markov oculto. Considerando CR a recompensa acumulada e x uma quantidade previamente escolhida, propomos uma equação para a obtenção de $P[CR \leq x] = \lim_{t \rightarrow \infty} P[CR(t) \leq x]$, sendo $0 \leq CR \leq B$. A solução aproximada proposta é baseada em um algoritmo iterativo, cuja acurácia é validada através de diversos exemplos. Nosso método considera que a recompensa acumulada possui limites inferior e superior, pois este é o caso de grande parte dos modelos de sistemas de comunicação. No entanto, o caso onde a recompensa não é limitada superiormente é facilmente obtido a partir da equação diferencial obtida, diferindo somente nas condições de contorno que devem ser consideradas.

Este capítulo está estruturado da seguinte forma. Na Seção 5.2 apresentamos a descrição dos Modelos de Markov oculto e expandido. Uma equação diferencial que descreve o comportamento da variável aleatória CR é obtida na Seção 5.3. O objetivo da Seção 5.4 é descrever o algoritmo iterativo proposto, que fornece uma solução aproximada para a medida de interesse $P[CR \leq x] = \lim_{t \rightarrow \infty} P[CR(t) \leq x]$. O conjunto de exemplos que estudam a acurácia da aproximação e a taxa de convergência do algoritmo iterativo é apresentado na Seção 5.5. As principais contribuições deste capítulo são descritas na Seção 5.6.

5.2 Descrição dos Modelos de Markov oculto e expandido

De uma forma geral, um modelo de Markov oculto é composto por dois processos estocásticos dependentes entre si. O primeiro dos processos é uma cadeia de Markov e o segundo componente de um HMM é um processo de observações, cuja distribuição, a qualquer instante de tempo, é completamente determinada pelo estado atual da cadeia correspondente. Uma referência completa sobre modelos de Markov ocultos pode ser encontrada em [119]. As definições que iremos apresentar nesta seção consideram modelos ocultos em tempo discreto, facilmente obtidos a partir dos modelos ocultos em tempo

5.2 Descrição dos Modelos de Markov oculto e expandido

contínuo após aplicação da técnica de uniformização, aplicada tanto ao processo oculto quanto ao de observações (Capítulo 2). Ou seja, a cada evento do processo uniformizado, com taxa de uniformização Λ , o valor do processo de observação também é modificado, de acordo com a distribuição de probabilidades que depende somente do estado do processo oculto.

Seja $\mathcal{X}^* = \{X^*(k); k \geq 0\}$ a cadeia de Markov em tempo discreto com estados pertencentes ao conjunto \mathcal{S} , com cardinalidade $|\mathcal{S}|$, sendo esta equivalente a cadeia de Markov em tempo contínuo $\mathcal{X} = \{X(t); t \geq 0\}$. A distribuição do estado inicial é dada pelo vetor $\boldsymbol{\pi}$ com $|\mathcal{S}|$ elementos:

$$\pi_i = P[X^*(0) = i]. \quad (5.1)$$

As probabilidades de transição entre estados são definidas pela matriz de ordem $|\mathcal{S}| \times |\mathcal{S}|$, $\mathbf{P} = \{p_{ij}\}$, onde:

$$p_{ij} = P[X^*(k) = j | X^*(k-1) = i]. \quad (5.2)$$

O processo de observações $\mathcal{Y} = \{Y(k); k \geq 0\}$, tem $|\mathcal{R}|$ estados pertencentes ao conjunto \mathcal{R} de recompensas de taxa¹ (no contexto das aplicações deste capítulo, o termo *emissões* será substituído por recompensas), sendo governado pela matriz $|\mathcal{S}| \times |\mathcal{R}|$, $\boldsymbol{\Gamma} = \{\gamma_i^j\}$, com:

$$\gamma_i^j = P[Y(k) = j | X^*(k) = i]. \quad (5.3)$$

Por questão de notação, o conjunto de parâmetros que define um modelo particular é dado pela tripla $\boldsymbol{\theta} = (\boldsymbol{\pi}, \mathbf{P}, \boldsymbol{\Gamma}, \Lambda)$.

Seja uma cadeia discreta de Markov $\mathcal{W} = \{W(k); k \geq 0\}$, com o conjunto de estados \mathcal{T} , com cardinalidade $|\mathcal{S}| \times |\mathcal{R}|$, sendo cada estado constituído por um par entre todas as combinações dos elementos dos conjuntos \mathcal{S} e \mathcal{R} . A matriz de probabilidades de transição é definida por $\hat{\mathbf{P}}$. Estas probabilidades são obtidas através da seguinte expressão:

$$\hat{p}_{i,(j,s)} = p_{ij} \gamma_j^s. \quad (5.4)$$

A equivalência entre as cadeias \mathcal{X}^* e \mathcal{W} , em relação a probabilidade de emitir uma seqüência de recompensas $R = r_1 r_2 r_3 \dots r_K$, com tamanho K , é garantida pelo Lema 5.1. A equivalência é provada em relação a emissão de uma seqüência de recompensas dado que a medida de interesse refere-se a variável aleatória CR , para o caso onde $t \rightarrow \infty$.

¹Por questão de brevidade, consideraremos “recompensa” significando recompensa de taxa.

5.2 Descrição dos Modelos de Markov oculto e expandido

Lema 5.1 *Seja uma determinada seqüência de observações $R = r_1 r_2 r_3 \dots r_K$, emitida por um modelo de Markov oculto θ . A probabilidade de emitir esta seqüência, dado o modelo θ , ou seja, $P(R|\theta)$, é igual a probabilidade de emissão desta mesma seqüência quando consideramos a cadeia de Markov discreta \mathcal{W} (que denominamos cadeia de Markov expandida \mathcal{W}), cujo os elementos da matriz de probabilidade de transição entre estados é definida por 5.4.*

Prova: Para esta prova, primeiramente consideremos a probabilidade de emitir a seqüência R para o modelo de Markov oculto. Seja uma determinada seqüência de estados $c = \{s_1 s_2 \dots s_K\}$ no conjunto de todas as seqüências \mathcal{C} possíveis de tamanho K na cadeia \mathcal{X}^* . Seja $c(k)$ uma função cujo valor é o índice de uma das possíveis recompensas que podem ser atribuídas a um estado. Portanto, a cada entrada no estado s_k , a recompensa $r_{c(k)}$ é escolhida com probabilidade $\gamma_{s_k}^{r_{c(k)}}$. A probabilidade da emissão da seqüência de observação R dada a seqüência de estados c é dada por (conforme [119]):

$$P[R | c, \theta] = \prod_{k=1}^K P[r_k | s_k, \theta].$$

Assumindo independência estatística das observações, temos que:

$$P[R | c, \theta] = \gamma_{s_1}^{r_{c(1)}} \gamma_{s_2}^{r_{c(2)}} \gamma_{s_3}^{r_{c(3)}} \dots \gamma_{s_K}^{r_{c(K)}},$$

e a probabilidade de uma determinada seqüência de estados c é dada por:

$$P[c | \theta] = \pi_{s_1} p_{s_1 s_2} p_{s_2 s_3} \dots p_{s_{K-1} s_K}.$$

A probabilidade conjunta das seqüências R e c é o produto dos termos $P[R | c, \theta]$ e $P[c | \theta]$. A probabilidade da seqüência R é obtida através da soma de todas as possíveis seqüências de estados pertencentes ao conjunto \mathcal{C} :

$$P[R | \theta] = \sum_{\forall c \in \mathcal{C}} P[R | c, \theta] P[c, \theta] = \sum_{s_1 s_2 \dots s_K} \pi_{s_1} \gamma_{s_1}^{r_{c(1)}} p_{s_1 s_2} \gamma_{s_2}^{r_{c(2)}} p_{s_2 s_3} \dots p_{s_{K-1} s_K} \gamma_{s_K}^{r_{c(K)}}. \quad (5.5)$$

A segunda parte da prova obtém a probabilidade de emissão da seqüência R considerando o processo de Markov expandido \mathcal{W} . Neste caso, cada estado está associado explicitamente a uma recompensa. Seja um determinado caminho de tamanho K representado por m pertencente ao conjunto de caminhos \mathcal{M} , cuja seqüência das recompensas emitidas é igual a R . A probabilidade da emissão desta seqüência está atrelada a probabilidade de percorrer o caminho m e é dada por:

$$P[R, m | \mathcal{W}] = \pi_{s_1} \hat{p}_{s_1, (s_2, r_{c(2)})} \hat{p}_{s_2, (s_3, r_{c(3)})} \dots \hat{p}_{s_{K-1}, (s_K, r_{c(K)})}. \quad (5.6)$$

5.2 Descrição dos Modelos de Markov oculto e expandido

Substituindo a Equação 5.4 na Equação 5.6, temos:

$$P[R, m | \mathcal{W}] = \pi_{s_1} \gamma_{s_1}^{r_{c(1)}} p_{s_1 s_2} \gamma_{s_2}^{r_{c(2)}} p_{s_2 s_3} \gamma_{s_3}^{r_{c(3)}} \cdots p_{s_{K-1} s_K} \gamma_{s_K}^{r_{c(K)}}. \quad (5.7)$$

Somando para todos os caminhos possíveis pertencentes ao conjunto \mathcal{M} , temos:

$$P[R | \mathcal{W}] = \sum_{\forall m \in \mathcal{M}} P[R, m | \mathcal{W}] = \sum_{s_1 s_2 \cdots s_K} \pi_{s_1} \gamma_{s_1}^{r_{c(1)}} p_{s_1 s_2} \gamma_{s_2}^{r_{c(2)}} p_{s_2 s_3} \gamma_{s_3}^{r_{c(3)}} \cdots p_{s_{K-1} s_K} \gamma_{s_K}^{r_{c(K)}}. \quad (5.8)$$

As Equações 5.5 e 5.8 são iguais, o que prova a equivalência do processo de Markov expandido \mathcal{W} e o processo de Markov oculto θ , quando consideramos a emissão de uma seqüência R de recompensas (ou observações). ●

A principal característica do processo de Markov \mathcal{W} , é que o seu espaço de estados é uma expansão do espaço de estados do processo oculto. A Figura 5.2(b) mostra a representação gráfica do modelo expandido equivalente a cadeia de Markov oculta da Figura 5.2(a).

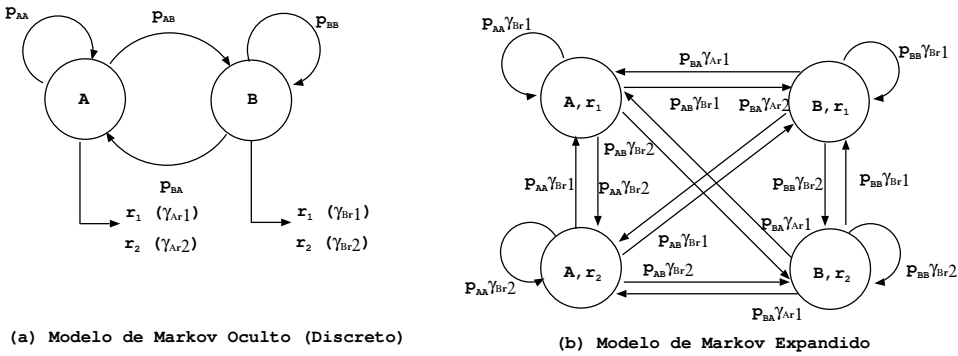


Figura 5.2: Representação gráfica do modelo de Markov expandido e o modelo de Markov oculto, para um pequeno exemplo com dois estados ocultos e emissão de recompensas $\{r_1, r_2\}$.

A aplicação direta do método descrito em [5], sem um prévio estudo da estrutura da matriz \hat{P} do modelo expandido pode resultar em um alto custo computacional. Este alto custo computacional é devido a dimensão $|\mathcal{S}| \times |\mathcal{R}|$ do espaço de estados do modelo expandido que pode ser muito grande, inviabilizando a aplicação dos métodos citados anteriormente, seja por questão de custo computacional ou até mesmo por problemas numéricos, devido ao cálculo de autovalores ou autovetores.

5.3 Uma Equação Diferencial para o Modelo de Markov oculto

Esta seção apresenta uma equação diferencial que representa o processo HMM, quando a medida de interesse a ser calculada é a distribuição da probabilidade acumulada, em estado estacionário, da variável aleatória CR.

Seja $\mathcal{X} = \{X(t); t \geq 0\}$ uma cadeia de Markov homogênea de tempo contínuo e espaço de estados finito, \mathcal{S} , com cardinalidade $|\mathcal{S}|$ e gerador infinitesimal $\mathbf{Q} = \{q_{ij}\}$. Seja $\mathcal{Y} = \{Y(t); t \geq 0\}$ uma função probabilística de \mathcal{X} com espaço de estados no conjunto \mathcal{R} , com $|\mathcal{R}|$ elementos. Como nos capítulos anteriores, a cadeia \mathcal{X} é transformada em uma cadeia em tempo discreto através da técnica de uniformização [28], onde $\Lambda > \max_i q_{ii}$, sendo este processo uniformizado definido como \mathcal{X}^* . A matriz $\mathbf{P} = \mathbf{I} + \mathbf{Q}/\Lambda$ representa a matriz de probabilidades de transição entre os estados da cadeia uniformizada.

A cada evento do processo uniformizado, uma nova recompensa é escolhida de acordo com a distribuição de probabilidades que depende somente do estado do processo \mathcal{X} . Por exemplo, se $X(t) = j$, então $Y(t) = l$, com probabilidade γ_j^l . O valor de $Y(t)$ determina a recompensa $r_{Y(t)}$, e, por conseqüência, a taxa de mudança da recompensa acumulada $CR(t)$. A recompensa acumulada em um instante de tempo t é definida como:

$$CR(t) = \int_0^t r_{Y(t)} dt. \quad (5.9)$$

Sejam $\hat{w}_{ij}(y, t)$ igual a $P[CR(t) = y, X(t) = j | X(0) = i]^2$ e $w_{ij}(y, t)$ igual a $P[CR(t) \leq y, X(t) = j | X(0) = i]$ que será obtida a partir de $\hat{w}_{ij}(y, t)$. Sejam, também, $\hat{w}_{ijl}(y, t)$ definida como $P[CR(t) = y, X(t) = j, Y(t) = l | X(0) = i]$ e $w_{ijl}(y, t)$ definida como $P[CR(t) \leq y, X(t) = j, Y(t) = l | X(0) = i]$. Finalizando, temos que $\hat{w}_{ij}(y, t) = \sum_{\forall l} \hat{w}_{ijl}(y, t)$ e, similarmente, que $w_{ij}(y, t) = \sum_{\forall l} w_{ijl}(y, t)$. O cálculo de $\hat{w}_{ij}(y, t)$ e $w_{ijl}(y, t)$ será feito seguindo os passos apresentados em [28].

Para um determinado tempo t e um nível de recompensa y , sendo o ponto correspondente $p_1 = (y, t)$, temos que a expansão de Taylor em primeira-ordem de $\hat{w}_{ijl}(y, t)$, em torno do ponto $p_0 = (y - hr_{Y(t-h)}, t - h)$ é dada por³:

$$\hat{w}_{ijl}(p_1) = \hat{w}_{ijl}(p_0) + \frac{\partial \hat{w}_{ijl}(p_0)}{\partial t} h + \frac{\partial \hat{w}_{ijl}(p_0)}{\partial y} hr_{Y(t-h)} + o(h^2), \quad (5.10)$$

²A variável aleatória CR é contínua. Portanto, $P[CR(t) = y]$ é um abuso de notação. Entretanto, a notação facilitará a explanação, da mesma maneira apresentada em [28].

³Ou seja, ao realizar um deslocamento para trás, a partir de p_1 , de h unidades de tempo para o passado, e respectivamente $hr_{Y(t-h)}$ unidades de recompensa, obtemos o ponto p_0 .

5.3 Uma Equação Diferencial para o Modelo de Markov oculto

onde $o(h^2)$ é uma função que representa o erro de truncamento da série de Taylor, tal que:

$$\lim_{h \rightarrow 0} \frac{o(h^2)}{h} = 0$$

com a função $o(h^2)$ tendendo mais rapidamente para zero do que o valor de h .

A partir da expansão de Taylor, a derivada parcial da função $\hat{w}_{ijl}(y, t)$ é dada por:

$$\frac{\partial \hat{w}_{ijl}(p_0)}{\partial t} = \frac{\hat{w}_{ijl}(p_1) - \hat{w}_{ijl}(p_0)}{h} - \frac{\partial \hat{w}_{ijl}(p_0)}{\partial y} r_{Y(t-h)} + \frac{o(h^2)}{h} \quad (5.11)$$

Passemos a definição da expressão que determina $\hat{w}_{ijl}(y, t)$. Aplicaremos a propriedade básica de processos Markovianos, que consiste no fato de que, se for conhecido o estado da cadeia no instante de tempo $t - h$, ao avaliar o estado da cadeia no tempo t não é necessário recorrer a nenhum instante de tempo anterior a $t - h$. Condicionando o valor da função $\hat{w}_{ijl}(y, t)$ no valor das variáveis aleatórias CR , X e Y no tempo $t - h$, podemos escrever:

$$\hat{w}_{ijl}(y, t) = \sum_{k=1}^{|\mathcal{S}|} \sum_{n=1}^{|\mathcal{R}|} \int_z P[CR(t) = y, X(t) = j, Y(t) = l | CR(t-h) = z, X(t-h) = k, Y(t-h) = n, X(0) = i] P[CR(t-h) = z, X(t-h) = k, Y(t-h) = n | X(0) = i] dz \quad (5.12)$$

Consideremos h suficientemente pequeno. A probabilidade de ocorrência de mais de um evento no intervalo $(t - h, t]$, representada por $o(h)$ tende a zero, quando h tende a zero (como definido em [133], página 407):

$$\lim_{h \rightarrow 0} \frac{o(h)}{h} = 0$$

$CR(t)$ torna-se independente de $X(t)$ e $Y(t)$, condicionado nos processos \mathcal{CR} , \mathcal{X} e \mathcal{Y} , no instante de tempo $t - h$. $X(t)$ e $Y(t)$ dependem somente de $X(t - h)$ e $Y(t - h)$. Podemos reescrever a equação 5.12 como:

$$\hat{w}_{ijl}(y, t) = \sum_{k=1}^{|\mathcal{S}|} \sum_{n=1}^{|\mathcal{R}|} \int_z P[CR(t) = y | CR(t-h) = z, X(t-h) = k, Y(t-h) = n] P[X(t) = j, Y(t) = l | X(t-h) = k, Y(t-h) = n] \hat{w}_{ikn}(z, t-h) dz. \quad (5.13)$$

Com h tendendo a zero, temos que:

$$CR(t) = CR(t-h) + r_{Y(t-h)} h + o(h). \quad (5.14)$$

Intuitivamente, esta última equação indica que a recompensa acumulada no instante t é aquela acumulada no instante $t - h$ acrescida da recompensa associada ao estado

$X(t-h)$ multiplicada por h . A função $o(h)$ novamente representa os termos relacionados com mudanças no intervalo de tempo $(t-h, t]$, e tende mais rapidamente a zero, dado que h tende a zero.

Baseado em 5.14, a distribuição de probabilidades $P[CR(t) = y | CR(t-h) = z, X(t-h) = k, Y(t-h) = n]$, pode ser representada pela função δ de Dirac, $\delta(x)$, com a seguinte propriedade de amostragem [123]:

$$\int_x \delta(x) f(x) dx = f(0). \quad (5.15)$$

Seja $N(t)$ o número de transições ocorridas no processo \mathcal{X} durante o intervalo de tempo $[0, t]$. Através da técnica de uniformização, os eventos de transições em \mathcal{X} ocorrem de acordo com um processo de *Poisson* com taxa Λ . A medida $P[X(t) = j, Y(t) = l | X(t-h) = k, Y(t-h) = n]$ é definida por:

$$P[X(t) = j, Y(t) = l | X(t-h) = k, Y(t-h) = n] = \mathbb{1}\{k = j, n = l\} e^{-\Lambda h} + (\mathbb{1}\{k = j\} + \frac{q_{kj}}{\Lambda}) \gamma_j^l \Lambda h e^{-\Lambda h} + o(h) \quad (5.16)$$

sendo $\mathbb{1}\{\}$ a função indicadora que assume valor igual a 1 se a condição descrita em $\{\}$ é verdadeira e zero caso contrário.

O primeiro termo do lado direito da Equação 5.16 representa o caso onde não existem transições no processo \mathcal{X} , sendo que este se encontra no estado j , emitindo a recompensa l . O segundo termo está relacionado com a ocorrência de um entre dois tipos de eventos: no primeiro tipo, o processo \mathcal{X} se encontra no estado final j e somente o processo de observação muda, sendo l a recompensa escolhida. No segundo tipo, existe uma transição entre estados do processo \mathcal{X} e do processo \mathcal{Y} , com a emissão da l -ésima recompensa. O termo $o(h)$ representa a probabilidade da ocorrência de dois ou mais eventos e tende a zero, dado que estamos considerando h pequeno o suficiente.

Usando a função δ de Dirac e aplicando a Equação 5.16 em 5.13, temos que:

$$\begin{aligned} \hat{w}_{ijl}(y, t) &= \sum_{k=1}^{|\mathcal{S}|} \sum_{n=1}^{|\mathcal{R}|} \int_z \delta(y - (z + hr_n + o(h))) \\ &\quad [\mathbb{1}\{k = j, n = l\} e^{-\Lambda h} + (\mathbb{1}\{k = j\} + \frac{q_{kj}}{\Lambda}) \gamma_j^l \Lambda h e^{-\Lambda h} + o(h)] \\ &\quad \hat{w}_{ikn}(z, t-h) dz \\ &= \sum_{k=1}^{|\mathcal{S}|} \sum_{n=1}^{|\mathcal{R}|} [\mathbb{1}\{k = j, n = l\} e^{-\Lambda h} + (\mathbb{1}\{k = j\} + \frac{q_{kj}}{\Lambda}) \gamma_j^l \Lambda h e^{-\Lambda h} + o(h)] \\ &\quad \hat{w}_{ikn}(y - hr_n + o(h), t-h) \end{aligned} \quad (5.17)$$

5.3 Uma Equação Diferencial para o Modelo de Markov oculto

Considerando os termos determinados pelas funções indicadoras $\mathbb{1}\{k = j, n = l\}$ e $\mathbb{1}\{k = j\}$, temos a seguinte equação para o termo $\hat{w}_{ijl}(y, t)$:

$$\begin{aligned} \hat{w}_{ijl}(y, t) &= e^{-\Lambda h} \hat{w}_{ijl}(y - hr_l + o(h), t - h) + \gamma_j^l \Lambda e^{-\Lambda h} \sum_{n=1}^{|\mathcal{R}|} \hat{w}_{ijn}(y - hr_n + o(h), t - h) + \\ &\quad \sum_{k=1}^{|\mathcal{S}|} q_{kj} \gamma_j^l h e^{-\Lambda h} \sum_{n=1}^{|\mathcal{R}|} \hat{w}_{ikn}(y - hr_n + o(h), t - h) + o(h) \end{aligned} \quad (5.18)$$

Após a divisão da Equação 5.18 por h , obtemos:

$$\begin{aligned} \frac{\hat{w}_{ijl}(y, t)}{h} &= \frac{e^{-\Lambda h} \hat{w}_{ijl}(y - hr_l + o(h), t - h)}{h} + \gamma_j^l \Lambda e^{-\Lambda h} \sum_{n=1}^{|\mathcal{R}|} \hat{w}_{ijn}(y - hr_n + o(h), t - h) + \\ &\quad \sum_{k=1}^{|\mathcal{S}|} q_{kj} \gamma_j^l e^{-\Lambda h} \sum_{n=1}^{|\mathcal{R}|} \hat{w}_{ikn}(y - hr_n + o(h), t - h) + \frac{o(h)}{h} \end{aligned} \quad (5.19)$$

Seja $p_0(y_n, t)$ o ponto onde o tempo é igual a $t - h$ e o nível de recompensa é igual a $y - hr_n + o(h)$, que, por brevidade de notação, escreveremos $p_0(y_n)$. Aplicando (5.19) na Equação (5.11), temos:

$$\begin{aligned} \frac{\partial \hat{w}_{ijl}(p_0)}{\partial t} &= \frac{e^{-\Lambda h} \hat{w}_{ijl}(p_0(y_l)) - \hat{w}_{ijl}(p_0)}{h} + \gamma_j^l \Lambda e^{-\Lambda h} \sum_{n=1}^{|\mathcal{R}|} \hat{w}_{ijn}(p_0(y_n)) + \\ &\quad \sum_{k=1}^{|\mathcal{S}|} q_{kj} \gamma_j^l e^{-\Lambda h} \sum_{n=1}^{|\mathcal{R}|} \hat{w}_{ikn}(p_0(y_n)) - \frac{\partial \hat{w}_{ijl}(p_0)}{\partial y} r_{Y(t-h)} + \frac{o(h)}{h}. \end{aligned} \quad (5.20)$$

Como consideramos um valor suficientemente pequeno para h , a recompensa em $Y(t - h)$ se iguala a $Y(t)$, ou equivalentemente, a l . Adicionalmente, quando h tende a zero, os pontos p_0 e $p_0(y_n)$, para todo valor n , convergem para $p_1 = (y, t)$. Assim, a Equação (5.20) é simplificada da seguinte forma:

$$\frac{\partial \hat{w}_{ijl}(y, t)}{\partial t} = \hat{w}_{ij}(y, t) \Lambda \gamma_j^l + \sum_{k=1}^{|\mathcal{S}|} \hat{w}_{ik}(y, t) q_{kj} \gamma_j^l - \frac{\partial \hat{w}_{ijl}(y, t)}{\partial y} r^l - \hat{w}_{ijl}(y, t) \Lambda. \quad (5.21)$$

A medida de interesse $w_{ij} = P[CR(t) \leq y, X(t) = j | X(0) = i]$ pode ser obtida através da Equação 5.21:

$$\begin{aligned} \frac{\partial \int_{-\infty}^y \hat{w}_{ijl}(y, t)}{\partial t} &= \int_{-\infty}^y \hat{w}_{ij}(y, t) \Lambda \gamma_j^l + \sum_{k=1}^{|\mathcal{S}|} \int_{-\infty}^y \hat{w}_{ik}(y, t) q_{kj} \gamma_j^l - \frac{\partial \int_{-\infty}^y \hat{w}_{ijl}(y, t)}{\partial y} r^l \\ &\quad - \int_{-\infty}^y \hat{w}_{ijl}(y, t) \Lambda \end{aligned}$$

5.3 Uma Equação Diferencial para o Modelo de Markov oculto

Utilizando a propriedade: $F_X(x) = P[X \leq x] = \int_{-\infty}^x f_X(t)dt$, onde $F_X(x)$ é a distribuição de probabilidade (representada por $w_{ij}(y, t)$ e $w_{ijl}(y, t)$) e f_X é a função densidade de probabilidade (representada por $\hat{w}_{ij}(y, t)$ e $\hat{w}_{ijl}(y, t)$), temos:

$$\frac{\partial w_{ijl}(y, t)}{\partial t} = w_{ij}(y, t)\Lambda\gamma_j^l + \sum_{k=1}^{|S|} w_{ik}(y, t)q_{kj}\gamma_j^l - \frac{\partial w_{ijl}(y, t)}{\partial y}r^l - w_{ijl}(y, t)\Lambda. \quad (5.22)$$

Seja a matriz $\mathbf{W}(y)$, cujo o elemento (i, j) é igual a $\{w_{ij}(y)\}$ e a matriz $\mathbf{W}^l(y)$, com elemento (i, j) igual a $w_{ijl}(y)$. A Equação 5.22 pode ser reescrita como:

$$\frac{\partial \mathbf{W}^l(y, t)}{\partial t} + \frac{\partial \mathbf{W}^l(y, t)}{\partial y}r^l = \mathbf{W}(y, t) [\mathbf{Q} + \Lambda\mathbf{I}] \text{diag}\{\gamma_j^l\}_j - \Lambda \mathbf{W}^l(y, t) \quad (5.23)$$

onde $\text{diag}\{\gamma_j^l\}_j$ é a matriz diagonal com o j -ésimo elemento na diagonal igual a γ_j^l .

Considerando a análise em estado estacionário, com t tendendo ao infinito, a diferencial $\partial \mathbf{W}^l / \partial t$, tende a 0 e todos os termos se tornam independentes de t . Adicionalmente, as medidas de interesse se tornam independentes do estado inicial do processo [133]. A partir da Equação 5.23, temos:

$$\frac{d\mathbf{w}^l(y)}{dy} = \mathbf{w}(y) [\mathbf{Q} + \Lambda\mathbf{I}] \text{diag}\left\{\frac{\gamma_j^l}{r^l}\right\}_j - \frac{\Lambda}{r^l} \mathbf{w}^l(y), \quad (5.24)$$

onde o j -ésimo elemento do vetor $\mathbf{w}^l(y)$ é igual a $\lim_{t \rightarrow \infty} P[CR(t) \leq y, X(t) = j, Y(t) = l]$ e o j -ésimo elemento do vetor $\mathbf{w}(y)$ é igual a $\lim_{t \rightarrow \infty} P[CR(t) \leq y, X(t) = j]$.

A Equação 5.24 representa a medida de interesse, em estado estacionário, considerando cada uma das possíveis recompensas l que podem ser emitidas em um determinado estado j do processo oculto \mathcal{X} .

A Equação 5.24, pode ser reescrita da seguinte forma:

$$\frac{d\mathbf{w}^{(l)}}{dy}(y) = \mathbf{w}(y) \mathbf{Q} \text{diag}\left\{\frac{\gamma_j^l}{r^l}\right\}_j + \frac{\Lambda}{r^l} (\mathbf{w}(y) \bullet \gamma^l - \mathbf{w}^{(l)}(y)), \quad (5.25)$$

onde o vetor γ^l é um vetor linha, cujo j -ésimo elemento é a probabilidade de emitir a recompensa l , no estado j , e o símbolo \bullet representa a multiplicação termo a termo entre os vetores $\mathbf{w}(y)$ e γ^l .

Conforme citado anteriormente, para cada estado oculto do processo \mathcal{X}^* , $|\mathcal{R}|$ recompensas podem ser emitidas. Para a obtenção da medida de interesse final $P[CR(t) \leq y, X(t) = j]$, $|\mathcal{R}|$ equações semelhantes a equação 5.25 deverão ser resolvidas. Vale ressaltar que não é obrigatório que todos os estados ocultos emitam o mesmo conjunto

5.3 Uma Equação Diferencial para o Modelo de Markov oculto

de recompensas. Caso uma determinada recompensa l não seja emitida no estado j , γ_j^l é igual a zero e a j -ésima entrada no vetor $\mathbf{w}^l(y)$ é igual a zero.

Dado que a medida final a ser calculada é independente da recompensa sendo emitida no estado, a manipulação algébrica mais natural a ser realizada no conjunto com $|\mathcal{R}|$ equações é fixar o estado final j e somar as contribuições de todas as recompensas. Desta forma, temos o seguinte sistema de equações diferenciais ordinárias, cuja a solução fornece o valor para a medida $\mathbf{w}(y)$:

$$\frac{d\mathbf{w}(y)}{dy} = \mathbf{w}(y) [\mathbf{Q} + \Lambda \mathbf{I}] \text{diag} \left\{ \sum_{\forall l} \left(\frac{\gamma_j^l}{r^l} \right) \right\}_j - \Lambda \sum_{\forall l} \left(\frac{\mathbf{w}^{(l)}(y)}{r^l} \right), \quad (5.26)$$

sendo este sistema equivalente ao seguinte sistema de equações diferenciais ordinárias⁴

$$\frac{d(\mathbf{w}(y))^T}{dy} = \text{diag} \left\{ \sum_{\forall l} \left(\frac{\gamma_j^l}{r^l} \right) \right\}_j [\mathbf{Q} + \Lambda \mathbf{I}]^T (\mathbf{w}(y))^T - \Lambda \sum_{\forall l} \left(\frac{(\mathbf{w}^{(l)}(y))^T}{r^l} \right) \quad (5.27)$$

onde os vetores $(\mathbf{w}(y))^T$ e $(\mathbf{w}^l(y))^T$ são vetores colunas, e $\mathbf{w}(y) = \sum_{\forall l} \mathbf{w}^{(l)}(y)$. Durante todo o restante do capítulo, assumiremos, por brevidade de notação, que os vetores $\mathbf{w}(y)$ e $\mathbf{w}^l(y)$, sem o símbolo T de transposição, são vetores colunas. A Equação 5.25 também pode ser reescrita de maneira análoga.

A Equação 5.26 é semelhante a Equação 3.4, apresentada no Capítulo 3. No entanto, para a Equação 5.26 não é possível obtermos, diretamente, uma solução semelhante a solução da Equação 3.4 escrita através da expansão espectral da matriz \mathbf{QR}^{-1} .

Para obtermos a solução da Equação 5.26, entre as diversas possíveis decorrentes das constantes de integração relacionadas as equações diferenciais, devemos estabelecer as condições de contorno inerentes ao problema em estudo. Para o caso onde várias recompensas podem ser emitidas em cada estado da cadeia de Markov homogênea em tempo contínuo \mathcal{X} , as condições de contorno são mais complexas do que as apresentadas no Capítulo 3, que são relacionadas com o caso onde somente uma recompensa é atribuída a cada estado. Um determinado estado oculto pode emitir recompensas positivas e/ou negativas, não sendo possível classificá-lo como um estado *filling*, onde somente recompensas positivas podem ser emitidas ou como um estado *emptying*, onde somente recompensas negativas são emitidas.

Iremos considerar modelos HMM, onde cada um dos estados ocultos emite somente recompensas positivas OU somente recompensas negativas. Esta consideração não limita

⁴Esta equivalência será importante para a aplicação do algoritmo iterativo proposto.

a aplicação dos resultados aqui apresentados, dado que ao definirmos a estrutura do modelo de Markov oculto podemos optar por uma estrutura em que seja considerada a separação dos valores das recompensas entre positivos e negativos [119]. Como consequência da especificação do HMM, onde recompensas positivas são agrupadas separadamente das recompensas negativas, podemos aplicar as condições de contorno para a definição das constantes de integração, apresentadas no Capítulo 3 e previamente definidas em [5].

5.4 Algoritmo Iterativo para Obtenção da Solução Aproximada da Equação 5.26

A equação diferencial 5.26 não possui solução fechada. A equação final que representa o HMM é composta por vetores que descrevem o comportamento do processo oculto \mathcal{X} (vetor $\mathbf{w}(y)$) e o comportamento do processo de observação de recompensas \mathcal{Y} (vetor $\mathbf{w}^l(y)$).

Segundo o resultado do Lema 5.1, que mostra a equivalência entre os processos \mathcal{X} e o processo \mathcal{W} , poderíamos utilizar o processo expandido \mathcal{W} e resolvermos o modelo onde cada estado possui uma única recompensa usando a metodologia apresentada em [5]. No entanto, nos casos onde as cardinalidades de \mathcal{S} e \mathcal{R} são elevadas, esta abordagem não é atraente, dado o alto custo computacional da obtenção dos autovalores e autovetores e também da instabilidade numérica inerente ao cálculo destes. A abordagem apresentada no Capítulo 3, utilizando processos QBD, também é uma alternativa para solução da medida de interesse em estudo. No entanto, o fato de expandirmos o processo de Markov oculto pode resultar em um grande espaço de estados, aumentando consideravelmente o custo computacional do método proposto no Capítulo 3. No que se segue, iremos propor um método de solução aproximado que resulte em diminuição do custo computacional para obter a medida de interesse $\lim_{t \rightarrow \infty} P[CR(t) \leq y, X(t) = j]$.

Nesta seção apresentamos o algoritmo iterativo proposto para obter a solução da medida de interesse supracitada. O algoritmo fornece um resultado aproximado para a solução da Equação 5.26. Em linhas gerais, o sistema de equações 5.26 será solucionado em três etapas diferentes, onde em cada uma delas transformamos termos variáveis em constantes calculadas em iterações anteriores. Desta forma, podemos escrever a solução da referida equação através de fórmulas fechadas para sistemas de equações diferenciais. A cada passo do algoritmo iterativo, transformaremos o sistema de equações diferenciais

5.4 Algoritmo Iterativo para Obtenção da Solução Aproximada da Equação 5.26

original em um sistema de equações lineares não-homogêneo [14], com solução conhecida.

A principal vantagem do algoritmo que propomos refere-se ao fato de reduzirmos a ordem da matriz da qual devemos calcular os autovalores e autovetores. Adicionalmente, reduzimos a dimensão do sistema de equações que deve ser resolvido para a obtenção das constantes de integração que compõe o resultado final, diminuindo prováveis ocorrências de erros numéricos. Na Seção 5.5 mostraremos um caso onde a solução do processo expandido \mathcal{W} , através do método apresentado em [5] resulta em erros numéricos que violam as condições de contorno inerentes ao problema.

5.4.1 Notação

Como definido no Capítulo 4, a notação em negrito é utilizada para matrizes, em letra maiúscula, como por exemplo Γ , e para vetores, em letra minúscula, como por exemplo, γ . O elemento de um vetor é denotado por γ_i . Para cada recompensa distinta do modelo em estudo, indexaremos vetores e recompensas com a letra l , como por exemplo, γ^l e Γ^l . O valor dos elementos de um vetor, em uma uma iteração particular do algoritmo é denotado, por exemplo, por $\alpha^{(k)}$, onde k representa a k -ésima iteração. A Tabela 5.1 apresenta a notação utilizada pelo algoritmo iterativo proposto.

5.4.2 O Algoritmo

Para apresentação do algoritmo e de seu custo computacional, primeiramente abordaremos, em três subseções distintas, as três etapas principais que o compõe: a definição da solução inicial, a determinação do vetor $\mathbf{w}^l(y)$ e a determinação do vetor $\mathbf{w}(y)$. Em seguida, apresentamos o algoritmo propriamente dito e o seu custo computacional. Finalizando, discutiremos questões referentes a sua acurácia.

5.4.3 Solução Inicial

Primeiramente, como em todo o algoritmo iterativo, devemos estabelecer uma solução, com a qual o algoritmo iniciará o processo de iterações sucessivas. Para diversos métodos iterativos encontrados na literatura, a solução inicial pode definir a taxa de convergência e a acurácia do algoritmo.

5.4 Algoritmo Iterativo para Obtenção da Solução Aproximada da Equação 5.26

Tabela 5.1: Descrição das matrizes e vetores utilizados na apresentação do algoritmo iterativo proposto.

Termo	Descrição
k	k -ésima iteração do algoritmo
r^l	Valor da l -ésima recompensa emitida
γ_j^l	Probabilidade de emitir a l -ésima recompensa no j -ésimo estado do processo \mathcal{X}
$\boldsymbol{\gamma}^l$	Vetor coluna onde o j -ésimo elemento é igual a γ_j^l
\mathbf{R}^l	Matriz diagonal onde o j -ésimo elemento é igual a $\frac{\gamma_j^l}{r_l}$
$\mathbf{w}^l(y)$	Vetor coluna onde o j -ésimo elemento é igual a $P[CR \leq y, X = j, Y = l]$
$\mathbf{w}(y)$	Vetor coluna onde o j -ésimo elemento é igual a $P[CR \leq y, X = j]$
$\mathbf{\Gamma}^l$	Matriz diagonal cujo os elementos diagonais são iguais aos elementos do vetor $\boldsymbol{\gamma}^l$
$\boldsymbol{\varphi}_j^l$	j -ésimo autovetor da matriz diagonal \mathbf{R}^l
$\alpha_j^l(y)$	Constante de integração do j -ésimo estado do processo \mathcal{W} , para l -ésima recompensa
\mathbf{R}	Matriz diagonal cujo o j -ésimo elemento é dado por $\sum_{\forall l} \left(\frac{\gamma_j^l}{r_l} \right)$
\mathbf{V}	Matriz com autovetores colunas da matriz \mathbf{RQ}^T
$\boldsymbol{\phi}_j$	j -ésimo autovetor da matriz \mathbf{RQ}^T
$\boldsymbol{\rho}_j(y)$	Vetor coluna com elementos iguais a constante de integração $a_j(y)$
λ_j	j -ésimo autovalor da matriz \mathbf{RQ}^T
$\boldsymbol{q}_j(y)$	Vetor coluna com os elementos iguais a $a_j^{(k)}(y)e^{\lambda_j y}$
$\boldsymbol{\Psi}_j$	Matriz cujas linhas são iguais ao vetor $\mathbf{V}_{i_*}^{(-1)}$ (i -ésima linha da matriz $\mathbf{V}^{(-1)}$)
$\boldsymbol{\Upsilon}_j$	Matriz cujas linhas são iguais ao vetor $(\frac{1}{\lambda_i})\mathbf{V}_{i_*}^{(-1)}$ (i -ésima linha da matriz $\mathbf{V}^{(-1)}$)
$\boldsymbol{\Phi}_i$	Matriz cujas as colunas são iguais ao vetor $\boldsymbol{\phi}_i$
$\boldsymbol{\Omega}^l$	Matriz diagonal com o j -ésimo elemento igual a $\frac{\gamma_j^l}{\lambda}$
\mathbf{D}_i	Matriz diagonal cujo os elementos diagonais são iguais aos elementos do vetor $\boldsymbol{\phi}_i$
$\boldsymbol{\delta} \bullet \boldsymbol{\zeta}$	Multiplicação elemento a elemento entre os vetores $\boldsymbol{\delta}$ e $\boldsymbol{\zeta}$

5.4 Algoritmo Iterativo para Obtenção da Solução Aproximada da Equação 5.26

Relembremos o sistema de equações que representa o processo HMM em estudo:

$$\frac{d\mathbf{w}(y)}{dy} = \mathbf{RQ}^T \mathbf{w}(y) + \Lambda \mathbf{Rw}(y) - \Lambda \left\{ \sum_{\forall l} \left(\frac{\mathbf{w}^l(y)}{r^l} \right) \right\}_j, \quad (5.28)$$

com $\mathbf{w}(y) = \sum_{\forall l} \mathbf{w}^{(l)}(y)$.

Analisando este sistema, podemos observar que o subsistema formado por:

$$\frac{d\mathbf{w}(y)}{dy} = \mathbf{RQ}^T \mathbf{w}(y) \quad (5.29)$$

possui solução da forma $\mathbf{w}(y) = e^{\mathbf{RQ}^T y}$, que pode ser escrita através da seguinte expansão espectral:

$$\mathbf{w}(y) = \sum_{\forall i \in \mathcal{S}} a_i \phi_i e^{\lambda_i y},$$

onde λ_i é o i -ésimo autovalor da matriz \mathbf{RQ}^T com o seu autovetor ϕ_i correspondente.

Ao resolvermos o Sistema 5.29, consideramos o termo:

$$\Lambda \mathbf{Rw}(y) - \Lambda \left\{ \sum_{\forall l} \left(\frac{\mathbf{w}^l(y)}{r^l} \right) \right\}_j \quad (5.30)$$

igual a zero. A seguinte interpretação probabilística pode ser feita em relação a esta aproximação que é utilizada como a solução inicial para o algoritmo iterativo: assumimos independência entre a variável aleatória que descreve o nível de recompensa acumulada e a variável aleatória que indica a recompensa emitida pelo estado final j do processo oculto \mathcal{X} . Em outras palavras, consideramos $\mathbf{w}^l(y) = \mathbf{w}(y) \bullet \gamma^l$.

A principal vantagem desta primeira aproximação é que a dimensão das matrizes envolvidas na obtenção destes resultados é da ordem de $|\mathcal{S}| \times |\mathcal{S}|$, bem como a dimensão do sistema linear para a determinação das constantes de integração a_i obtidas através das condições de contorno previamente discutidas no Capítulo 3.

Conforme veremos na Seção 5.5, a escolha de uma solução inicial razoável que descreve o comportamento do processo oculto \mathcal{X} , é fundamental para obtenção de resultados próximos aos resultados exatos do modelo de Markov oculto. Na seção 5.5 veremos, através de exemplos que, assumir independência entre as variáveis aleatórias listadas anteriormente, resulta em boa acurácia da aproximação inicial.

5.4.4 Definindo um valor aproximado para o vetor $\mathbf{w}^l(y)$

A próxima etapa do algoritmo iterativo proposto consiste em obter aproximações sucessivas para o valor do vetor $\mathbf{w}^l(y)$, onde o j -ésimo elemento representa $\lim_{t \rightarrow \infty} P[CR(t) \leq$

$y, X(t) = j, Y(t) = l]$.

Conforme mostrado na Seção 5.3, o seguinte sistema de equações diferenciais :

$$\frac{d\mathbf{w}^l(y)}{dy} = \mathbf{R}^l \mathbf{Q}^T \mathbf{w}(y) + \frac{\Lambda}{r^l} [(\mathbf{w}(y) \bullet \boldsymbol{\gamma}^l) - \mathbf{w}^l(y)]. \quad (5.31)$$

deve ser resolvido para cada uma das $|\mathcal{R}|$ recompensas distintas que podem ser atribuídas ao processo de observação \mathcal{Y} . Analisando a Equação 5.31, o conjunto de $|\mathcal{R}|$ sistemas lineares descrevem o comportamento do processo expandido \mathcal{W} , cuja a equivalência com o processo oculto \mathcal{X} é provada pelo Lema 5.1.

Seja o termo:

$$\mathbf{R}^l \mathbf{Q}^T \mathbf{w}(y) + \frac{\Lambda}{r^l} (\mathbf{w}(y) \bullet \boldsymbol{\gamma}^l)$$

uma função $\mathbf{g}(y)$, onde os elementos do vetor $\mathbf{w}(y)$ são conhecidos através do valor aproximado obtido na iteração anterior. A Equação 5.31 pode ser reescrita da seguinte forma:

$$\frac{d\mathbf{w}^l(y)}{dy} = -\frac{\Lambda}{r^l} \mathbf{w}^l(y) + \mathbf{g}(y) \quad (5.32)$$

com a seguinte solução associada:

$$\mathbf{w}^l(y) = e^{\left(\frac{-\Lambda}{r^l}\right)y} \sum_{\forall j \in \mathcal{S}} (\alpha_j^l \boldsymbol{\varphi}_j^l) + \left(\frac{r^l}{\Lambda}\right) [\mathbf{R}^l \mathbf{Q}^T \mathbf{w}(y)] + (\mathbf{w}(y) \bullet \boldsymbol{\gamma}^l) \quad \forall l, \quad (5.33)$$

onde as constantes de integração α_j^l são obtidas através da utilização das condições de contorno, apresentadas no Capítulo 3. Cada uma das equações do Sistema 5.32 é uma equação do tipo: $\frac{dy}{dt} = -ay + b$, com solução $y = Ce^{-a} + b/a$.

5.4.5 Definindo um valor aproximado para o vetor $\mathbf{w}(y)$

Antes de prosseguirmos com o detalhamento do algoritmo iterativo proposto, seja o seguinte sistema de equações diferenciais:

$$\frac{d\mathbf{x}(y)}{dy} = \mathbf{A}\mathbf{x}(y) + \mathbf{g}(y). \quad (5.34)$$

A solução deste sistema pode ser brevemente descrita da seguinte forma [14]. Seja \mathbf{T} a matriz cujas colunas são os autovetores $\boldsymbol{\xi}^{(1)}, \dots, \boldsymbol{\xi}^{(n)}$ da matriz \mathbf{A} , da ordem $n \times n$, e uma nova variável dependente $\boldsymbol{\zeta}(y)$ definida por:

$$\mathbf{x}(y) = \mathbf{T}\boldsymbol{\zeta}(y). \quad (5.35)$$

5.4 Algoritmo Iterativo para Obtenção da Solução Aproximada da Equação 5.26

Substituindo a Equação 5.35 em 5.34 e multiplicando por \mathbf{T}^{-1} , obtemos:

$$\frac{\zeta(y)}{dy} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\zeta(y) + \mathbf{T}^{-1}\mathbf{g}(y) = \mathbf{D}\zeta(y) + \mathbf{h}(y), \quad (5.36)$$

onde $\mathbf{h}(y) = \mathbf{T}^{-1}\mathbf{g}(y)$ e \mathbf{D} é a matriz diagonal cujos elementos diagonais são os autovalores r_1, \dots, r_n da matriz \mathbf{A} . A Equação 5.36 é um sistema de n equações independentes para $\zeta_1(y), \dots, \zeta_n(y)$; em conseqüência, as equações podem ser resolvidas separadamente. Em forma escalar, a Equação 5.36 fica:

$$\frac{\zeta_j(y)}{dy} = r_j\zeta_j(y) + h_j(y), \quad j = 1, \dots, n \quad (5.37)$$

onde $h_j(y)$ é uma determinada combinação linear de $g_1(y), \dots, g_n(y)$.

A Equação 5.37 é uma equação linear de primeira ordem, com a seguinte solução:

$$\zeta_j(y) = e^{\lambda_j y} \int_{y_0}^y e^{-\lambda_j s} h_j(s) ds + c_j e^{\lambda_j y}, \quad j = 1, \dots, n \quad (5.38)$$

onde c_j é uma constante de integração arbitrária. A solução $\mathbf{x}(y)$ da Equação 5.34 é obtida a partir da Equação 5.35.

Considere o sistema de equações lineares mostrado na Equação 5.28. Seja o seguinte termo:

$$\Lambda \mathbf{R} \sum_{\forall l} \mathbf{w}^l(y) - \Lambda \left\{ \sum_{\forall l} \left(\frac{\mathbf{w}^l(y)}{r^l} \right) \right\}_j,$$

uma função $f(y)$, onde o vetor $\mathbf{w}(y)$ é substituído por $\sum_{\forall l} \mathbf{w}^l(y)$ (conforme definição apresentada na Seção 5.3). O valor dos elementos do vetor $\mathbf{w}^l(y)$ são conhecidos em iterações anteriores do algoritmo proposto. Desta forma, podemos reescrever o sistema 5.28 da seguinte forma:

$$\frac{d\mathbf{w}(y)}{dy} = \mathbf{R}\mathbf{Q}^T \mathbf{w}(y) + \mathbf{f}(y). \quad (5.39)$$

A solução de cada uma das $|\mathcal{S}|$ equações independentes pertencentes ao sistema de equações 5.39 é dada pela Equação 5.38, com os autovalores λ_j referentes a matriz $\mathbf{R}\mathbf{Q}^T$, bem como os autovetores que formam as colunas da matriz \mathbf{T} .

A solução $\mathbf{w}(y)$, cujo o j -ésimo elemento é igual a $P[CR \leq y, X = j]$, após a

5.4 Algoritmo Iterativo para Obtenção da Solução Aproximada da Equação 5.26

substituição na Equação 5.39 do vetor $\mathbf{w}^l(y)$ pela Equação 5.33, é dada por:

$$\begin{aligned} \mathbf{w}(y) &= \phi_1 \bullet \\ &\left\{ \rho_1 + y \Psi_1 \left[\Lambda \mathbf{R} \mathbf{w}^{(k,l)}(y) - \Lambda \sum_{\forall l} \left(\sum_{\forall j} (\alpha_j^l \varphi_j^l) \frac{e^{\left(\frac{-\Lambda}{r^l} y\right)}}{r^l} + \frac{[\Omega^l \mathbf{Q}^T \mathbf{w}(y) + (\mathbf{w}(y) \bullet \gamma^l)]}{r^l} \right) \right] \right\} + \\ &\sum_{\forall i \neq 1} \left\{ \phi_i \bullet \left[\varrho_i - \Upsilon_j \left(\Lambda \mathbf{R} \mathbf{w}^{(k,l)}(y) - \Lambda \sum_{\forall l} \left(\sum_{\forall j} (\alpha_j^l \varphi_j^l) \frac{e^{\left(\frac{-\Lambda}{r^l} y\right)}}{r^l} + \frac{[\Omega^l \mathbf{Q}^T \mathbf{w}(y) + (\mathbf{w}(y) \bullet \gamma^l)]}{r^l} \right) \right) \right] \right\}. \end{aligned} \quad (5.40)$$

Com o objetivo de simplificação da Equação 5.40, seja a seguinte equação, obtida a partir da propriedade distributiva:

$$\begin{aligned} \mathbf{w}(y) &= (\phi_1 \bullet \rho_1) + \Lambda y \phi_1 \bullet \left\{ \Psi_1 \mathbf{R} \sum_{\forall l} \left(\sum_{\forall j} (\alpha_j^l \varphi_j^l) e^{\left(\frac{-\Lambda}{r^l} y\right)} + [\Omega^l \mathbf{Q}^T \mathbf{w}(y) + (\mathbf{w}(y) \bullet \gamma^l)] \right) \right\} \\ &- \Lambda y \phi_1 \bullet \left\{ \Psi_1 \left[\sum_{\forall l} \left(\left(\sum_{\forall j \in \mathcal{S}} (\alpha_j^l \varphi_j^l) \frac{e^{\left(\frac{-\Lambda}{r^l} y\right)}}{r^l} \right) + \frac{[\Omega^l \mathbf{Q}^T \mathbf{w}(y) + (\mathbf{w}(y) \bullet \Omega^l)]}{r^l} \right) \right] \right\} + \sum_{\forall i \neq 1} (\phi_i \bullet \varrho_i) \\ &- \Lambda \sum_{\forall i \neq 1} \phi_i \bullet \left\{ \Upsilon_i \mathbf{R} \sum_{\forall l} \left(\sum_{\forall j} (\alpha_j^l \varphi_j^l) e^{\left(\frac{-\Lambda}{r^l} y\right)} + [\Omega^l \mathbf{Q}^T \mathbf{w}(y) + (\mathbf{w}(y) \bullet \gamma^l)] \right) \right\} \\ &+ \Lambda \sum_{\forall i \neq 1} \phi_i \bullet \left\{ \Upsilon_i \left[\sum_{\forall l} \left(\left(\sum_{\forall j \in \mathcal{S}} (\alpha_j^l \varphi_j^l) \frac{e^{\left(\frac{-\Lambda}{r^l} y\right)}}{r^l} \right) + \frac{[\Omega^l \mathbf{Q}^T \mathbf{w}(y) + (\mathbf{w}(y) \bullet \gamma^l)]}{r^l} \right) \right] \right\}. \end{aligned} \quad (5.41)$$

Seja a matriz Φ_i conforme definida na Tabela 5.1. Podemos reescrever o lado esquerdo da equação 5.41 como:

$$\begin{aligned} \mathbf{w}(y) &= (\phi_1 \bullet \rho_1) \\ &+ \Lambda y \Phi_1 \bullet (\Psi_1 \mathbf{R}) \sum_{\forall l} \left(\sum_{\forall j} (\alpha_j^l \varphi_j^l) e^{\left(\frac{-\Lambda}{r^l} y\right)} + [\Omega^l \mathbf{Q}^T \mathbf{w}(y) + (\mathbf{w}(y) \bullet \gamma^l)] \right) \\ &- \Lambda y (\Phi_1 \bullet \Psi_1) \left\{ \sum_{\forall l} \left[\left(\sum_{\forall j \in \mathcal{S}} (\alpha_j^l \varphi_j^l) \frac{e^{\left(\frac{-\Lambda}{r^l} y\right)}}{r^l} \right) + \frac{[\Omega^l \mathbf{Q}^T \mathbf{w}(y) + (\mathbf{w}(y) \bullet \Omega^l)]}{r^l} \right] \right\} + \sum_{\forall i \neq 1} (\phi_i \bullet \varrho_i) \\ &- \Lambda \sum_{\forall i \neq 1} \Phi_i \bullet (\Upsilon_i \mathbf{R}) \sum_{\forall l} \left(\sum_{\forall j} (\alpha_j^l \varphi_j^l) e^{\left(\frac{-\Lambda}{r^l} y\right)} + [\Omega^l \mathbf{Q}^T \mathbf{w}(y) + (\mathbf{w}(y) \bullet \gamma^l)] \right) \\ &+ \Lambda \sum_{\forall i \neq 1} \left\{ (\Phi_i \bullet \Upsilon_i) \left[\sum_{\forall l} \left(\left(\sum_{\forall j \in \mathcal{S}} (\alpha_j^l \varphi_j^l) \frac{e^{\left(\frac{-\Lambda}{r^l} y\right)}}{r^l} \right) + \frac{[\Omega^l \mathbf{Q}^T \mathbf{w}(y) + (\mathbf{w}(y) \bullet \gamma^l)]}{r^l} \right) \right] \right\}. \end{aligned} \quad (5.42)$$

5.4 Algoritmo Iterativo para Obtenção da Solução Aproximada da Equação 5.26

Substituindo o produto elemento a elemento por produto matricial, através de matrizes diagonais D_i , temos:

$$\begin{aligned}
 \mathbf{w}(y) = & \mathbf{D}_1 \boldsymbol{\rho}_1 + \Lambda y \mathbf{D}_1 \Psi_1 \mathbf{R} \sum_{\forall l} \left(\sum_{\forall j} (\alpha_j^l \boldsymbol{\varphi}_j^l) e^{\left(\frac{-\Lambda}{r^l} y\right)} + [\boldsymbol{\Omega}^l \mathbf{Q}^T \mathbf{w}(y) + \boldsymbol{\Gamma}^l \mathbf{w}(y)] \right) \\
 & - \Lambda y (\mathbf{D}_1 \Psi_1) \left\{ \sum_{\forall l} \left[\left(\sum_{\forall j \in \mathcal{S}} (\alpha_j^l \boldsymbol{\varphi}_j^l) \frac{e^{\left(\frac{-\Lambda}{r^l} y\right)}}{r^l} \right) + \frac{[\boldsymbol{\Omega}^l \mathbf{Q}^T \mathbf{w}(y) - \boldsymbol{\Gamma}^l \mathbf{w}(y)]}{r^l} \right] \right\} + \sum_{\forall i \neq 1} \mathbf{D}_i \boldsymbol{\rho}_i \\
 & - \Lambda \sum_{\forall i \neq 1} (\mathbf{D}_i \Upsilon_i \mathbf{R}) \sum_{\forall l} \left(\sum_{\forall j} (\alpha_j^l \boldsymbol{\varphi}_j^l) e^{\left(\frac{-\Lambda}{r^l} y\right)} + [\boldsymbol{\Omega}^l \mathbf{Q}^T \mathbf{w}(y) + \boldsymbol{\Gamma}^l \mathbf{w}(y)] \right) \\
 & + \Lambda \sum_{\forall i \neq 1} (\mathbf{D}_i \Upsilon_i) \left\{ \sum_{\forall l} \left[\left(\sum_{\forall j \in \mathcal{S}} (\alpha_j^l \boldsymbol{\varphi}_j^l) \frac{e^{\left(\frac{-\Lambda}{r^l} y\right)}}{r^l} \right) + \frac{[\boldsymbol{\Omega}^l \mathbf{Q}^T \mathbf{w}(y) + \boldsymbol{\Gamma}^l \mathbf{w}(y)]}{r^l} \right] \right\}.
 \end{aligned} \tag{5.43}$$

Desta forma, a cada iteração do algoritmo proposto, obteremos a solução da Equação 5.43, sendo esta uma aproximação para a solução exata do Sistema de equações 5.28.

5.4.6 Visão Geral do Algoritmo Iterativo Proposto e o Custo Computacional

O seguinte algoritmo pode ser usado para obter, iterativamente, as soluções das Equações 5.33 e 5.43:

1. Calcule a solução inicial dada pelo sistema de equações:

$$\frac{d\mathbf{w}(y)}{dy} = \mathbf{RQ}^T \mathbf{w}(y)$$

com

$$\mathbf{w}^{(0)}(y) = \sum_{\forall i \in \mathcal{S}} a_i \boldsymbol{\phi}_i e^{\lambda_i y}$$

sendo λ_i o i -ésimo autovalor da matriz \mathbf{RQ}^T com o seu autovetor $\boldsymbol{\phi}_i$ correspondente;

2. Calcule as constantes de integração $\alpha_j^{(k,l)}(y)$ a partir da aplicação das condições de contorno nos $|\mathcal{R}|$ sistemas de equações lineares:

$$\mathbf{w}^{(k,l)}(y) = e^{\left(\frac{-\Lambda}{r^l} y\right)} \sum_{\forall j \in \mathcal{S}} (\alpha_j^l \boldsymbol{\varphi}_j^l) + \left(\frac{r^l}{\Lambda}\right) [\mathbf{R}^l \mathbf{Q}^T \mathbf{w}^{(k-1)}(y)] + (\mathbf{w}^{(k-1)}(y) \bullet \boldsymbol{\gamma}^l);$$

5.4 Algoritmo Iterativo para Obtenção da Solução Aproximada da Equação 5.26

3. Calcule os elementos dos vetores $\rho_j^{(k)}(y)$ e $\varrho_j^{(k)}(y)$, através da aplicação das condições de contorno na Equação 5.43;
4. Calcule $\mathbf{w}^{(k)}(y)$, a partir de $\mathbf{w}^{(k-1)}(y)$, usando a seguinte equação:

$$\begin{aligned}
 \mathbf{w}^{(k)}(y) = & \mathbf{D}_1 \boldsymbol{\rho}_1 + \Lambda y \mathbf{D}_1 \Psi_1 \mathbf{R} \sum_{\forall l} \left(\sum_{\forall j} (\alpha_j^l \boldsymbol{\varphi}_j^l) e^{\left(\frac{-\Lambda}{r^l} y\right)} + [\boldsymbol{\Omega}^l \mathbf{Q}^T \mathbf{w}^{(k-1)}(y) + \boldsymbol{\Gamma}^l \mathbf{w}^{(k-1)}(y)] \right) \\
 & - \Lambda y (\mathbf{D}_1 \Psi_1) \left\{ \sum_{\forall l} \left[\left(\sum_{\forall j \in \mathcal{S}} (\alpha_j^l \boldsymbol{\varphi}_j^l) \frac{e^{\left(\frac{-\Lambda}{r^l} y\right)}}{r^l} \right) + \frac{[\boldsymbol{\Omega}^l \mathbf{Q}^T \mathbf{w}^{(k-1)}(y) - \boldsymbol{\Gamma}^l \mathbf{w}^{(k-1)}(y)]}{r^l} \right] \right\} + \sum_{\forall i \neq 1} \mathbf{D}_i \boldsymbol{\varrho}_i \\
 & - \Lambda \sum_{\forall i \neq 1} (\mathbf{D}_i \Upsilon_i \mathbf{R}) \sum_{\forall l} \left(\sum_{\forall j} (\alpha_j^l \boldsymbol{\varphi}_j^l) e^{\left(\frac{-\Lambda}{r^l} y\right)} + [\boldsymbol{\Omega}^l \mathbf{Q}^T \mathbf{w}^{(k-1)}(y) + \boldsymbol{\Gamma}^l \mathbf{w}^{(k-1)}(y)] \right) \\
 & + \Lambda \sum_{\forall i \neq 1} (\mathbf{D}_i \Upsilon_i) \left\{ \sum_{\forall l} \left[\left(\sum_{\forall j \in \mathcal{S}} (\alpha_j^l \boldsymbol{\varphi}_j^l) \frac{e^{\left(\frac{-\Lambda}{r^l} y\right)}}{r^l} \right) + \frac{[\boldsymbol{\Omega}^l \mathbf{Q}^T \mathbf{w}^{(k-1)}(y) + \boldsymbol{\Gamma}^l \mathbf{w}^{(k-1)}(y)]}{r^l} \right] \right\};
 \end{aligned} \tag{5.44}$$

5. Retorne ao passo 2, caso o critério de parada não seja alcançado.

Em relação ao algoritmo proposto, a primeira questão a ser detalhada refere-se ao critério de parada utilizado. Diversos paradigmas podem ser considerados, como número total de iterações realizadas, erro absoluto entre iterações, entre outras [132]. Por exemplo, considerar o critério do número total de iterações pode interromper o algoritmo de maneira prematura, gerando resultados distantes do real, principalmente quando a taxa de convergência do algoritmo iterativo é muito lenta. Para a versão do algoritmo proposto neste trabalho, consideramos como critério de parada a análise do máximo erro absoluto entre duas iterações consecutivas.

A segunda questão que deve ser abordada quando algoritmos iterativos são propostos, refere-se a convergência. Por exemplo, os métodos iterativos Gauss-Seidel, Jacobi, Power, SOR entre outros, convergem para o vetor de distribuição de probabilidades em estado estacionário $\boldsymbol{\pi}$ sob certas condições estabelecidas. Vale ressaltar que a convergência de um método iterativo não significa que este produza a resposta exata do sistema em estudo, mas, simplesmente que este não oscila entre soluções distintas, ou, adicionalmente, que a solução não muda de valor indefinidamente.

A análise de convergência do algoritmo proposto foi realizada de forma empírica. Em todos os 96 testes feitos, o algoritmo convergiu, para uma solução *aproximada* da solução

5.4 Algoritmo Iterativo para Obtenção da Solução Aproximada da Equação 5.26

exata do modelo. Diferentemente do capítulo 4, não estabelecemos limites superior e inferior para o resultado aproximado.

Ainda sobre a análise de convergência, em menos de 10% dos casos, a solução aproximada não resultou em uma distribuição de probabilidades, devido a erros numéricos na determinação das condições de contorno. Por exemplo, a alguns níveis de recompensa foram associados resultados negativos, ou em alguns casos as funções geradas não eram monotônicas crescentes em função de y , propriedade da função distribuição de probabilidade acumulada. A característica em comum em todos estes casos era que as recompensas emitidas em um determinado estado diferiam em valores onde a maior recompensa era uma ordem de magnitude superior que a menor recompensa. Para alguns casos, definimos novos estados para agrupar recompensas da mesma ordem de grandeza. No entanto, maiores investigações devem ser feitas para analisar a relação entre os valores atribuídos às recompensas e os problemas numéricos ocorridos.

Acoplada a questão da convergência do método iterativo, está a questão da rapidez de convergência. Métodos tradicionais como os citados nos parágrafos anteriores possuem taxa de convergência proporcional ao módulo do autovalor subdominante da matriz de transição de probabilidades \mathbf{P} . Quanto mais próximo este valor é de 1, que é o raio espectral da matriz \mathbf{P} , mais lentamente o algoritmo converge para o resultado final.

Conforme mostraremos na Seção 5.5, estudos empíricos mostraram uma forte correlação entre o total de iterações realizadas pelo algoritmo proposto e a maior ordem dos autovalores do modelo em estudo. A identificação de condições para a convergência ou não do algoritmo, bem como a rapidez desta convergência são tópicos para pesquisas futuras.

O último tópico a ser abordado é o custo computacional, em total de multiplicações, do algoritmo proposto. Para tal, iremos apresentar o custo para cada uma das etapas do algoritmo descrito no início desta seção. Iremos considerar que o custo computacional para o cálculo dos autovalores e autovetores é de $10M^3$ multiplicações, que a solução de sistemas lineares é da ordem de M^3 multiplicações, sendo M a dimensão das matrizes envolvidas. Assim:

1. Para o cálculo da solução inicial, devemos definir os autovalores e autovetores de uma matriz da ordem $|\mathcal{S}|$, com um custo total de $10|\mathcal{S}|^3$ multiplicações;
2. Para o cálculo das constantes de integração $\alpha_j^{(k,l)}(y)$ são necessárias $c(|\mathcal{S}|^2 + |\mathcal{S}|) + |\mathcal{R}||\mathcal{S}|^2$ multiplicações, sendo c o total de pontos dos quais a medida de interesse

está sendo calculada;

3. Para o cálculo dos elementos dos vetores ρ_j e q_j , deve-se resolver c sistemas lineares, com total de multiplicações da ordem de $|\mathcal{S}|^3$.
4. Para obter a solução da Equação 5.43, que também é utilizada para obter as constantes de integração na etapa 3, são necessárias $c(2|\mathcal{S}|^3 + |\mathcal{S}|^2 + |\mathcal{R}||\mathcal{S}|)$ multiplicações. O fato de manipulação de matrizes diagonais e vetores, reduz a complexidade computacional de maneira significativa.

Assim, a equação de custo para o algoritmo iterativo é de:

$$T = I(10|\mathcal{S}|^3 + c(|\mathcal{S}|^2 + |\mathcal{S}|) + |\mathcal{R}||\mathcal{S}|^2 + c(2|\mathcal{S}|^3 + |\mathcal{S}|^2 + |\mathcal{R}||\mathcal{S}|)),$$

com I denotando o número total de iterações até o algoritmo atingir o critério de parada utilizado.

5.5 Exemplos

Nesta seção apresentaremos alguns exemplos com o objetivo de avaliar questões importantes inerentes ao algoritmo iterativo proposto: a sua acurácia, dado que a solução retornada pelo algoritmo é uma aproximação da solução real; estudo empírico da taxa de convergência, em número total de iterações; e custo computacional para a obtenção da solução. Os exemplos apresentados serão divididos em 3 classes distintas. Em todos os gráficos, estamos considerando a medida $P[CR \leq B^-]$, não representando o caso em que $CR = B$ e recompensas positivas sendo emitidas. A probabilidade da recompensa $CR = B$ quando são emitidas recompensas positivas foi apresentada na Seção 3.2.1 do Capítulo 3.

Para a análise da acurácia do método, usaremos a métrica de erro absoluto e erro relativo. Seja $P[CR \leq x]$ a solução exata da medida de interesse, $\hat{P}[CR \leq x]$ a solução aproximada dada pelo algoritmo iterativo e $\hat{P}_I[CR \leq x]$ a solução inicial. O erro absoluto da solução inicial é dado por $|P[CR \leq x] - \hat{P}_I[CR \leq x]|$ e o erro da solução aproximada final do algoritmo é dado por $|P[CR \leq x] - \hat{P}[CR \leq x]|$. O erro absoluto máximo refere-se ao maior erro absoluto quando consideramos todos os níveis de recompensa, para os quais a medida foi calculada, entre o limite inferior e o limite superior para a variável aleatória CR . Similarmente, o erro relativo é dado por $|P[CR \leq x] - \hat{P}_I[CR \leq x]|/P[CR \leq x]$ para a solução inicial e $|P[CR \leq x] - \hat{P}[CR \leq x]|/P[CR \leq x]$ para

a solução aproximada. O máximo erro relativo é definido similarmente a definição do máximo erro absoluto.

Na primeira classe de exemplos, mostraremos a aplicação do algoritmo proposto para obtenção da medida de interesse $P[CR \leq y] = \sum_{\forall i \in \mathcal{S}} P[CR \leq y, X = i]$, em uma cadeia simples com dois estados ocultos, representados pela variável aleatória X e três recompensas distintas $\{r_1, r_2, r_3\}$, representadas pela variável aleatória Y . O principal objetivo desta classe de exemplos é mostrar o comportamento do erro absoluto com a variação da distribuição de probabilidade da emissão das recompensas em um determinado estado oculto, bem como com a modificação dos valores das recompensas sendo emitidas.

Na segunda classe de exemplos, abordaremos a influência da mudança de carga no sistema⁵ no resultado aproximado obtido através do algoritmo proposto. Consideraremos dois tipos distintos de modelos HMM: no primeiro conjunto de cenários, o HMM possui 2 estados ocultos, sendo 3 recompensas emitidas. No segundo conjunto de cenários, ampliamos a cardinalidade de estados ocultos para 3 e de recompensas emitidas para 9.

Na terceira classe, apresentaremos um exemplo cujas as recompensas atribuídas ao processo de observação \mathcal{Y} pertencem ao conjunto de valores utilizados para a modelagem do canal de comunicação do Departamento de Engenharia de Sistemas e Computação COPPE/UFRJ [37]. O total de recompensas distintas emitidas é de $|\mathcal{R}| = 15$. Para este caso, a solução da medida de interesse escrita através da expansão espectral de uma matriz de dimensão 15, conforme apresentada em [5], resulta em erro numérico, não sendo possível obter o resultado exato através desta metodologia. O resultado exato foi calculado com o método apresentado em [88], considerando diversos tempos de observação até que o sistema estivesse em equilíbrio. Através do algoritmo proposto, a solução aproximada, com boa acurácia, foi obtida sem erros numéricos.

Finalizando, discutiremos, brevemente, alguns resultados em relação a acurácia e a taxa de convergência do algoritmo proposto. Estes resultados, apesar de empíricos, fornecem informações importantes sobre o comportamento do algoritmo proposto, e, principalmente, direções para que resultados analíticos possam ser posteriormente definidos.

⁵Se a taxa média de chegada de informações é de λ e a capacidade de serviço é de μ , temos que a carga no sistema é de $\rho = \frac{\lambda}{\mu}$.

5.5.1 Classe 1: Dois estados ocultos

Seja o seguinte modelo de Markov oculto a ser estudado. O processo oculto \mathcal{X} pode se encontrar em um de dois estados diferentes, A e B , com taxas de transição de $A \rightarrow B$ igual a 5 e de $B \rightarrow A$ igual a 10. No estado oculto A , o processo de observação \mathcal{Y} emite uma única recompensa igual a -50000 , enquanto no estado B a recompensa r_2 é emitida com probabilidade p_2 e a recompensa r_3 é emitida com probabilidade p_3 . A Figura 5.3 mostra o HMM utilizado nesta classe de exemplos.

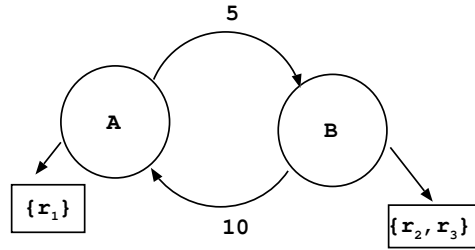


Figura 5.3: HMM utilizado para o primeiro conjunto de exemplos. O processo possui dois estados ocultos e três recompensas distintas.

Primeiramente, seja o caso onde a recompensa r_3 varia dentro do seguinte intervalo $[r_2, 2r_2, 3r_2, \dots, 15r_2]$, permanecendo constante as probabilidades p_2 e p_3 . O principal objetivo em estudarmos este cenário é mostrar o impacto de consideramos o termo:

$$\Lambda R w(y) - \Lambda \left\{ \sum_{\forall l} \left(\frac{w^l(y)}{r^l} \right) \right\}_j = 0,$$

ou, em termos probabilísticos, independência entre o valor da recompensa sendo emitida e o seu impacto no nível de recompensa acumulada.

A Figura 5.4 mostra o resultado obtido, quando variamos a recompensa r_3 e $r_2 = 1000$, $p_2 = p_3 = 0.5$ com tamanho da fila de fluido igual a 1000, para o comportamento do erro absoluto máximo entre a solução inicial do algoritmo iterativo e a solução exata, sendo a última calculada com o algoritmo apresentado em [5]. Conforme podemos observar, quanto maior a diferença entre as recompensas emitidas no estado oculto B , maior o erro absoluto máximo. Este resultado confirma a dependência que existe entre o valor da recompensa acumulada e a recompensa sendo emitida: quanto maior a diferença entre as recompensas pertencentes a um estado oculto, maior proporção da medida $w_j(y)$ é atribuída ao resultado $w_j^l(y)$, com l relacionado com as recompensas de menores valores. Os demais exemplos, confirmam esta característica para a solução inicial usada pelo algoritmo.

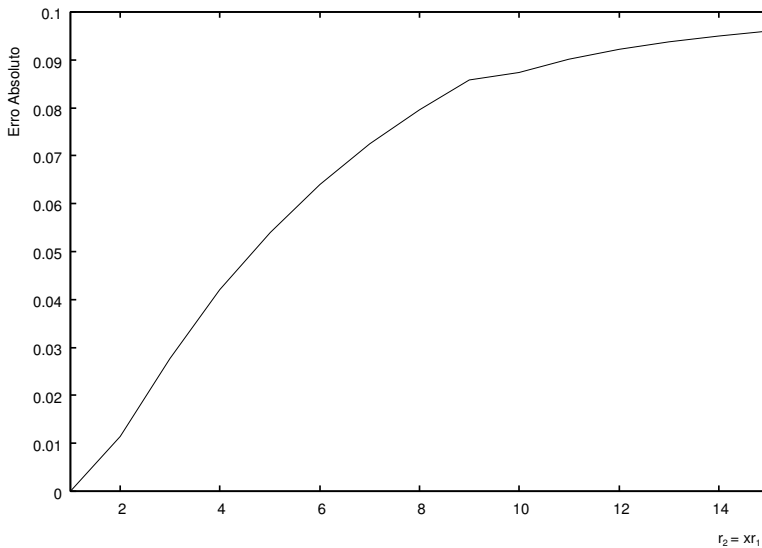


Figura 5.4: Comportamento do erro absoluto quando a diferença entre os valores de recompensas atribuídas a um estado oculto varia dentro de um determinado intervalo.

Consideremos o mesmo processo HMM, com $r_1 = -50000$, $r_2 = 1000$, $r_3 = 2000$, $p_1 = 0.5$, $p_2 = 0.5$ e tamanho de fila $B = 1000$. A Figura 5.5 mostra o resultado obtido pelo algoritmo iterativo proposto. O erro absoluto máximo entre a solução inicial e a solução exata é de $1.1493e-2$ e é igual a $3.6418e-3$ para a solução aproximada. Para o erro relativo máximo, temos $1.2234e-2$ para a solução inicial e $4.3014e-3$ para a solução aproximada. Ou seja, a aplicação do algoritmo iterativo proposto resulta em uma diminuição de aproximadamente uma ordem de grandeza no erro absoluto máximo e no erro relativo máximo. Adicionalmente, analisamos o impacto da modificação da distribuição de probabilidades na métrica de desempenho *erro absoluto máximo*, quando $p_3 \geq p_2$, com $p_3 = \{0.9, 0.8, 0.7, 0.6, 0.5\}$. Observamos, conforme esperado, que o erro absoluto máximo variou no intervalo $[1.0317e-2, 1.1569e-2]$. O erro absoluto entre a solução exata e a solução aproximada fornecida pelo algoritmo, se encontram no intervalo $[3.6418e-3, 3.4764e-2]$. Para o caso onde atribuímos $p_3 = 0.9$, o valor máximo do erro absoluto foi maior do que o encontrado para a solução inicial. Este comportamento de aumento do erro, ocorreu em poucos casos entre os 96 testes realizados. No entanto, mesmo para os casos de aumento do erro absoluto máximo da solução final quando comparado ao erro absoluto máximo da solução inicial, estes não ultrapassam 3% da solução exata. As propriedades do algoritmo em termos do erro de solução e convergência ainda estão sendo estudadas.

Passemos ao caso onde $r_2 = 1000$ e $r_3 = 6000$, sendo mantidos os valores dos demais parâmetros. A Figura 5.6 mostra o resultado obtido, após a aplicação do método proposto.

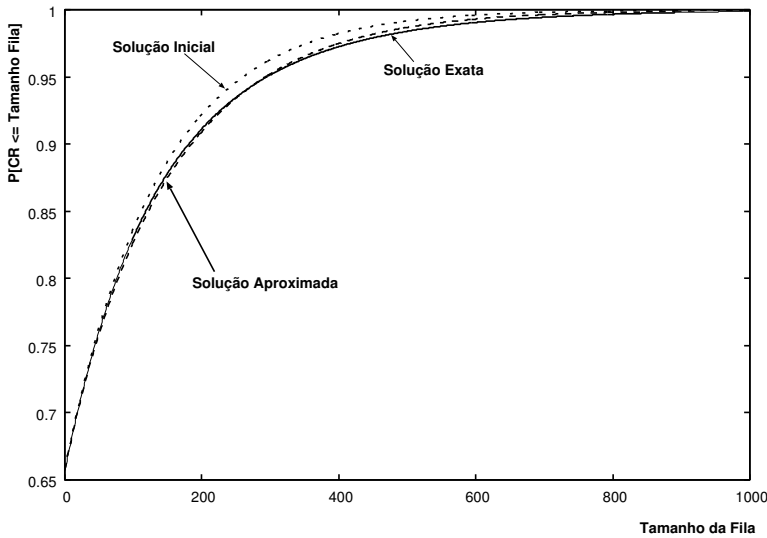


Figura 5.5: Distribuição da recompensa acumulada em uma fila de tamanho $B = 1000$ e $r_2 \approx r_3$.

O erro absoluto máximo da solução inicial, em relação a solução exata é de $6.4016e-2$ e é igual a $1.3655e-2$ para a solução aproximada. O erro relativo máximo é igual a $7.1098e-2$ para a solução inicial, e $1.6543e-2$ para a solução aproximada. Dado que a recompensa r_3 é seis vezes maior que a recompensa r_2 , o erro máximo da solução inicial é maior do que o encontrado na caso de estudo anterior. Diferentemente do caso anterior, a variação da distribuição de probabilidades da emissão das recompensas r_2 e r_3 resulta em um intervalo mais largo de erro relativo, sendo este igual a $[3.9716e-2, 6.8623e-2]$. Para a solução aproximada, estes erros se encontram no intervalo $[1.3655e-2, 6.8046e-2]$.

Para os casos abordados nesta primeira classe de exemplos, o total de iterações para que a mudança do valor da medida de interesse, entre iterações sucessivas, seja menor que $1e-8$ foi de 2 iterações. Os autovalores são da ordem $1e-3$ e $1e-4$. Baseado em testes empíricos, conforme mostraremos ao final desta seção de exemplos, o número total de iterações necessárias para atingir o critério de parada estipulado está fortemente relacionado com a maior ordem dos autovalores envolvidos no cálculo da medida de interesse⁶.

⁶Os autovalores são relacionados com os parâmetros do modelo. A definição prévia destes não é uma tarefa trivial, conforme discutido na literatura ([87] e referências).

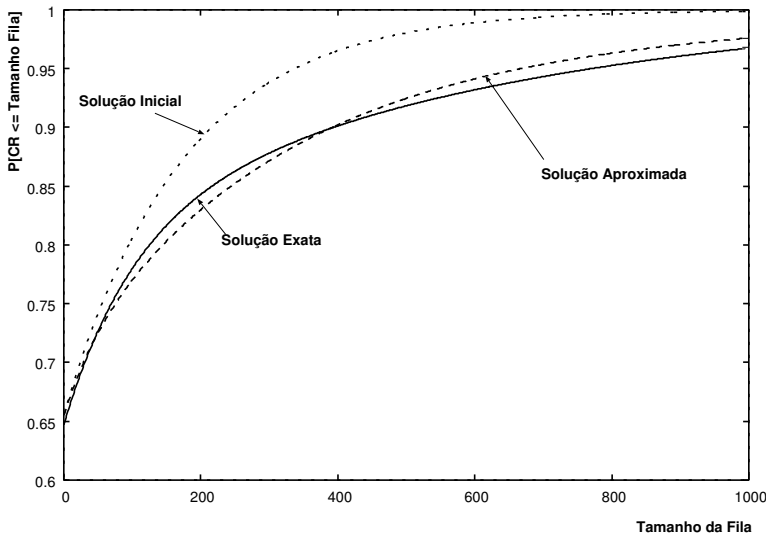


Figura 5.6: Distribuição da recompensa acumulada em uma fila de tamanho $B = 1000$ e $r_2 \ll r_3$.

5.5.2 Classe 2: Variando a carga do sistema

Nesta classe de exemplos consideraremos o estudo da influência do desempenho do algoritmo proposto quando a carga do sistema é variada. Esta variação é realizada através do parâmetro μ , que representa a capacidade de serviço destinada a fila de fluido.

Primeiramente, seja o HMM, com dois estados ocultos A e B , com taxa de chegada de informações em A igual a $\delta_1 = 0$ e taxas de chegada em B , $\delta_2 = 10000$ e $\delta_3 = 15000$, com probabilidades $\{p_2 = 0.2, p_3 = 0.8\}$ (como mostrado na Figura 5.7). A taxa de transição de $A \rightarrow B$ é igual a 5 e de $B \rightarrow A$ igual a 10. Desta forma, a taxa média de chegada de informações na fila de fluido com tamanho $B = 500$ é de $\delta_{\text{méd}} = 4666$ unidades de informação e $\mu = 4713$. A Figura 5.8 apresenta a solução aproximada para carga de $\rho = 0.99$. O seguinte conjunto de recompensas foi atribuído ao processo de observações \mathcal{Y} : $\{r_1 = -4713, r_2 = 5287, r_3 = 10287\}$.

O erro absoluto máximo para a solução inicial usada pelo algoritmo aproximativo foi de $2.4266e-3$, sendo esta mesma métrica, quando utilizada a solução aproximada, igual a $4.62876e-3$. Neste cenário a solução inicial já está próxima ao resultado exato. Note que a recompensa r_3 emitida no estado oculto B , é aproximadamente apenas 2 vezes maior que a recompensa r_2 emitida neste mesmo estado. O total de iterações necessárias para atingir o critério de parada utilizado foi de 3, para valor máximo de autovalores envolvidos na ordem de $1e-2$.

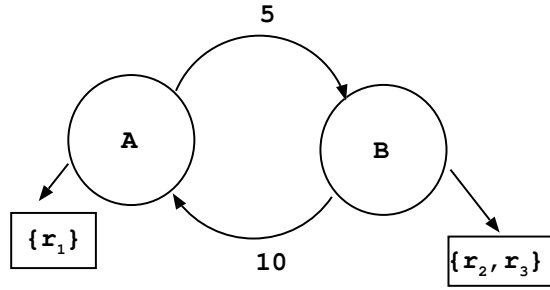


Figura 5.7: HMM utilizado para o segundo conjunto de exemplos. O processo possui dois estados ocultos três recompensas distintas.

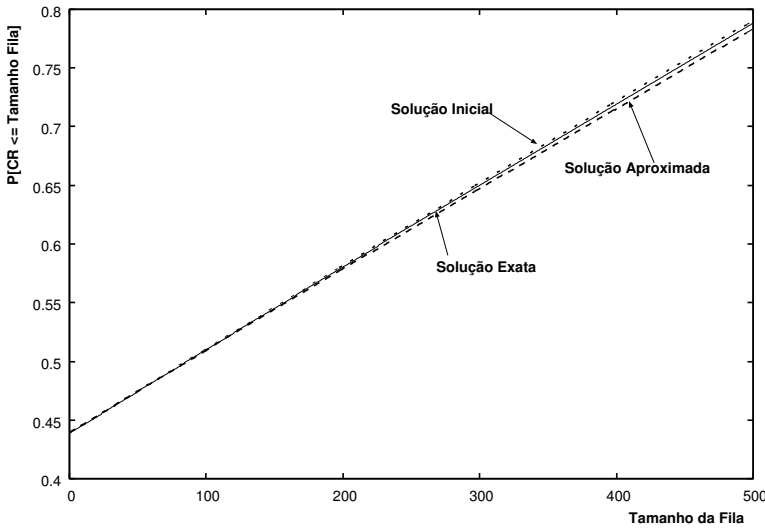


Figura 5.8: Resultados obtidos pelo algoritmo iterativo para um modelo de Markov oculto com dois estados ocultos e carga no sistema de $\rho = 0.99$ e $B = 500$.

Consideremos uma carga no sistema igual a $\rho = 0.7$. Com as mesmas taxas de transmissão descritas anteriormente, a capacidade do canal que serve as informações enviadas a fila de fluido é de $\mu = 6665$. As seguintes recompensas são emitidas pelos estados ocultos: $\{r_1 = -6665, r_2 = 3335, r_3 = 10334\}$. A Figura 5.9 mostra os resultados obtidos para este caso em estudo. O erro absoluto máximo para a medida de interesse, considerando a solução inicial foi de $1.0425e-2$. Quando comparado com a solução aproximada decorrente da aplicação do algoritmo proposto, o erro foi de $1.5031e-2$, com um total de 3 iterações executadas.

Seja o mesmo HMM descrito anteriormente, para um novo tamanho de fila $B = 1000$, carga no sistema de $\rho = 0.7$ e $\mu = 5951$. As seguintes recompensas são atribuídas ao processo de observação \mathcal{Y} : $\{r_1 = -5951, r_2 = 4049, r_3 = 9049\}$, com probabilidades $\{p_1 = 1, p_2 = 0.5, p_3 = 0.5\}$, com $\delta_{\text{méd}} = 4166$. A Figura 5.10 apresenta os re-

5.5 Exemplos

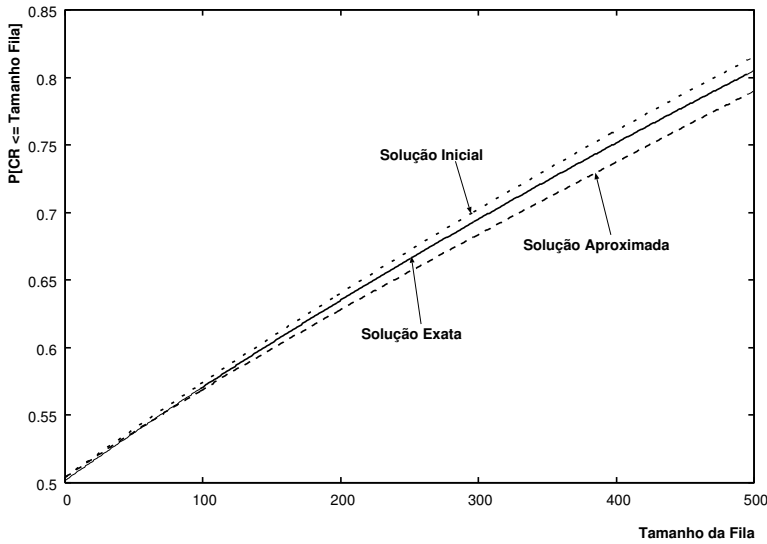


Figura 5.9: Resultados obtidos pelo algoritmo iterativo para um HMM com dois estados ocultos e carga no sistema de $\rho = 0.7$ e $B = 500$.

sultados. Após a aplicação do algoritmo, o máximo erro absoluto diminuiu em 2 vezes, passando de $1.7182e-2$ para $7.5870e-3$, quando comparado com o erro absoluto máximo da solução inicial.

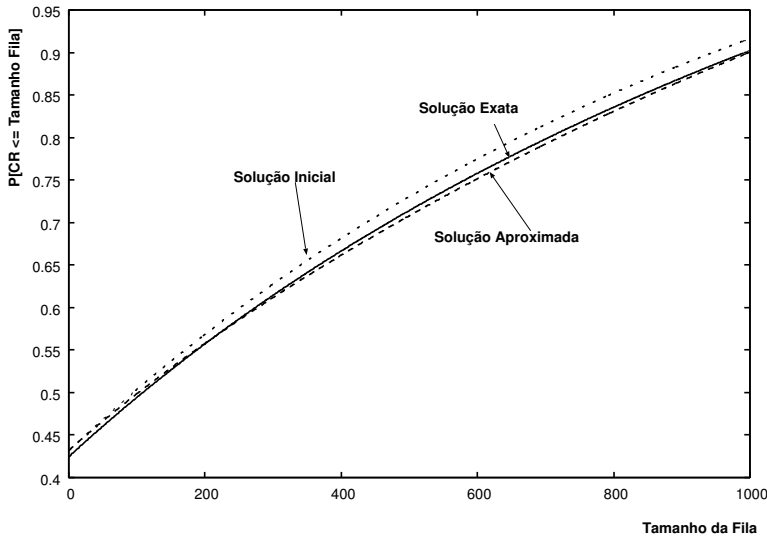


Figura 5.10: Resultados obtidos pelo algoritmo iterativo para um HMM com dois estados ocultos, $B = 1000$ e carga no sistema de $\rho = 0.7$.

Passemos a seguinte carga no sistema: $\rho = 0.5$ e $\mu = 8332$. Para este sistema, temos os seguintes valores de recompensas podem ser emitidos: $\{r_1 = -8332, r_2 = 1668, r_3 = 6668\}$. A Figura 5.11 apresenta os resultados obtidos. Podemos observar que a solução inicial, onde o termo dado pela Equação 5.30 é igualado a zero encontra-se mais distante

5.5 Exemplos

do que todos os demais resultados aproximados, com máximo erro absoluto da solução inicial igual a $6.5117e-2$. Confirmando o comportamento mostrado na Seção 5.5.1, quanto maior a diferença entre os valores de recompensas emitidos em um mesmo estado oculto, mais distante encontra-se a solução inicial. O erro da aproximação proposta, que neste caso foi de $2.8262e-2$. Apesar de o valor ser maior que nos casos anteriores, este erro equivale a aproximadamente 5% do menor valor da medida de interesse sendo calculada.

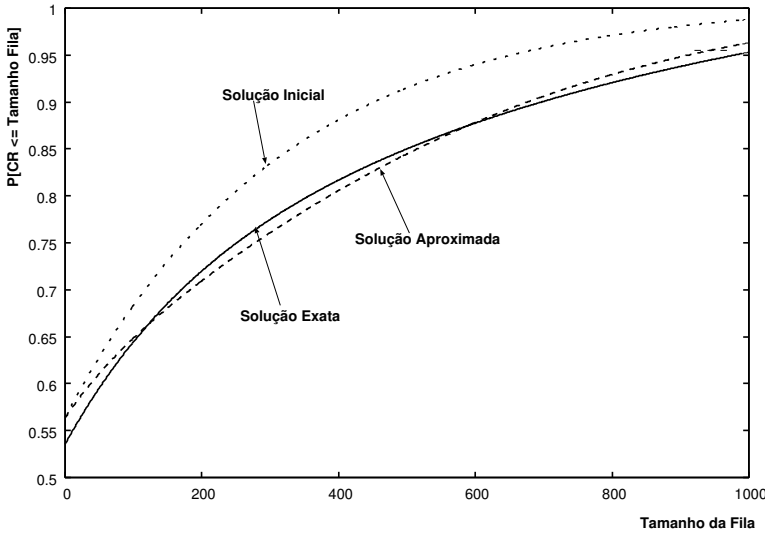


Figura 5.11: Resultados obtidos pelo algoritmo iterativo para um HMM com dois estados ocultos, $B = 1000$ e carga no sistema de $\rho = 0.5$.

Como um segundo conjunto de exemplos, seja um modelo de Markov oculto com $S = \{A, B, C\}$, e as seguintes taxas de emissão de informação: para o estado A , $\{\delta_1 = 0, \delta_2 = 20, \delta_3 = 50\}$, com probabilidades $\{p_1 = 0.3, p_2 = 0.3, p_3 = 0.4\}$; para o estado B , $\{\delta_4 = 900, \delta_5 = 800, \delta_6 = 500\}$, com probabilidades $\{p_4 = 0.2, p_5 = 0.3, p_6 = 0.5\}$; e estado C , $\{\delta_7 = 800, \delta_8 = 500, \delta_9 = 700\}$, com probabilidades $\{p_7 = 0.2, p_8 = 0.7, p_9 = 0.1\}$. O tamanho da fila de fluido associada ao HMM é de $B = 100$. A Figura 5.12 apresenta a representação gráfica do modelo.

Consideremos a análise do resultado da aplicação do algoritmo iterativo proposto quando a carga de informações no sistema é de $\rho = 0.99$, com capacidade de serviço de informações $\mu = 331$. Os valores de recompensas emitidas pelo processo \mathcal{Y} de observações são: $\mathcal{R} = \{r_1 = -331, r_2 = -311, r_3 = -271, r_4 = 569, r_5 = 469, r_6 = 169, r_7 = 469, r_8 = 169, r_9 = 369\}$. A aplicação do algoritmo iterativo aproximativo reduziu o erro máximo absoluto em aproximadamente 2 vezes, em relação ao erro absoluto máximo da solução inicial. Foram necessárias 15 iterações para obter a medida aproximada, sendo a máxima ordem dos autovalores envolvidos igual a $1e-2$. A Figura

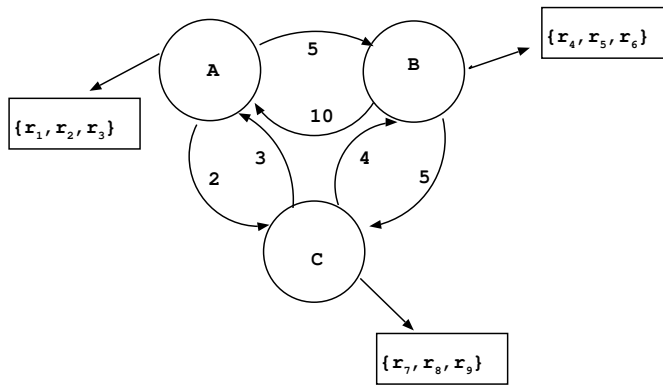


Figura 5.12: Modelo de Markov oculto utilizado para o segundo conjunto de exemplos. Neste, temos três estados ocultos, com a possibilidade de emissão de nove recompensas distintas.

5.13 mostra os resultados obtidos.

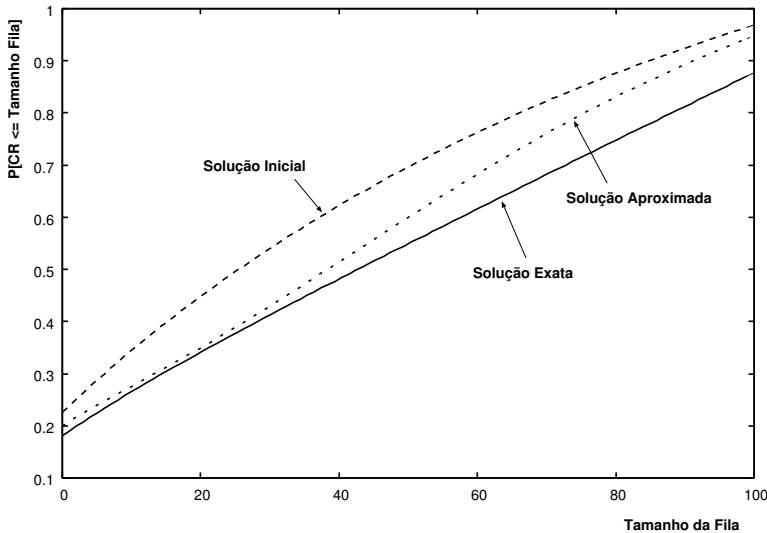


Figura 5.13: Resultados obtidos pelo algoritmo iterativo para um HMM com três estados ocultos, $B = 100$ e carga no sistema de $\rho = 0.99$.

Seja a carga no sistema igual a $\rho = 0.9$ e $\mu = 364$. As recompensas emitidas são as seguintes $\mathcal{R} = \{r_1 = -364, r_2 = -344, r_3 = -314, r_4 = 536, r_5 = 436, r_6 = 136, r_7 = 436, r_8 = 136, r_9 = 336\}$, onde podemos observar que a máxima diferença entre recompensas emitidas por um mesmo estado oculto não ultrapassa 3 vezes. A Figura 5.14 apresenta os resultados. O erro máximo da solução inicial ocorre no ponto $y \approx 50$, onde a solução aproximada é igual a $w(50) = 6.9998e-1$ e a solução exata é igual a $w(50) = 5.9259e-1$, ou seja, um erro absoluto de aproximadamente 18% da medida exata. Considerando a solução aproximada, temos que $w(50) = 6.0409e-1$, ou seja, um erro de

5.5 Exemplos

apenas 2% em relação a medida exata. Em contrapartida, conforme podemos observar, a diminuição do erro não é a mesma para todos os níveis de recompensa acumulada. Por exemplo, consideremos o ponto $y = 80$. Para este caso, o erro diminuiu de 14% para 6%.

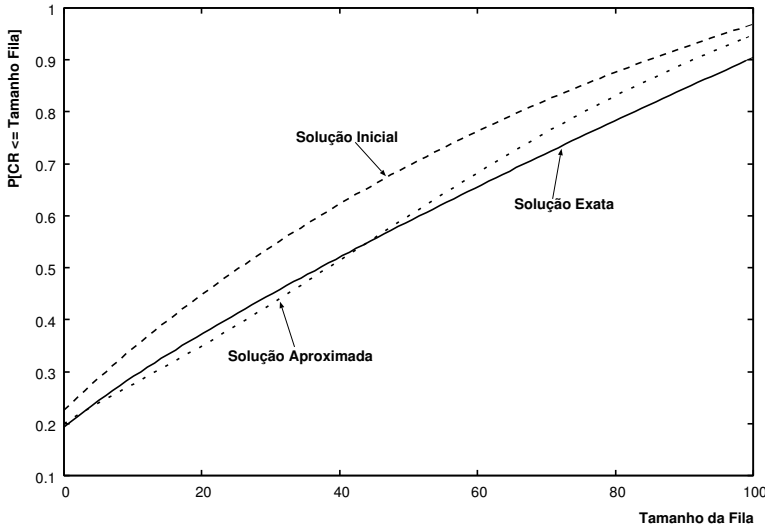


Figura 5.14: Resultados obtidos pelo algoritmo iterativo para um HMM com três estados ocultos, $B = 100$ e carga no sistema de $\rho = 0.9$.

Finalizando, consideremos o caso em que a diferença entre valores das recompensas emitidas em um determinado estado oculto é de, no máximo, aproximadamente 5.5 vezes. Este cenário ocorre quando consideramos $\rho = 0.8$ e $\mu = 410$, resultando no seguinte conjunto de recompensas $\mathcal{R} = \{r_1 = -410, r_2 = -390, r_3 = -360, r_4 = 490, r_5 = 390, r_6 = 90, r_7 = 390, r_8 = 90, r_9 = 290\}$. A Figura 5.15 ilustra os resultados obtidos. O erro absoluto máximo, quando considerada a solução inicial, foi de $1.7850e-1$, ocorrido no ponto $y \approx 50$, resultando em um erro de 25% da medida exata. Após a aplicação do algoritmo iterativo, este erro diminuiu para 7% da medida exata. O máximo erro absoluto da solução aproximada foi igual a $7.9628e-2$, ocorrido no ponto $y \approx 80$, e, portanto, inferior a 10% da medida real.

5.5.3 Classe 3: HMM baseada em [37]

Seja a seguinte descrição para o modelo de Markov oculto, cujas as recompensas que são observadas são referentes ao tráfego do canal de saída do Departamento de Engenharia de Sistemas e Computação/COPPE/UFRJ [37]. A cardinalidade das recompensas distintas atribuídas ao processo de observações \mathcal{Y} é de $|\mathcal{R}| = 15$, sendo a cardinalidade do espaço de estados $|\mathcal{S}| = 4$, com $\mathcal{S} = \{A, B, C, D\}$. O processo HMM

5.5 Exemplos

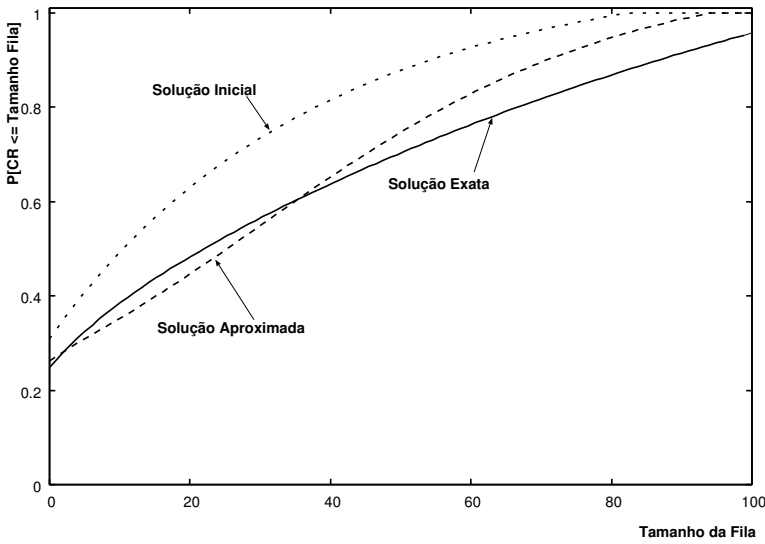


Figura 5.15: Resultados obtidos pelo algoritmo iterativo para um HMM com três estados ocultos, $B = 100$ e carga no sistema de $\rho = 0.8$.

que envia informações a uma fila de tamanho $B = 1000$ é mostrado na Figura 5.16. Os valores das recompensas emitidas por cada um dos estados ocultos do processo são iguais a: $\mathcal{R} = \{r_1 = -821, r_2 = -663, r_3 = -505, r_4 = -347, r_5 = -189, r_6 = -189, r_7 = -347, r_8 = -821, r_9 = 2181, r_{10} = 2655, r_{11} = 3919, r_{12} = 127, r_{13} = 285, r_{14} = 601, r_{15} = 759\}$. O estado A emite as recompensas $\{r_1, r_2, r_3, r_4, r_5\}$, com distribuição equiprovável de emissão de recompensas; o estado B emite as recompensas $\{r_6, r_7, r_8\}$, com as respectivas probabilidades $\{p_6 = 0.3, p_7 = 0.3, p_8 = 0.4\}$; o estado C emite as recompensas $\{r_9, r_{10}, r_{11}\}$, com as respectivas probabilidades $\{p_9 = 0.3, p_{10} = 0.3, p_{11} = 0.4\}$ e o estado D emite as recompensas complementares, com a seguinte distribuição de probabilidades $\{p_{12} = 0.3, p_{13} = 0.2, p_{14} = 0.1, p_{15} = 0.4\}$.

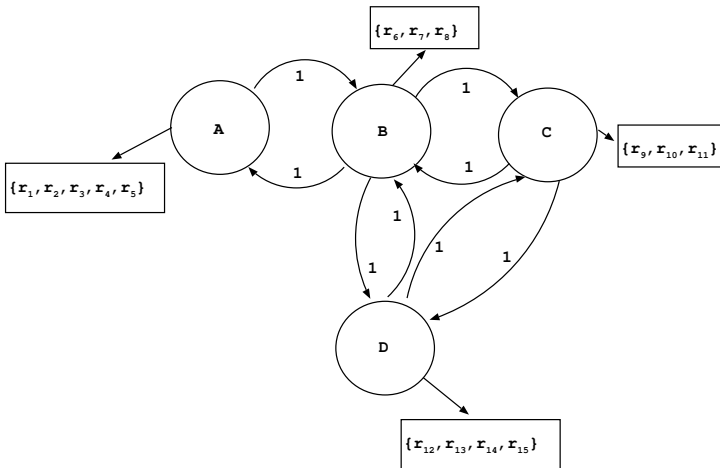


Figura 5.16: Processo HMM em estudo.

5.5 Exemplos

A Figura 5.17 mostra o resultado obtido pelo algoritmo iterativo proposto. Para este exemplo, a solução exata, a partir da solução do processo expandido \mathcal{W} , foi obtida através do método proposto em [88], pois a metodologia proposta em [5] não preservou as condições de contorno para o caso em que $w_{C,9}(0) = 0$, sendo $(C, 9)$ o estado da cadeia expandida que refere-se ao estado oculto C e recompensa r_9 . Devido a erro numérico na solução do sistema para os valores das constantes de integração, $w_{C,9}(0) = 0.1620$, resultando em uma solução inválida.

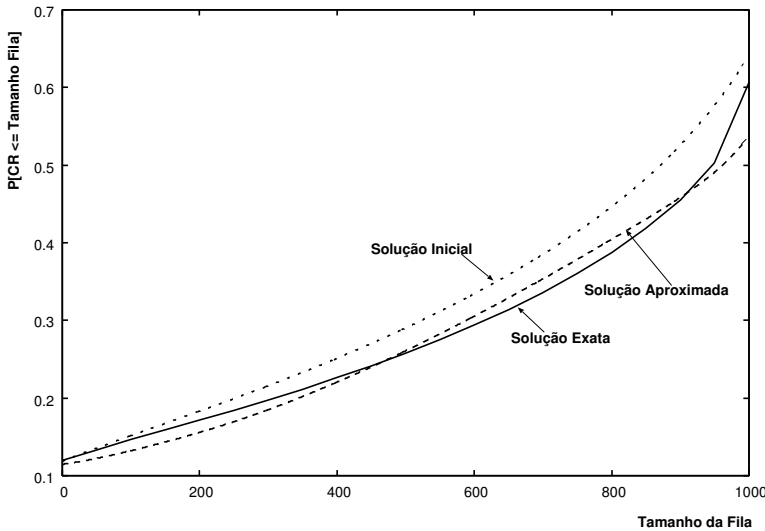


Figura 5.17: Distribuição da recompensa acumulada em uma fila de tamanho $B = 1000$, com valores de recompensas referentes ao tráfego real coletado em um canal do Departamento de Engenharia de Sistemas e Computação/COPPE/UFRJ, e previamente mostrado em [37].

Em contrapartida, o algoritmo iterativo proposto fornece uma solução aproximada, cujo o valor máximo do erro absoluto é igual a $6.7188e-2$ no ponto $y = 1000$, sendo este erro relativo igual a $1.1018e-1$. Pelo algoritmo proposto, devemos resolver um sistema de equações de dimensão 4, o que diminui a probabilidade de erros numéricos semelhantes aos que ocorrem quando a metodologia apresentada em [5] é aplicada a modelos com maior número de estados. O total de iterações necessárias para atingir o critério de parada definido anteriormente foi de 11, sendo a ordem dos autovalores envolvidos na obtenção dos resultados igual a $1e-2$, com o total de $6.237e5$ multiplicações, considerando $c = 1000$, sendo c o total de níveis da recompensa acumulada a terem a medida de interesse calculada. Vale ressaltar que para obter o comportamento da distribuição de probabilidades acumulada, poderíamos considerar um conjunto menor de pontos. Em alguns casos, podemos somente estar interessados no valor da distribuição no ponto $y = 0$ ou $y = B$, diminuindo significativamente o custo computacional. Para obtermos a solução

exata através do método [88], cuja ordem do custo computacional é de $O(mdMN^3)$, com o parâmetro m igual ao total de recompensas do modelo, d o número médio de entradas não nulas da matriz de probabilidades \mathbf{P} , M a cardinalidade do espaço de estados e N o ponto de truncagem do termo de *Poisson*, foram necessárias $2.3918e15$ multiplicações. Este custo se explica principalmente pelo fato do total de multiplicações estar relacionado com o tempo t de observação da medida.

Finalizando, baseado no conjunto de exemplos das classes $\{1, 2, 3\}$, foram necessárias, em média 4.5 iterações para atingir o critério de parada quando a maior ordem dos autovalores do modelo era de $1e-3$ (o maior número de iterações para esta categoria de exemplos foi de 11 iterações, sendo, as demais, entre 3 e 4 iterações), 15 iterações para os casos onde esta ordem era igual a $1e-2$ e 17 para os casos onde a ordem era de $1e-1$. Este mesmo comportamento se repetiu para os demais testes realizados, porém não apresentados nesta tese. Estes resultados indicam uma correlação entre a ordem dos autovalores e a taxa de convergência do algoritmo proposto, fornecendo uma direção para o aprofundamento dos estudos analíticos sobre a taxa de convergência do algoritmo.

Sobre a precisão do algoritmo proposto, baseado nos testes empíricos, que quanto maior a acurácia da solução inicial, maior a acurácia da solução final, o que é esperado. Em todos os casos, o erro da solução final do algoritmo foi, no pior caso, igual a 10%, sendo que a grande maioria dos valores ficaram abaixo de 7%. Como no caso da análise da taxa de convergência, a formalização da acurácia da aproximação, através de uma possível definição de limites inferior e superior, é um tema para trabalhos futuros.

5.6 Sumário das Contribuições

Modelos de Markov ocultos, anteriormente aplicados a área de processamento de sinal de voz, têm sido estudados em diversas outras áreas, incluindo redes de computadores ([41],[37],[138],[50]). Uma das aplicações destes modelos é na caracterização de tráfego de informações em *backbones*, enviando estas informações para fila de roteadores. Nesta aplicação, uma das principais medidas de interesse é a distribuição de probabilidade $P[CR \leq x] = \lim_{t \rightarrow \infty} P[CR(t) \leq x]$ do conteúdo da fila. As principais contribuições apresentadas neste capítulo são as seguintes:

1. Formalização de uma equação diferencial que modela a variação da variável aleatória que representa a recompensa acumulada, com limite superior e inferior, em uma fila

de fluido alimentada por um modelo de Markov oculto;

2. Propomos um algoritmo iterativo, que fornece resultados com boa acurácia quando comparados com o resultado exato que utiliza a metodologia de expansão espectral apresentada em [5]. O algoritmo aqui apresentado, diminui a ocorrência de problemas numéricos, dado que as matrizes envolvidas são da dimensão do total dos estados pertencentes ao processo oculto \mathcal{X} . A principal vantagem do algoritmo é o de calcular autovalores e autovetores de matrizes menores do que se fosse utilizada a transformação do HMM para uma cadeia de Markov com uma recompensa em cada estado. Sendo R o número de recompensas associadas a um estado do HMM e M o número de estados, a expansão da matriz e a utilização do método de [5] implicaria no cálculo de autovalores e autovetores de uma matriz de ordem $MR \times MR$, ao passo que nosso algoritmo trabalha apenas com uma matriz de dimensão $M \times M$. Como “efeito colateral”, problemas numéricos podem, também, ser diminuídos;
3. Apresentamos um conjunto de testes que, empiricamente, mostram que a taxa de convergência do algoritmo proposto está correlacionada com os autovalores das matrizes envolvidas nas etapas do algoritmo, e indicam a boa acurácia do algoritmo proposto. Estes resultados, mesmo empíricos, são fundamentais para direcionar possíveis formalizações das questões sobre convergência do algoritmo e da sua acurácia.

Capítulo 6

Avaliando a Qualidade de Serviço para Aplicações Interativas de Voz na *Internet*

6.1 Introdução

Aplicações como voz sobre IP (*Internet Protocol*) [18], videoconferência, vídeo sob demanda, entre outras, estão cada vez mais comuns. Estas aplicações possuem requisitos de retardo, *jitter* e perda bastante distintos das aplicações de dados. Os requisitos das aplicações são, em geral, denominados de Qualidade de Serviço (QoS). A *Internet* de hoje não oferece a QoS exigida pelas aplicações multimídia. O serviço que a *Internet* provê é o *best-effort* (melhor esforço), não oferecendo, por exemplo, garantias de *jitter* ou retardo máximo entre o emissor e o receptor de uma aplicação.

Como a *Internet* atual não garante nenhuma qualidade ao usuário final, é importante que as aplicações possam implementar mecanismos para manter a QoS em um nível mínimo aceitável. Portanto, é essencial que sejam determinados quais são os parâmetros que influenciam a QoS, bem como entender como o efeito combinado destes parâmetros afetam a QoS percebida pelo usuário final das aplicações em tempo real.

Neste capítulo iremos estudar aplicações de voz sobre IP (VoIP) com interatividade. Mostraremos que a qualidade da conversa, percebida pelo usuário final, depende de uma combinação complexa de vários parâmetros. Estendemos a utilização da metodologia apresentada em [101] ao contexto de conversas interativas e mapeamos os valores de um conjunto de parâmetros em uma única nota que é próxima ao que um observador humano atribuiria a uma sessão de conversação interativa, caso este tivesse participado de um teste

subjetivo.

A análise de QoS pode ser realizada usando testes do tipo objetivo ou subjetivo. A grande maioria dos testes objetivos se baseiam na medição de parâmetros relacionados com a codificação da voz ou com a qualidade da rede (ex: *jitter*, retardo, perda) no receptor comparando-os, com os mesmos parâmetros no emissor ([8],[69] e [137]). Testes subjetivos são baseados em avaliações feitas por humanos sobre condições bem definidas e controladas [71].

A principal característica das métricas objetivas é que estas propõem diferentes maneiras para comparar as amostras de dados recebidas através da rede com as amostras de dados originais. Apesar deste tipo de avaliação ser menos custosa em termos de tempo e complexidade, a correlação com os resultados subjetivos é baixa, principalmente quando são incluídos parâmetros inerentes a rede, como retardo ou *jitter* [141]. Adicionalmente, dado que o sinal original é necessário para realizar a comparação com o sinal recebido, a grande maioria dos testes objetivos não são indicados para aplicações em tempo real. Em contrapartida, a análise em tempo real é essencial se a aplicação em questão deve estabelecer alguma forma de controle para aumentar a qualidade que está sendo oferecida ao usuário.

Os testes objetivos mais comuns encontrados na literatura destinados a avaliação da qualidade de aplicações multimídia são o *Signal-to-Noise Ratio* (SNR), *Segmental SNR*, *Perceptual Speech Quality Measure* (PSQM e PSQM+) ([8] e [7]), *Measuring Normalizing Blocks* (MNB) [137], *ITU-T E-model* [69] e *Perceptual Evaluation of Speech Quality* (PESQ)[72]. Exceto para o *ITU-T E-model* e para o algoritmo P.563 [114], todos os outros testes comparam o sinal original enviado pelo usuário e o sinal recebido pelo outro usuário participante da conversa. Adicionalmente, estes testes avaliam somente a qualidade da voz recebida através de comparação entre o sinal gerado pelo emissor e o sinal degradado no receptor. Estes testes não analisam a percepção dos usuários da sessão VoIP com relação a interatividade ou características relacionadas a redes de dados, como o retardo, perda, entre outras.

Avaliações subjetivas de qualidade medem a qualidade do áudio ou de uma conversa interativa a partir da perspectiva do usuário final, e, por isso, conseguem refletir as sensações inerentes ao seres humanos. Para fluxos multimídia interativos, a recomendação ITU-T P.920 [73] juntamente com a recomendação ITU-T P.800 [71] descrevem como as avaliações subjetivas devem ser realizadas. O resultado dos testes subjetivos é denominado, tecnicamente, como *Mean Opinion Score* (MOS), dado que assume a média de

todas as avaliações realizadas pelo conjunto de participantes e expressa a qualidade subjetiva em relação a um fluxo multimídia.

Apesar de a avaliação de qualidade das aplicações multimídia utilizando testes subjetivos ser altamente eficiente para analisar muitas das características inerentes a estas aplicações, estes testes possuem muitas limitações: (a) ambientes de testes devem ser estabelecidos; (b) o processo não pode ser automatizado (por definição) e; (c) testes subjetivos consomem muito tempo para realização, e, por conseguinte, não podem ser refeitos com grande frequência.

A metodologia PSQA (*Pseudo-Subjective Quality Assessment*) [101] é baseada na combinação de testes subjetivos com uma ferramenta estatística de aprendizado, chamada *Random Neural Network* (RNN) [53], que possibilita estimativas de qualidade, utilizando parâmetros da aplicação e medidos na própria rede. Com esta metodologia, os testes subjetivos são realizados uma única vez, fornecendo dados necessários para o treinamento da rede neural. Por conseguinte, futuras avaliações em tempo real utilizando esta rede neural poderão ser realizadas, “imitando” o comportamento de uma pessoa, caso esta estivesse avaliando a QoS da aplicação em curso.

Esta metodologia foi previamente utilizada para aplicações de fluxo de áudio transmitido pela *Internet* [101], sem considerar interatividade entre os participantes. Adicionalmente, diversos resultados, utilizando PSQA, para redes sem fio e de algoritmos de controle baseados na QoS percebida pelo usuário final podem ser encontrados em [136].

As principais contribuições deste capítulo, constituem na aplicação da metodologia PSQA para avaliar aplicações interativas de voz na *Internet* e na análise de diversos resultados que mostram o impacto dos parâmetros como codificação, esquema de redundância, percentual de pacotes perdidos (PP), tamanho médio da rajada de perda de pacotes (*Mean Burst Loss Size - MBLs*), retardo (*delay*) e *jitter* na qualidade percebida pelo usuário final da aplicação em questão. Como a teoria utilizada para obtenção dos resultados mostrados neste capítulo foi anteriormente proposta e desenvolvida em ([136] e [102]), iremos abordar brevemente as principais definições para a compreensão dos resultados, e focalizar na contribuição principal que é o desempenho da metodologia no caso das aplicações de VoIP com interatividade. A escolha desta última aplicação deve-se ao fato de que parte dos modelos teóricos estudados podem ser aplicados a modelos de taxa de erros em aplicação de voz sobre *Internet Protocol* (IP) que, por sua vez, podem ser baseados em cadeias ocultas de Markov ([48] e [50]).

Vale ressaltar que os demais estudos encontrados na literatura, relacionados com

testes subjetivos ou objetivos, consideram um ou dois parâmetros simultaneamente para aferirem a QoS percebida por um usuário, negligenciando o efeito conjunto dos parâmetros citados no parágrafo anterior. A metodologia apresentada neste capítulo é importante para melhor compreensão de como um conjunto de parâmetros previamente definidos influencia na qualidade das aplicações de voz na *Internet*.

Conforme apresentaremos neste capítulo, a metodologia PSQA mostrou-se adequada para avaliar, em tempo real, a QoS percebida pelo usuário final de uma aplicação de voz na *Internet*. Os estudos aqui realizados consideram um cenário “estático”, onde o grau de satisfação do usuário é aferido sem considerar o histórico de toda a sessão interativa analisada.

Este capítulo está estruturado da seguinte forma. Na Seção 6.2 apresentamos uma lista de trabalhos encontrados na literatura e que abordam este assunto. A metodologia de avaliação utilizada é descrita em 6.3. Os testes subjetivos realizados são apresentados na Seção 6.4 e os resultados da aplicação da metodologia em 6.5. Na Seção 6.6, descrevemos o sumário das contribuições deste capítulo.

6.2 Trabalhos Relacionados

O principal objetivo desta seção é apresentar os principais trabalhos encontrados na literatura sobre avaliação de QoS de aplicações multimídia na *Internet*. Este tópico de pesquisa tem sido extensivamente estudado na literatura e um grande grupo de artigos que apresentam diferentes metodologias para esta avaliação pode ser encontrado. A descrição dos trabalhos relacionados se restringirá a destacar as principais características de cada uma das metodologias de maior importância encontradas na literatura e que são mais próximas, em termos de objetivo final, da metodologia PSQA descrita e utilizada neste capítulo.

Conforme citado anteriormente, existe uma lista extensa de trabalhos que estudam o impacto na Qualidade de Serviço percebida pelo usuário quando o seguinte conjunto de parâmetros é considerado: codificação utilizada (Para transmissão através da *Internet*, o áudio deve ser digitalizado e compactado. Diversos codificadores podem ser encontrados como *Pulse Codec Modulation* (PCM) e *Global System for Mobile Communications* (GSM). A principal diferença entre eles reside na taxa de compactação das informações a serem enviadas. No trabalho apresentado em [101], o codificador PCM utilizado foi o do tipo linear, com amostras de 16 *bits.*), técnicas de redundância para corrigir erros

(FEC), intervalo entre geração de pacotes de dados (PI), percentual pacotes perdidos na rede (PP), tamanho médio da rajada de pacotes perdidos (MBLS), o retardo (*delay*) e *jit-ter*. No entanto, a grande maioria destes trabalhos estão relacionados a aplicações do tipo *one-way* (ou seja, com fluxos em uma direção) e sem interatividade, e são focados em estudar o efeito conjunto de no máximo dois dos parâmetros supracitados.

O trabalho mais tradicional de avaliação de qualidade de voz interativa pode ser encontrado em [80]. Este trabalho se difere do apresentado neste capítulo segundo a rede analisada: em [80] os autores descrevem os efeitos do retardo de transmissão na qualidade da voz em conversas telefônicas realizadas nas tradicionais redes públicas de telefonia (*Public Switched Telephone Network (PSTN)*). O principal objetivo do trabalho é obter avaliações subjetivas através da realização de uma série de “tarefas” que representem diferentes tipos de conversas que podem ser feitas, como contar números seguidos, descrever figuras e etc. A conclusão dos testes realizados mostra que atrasos superiores a $500ms$ resultam em dificuldades de compreensão por parte do usuário final, gerando insatisfação em relação a qualidade oferecida pela aplicação. Estes resultados são consistentes com outros trabalhos na literatura, principalmente com a recomendação ITU-T G.114 [70], que apresenta os mesmos limites de retardo aceitáveis durante uma conversa interativa. Todavia, os limites apresentados na recomendação ITU-T G.114 [70] vem sendo questionados na literatura [74]. Os resultados mostrados na Seção 6.5 reforçam o questionamento apresentado em [74].

Consideremos alguns trabalhos que avaliem a qualidade de voz transmitida pela *Internet*. A maior parte destes trabalhos analisam a influência da perda de pacotes na QoS percebida pelo usuário de aplicações multimídia na *Internet*. Em [66], a análise da qualidade de voz fica restrita ao efeito conjunto dos parâmetros percentual de perda de pacotes e o intervalo entre geração de amostras de voz. O estudo feito pelos autores é baseado em sistemas de reconhecimento automático de voz e utiliza um HMM em substituição aos testes subjetivos, com o objetivo de expressar a QoS oferecida pela aplicação de voz ao usuário final. Em contrapartida, a metodologia apresentada neste capítulo considera uma combinação mais complexa de parâmetros, como perda, retardo, codificação, taxa de envio de *bits* e *jit-ter*. Os autores em [76] apresentam uma discussão sobre quais os fatores que podem afetar a QoS de aplicações multimídia quando somente o parâmetro perda de pacotes é considerado. O principal objetivo do trabalho é a apresentação de modelos para caracterização de perda de pacotes, sem nenhuma realização de testes subjetivos que relacionem o quanto a perda de pacotes afeta a QoS percebida pelo usuário.

Em relação a literatura que descreve os testes objetivos, entre todos os citados na

Introdução, somente a técnica *E-model* e o algoritmo P.563 [114] não precisam ter acesso ao sinal original dos dados cuja a qualidade será avaliada. O *E-model* é computacionalmente simples [63] e pode ser facilmente utilizado para aplicações de tempo real. No entanto, o *E-model* não é um método desenvolvido para avaliar QoS (como é claramente especificado na recomendação ITU-T), mas sim, como uma ferramenta de planejamento de redes de dados [122], que considera diversos parâmetros que influenciam na QoS, sendo que a maior parte destes está intrinsecamente ligada à técnicas de processamento de sinal e não a parâmetros relacionados com redes de computadores, que é o campo em que estamos interessados. Somente trabalhos mais recentes estenderam o *E-model* com o intuito de considerar perdas em rajadas que ocorrem nas redes de pacotes. Todavia, mesmo com estas extensões, os resultados obtidos para a avaliação de qualidade percebida pelo usuário final, utilizando o *E-model*, são limitadas. Adicionalmente, os resultados obtidos são muito distantes dos resultados obtidos através de testes subjetivos, e, conseqüentemente, da métrica MOS [63].

O trabalho no qual nos baseamos publicado por Mohamed *et al* [101], utiliza a metodologia PSQA para avaliar a qualidade de fluxos VoIP em uma única direção, onde parâmetros como perda de pacotes e codificação do áudio (codificadores do tipo *Pulse Code Modulation* (PCM) e GSM) foram considerados. O sucesso na aplicação da metodologia PSQA foi uma das motivações para a extensão de seu escopo a aplicações multimídia interativas. Adicionalmente, entre as motivações apresentadas na seção 6.1, temos a importância de assegurarmos ao usuário final uma Qualidade de Serviço mínima para as aplicações sendo executadas utilizando a *Internet*.

Finalizando, algumas ferramentas mais atuais como o VQmon [134], PSM [117] e OPERA [113] avaliam a qualidade de voz baseadas nos padrões ITU-T E-model, P.862 e P.563, estendendo as funcionalidades destes para considerar parâmetros como retardo, perda, entre outros. No entanto, diferentemente da metodologia apresentada neste capítulo, estas ferramentas são proprietárias, impossibilitando a análise detalhada de suas implementações e posteriores modificações. Maiores detalhes destas ferramentas podem ser encontrados nas respectivas páginas *web* das empresas que fornecem estes serviços.

6.3 A Metodologia PSQA para Avaliação de Qualidade

Esta seção objetiva descrever o conjunto de passos a serem realizados com intuito de automatizar a avaliação de qualidade de uma conversa interativa utilizando a *Inter-*

6.3 A Metodologia PSQA para Avaliação de Qualidade

net. A metodologia PSQA não é uma contribuição original do nosso trabalho, mas sim a sua aplicabilidade no contexto de conversas interativas na *Internet* e no estudo do impacto da variação de um conjunto de parâmetros neste tipo de aplicação. Portanto, iremos restringir a apresentação dos principais conceitos da mesma. Maiores detalhes desta metodologia podem ser encontrados em [136] e [102].

A metodologia PSQA pode ser aplicada para avaliar a qualidade de aplicações multimídia em tempo real, podendo auxiliar na implementação de mecanismos de controle que visam oferecer uma melhor QoS ao usuário final. Como exemplo de mecanismo de controle, consideremos o caso em que o percentual de perda de pacotes na rede esteja muito alta. A metodologia PSQA fornece uma nota para esta qualidade (MOS) e através desta informação, a aplicação poderá aperfeiçoar, por exemplo, o seu esquema de FEC, resultando em uma melhor qualidade final. Esta metodologia está baseada na utilização da teoria de *Randon Neural Networks* (RNNs) ([53],[59]) para definição do MOS relacionado com a aplicação em questão.

Em linhas gerais, esta metodologia pode ser dividida em três etapas principais: (a) escolha dos parâmetros que afetam diretamente a QoS da aplicação considerada; (b) realização dos testes subjetivos e; (c) escolha da RNN, seu treinamento e sua posterior validação. O conjunto de parâmetros a ser escolhido depende da aplicação a ser considerada. Neste trabalho estamos interessados nas aplicações de VoIP interativas e em tempo real. Em relação aos testes subjetivos, duas ferramentas devem ser utilizadas: primeiramente a ferramenta VoIP propriamente dita, que será utilizada pelos usuários participantes dos testes (referenciados na literatura como *human subjects*) e um módulo que seja capaz de emular diferentes condições da rede de acordo com os parâmetros escolhidos (por exemplo, retardo na entrega de pacotes e possíveis perdas destes). A partir dos resultados destes testes, um banco de dados é criado, onde todos os resultados MOS, cuja a definição formal será apresentada posteriormente, são armazenados, cada qual relacionado com um conjunto de valores para os parâmetros escolhidos. Finalmente, uma arquitetura para a rede neural aleatória é selecionada, bem como um algoritmo para seu posterior treinamento. Após o treinamento, um procedimento de validação é realizado com o objetivo de verificar se o MOS obtido através da RNN está em conformidade com o MOS obtido a partir dos testes subjetivos. A Figura 6.1 mostra as três etapas principais da metodologia PSQA. A seguir, detalhamos cada uma destas etapas.

Os parâmetros a serem considerados são aqueles que, de certa forma, resultarão em um impacto significativo na qualidade da aplicação avaliada. Como neste trabalho estamos considerando aplicações VoIP interativas, os seguintes parâmetros são candidatos

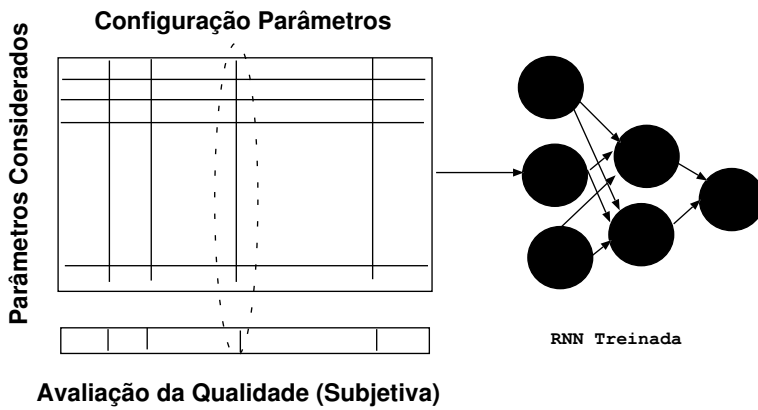


Figura 6.1: Representação gráfica das três etapas principais da metodologia PSQA.

imediatos:

1. Percentual de pacotes perdidos e tamanho médio da rajada de perda de pacotes: Um dos maiores problemas em não possuir uma rede de dados confiável é que alguns pacotes enviados podem ser perdidos. As perdas de pacotes podem ocorrer devido a várias razões, como problemas de roteamento, erros de transmissão e congestionamento na rede. Entre estas razões, perdas devido a congestionamento são as mais comuns. Na *Internet*, os roteadores normalmente possuem filas de armazenamento do tipo *drop-tail*, o que significa que quando a fila está cheia, novos pacotes são descartados quando chegam a fila, até que algum pacote seja servido e uma posição na fila seja “liberada”. Em relação a aplicação que estamos analisando, queremos avaliar a partir de que percentual de pacotes perdidos e de tamanho médio da rajada de perda a compreensão da conversa entre os participantes da sessão é prejudicada, afetando diretamente a QoS percebida pelo usuário.
2. *Forward Error Correction*: As técnicas classificadas como FEC têm como objetivo reconstruir pacotes descartados na rede a partir de informações adicionais. A utilização destas técnicas ajudam a aumentar a qualidade provida pela *Internet* à aplicação de voz. Na literatura diversos esquemas de FEC podem ser encontrados. A eficiência de cada um destes esquemas pode ser quantificada através do total de pacotes perdidos que pode ser recuperado, bem como da quantidade adicional de informação que deve ser transmitida para que estes sejam recuperados.
3. Retardo e *jitter*: O retardo fim-a-fim e o *jitter* possuem uma grande influência na qualidade percebida quando aplicações multimídia, principalmente interativas, são utilizadas. Isto ocorre dado que altos valores de retardo e uma variação grande nos

Retardo (ms)	Qualidade
< 150	Aceitável
150 – 400	Degradação na interatividade
> 400	Inaceitável

Tabela 6.1: Influência do retardo em uma direção na interatividade da comunicação segundo a recomendação G.114.

valores de *jitter* reduzem o grau de interatividade. Aplicações de voz exigem a entrega dos pacotes no destino dentro de um retardo máximo e com intervalo entre pacotes igual ao intervalo que os pacotes foram gerados na origem. Um grande conjunto de trabalhos na literatura ([80],[122] e referências) tem sido feito com o objetivo de determinar o impacto do retardo em comunicações com diversos participantes, em particular em aplicações de telefonia tradicional. Naqueles trabalhos, foram estabelecidos alguns limites para os valores do retardo que definem a qualidade da conversa. A Tabela 6.1 mostra a influência do retardo em uma direção, de acordo com a norma ITU-T G.114 [70]. Claramente, os efeitos do retardo na qualidade depende enormemente do tipo de aplicação e das expectativas dos usuários.

4. Taxa de Envio de *bits*: Este parâmetro influencia diretamente na qualidade do áudio enviado ao cliente. Maiores taxas de *bits* tendem a oferecer melhor qualidade ao usuário, quando o mesmo codificador está sendo considerado.

Devemos notar que alguns destes parâmetros são variáveis aleatórias, como por exemplo o retardo, e outros não, como o esquema de FEC utilizado. Para os parâmetros que não são variáveis aleatórias, um subconjunto de valores, dentre todos os possíveis, deve ser selecionado para a realização dos testes subjetivos. Para as variáveis aleatórias utilizamos um modelo para o parâmetro percentual de pacotes perdido, e uma distribuição para os parâmetros de retardo e *jitter*. Maiores detalhes serão apresentados no final desta seção.

Seja $\mathcal{P} = \{\pi_1, \dots, \pi_P\}$ o conjunto de parâmetros selecionados e $\{\nu_{i1}, \dots, \nu_{iH_i}\}$ um conjunto representativo dos valores associados ao parâmetro π_i , com $\nu_{i1} < \nu_{i2} < \dots < \nu_{iH_i}$. Denotaremos por *configuração*, um conjunto particular de valores atribuídos a todos os parâmetros que afetam a qualidade da aplicação em estudo e que foram listados na primeira etapa do método proposto. O conjunto \mathcal{C} para todas as possíveis configurações, possui cardinalidade $|\mathcal{C}| = \prod_{i=1}^P H_i$, que em termos reais é extremamente grande, dado

6.3 A Metodologia PSQA para Avaliação de Qualidade

que uma permutação entre todos os valores dos parâmetros deverá ser realizada. Em consequência desta grande cardinalidade e dos custos da realização da avaliação de QoS através de testes subjetivos, selecionamos um subconjunto de \mathcal{C} (com cardinalidade $|S| < |\mathcal{C}|$) para o treinamento e validação da rede neural que estimará o valor do MOS percebido pelo usuário final.

Com o objetivo de realizar os testes subjetivos, criamos um ambiente onde os participantes foram divididos em pares e uma conexão VoIP interativa entre eles foi estabelecida. A rede de comunicação usada pela conexão foi emulada, proporcionando o controle dos valores a serem associados a cada um dos parâmetros π_i . Este controle é essencial para a avaliação da qualidade, dado que devemos assegurar que a avaliação sendo realizada deve estar realmente relacionada com a configuração desejada. A utilização da *Internet* inviabilizaria este fato.

Cada sessão VoIP interativa corresponde a uma das S configurações $\{\gamma_1, \dots, \gamma_S\}$ onde $\gamma_s = (\nu_1^{(s)}, \dots, \nu_p^{(s)})$ e $\nu_p^{(s)}$ é o valor do parâmetro π_p na configuração γ_s .

Cada um dos pares de participantes realiza S sessões interativas cada uma correspondendo a um conjunto de valores para os parâmetros selecionados. A cada sessão, uma nota, que expressa a qualidade, é atribuída. A metodologia para esta avaliação é descrita nas recomendações ITU-T P.920 e ITU-T P.800. Basicamente, um grupo de participantes é convidado a realizar conversas interativas para uma dada configuração (isto é, cada participante deverá atribuir a cada sessão uma nota previamente definida dentro de uma escala de qualidade $[M_{min}, M_{max}]$). Os valores do conjunto de parâmetros não devem ser conhecidos pelos participantes, com o objetivo de não permitir nenhuma possível relação entre a qualidade percebida propriamente dita e os valores dos parâmetros, evitando influências prévias na avaliação final. Após a realização dos testes, uma análise estatística é realizada, removendo os participantes que porventura atribuíram notas não-confiáveis, evitando possíveis discrepâncias entre os valores atribuídos [68]. Uma nota média, atribuída a cada configuração é calculada a partir das demais notas, em uma métrica conhecida como *Mean Opinion Score* (MOS). Formalmente, o MOS é a qualidade associada a uma linha que transmite voz. Esta é uma métrica subjetiva, onde pessoas que participam de uma conversa utilizando a *Internet* ou uma rede telefônica tradicional, fornecem uma nota entre 1 e 5 à qualidade para esta conversa. O MOS associado a cada configuração γ_s será denotado formalmente por ψ_s . A Tabela 6.2 mostra as notas consideradas e o seus respectivos significados.

A próxima etapa após a realização dos testes subjetivos está relacionada com a es-

Notas	Significado
5	Excelente
4	Bom
3	Razoável
2	Ruim
1	Péssimo

Tabela 6.2: Descrição dos valores de MOS e seus respectivos significados.

colha de uma arquitetura para a rede neural, bem como de um algoritmo de treinamento (detalhes sobre a RNN utilizada serão abordados na próxima seção). As amostras pertencentes ao banco de dados são divididas em duas partes: um grupo utilizado para treinar a RNN e outro grupo para validação da precisão dos resultados fornecidos pela RNN. Após o treinamento e a sua validação, é esperado que a RNN seja capaz de calcular a medida de MOS para qualquer configuração possível, mesmo para as que não foram consideradas durante a etapa de treinamento, sendo esta característica que possibilita “automatizarmos” os testes subjetivos.

Assim, a RNN pode ser vista como uma função real f com P valores reais e com uma saída no intervalo $[1, 5]$, tal que:

- (i) para qualquer amostra γ_s , $f(\nu_1^{(s)}, \dots, \nu_P^{(s)}) \approx \psi_s$,
- (ii) para qualquer outra configuração (ν'_1, \dots, ν'_P) , $f(\nu'_1, \dots, \nu'_P)$ é próxima ao MOS que seria obtido, caso uma avaliação subjetiva, para a configuração ν'_1, \dots, ν'_P , fosse realizada.

A fase de validação é realizada com o intuito de verificar se a condição descrita em (ii) acontece (é válido ressaltar que as amostras usadas nesta fase não estavam presentes na fase de treinamento). Esta validação é realizada através da comparação entre a saída da RNN e o valor esperado do MOS para o conjunto de configurações que não foram utilizados na fase de treinamento. A precisão da RNN treinada é estimada através do cálculo da medida de *Mean Square Error* (MSE) entre o MOS definido pelos participantes dos testes subjetivos e o estimado pela RNN. A operação global da metodologia PSQA está representada na Figura 6.2.

Finalizando a descrição da metodologia PSQA, iremos discutir brevemente a RNN utilizada, o algoritmo de treinamento e as distribuições escolhidas para representar os parâmetros que são variáveis aleatórias. Vale ressaltar que maiores detalhes na teoria de

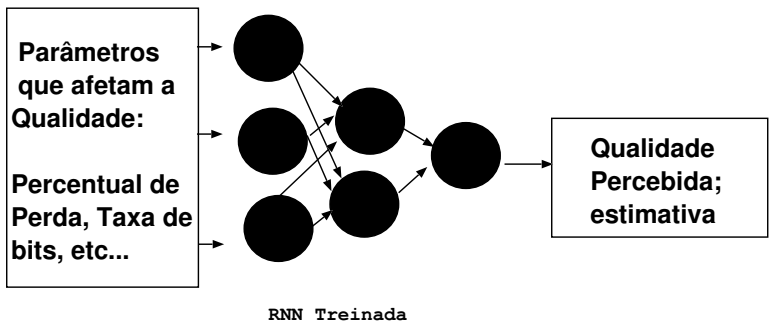


Figura 6.2: Visão global da metodologia PSQA aplicada a avaliação de aplicações interativas de VoIP.

RNN e de algoritmos de treinamento podem ser encontradas em ([53],[59],[136] e [102]).

6.3.1 Rede Neural Aleatória (*Random Neural Network*)

A Rede Neural utilizada neste trabalho é um tipo particular de *G-network* ([54], [59],[57] e [58]) onde cada neurônio se comporta segundo uma fila $M/1$ com taxa de serviço μ_i . Após deixar o neurônio i , ou equivalentemente a fila i , o cliente deixa a rede de filas com probabilidade d_i , passa para a fila j como um cliente positivo com probabilidade p_{ij}^+ ou como um cliente negativo com probabilidade p_{ij}^- . Quando um cliente negativo chega a uma fila, este “elimina” um dos clientes previamente instalados na fila e se “auto-destrói”. Na RNN utilizada neste trabalho, a chegada de clientes a rede de filas é uma variável aleatória com distribuição *Poisson* e é restrita aos clientes do tipo positivo. É mostrado em [56, 54] que este tipo de rede possui solução em forma de produto (*product form solution*), isto é, a distribuição conjunta do número total de clientes nas filas é dada por um produto da distribuição do número total de clientes em cada uma das filas no sistema.

O principal papel da RNN no contexto deste trabalho é como um estimador estatístico, ou seja, a rede neural “aprende” como avaliar a qualidade do áudio, gerando uma avaliação que é atribuída a um ser humano, caso este efetivamente a estivesse realizando. Seja λ_i^+ a taxa de chegada no neurônio i . A rede recebe como entrada $\vec{\lambda} = (\lambda_1^+, \dots, \lambda_P^+)$ e fornece como saída $\vec{q} = (q_1, \dots, q_i, \dots)$, onde $q_i = \lambda_i^+ / \mu_i$ é a carga na fila i , ou similarmente no neurônio i . A partir da teoria para *G-networks*, temos

$$q_i = \frac{\lambda_i^+ + \sum_{j=1}^M q_j w_{ji}^+}{\mu_i + \sum_{j=1}^M q_j w_{ij}^-}, \quad i = 1, \dots, M \tag{6.1}$$

onde $w_{ij}^+ = \mu_i p_{ij}^+$ e $w_{ij}^- = \mu_i p_{ij}^-$. A Figura 6.3 mostra a representação gráfica de uma rede

neural aleatória do tipo *feed-forward* com três camadas.

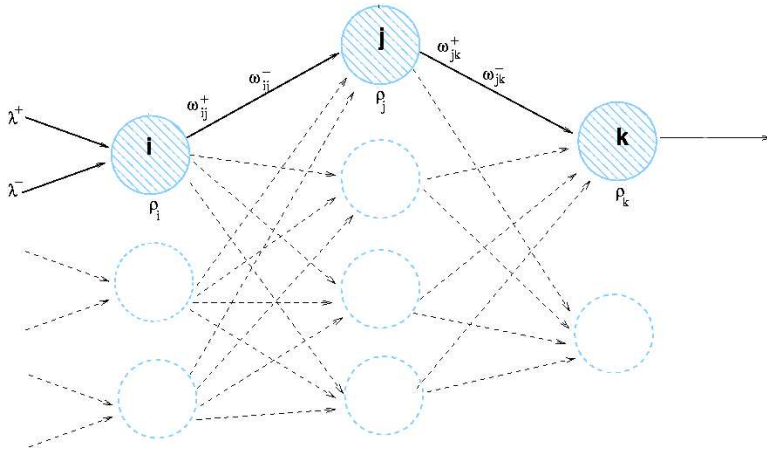


Figura 6.3: Componentes de uma rede neural aleatória. Este esquema representa uma parte de uma rede do tipo *feed-forward* com três camadas.

Para a nossa aplicação, é necessário somente um valor de saída, que corresponderá a qualidade percebida pelo usuário para qualquer configuração de parâmetros, inclusive para aquelas que não foram utilizadas na fase de treinamento. Este valor é definido como a carga ϱ_o para a fila de saída, ou na terminologia de redes neurais, para o neurônio de saída o .

Para o treinamento, são utilizados S pares $(\vec{\nu}^s, \psi_s)_{s=1, \dots, S}$, com $\vec{\nu}^s = (v_1^{(s)}, \dots, v_P^{(s)})$. A rede utilizada se comporta da seguinte maneira. São utilizados P neurônios de entrada e um único neurônio de saída. Quando a taxa de chegada no neurônio i é $\lambda_i^+ = \nu_i^{(s)}$ a taxa de ocupação no neurônio o , ϱ_o , deve aproximar a nota média ψ_s . A saída ϱ_o é uma função dos w_{ij}^+ 's, dos w_{ij}^- 's (que são definidos como *pesos* na terminologia padrão de redes neurais), e das taxas de entrada λ^+ . O treinamento objetiva resolver o seguinte problema de otimização, ou seja, obter o vetor de *pesos* \vec{w}_0 definido por:

$$\vec{w}_0 = \operatorname{argmin}_{\vec{w} \geq \vec{0}} \frac{1}{2} \sum_{s=1}^S (\varrho_o(\vec{w}, \vec{\nu}^s) - \psi_s)^2 \quad (6.2)$$

onde a minimização é realizada no universo de vetores de *pesos* \vec{w}_* com coordenadas positivas.

Em aplicações que envolvem aprendizado, como a abordada neste capítulo, é comum o uso do tipo de topologia *feed-forward*, onde a saída da rede é uma função das variáveis de entrada e de seus *pesos*. A utilização deste tipo de arquitetura implica na simplificação

da computação realizada, resultando em procedimentos mais eficientes de aprendizado. Maiores detalhes podem ser encontrados em ([56],[53], [54],[55],[57]).

Resumidamente, a RNN utilizada neste trabalho possui P neurônios de entrada. Seja \mathcal{I} o conjunto destes neurônios. O fluxo externo que entra na rede corresponde aos parâmetros selecionados na primeira etapa da metodologia. A saída somente é permitida a partir de um único neurônio. Entre este neurônio de saída e o conjunto de neurônios de entrada, existe um conjunto de \mathcal{H} neurônios, chamados neurônios *ocultos* (*hidden neurons*), que recebem o fluxo que chega a partir do conjunto \mathcal{I} e envia-o ao neurônio de saída o . A condição para estabilidade desta rede é de $\varrho_j < 1$ para todos os neurônios j . Sem perda de generalidade, todos os valores de entrada para todo $i \in \mathcal{I}$ são reescalados, de tal forma que tenhamos $0 \leq \lambda_i^+ \leq 1$. Assim, uma condição suficiente para a estabilidade é que, para qualquer neurônio de entrada $i \in \mathcal{I}$, $\lambda_i^+ < \mu_i$, e, para qualquer neurônio escondido $h \in \mathcal{H}$, $\mu_h \geq P$ e também $\mu_o \geq P$.

6.3.2 Estudo dos Parâmetros que são Variáveis Aleatórias

Parâmetros que são variáveis aleatórias são um pouco mais complexos de serem tratados do que os outros parâmetros dado que é necessário escolher previamente um modelo apropriado ou uma distribuição que os represente. A parametrização do modelo escolhido deve ser simples o suficiente para manter baixa a complexidade computacional da metodologia proposta.

Para o parâmetro de retardo dos pacotes na rede, utilizamos a ferramenta NetEm [129], que será descrita na Seção 6.4. Na ferramenta NetEm, o retardo é representado por um valor constante, com aumento/diminuição aleatórios. Neste modelo, também é possível definir uma correlação entre duas amostras de retardo sucessivas. Esta correlação aproxima a dependência temporal, limitando o valor da próxima amostra dentro de um intervalo centralizado na amostra atual. Conseqüentemente, o *jitter* no destino é imediatamente definido pelos valores de retardo. Para os testes realizados, utilizamos a distribuição uniforme para o retardo de pacotes na rede, sem a representação da correlação de retardo entre pacotes sucessivos.

Vários *traces* reais de retardo e *jitter* foram coletados na *Internet*, usando o módulo de Engenharia de Tráfego da ferramenta Tangram-II ([34],[36]). Estes foram coletados entre a *University of Massachussets* (UMASS) e o LAND/UFRJ durante vários dias no mês de Março de 2005, em três horários diferentes. Estes *traces* foram utilizados para obter os

6.3 A Metodologia PSQA para Avaliação de Qualidade

parâmetros da distribuição de retardo (média, variância) e para a validação da distribuição que utilizamos no emulador de redes. Os parâmetros de retardo e *jitter* obtidos a partir do NetEm mostraram-se estatisticamente similares aos *traces* reais coletados na *Internet*, considerando as estatísticas de média e variância.

Dois outros parâmetros que são variáveis aleatórias são o percentual de pacotes perdidos e o tamanho da rajada de perdas de pacotes. Diversos modelos matemáticos podem ser encontrados na literatura, cujo o objetivo é representar o processo de perda de pacotes na *Internet* ([2],[11],[127]).

Muitos autores apresentam modelos relativamente simples, como cadeias de Markov com dois estados ([2],[12],[127]) para modelar o processo de perda de pacotes. Para modelar este processo, escolhemos um modelo de *Gilbert* simples, que é uma cadeia de Markov discreta com dois graus de liberdade. A escolha deste caso particular de modelo de *Gilbert* ao invés do modelo original de *Gilbert* [60], baseia-se no fato de que o caso particular requer somente dois parâmetros, ao invés de três, simplificando a computação e o processo de parametrização a serem realizados [101]. Apesar da diminuição dos parâmetros a serem estimados, o caso particular do modelo de *Gilbert* provê boa aproximação para o processo de perda de pacotes.

No modelo de *Gilbert*, o canal de dados possui dois estados, um no qual a transmissão é perfeita e um outro em que erros ocorrem com probabilidade d . A Figura 6.4 mostra a cadeia de Markov do modelo de *Gilbert*. No modelo simplificado, todos os pacotes são perdidos no estado 1, ao invés de considerar que neste estado exista uma probabilidade d de perda. Este modelo, mesmo simplificado, permite capturar a rajada de perda dos pacotes na rede. A Figura 6.5 mostra a cadeia simplificada.

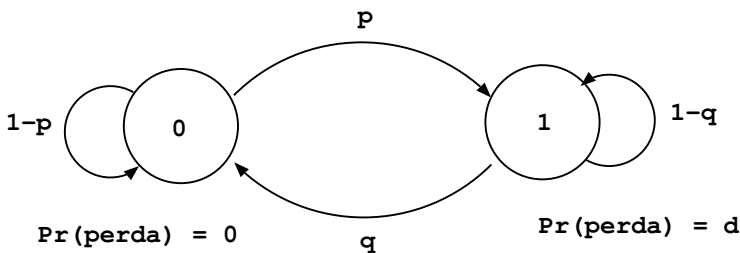


Figura 6.4: Modelo de *Gilbert*. Quando o processo está no estado 0, a transmissão é sem erros. No estado 1, erros ocorrem com probabilidade d .

O modelo de *Gilbert* foi parametrizado usando duas medidas obtidas de *traces* reais. A primeira delas é o tamanho médio da rajada de perdas (MLBS) e a outra a probabilidade de perda de um pacote.

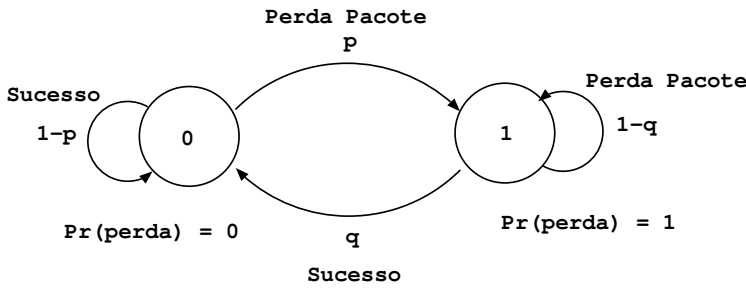


Figura 6.5: Modelo simplificado de *Gilbert*. Quando o processo está no estado 0, a transmissão é sem erros. No estado 1, todos os pacotes são perdidos.

Considere a LBS a variável aleatória que representa o tamanho da rajada de perda. No modelo de *Gilbert* temos que:

$$P[LBS = k] = (1 - q)^{k-1}q, \quad k \geq 1, \tag{6.3}$$

e, portanto, $MLBS = \frac{1}{q}$.

De forma similar, a probabilidade de perda de um pacote no modelo de *Gilbert* é dada por:

$$PER = \pi_0 p + \pi_1 (1 - q) \tag{6.4}$$

onde (π_0, π_1) é a distribuição, em estado estacionário, do processo \mathcal{X} . Temos que $\pi_0 = q(p + q)^{-1}$ e $\pi_1 = p(p + q)^{-1}$. Temos então:

$$PER = \frac{q}{p + q}p + \frac{p}{p + q}(1 - q) = \frac{p}{p + q} = \pi_1.$$

Dado que $q = MLBS^{-1}$, temos

$$p = q \frac{PER}{1 - PER} = \frac{1}{MLBS} \frac{PER}{1 - PER}.$$

A partir destas relações, podemos parametrizar o modelo, através de *traces* reais. Assim, considerando os valores PER_m e $MLBS_m$, retirados dos *traces*, teremos:

$$p = \frac{1}{MLBS_m} \frac{PER_m}{1 - PER_m}, \quad q = \frac{1}{MLBS_m},$$

e o modelo de perda está completamente caracterizado.

Os parâmetros PER_m e $MLBS_m$ foram estimados usando *traces reais* de perda de pacotes coletados por Moon *et al* [140]. Para a validação do modelo de *Gilbert*, foram gerados *traces* a partir do modelo e foram comparados o primeiro e segundo momentos com as mesmas estatísticas calculadas para o *trace* real. O processo de perda obtido pelo modelo proposto mostrou-se estatisticamente similar quando comparado com os *traces* reais coletados.

6.4 Descrição dos Experimentos Realizados

Esta seção destina-se a apresentar o ambiente dos testes subjetivos que fazem parte da primeira etapa da metodologia PSQA para a avaliação da Qualidade de Serviço percebida pelo usuário final de uma aplicação de voz interativa na *Internet*. Descreveremos a ferramenta de voz escolhida para a realização dos experimentos e adicionalmente apresentaremos as principais características e motivações da escolha do codificador Speex [135].

6.4.1 Ferramenta Interativa de Voz - VivaVoz

O VivaVoz é uma ferramenta de código aberto que vem sendo desenvolvida no LAND (UFRJ) desde 1997 [84], cujo o maior objetivo é a implementação e testes de novos algoritmos para melhorar a qualidade de serviço percebida pelo usuário.

Esta ferramenta permite que duas, ou mais pessoas, se comuniquem por voz utilizando a *Internet*, possibilitando a coleta de uma variedade de estatísticas usadas para análise de tráfego de voz, auxiliando no desenvolvimento, testes e aprimoramento de estudos sobre a qualidade da transmissão de voz.

O estado da *Internet* pode variar no tempo. Durante a execução de uma aplicação, o percentual de perda de dados ou o retardo pode alcançar níveis inaceitáveis. Estes problemas devem ser contornados, objetivando oferecer uma melhor qualidade do áudio entregue ao usuário final.

No caso específico da ferramenta VivaVoz, podemos citar dois problemas principais que afetam a qualidade percebida pelo usuário final e as medidas preventivas que são implementadas para evitar ao máximo a degradação da qualidade percebida por este usuário:

1. Recuperação de Pacotes: Para recuperar pacotes perdidos devido ao problema de congestionamento que ocorre na *Internet* [43], o VivaVoz utiliza uma técnica de *Forward Error Correction* (FEC). As técnicas classificadas como FEC têm como objetivo reconstruir pacotes descartados na rede a partir de informações adicionais, denominadas **redundâncias**, enviadas junto com o fluxo de dados pelo emissor. Tais métodos não utilizam retransmissão e por isso apresentam um limite com relação ao número de pacotes que podem ser recuperados. A eficácia dos algoritmos de FEC é medida pela capacidade que estes têm de recuperar perdas de pacotes, sem

6.4 Descrição dos Experimentos Realizados

o aumento significativo do uso da capacidade do canal de dados.

O VivaVoz implementa um algoritmo de FEC baseado em operações de XOR entre grupos de pacotes de diferentes tamanhos [47]. Este novo esquema divide os pacotes entre janelas de uma dado tamanho l , e divide cada janela em s subconjuntos distintos. Uma operação de XOR é realizada entre os pacotes de cada subconjunto, e os resultados são incorporados nos primeiros s pacotes da próxima janela (este tipo de configuração é denotada por $s : l$). Na Figura 6.6 podemos ver um exemplo do esquema em que cada janela tem 6 pacotes, com dois subconjuntos de pacotes (2:6). O esquema 1:2::3:6 é uma combinação de dois esquemas diferentes, resultando em maior redundância e possibilidade de recuperação de pacotes perdidos. A Figura 6.7 mostra a representação deste esquema usado durante os testes subjetivos.

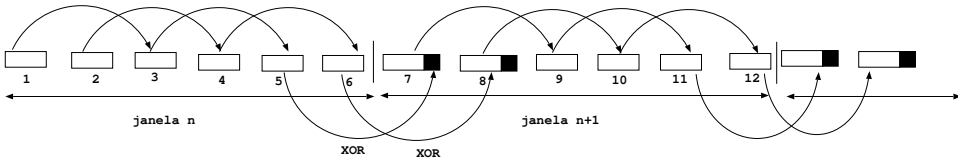


Figura 6.6: Esquema de redundância do tipo 2:6.

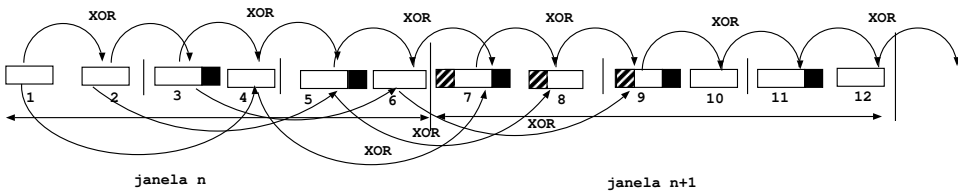


Figura 6.7: Esquema 1:2::3:6 utilizado durante os testes subjetivos realizados. O retângulo em preto representa a redundância do esquema 1:2, enquanto o listrado representa a redundância do esquema 3:6. Os arcos acima dos pacotes representam o esquema 1:2, e os abaixo, o esquema 3:6.

2. Eliminação do *Jitter*: Para eliminar o *jitter*, o VivaVoz utiliza uma técnica amplamente encontrada na literatura, denominada *playout buffer* ([33] e referências neste artigo), onde um *buffer* com capacidade para armazenar N pacotes é acoplado a ferramenta. Quando o VivaVoz recebe os primeiros pacotes de áudio, estes não são imediatamente tocados, mas sim, armazenados no *playout buffer*. Quando este atinge um tamanho pré-definido X , o VivaVoz inicia a apresentação do áudio para o usuário. Todo pacote de áudio que chega é armazenado neste *buffer*. Esta técnica implementada na ferramenta diminui o impacto do problema do *jitter* na qualidade de serviço percebida pelo usuário final, conforme mostraremos na seção 6.5.2.

6.4.2 *Speex*

O *Speex* é um codificador de voz [135], de código aberto, que possui diversos níveis de compressão, com taxas que variam de $2.4Kbps$ até $44Kbps$. Esta alta taxa de compressão permite que este seja utilizado por usuários que dispõem de conexões lentas, como as *dial up*.

As principais motivações para a utilização deste codificador na realização dos testes subjetivos são as seguintes:

1. O *Speex* é um codificador utilizado em projetos de diversas ferramentas de voz sobre a *Internet*, como *LinPhone* [83], *GnomeMeeting* [51] e o projeto de PBX em *software* denominado *Asterisk* [77]. O aumento da importância e do uso deste codificador em diferentes ferramentas resulta em uma nova direção de pesquisa para compreender e estudar o seu comportamento em relação a QoS percebida pelo usuário final de uma aplicação de voz;
2. Estudos anteriores para avaliação de qualidade de aplicações de voz utilizando a *Internet* mostram o comportamento dos codificadores PCM e GSM ([101],[80],[92]). No entanto, existem poucas avaliações na literatura em relação ao codificador *Speex*;
3. Tecnicamente, o codificador *Speex* permite transmitir áudio em diversas taxas de *bits*, possibilitando um amplo conjunto de cenários a serem avaliados.

6.4.3 Metodologia e Ambiente dos Experimentos

Avaliar a qualidade de fluxos VoIP interativos é uma tarefa mais complicada que o caso onde queremos avaliar a qualidade de fluxos em uma direção (*one-way*). Além da necessidade de mais parâmetros, como retardo e *jitter*, outros fatores devem ser considerados. Por exemplo, devemos verificar se uma determinada conversa possui alta interatividade ou se um dos participantes a monopolizou. A metodologia utilizada para a realização dos testes subjetivos obedece as recomendações da ITU-T P.920 [73] que lista o tipo de conversa e o grau de interatividade entre os participantes do teste e a P.800 que descreve a estrutura física dos testes [71].

Seis parâmetros que afetam a qualidade percebida pelo usuário final foram considerados. Quatro deles estão relacionados ao estado da rede e os restantes estão relacionados com os esquemas de codificação de voz utilizados. Os parâmetros de rede utilizados

6.4 Descrição dos Experimentos Realizados

Parâmetros	Valores
Percentual de Perda	0%... 60%
Tamanho Médio da Rajada de Perdas	1... 5
Retardo em uma direção	0ms... 600ms
<i>Jitter</i> (em função do retardo)	0%... 40%
Taxa de envio de <i>bits</i> (codificador Speex)	2.4kbps... 24.8kbps
FEC	desligado, 1:2, 1:2::3:6

Tabela 6.3: Parâmetros de rede e de codificação utilizados durante os testes subjetivos.

são: percentual de pacotes perdidos (PP), tamanho médio da rajada de perda de pacotes (MBLS), o valor médio para o retardo em uma direção e o *jitter* (em função do retardo). Relacionados ao processo de codificação, consideramos o esquema de FEC proposto em [47] e a taxa de transmissão de pacotes gerada pelo codificador *Speex*. Somente o codificador *Speex* foi utilizado durante os experimentos [135], dado que este codificador resulta em uma boa qualidade para voz, mesmo com baixas taxas de envio de *bits*, além de possibilitar diversos valores para a taxa de envio. A Tabela 6.3 mostra o conjunto de valores utilizados para os parâmetros considerados. Nesta tabela, o esquema de FEC 1:2 é capaz de corrigir rajadas de perda de pacotes com tamanho 1 e o esquema 1:2::3:6 pode corrigir rajadas de no máximo 4 pacotes perdidos.

Para a realização dos testes é necessário simular uma rede com as mesmas condições de cada uma das configurações consideradas no modelo da rede neural. Com esta finalidade, utilizamos um computador, executando o sistema operacional Linux, com o papel de roteador entre dois computadores que executavam o programa VivaVoz. Uma versão modificada do NetEm [129] foi utilizada para emular as condições de rede de cada uma das configurações. Esta versão modificada do NetEm incorpora o modelo de *Gilbert* para o representar o processo de perda de pacotes. Para o *jitter*, a definição usada pelo NetEm é o valor esperado da diferença entre o retardo do pacote atual e do retardo médio. Seja a variável aleatória D_i que representa o retardo do i -ésimo pacote. Como consideramos valores de *jitter* que são dependentes do retardo médio \bar{D} , normalizamos este valor por \bar{D} . Em outras palavras, se D_i é a variável aleatória que assume valores a partir de um conjunto atual de valores de retardo para os pacotes pertencentes ao fluxo de voz, então o *jitter* é definido como $E[|D_i - \bar{D}|]/\bar{D}$.

Os seis parâmetros considerados geram diversas configurações possíveis a serem posteriormente avaliadas. Escolhemos um conjunto de valores para os parâmetros de tal maneira que tenhamos um número de configurações razoável a serem avaliadas, e que

6.4 Descrição dos Experimentos Realizados

ao mesmo tempo, tenhamos dados suficientes para o treinamento da rede neural. No contexto de testes subjetivos, por razoável entende-se um número de configurações que possam ser avaliadas sem que o cansaço das pessoas que participam dos testes influenciem na avaliação subjetiva da qualidade de serviço.

Para a realização dos testes subjetivos, selecionamos 120 configurações entre todos os valores possíveis. Este número foi determinado pela duração dos testes subjetivos: considerando que cada sessão relacionada a cada configuração tenha duração de, em média, 3 minutos, 120 configurações serão avaliadas em 6 horas de testes. Achemos que, em termos práticos, testes superiores a 6 horas possuíam alta chance da fadiga dos participantes, podendo influenciar a avaliação da qualidade de serviço. Vale ressaltar que as 120 configurações foram divididas em 4 grupos de 30 configurações cada.

Dentre as 120 configurações, 48 correspondem a condições limites, por exemplo com percentual de perda entre 40% – 60% e retardo entre 400ms – 600ms. As demais foram selecionadas seguindo uma tendência na direção do que consideramos estados normais de operação na *Internet*, considerando o conjunto de *traces* obtidos entre a *UMASS* e a *UFRJ*: percentual de perda 1% – 3%, tamanho médio de rajadas de perda 1 – 5 e taxa de envio de *bits* média e alta.

Os experimentos foram realizados paralelamente com 12 participantes no *IRISA/França* e no *LAND/UFRJ*. Cada um dos participantes de um determinado par realizou os testes em uma sala diferente, utilizando *headsets*. A utilização de *headsets* minimiza o problema de eco que ocorre em aplicações de voz. Portanto, podemos desconsiderar este parâmetro no treinamento da rede neural aleatória.

Conforme sugerido nas recomendações *ITU-T P.800* [71] e *P.920* [73], ao início de cada sessão de testes algumas configurações denominadas *warm-up* devem ser realizadas. Desta forma, o participante se familiariza com os testes que serão feitos, podendo analisar configurações com qualidades boa, média ou ruim e estabelecer um critério individual de classificação. As avaliações realizadas nesta etapa inicial não são consideradas no cálculo do *MOS*.

Todos os participantes possuem experiência prévia na utilização de aplicativos de *VoIP*. Para cada configuração, dois tipos de tarefas foram realizadas, objetivando conversas com diferentes níveis de interatividade[73]:

- contar até 20, alternadamente, o mais rápido possível;
- diálogos, baseados na representação de um cenário, na descrição de uma figura ou

em uma conversa livre.

Para cada uma das 120 configurações, as notas fornecidas pelos participantes são agrupadas, uma média é calculada e um tratamento estatístico, conhecido como teste β_2 descrito em [136] e originalmente proposto em [68], é realizado para detectar os participantes não confiáveis, cujos resultados são extremamente diferentes dos resultados dos demais, sugerindo um comportamento não correto ao realizar os testes. A média final obtida para cada configuração considerada é o *Mean Opinion Score*.

6.5 Resultados Obtidos

Nesta seção mostraremos os resultados obtidos a partir da aplicação da metodologia PSQA para avaliação de qualidade de aplicações interativas de voz na *Internet*. Primeiramente, avaliaremos o desempenho da metodologia PSQA proposta. Posteriormente, apresentaremos um conjunto amplo de gráficos que mostram como a variação dos parâmetros afeta a qualidade percebida pelo usuário (MOS).

6.5.1 Desempenho da Metodologia PSQA

Conforme descrito na Seção 6.3, após obtermos os resultados dos testes subjetivos realizados por um conjunto de participantes, devemos treinar várias redes neurais que serão candidatas a estimar o valor do coeficiente MOS. Para o sucesso da aplicação da metodologia, é essencial que a estimativa retornada pela rede, represente, da melhor maneira possível, o comportamento de um usuário final, caso este estivesse realizando a avaliação de qualidade.

Seja o conjunto \mathcal{T} , denominado *conjunto de treinamento* utilizado para a definição do vetor de pesos \vec{w} . O total de configurações pertencentes a este conjunto é de 100 configurações. Seja o conjunto \mathcal{V} , formado por um conjunto de 20 configurações, denominado *conjunto de validação*. A utilização do conjunto de \mathcal{V} tem como principal objetivo evitar o problema de *overtraining*, característico de redes neurais artificiais (ANNs). O problema de *overtraining* ocorre quando, após o treinamento, os pesos associados a rede neural estão extremamente correlacionados com as configurações de treinamento, resultando em uma rede neural com pequeno grau de generalização. Nestes casos, os resultados desta rede para configurações não utilizadas anteriormente tornam-se pouco

precisos.

A cada iteração de treinamento a minimização da função apresentada na Equação 6.2 é calculada para o conjunto \mathcal{T} de configurações. A partir deste resultado, o MSE é calculado tanto para o conjunto \mathcal{V} quanto para o conjunto \mathcal{T} , e o treinamento deve ser parado caso exista um superadaptação em relação ao conjunto \mathcal{T} . Obviamente, se esta superadaptação ocorrer e o erro em relação ao conjunto \mathcal{V} estiver alto, deve-se redefinir os conjuntos \mathcal{T} e \mathcal{V} e novos treinamentos devem ser realizados. No entanto, para todas as arquiteturas utilizadas, o MSE para o conjunto \mathcal{V} atingiu valores dentro do erro estipulado anteriormente a diminuição drástica do MSE para o conjunto \mathcal{T} .

Resultados anteriores na avaliação de fluxos de voz unidirecionais [124] mostram que arquiteturas muito simples, com duas camadas de neurônios (neurônios de entrada e o de saída) resultam em boa estimativa da qualidade da aplicação sendo estudada. Estes resultados são extremamente atrativos, dado que a partir de redes neurais com arquiteturas simplificadas, fórmulas fechadas, em função do conjunto de parâmetros usados, podem ser derivadas para estimar a qualidade percebida.

Com o intuito de verificar se uma arquitetura simplificada de rede neural fornecia boas estimativas para o MOS, repetimos o mesmo procedimento comparando o desempenho entre diferentes níveis de complexidade para a arquitetura da rede neural. Baseado na métrica de MSE, a arquitetura mais simples, com 6 neurônios de entrada e 1 de saída, resultou em um valor duas ordens de grandeza maior que a arquitetura mais complexa. Desta forma, utilizamos uma arquitetura mais complexa, com 6 neurônios de entrada, 13 neurônios ocultos e 1 neurônio de saída. O aumento da complexidade da RNN a ser utilizada é um reflexo do aumento de complexidade da própria aplicação que estamos estudando.

A Figura 6.8 apresenta a comparação do desempenho das duas arquiteturas citadas anteriormente. Considerando o conjunto \mathcal{V} , a arquitetura com 3 camadas resulta em um MSE de 0.0025. Em contrapartida, a arquitetura com duas camadas resulta em um MSE de 0.3928.

Apesar de optarmos pela utilização da arquitetura mais complexa para obtenção dos resultados deste trabalho, ressaltamos que a arquitetura simplificada também captura o mesmo comportamento da arquitetura mais complexa, conforme pode ser visto na Figura 6.9. Podemos notar que para ambos os gráficos, a mesma conclusão pode ser evidenciada: a qualidade da aplicação torna-se insensível ao retardo, quando ocorre um alto percentual de perda de pacotes. Desta forma, caso o objetivo seja capturar o comportamento, a

6.5 Resultados Obtidos

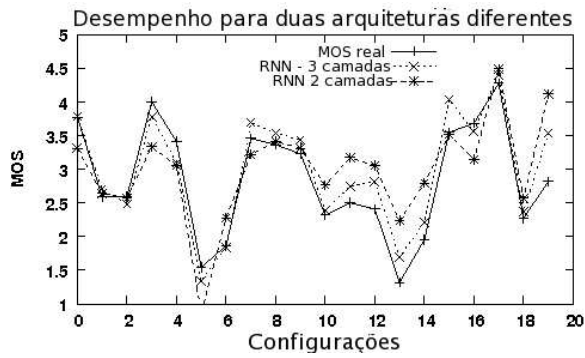


Figura 6.8: MOS real, obtido a partir dos testes subjetivos e MOS estimado usando duas arquiteturas diferentes para a rede neural.

arquitetura simplificada pode ser escolhida e uma fórmula fechada para a obtenção da QoS, semelhante a apresentada em [124], pode ser utilizada.

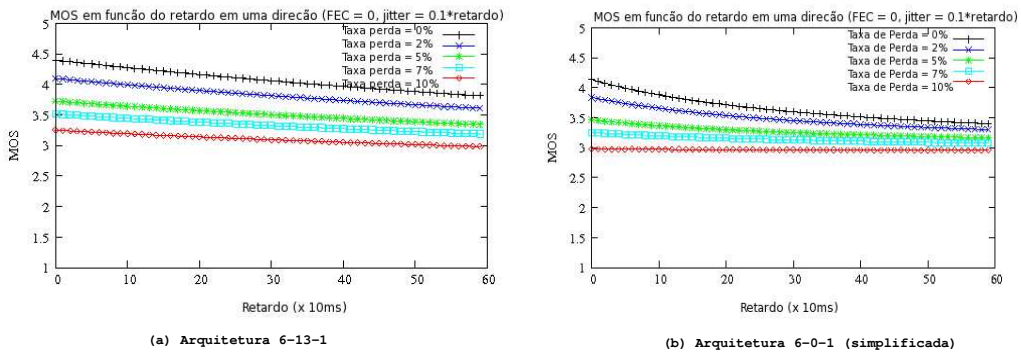


Figura 6.9: MOS em função do retardo para as arquiteturas estudadas.

Uma outra questão relacionada com o desempenho da rede neural, é o comportamento para configurações extremamente diferentes das encontradas no conjunto de treinamento ou validação. Em [136], esta questão foi amplamente discutida para o caso de aplicações de voz sem interatividade (*audio streaming*). No caso da aplicação estudada neste capítulo, consideramos um amplo conjunto de valores para os parâmetros relacionados com a rede, com casos extremos onde a perda é de 60% e delay de 600 ms. Desta forma, considerando o *Speex*, a rede neural pode capturar o comportamento do usuário de uma aplicação de voz de maneira eficiente para o conjunto de dados considerados na Tabela 6.3.

6.5.2 Estudo da Influência dos Parâmetros na Avaliação de Qualidade (MOS)

Nesta seção apresentamos os resultados que descrevem a influência dos parâmetros de codificação e da rede na qualidade, da aplicação de VoIP interativa, percebida pelo usuário final, através das estimativas fornecidas pela metodologia PSQA. Para estas análises, selecionamos diferentes combinações entre estes parâmetros, propiciando um amplo estudo do comportamento do MOS para as condições da rede e perfil de usuários conforme descrito na Seção 6.4.

Percentual de Perda, Retardo e esquema de FEC

Inicialmente avaliaremos os valores obtidos para o MOS variando o percentual de perda e o retardo, considerando a presença ou não de FEC.

As Figuras 6.10 e 6.11 mostram a variação de qualidade em função do retardo para cinco valores diferentes de percentual de perda, com a presença ou não do esquema de FEC, respectivamente. Podemos observar que quando o percentual é de 0%, o valor do MOS diminui em aproximadamente 0.6 pontos quando o retardo, em *UMA* direção, varia de *0ms* a *600ms*. Todavia, quando o percentual de perda aumenta, o impacto na qualidade percebida pelo usuário, em função do retardo, diminui consideravelmente. A qualidade da voz torna-se insensível em relação ao retardo, no cenário em que na rede ocorrem altos percentuais de perda. Quando o percentual de perda é maior que 7% ou 10%, o impacto do retardo na qualidade percebida pelo usuário é praticamente nulo. Estes resultados mostram que o percentual de perda possui, expressivamente, um maior impacto na qualidade do que o impacto produzido pelo retardo. As figuras também mostram que o uso de esquemas de FEC resultam em aumento significativo na qualidade.

A Figura 6.12 mostra a influência gerada pelo percentual de perda, para diversos valores de retardo. Podemos observar que a diferença na qualidade para diferentes valores de retardo diminui com o aumento do percentual de perda. Este resultado reafirma a suspeita de que o parâmetro percentual de perda possui uma maior influência na qualidade, para o perfil de participantes dos testes subjetivos, ou seja, usuários assíduos de aplicações de voz sobre IP.

A Figura 6.13 reafirma os resultados apresentados previamente, mostrando como a qualidade se comporta quando o percentual de perda e o retardo variam simultaneamente.

6.5 Resultados Obtidos

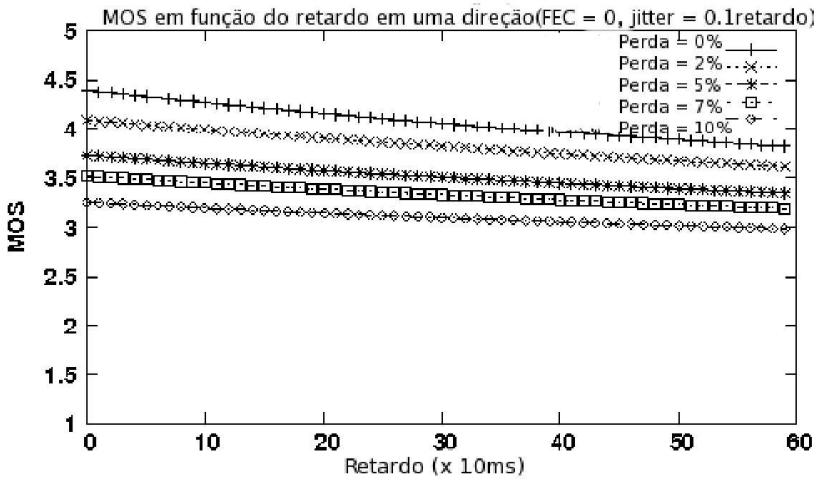


Figura 6.10: Qualidade percebida pelo usuário como função do retardo, para algumas taxas de perda (sem esquema de FEC). O valor do jitter é de 10% do retardo.

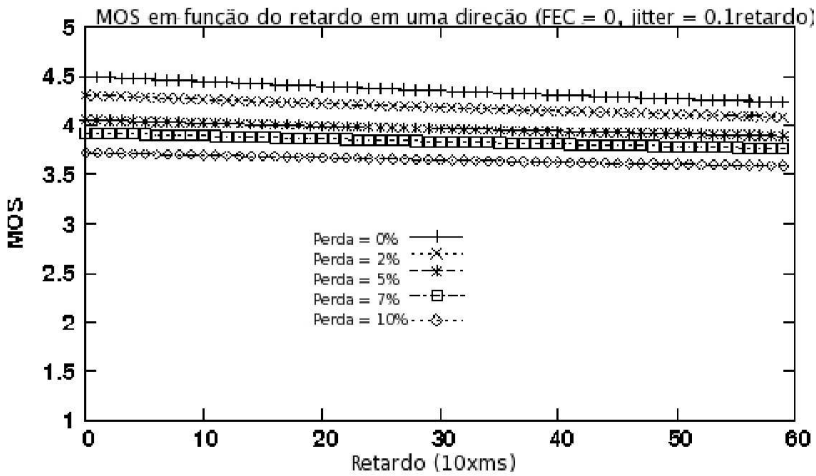


Figura 6.11: Qualidade percebida pelo usuário como função do retardo, para algumas taxas de perda (com esquema de FEC - 1:2::3:6). O valor do jitter é de 10% do retardo.

O impacto gerado por grandes valores de retardo quando perdas não ocorrem é similar ao caso quando temos 5% de percentual de perda e pequenos valores de retardo. Adicionalmente, quando o percentual de perda é maior que 10%, o impacto em função do retardo diminui significativamente.

O esquema de FEC usado desempenha um papel importante na qualidade percebida pelo usuário. Seja a seguinte definição de FEC: FEC = 0, significa que não existem mecanismos de recuperação de pacotes; FEC = 1, significa o tipo 1:2; e o FEC = 2, o tipo 1:2:3:6. A utilização destes esquemas, proporciona qualidades a níveis aceitáveis, mesmo em cenários com grande valores de taxas de perda. Por exemplo, na Figura 6.14

6.5 Resultados Obtidos

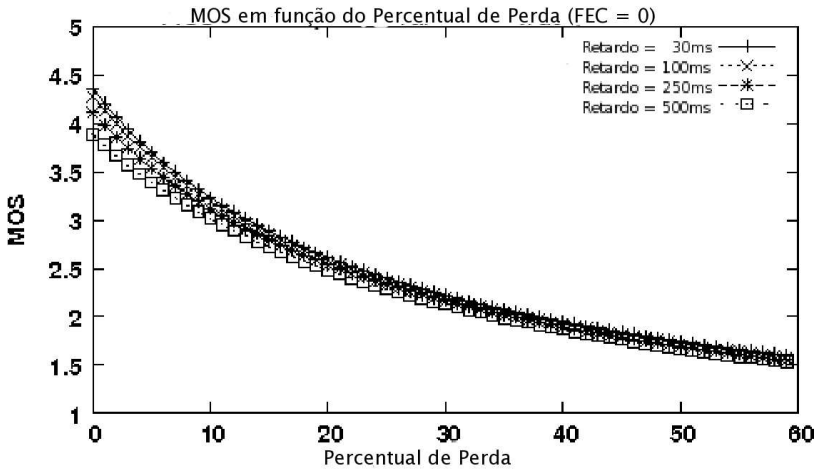


Figura 6.12: Qualidade percebida pelo usuário em função da taxa de perda, para alguns valores de retardo (sem esquema de FEC).

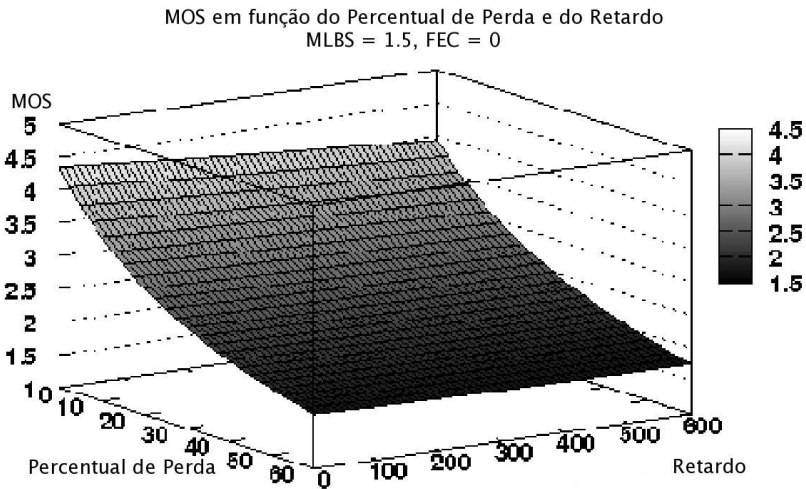


Figura 6.13: Qualidade percebida pelo usuário em função do percentual de perda e do retardo, para cenários sem FEC.

podemos observar que quando um esquema de FEC é utilizado, a qualidade permanece aceitável para taxas de perdas de até 22%. Em contrapartida, para cenários sem a utilização de esquemas de FEC, a qualidade é aceitável para taxas de perda de até 10%. O uso do esquema de FEC proposto em [47] gera bons resultados, mesmo quando as condições na rede estão extremamente degradadas.

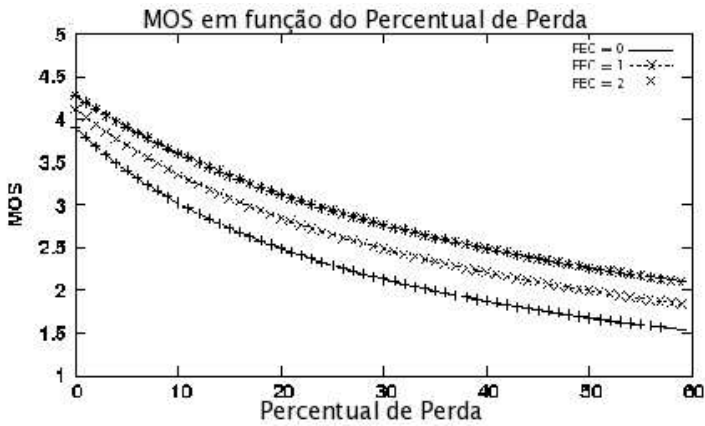


Figura 6.14: Qualidade percebida pelo usuário em função do percentual de perda, para três diferentes esquemas de FEC [47], retardo igual a $200ms$, jitter = $0.1 \times$ retardo e taxa *bits* igual a 14.2.

Percentual de Perda, Tamanho Médio da Rajada de Perda

Nesta seção avaliamos o comportamento do MOS em função do percentual de perda e do tamanho médio da rajada de perda (MLBS). Para os valores de MLBS considerados, a qualidade percebida pelo usuário final é praticamente independente do valor deste parâmetro. Pequenas variações na qualidade ocorreram, mas elas foram imperceptíveis para grande parte dos usuários.

Os resultados obtidos são diferentes dos resultados prévios para o caso de fluxos de voz sem interatividade. Uma explicação plausível para esta independência na qualidade em relação ao parâmetro de MLBS reside no bom desempenho do mecanismo de cancelamento de perda do codificador *Speex*. Podemos também notar que, para um mesmo valor de percentual de perda, a medida que o valor de MLBS aumenta, o usuário irá observar rajadas de perda mais longas, no entanto, em menor quantidade. Provavelmente, a diminuição do número de rajadas de perda (juntamente com o algoritmo implementado no *Speex*) suaviza o impacto na qualidade gerado pelo aumento do valor no parâmetro MLBS nos cenários estudados. A Figura 6.15 mostra os resultados obtidos com retardos no valor de $100ms$ e com $10ms$ para o parâmetro *jitter*.

Jitter

O próximo estudo realizado foi o impacto de diferentes valores do parâmetro *jitter* na qualidade percebida pelo usuário final. Consideramos o *jitter* como uma fração do

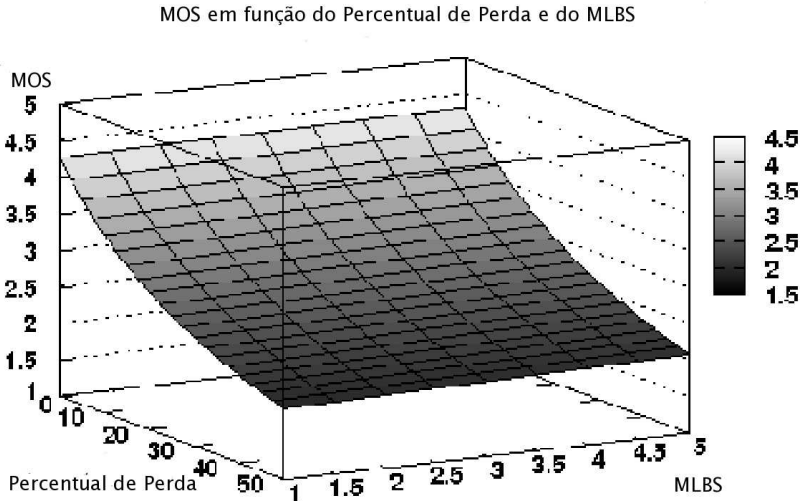


Figura 6.15: Qualidade percebida pelo usuário em função do percentual de perda e de valores para o MLBS, sem esquema de FEC.

retardo, conforme mencionado na Seção 6.4.

Similarmente ao observado para o parâmetro de retardo, o impacto do parâmetro *jitter* na qualidade para conversa interativa não é significativo. Para pequenos valores de percentual de perda e para altos valores de retardo, alguma variação na qualidade ocorre, mas não é perceptível para a maioria dos usuários. Adicionalmente, a ferramenta *VivaVoz* possui um mecanismo de *playout buffer*, que reduz o impacto do parâmetro *jitter* na qualidade, o que também explica os resultados. As Figuras 6.16 e 6.17 mostram a variação da qualidade como função do retardo e do *jitter*, para taxas de perda de 0% e 5%, respectivamente.

Taxa de Envio de *Bits*

Consideramos 8 diferentes taxas de envio de *bits*, com valores entre 2.4kbps a 24.8kbps. Como esperado, a qualidade percebida varia significativamente com a taxa de *bits* utilizada. Todavia, níveis de qualidade aceitáveis para todos os valores de taxa de *bits* acontecem para valores de percentual de perda de até 5%, quando o esquema de FEC é usado. Altas taxas de *bits* oferecem qualidade aceitável quando o percentual de perda é menor que 20%. Podemos notar que um bom compromisso entre qualidade e economia de banda ocorre para taxa de *bits* de 11.2 ou 14.2 kbps.

As Figuras 6.18 e 6.19 mostram a qualidade percebida pelo usuário em função da

6.5 Resultados Obtidos

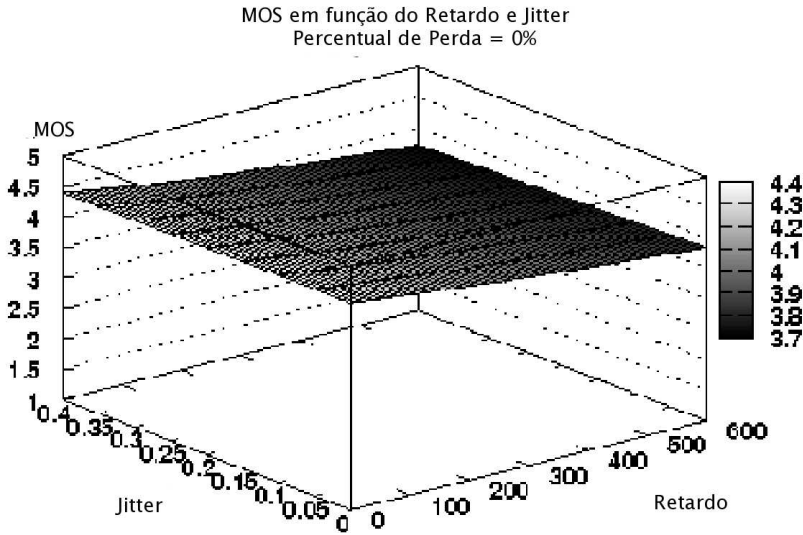


Figura 6.16: Qualidade percebida pelo usuário em função do retardo e do *jitter*, para percentual de perda de 0%.

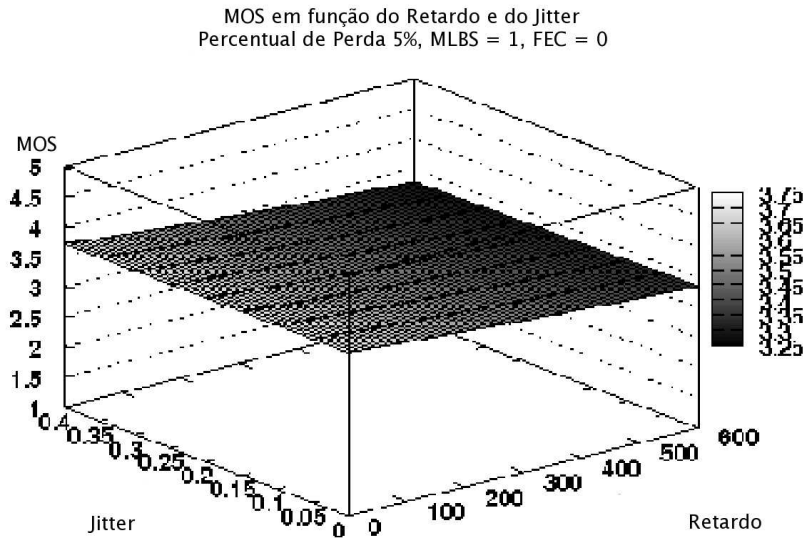


Figura 6.17: Qualidade percebida pelo usuário em função do retardo e do *jitter*, para percentual de perda de 5%.

taxa de envio de *bits*, com e sem esquema de FEC, respectivamente. Para o caso onde o esquema de FEC não é utilizado, a taxa de *bits* enviados dever ser de no mínimo 6 kbps, objetivando uma qualidade aceitável para um percentual de perda em torno de 5%.

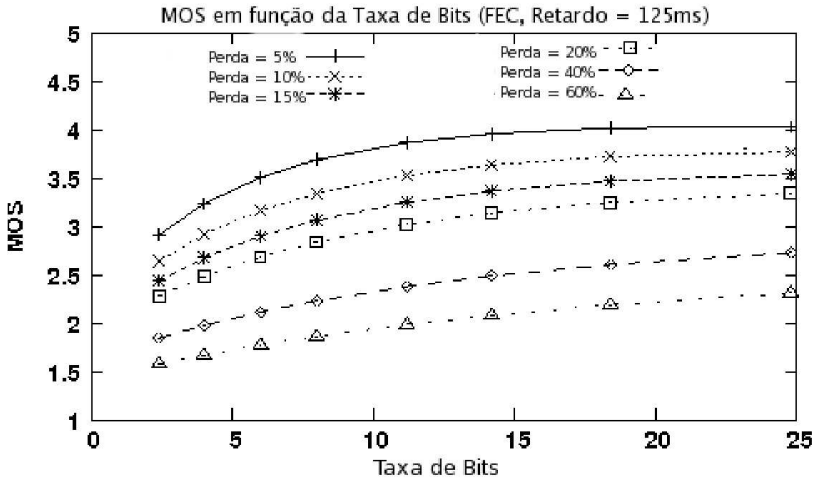


Figura 6.18: Qualidade percebida em função da taxa de *bits*, para diversos valores de percentual de perda, com utilização de esquema de FEC (1:2:3:6). O retardo considerado foi de 125 ms.

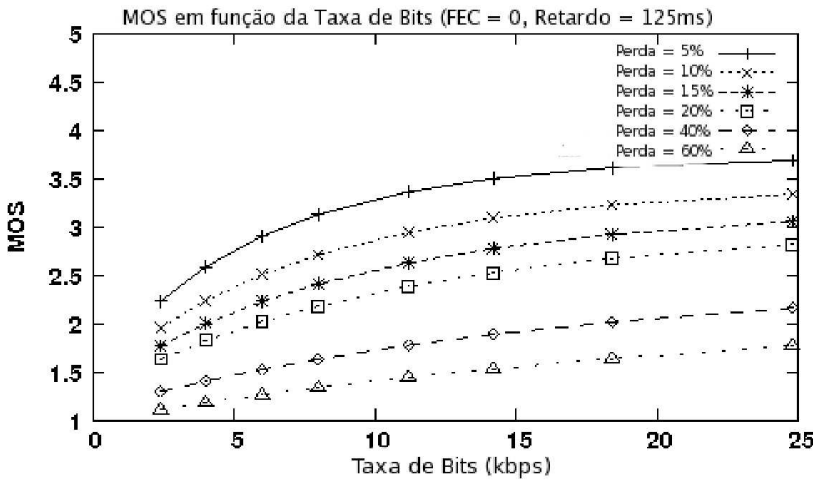


Figura 6.19: Qualidade percebida em função da taxa de *bits*, para diversos valores de percentual de perda, sem esquema de FEC.

6.6 Sumário das Contribuições

As principais contribuições referentes a este capítulo podem ser resumidas da seguinte forma:

1. Uma análise detalhada da qualidade percebida pelo usuário final de uma aplicação de VoIP interativa foi apresentada, com o estudo da influência de seis parâmetros nesta qualidade. Os parâmetros relacionados com a camada de aplicação foram a taxa de envio de *bits* e o esquema de FEC utilizado. Na camada de rede, os

parâmetros foram o percentual de perda, o tamanho médio da rajada de perdas, o retardo em uma direção e o *jitter*;

2. Nos demais trabalhos encontrados na literatura, somente o comportamento dos codificadores de áudio PCM e GSM, como fatores que influenciam a QoS percebida pelo usuário, foi considerado. No trabalho apresentado neste capítulo, realizamos o estudo do comportamento do codificador *Speex*, considerando as diversas taxas de *bits* deste codificador;
3. A partir dos testes subjetivos e da utilização da metodologia PSQA, podemos constatar que para o perfil dos participantes dos testes e para o ambiente utilizado, o fator que mais influencia negativamente a percepção os usuários de uma aplicação interativa, quando o retardo e a perda são considerados, é a perda. Para o perfil dos usuários participantes dos testes, retardos de até aproximadamente 300ms são aceitáveis, fornecendo novos limites para os considerados na recomendação ITU-T G.114 [70]. Estes resultados reforçam os questionamentos apresentados em [74], motivando maiores estudos, considerando outros codificadores e múltiplos participantes em uma sessão interativa;
4. A metodologia PSQA, anteriormente aplicada somente aos casos de VoIP em uma direção, foi utilizada para analisar como diversos parâmetros podem influenciar na QoS percebida por um usuário de uma aplicação de voz utilizando a *Internet*. Através da metodologia PSQA, apresentamos um conjunto de gráficos que mostram como diversos parâmetros influenciam a QoS oferecida, pela *Internet*, ao usuário de uma aplicação de voz. Os resultados aqui apresentados oferecem informações importantes para que mecanismos de controle possam ser implementados para garantir a QoS mínima a um usuário final de uma aplicação VoIP interativa.

Capítulo 7

Conclusões e Trabalhos Futuros

Modelos Markovianos têm sido amplamente utilizados para descrever sistemas e obter diversas medidas de interesse. O principal enfoque desta tese foi propor um conjunto de métodos de solução para modelos Markovianos com recompensas. As aplicações destes modelos são na área de redes de computadores e sistemas computacionais em geral.

No Capítulo 3 apresentamos uma nova abordagem para a solução de modelos estocásticos de filas de fluido, onde a distribuição complementar do conteúdo de informação na fila é obtida através da solução de um processo do tipo *Quasi-Birth-Death*. A modelagem proposta é baseada em recompensas de impulso e gera um processo QBD que possui uma estrutura especial, onde submatrizes na diagonal são zeradas. Foi proposto um algoritmo de eliminação de estados, denominado *Algoritmo Bloco Diagonal Zero*, visando a solução eficiente do QBD. Este algoritmo possui menor custo computacional para resolver o QBD obtido quando comparado com outros algoritmos da literatura. A principal vantagem em relação ao método proposto em [3] e [4] é que não é necessário o cálculo de exponenciais de matrizes e, adicionalmente, o fato do algoritmo de eliminação depender somente do número total de níveis modelados, em contrapartida aos métodos para obter a matriz \mathbf{R} , que são dependentes dos autovalores da matriz \mathbf{G} (conforme mostrado no Capítulo 3). O método proposto é uma aproximação do resultado exato da medida $P[CR > x] = \lim_{t \rightarrow \infty} P[CR(t) > x]$. Um conjunto de exemplos avaliaram a acurácia do método. A metodologia apresentada é simples e pôde ser estendida para o caso de duas filas de fluido e um único servidor, com a política de serviço *Generalized Processor Sharing*.

O principal objetivo do Capítulo 4 foi apresentar um método aproximado, para estimativa dos limites inferior e superior do valor esperado da recompensa acumulada para

sistemas com estados absorventes (ECRA) e com cardinalidade da ordem de milhões de estados. Baseado nas teorias de complementação estocástica e de *path-based*, os resultados obtidos mostraram que os limites inferior e superior do erro para a medida são estreitos, principalmente para sistemas que passam a maior parte do tempo, antes da absorção, em poucos estados. A complexidade computacional para a obtenção da medida de interesse supracitada é extremamente menor quando comparada aos métodos que utilizam todo o espaço de estados modelado. O conjunto de medidas de interesse que podem ser obtidas por este método é bem amplo, variando de medidas importantes de confiabilidade, como o *Mean Time to Failure*, a medidas de desempenho, como a média do tamanho de uma fila antes da ocorrência de um evento, por exemplo um estado sinalizando congestionamento, etc. Como trabalhos futuros relacionados ao Capítulo 4 podemos citar a influência, nos limites estabelecidos, da utilização de outros algoritmos de busca para determinar o complemento estocástico e o estudo sobre a possibilidade da utilização dos métodos citados em [104] e [91] para a medida ECRA e posterior comparação com o método proposto.

Quanto ao Capítulo 5, obtivemos uma equação diferencial que descreve o comportamento da recompensa de fluido acumulada em filas finita e infinita quando a transmissão de fluido é representada por um modelo de Markov oculto. Diferentemente dos demais métodos encontrados na literatura, que calculam medidas de interesse de modelos que associam uma única recompensa em cada estado, a equação obtida leva em consideração a existência de várias recompensas associadas a cada um dos estados. Apresentamos um algoritmo iterativo que fornece um resultado aproximado para a medida de interesse $P[CR \leq x] = \lim_{t \rightarrow \infty} P[CR(t) \leq x]$. O algoritmo proposto possui como maior vantagem a redução da ordem das matrizes a terem os autovalores e autovetores calculados, bem como do sistema de equações que calculam as constantes de integração relacionadas as condições de contorno tanto para fila finita, quanto para infinita. Como trabalhos futuros, podemos citar a formalização da convergência do algoritmo e da acurácia da solução aproximada. Adicionalmente, podemos estudar outros métodos de solução da equação proposta.

No Capítulo 6 apresentamos a metodologia PSQA (*Pseudo-Subjective Quality Assessment*) baseada na combinação de testes subjetivos com uma ferramenta estatística de aprendizado, chamada *Random Neural Network* (RNN) [53] que possibilita estimativas de qualidade, utilizando parâmetros da aplicação e medidos na própria rede. Com esta metodologia, os testes subjetivos são realizados uma única vez, fornecendo dados necessários para o treinamento da rede neural. Por conseguinte, futuras avaliações em tempo real utilizando esta rede neural poderão ser realizadas, “imitando” o comporta-

mento de uma pessoa, caso esta estivesse avaliando a QoS da aplicação em curso. Como principais contribuições deste capítulo, podemos citar a análise detalhada da qualidade percebida pelo usuário final de uma aplicação VoIP interativa, considerando a influência conjunta de vários parâmetros, como perda, retardo, *jitter*, FEC e taxa de *bits* enviada pelo codificador *Speex*.

A partir dos testes subjetivos e da metodologia PSQA, constatamos que para o perfil dos participantes dos testes e para o ambiente utilizado, o fator que mais influencia negativamente a percepção dos usuários de uma aplicação interativa, quando o retardo e a perda são considerados, é a perda. Os resultados apresentados fornecem novos limites para o valor do retardo considerado na ITU-T G.114 [70]. Direções futuras de pesquisa incluem a extensão deste trabalho seguindo a mesma metodologia apresentada em [124], onde a metodologia tradicional de análise de desempenho é combinada a metodologia PSQA, considerando aplicações de VoIP interativas. Estes resultados servirão na parametrização de modelos cuja solução poderá ser obtida utilizando os métodos propostos nesta tese.

Referências Bibliográficas

- [1] ADVE, V., AGBARIA, A., HILTUNEN, M., IYER, R., JOSHI, K., KALBAREZYK, Z., LEFEVER, R., PLANTE, R., SANDERS, W. H., E SCHLICHTING, R. D. A Compiler-Enabled Model- and Measurement-Driven Adaptation Environment for Dependability and Performance. In *Next Generation Software (NGS) Workshop at the International Parallel & Distributed Processing Symposium (IPDPS)* (2005).
- [2] AHLGREN, B., ANDERSSON, A., HAGSAND, O., E MARSH, I. Dimensioning Links for IP Telephony. Relatório Técnico T2000-09, Swedish Institute of Computer Science (SICS), 2000.
- [3] AHN, S., E RAMASWAMI, V. Fluid Flow Models and Queues - A Connection by Stochastic Coupling. *Stochastic Models* 19, 3 (2003), 325-348.
- [4] AHN, S., E RAMASWAMI, V. Steady State Analysis of Finite Fluid Flow Models Using Finit QBDs. *Queueing Systems*, 49 (2005), 223-259.
- [5] ANICK, D., MITRA, D., E SONDHI, M. M. Stochastic Theory of a Data-Handling System with Multiple Sources. *The Bell System Technical Journal* 61, 8 (1982), 1871-1894.
- [6] BASHARIN, G. P., LANGVILLE, A. N., NAUMOV, V. A., GRASSMANN, W., MEYER, C., STEWART, B., E SZYLD, D. The Life and Work of A. A. Markov. *Linear Algebra and its Applications* 386, 34 (2004), 3-26.
- [7] BEERENDS, J. Improvement of the P.861 Perceptual Speech Quality Measure. ITU-T SG12 COM-34E, dezembro de 1997.
- [8] BEERENDS, J., E STEMERDINK, J. A Perceptual Speech Quality Measure Based on a Psychoacoustic Sound Representation. *Journal of Audio Eng. Soc.* 42 (dezembro de 1994), 115-123.

- [9] BLADT, M., MEINI, B., NEUTS, M. F., E SERICOLA, B. Distributions of Reward Functions on Continuous-Time Markov Chains. In *Matrix-Analytic Methods*. 2002, pp. 39–62.
- [10] BOBBIO, A., E TRIVEDI, K. S. An Aggregation Technique for the Transient Analysis of Stiff Markov Chains. *IEEE Trans. Computers C-35*, 9 (1986), pp. 803–814.
- [11] BOLOT, J.-C., E CRÉPIN, H. Analysis and Control of Audio Packet Loss over Packet-Switched Networks. In *LNCS 1018 - Network and Operating System Support for Digital Audio and Video - Fifth International Workshop, NOSSDAV'95* (1995), pp. 163–174.
- [12] BOLOT, J.-C., FOSSE-PARISIS, S., E TOWSLEY, D. Adaptive FEC-Based Error Control for Internet Telephony. In *Proceedings of INFOCOM '99* (New York, NY, USA, março de 1999), pp. 1453–1460.
- [13] BOUISSOU, M., E LEFEBVRE, Y. A Path-Based Algorithm to Evaluate Asymptotic Unavailability for Large Markov Models. In *Annual Reliability and Maintainability Symposium* (2002), pp. 32–39.
- [14] BOYCE, W. E., E DIPRIMA, R. C. *Elementary Differential Equations and Boundary Value Problems*. John Wiley & Sons, 2001.
- [15] CANCELA, H., RUBINO, G., E TUFFIN, B. MTTF Estimation by Monte Carlo Methods using Markov Models. *Monte Carlo Methods and Applications* 8, 4 (2002), 312–341.
- [16] C.D.MEYER. Uncoupling the Perron Eigenvalue Problem. Relatório Técnico 1, Center for Research in Scientific Computation, North Carolina State University, Raleigh, NC, Março 1989.
- [17] COURTOIS, P., E SEMAL, P. Bounds for the Positive Eigenvectors of Nonnegative Matrices and for their Approximation by Decomposition. *J. ACM* 31 (1984), 804–825.
- [18] CRAY, A. Voice Over IP: Hear's How. *Data Communications International* 27, 5 (abril de 1998), 44–59.
- [19] DA SILVA, A. C., LEÃO, R., E DE SOUZA E SILVA, E. An Efficient Approximate Technique for Solving Fluid Models. *Performance Evaluation Review* 32, 2 (2004), 6–8.

- [20] DA SILVA, A. P. C., E RUBINO, G. Computing Bounds of the Expected Cumulative Reward to Absorption. In *Proceedings of 7th International Workshop on Performability of Computer and Communication Systems* (September 2005).
- [21] DA SILVA, A. P. C., E RUBINO, G. Bounding the Mean Cumulated Reward up to Absorption. In *Proceedings of A. A. Markov Anniversary Meeting* (2006), A. N. Langville e W. J. Stewart, Eds., Bosen Books, NC, USA, pp. 169–188.
- [22] DA SILVA SOARES, A., E LATOUCHE, G. Matrix-analytic Methods for Fluid Queues with Finite Buffers. *Performance Evaluation* 63 (2006), 295–314.
- [23] DA SILVA SOARES E GUY LATOUCHE, A. Futher Results on the Similarity between Fluid Queues and QBDS. In *Fourth International Conference in Matrix-Analytic Methods*. (2003), pp. 89–106.
- [24] DALY, D., DEAVOURS, D. D., DOYLE, J. M., WEBSTER, P. G., E SANDERS, W. H. Mobius: An Extensible Tool For Performance and Dependability Modeling. In *Computer Performance Evaluation - Modelling Techniques and Tools - 11th International Conference (TOOLS2000)*, vol. 1786. Springer, Março 2000, pp. 332–336.
- [25] DE F. REINHARDT, K., DE SOUZA E SILVA, E., , E LEÃO, R. M. M. Um Ambiente de Simulação de Fluido para Redes Multimídia. In *21^o Simpósio Brasileiro de Redes de Computadores* (Maio 2003).
- [26] DE SOUZA E SILVA, E., E GAIL, H. Calculating availability and performability measures of repairable computer systems using randomization. *J. ACM* 36, 1 (1989), 171–193.
- [27] DE SOUZA E SILVA, E., E GAIL, H. An Algorithm to Calculate Transient Distributions of Cumulative Rate and Impulse based Reward. *Communications in Statistics - Stochastic Models* 14, 3 (1998), 509–536.
- [28] DE SOUZA E SILVA, E., E GAIL, H. Transient Solutions for Markov Chains. In *Computational Probability*, W. Grassmann, Ed. Kluwer, 2000, pp. 44–79.
- [29] DE SOUZA E SILVA, E., E GAIL, H. R. Calculating Cumulative Operational Time Distributions of Repairable Computer Systems. *IEEE Transactions on Computers* 35, 4 (1986), 322–332.

- [30] DE SOUZA E SILVA, E., E GAIL, H. R. Calculating Availability and Performability Measures of Repairable Computer Systems Using Randomization. *Journal of the Association for Computing Machinery* 36, 1 (1989), 171–193.
- [31] DE SOUZA E SILVA, E., GAIL, H. R., E CAMPOS, R. V. Calculating Transient Distributions of Cumulative Reward. In *ACM Sigmetrics Performance Evaluation Review* (1995), vol. 23, pp. 231–240.
- [32] DE SOUZA E SILVA, E., GAIL, R., E GUEDES, J. Transient Distributions of Cumulative Rate and Impulse Rewards with Applications. In *Proceedings of the 18th IFIP TC7 Conference on System Modelling and Optimization* (1999), Chapman and Hall/CRC Press, pp. 298–306.
- [33] DE SOUZA E SILVA, E., LEAO, R. M., RIBEIRO-NETO, B., E CAMPOS, S. Performance Issues of Multimedia Applications. *Performance Evaluation of Complex Systems: Techniques and Tools* 2459 (2002), 374–405.
- [34] DE SOUZA E SILVA, E., LEÃO, R., E MUNTZ, R. Analytical, Simulation and Measurement Techniques: Experiences with an Integrated Modeling Environment. In *Performance Evaluation - Stories and Perspectives*, G. Kotsis, Ed. Austrian Computer Society, series: OCG Schriftenreihe, 2003, pp. 27–46.
- [35] DE SOUZA E SILVA, E., E LEÃO, R. M. M. The TANGRAM-II Environment. In *Computer Performance Evaluation - Modelling Techniques and Tools - 11th International Conference (TOOLS2000)*, vol. 1786. Springer, Março 2000, pp. 366–369.
- [36] DE SOUZA E SILVA, E., LEÃO, R. M. M., DA SILVA, A. P. C., ROCHA, A. A. A., DUARTE, F. P., SILVEIRAFILHO, F. J., JAIME, G. D. G., E MUNTZ, R. R. Modeling, Analysis, Measurement and Experimentation with the Tangram-II Integrated Environment. ValueTools (2006).
- [37] DE SOUZA E SILVA, E., LEÃO, R. M. M., TRINDADE, M. B., DA SILVA, A. P. C., RIBEIRO, B. F., DUARTE, F. P., E AZEVEDO, J. A. Uma Metodologia de Dimensionamento com QoS usando Cadeias de Markov Ocultas. In *Proceedings XXI Simpósio Brasileiro de Telecomunicações - SBT* (2004).
- [38] DE SOUZA E SILVA, E., M. LEÃO, R. M., E MARIE, R. Efficient Solutions for an Approximation Technique for the Transient Analysis of Markovian Models. Relatório Técnico 11, Institut National de Recherche en Informatique et en Automatique, Centre de Diffusion, INRIA, BP 105 - 78153 Le Chesnay Cedex, France, 1996.

- [39] DE SOUZA E SILVA, E., E MUNTZ, R. R. *Métodos Computacionais de Solução de Cadeias de Markov: Aplicações a Sistemas de Computação e Comunicação*, vol. 1. Rio de Janeiro: Escola de Computação, 1992.
- [40] DE SOUZA E SILVA, E., E OCHOA, P. M. State Space Exploration in Markov Models. *Performance Evaluation* 20 (1992), 152–166.
- [41] DUARTE, F. P., DE SOUZA E SILVA, E., E TOWSLEY, D. An Adaptive FEC Algorithm Using Hidden Markov Chains. *ACM SIGMETRICS Performance Evaluation Review* (2003).
- [42] DUGAN, J. B., E TRIVEDI, K. S. Covarage Modeling for Dependability Analysis of Fault-Tolerant Systems. *IEEE Trans. Computers* 38, 6 (1989), pp. 775–787.
- [43] E KEITH W. ROSS, J. F. K. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison Wesley, 2005.
- [44] E SILVA, E. S., E GAIL, H. R. The Uniformization Method in Performability Analysis. In *Performability Modelling: Techniques and Tools*, G. R. B. R. Haverkort, R. Marie e K. S. Trivedi, Eds. Wiley, 2001, ch. 3, pp. 31–58.
- [45] ELWALID, A. I., E MITRA, D. Fluid Models for the Analysis and Design of Statistical Multiplexing with Loss Priorities on Multiple Classes of Bursty Traffic. In *Infocom* (1992).
- [46] FERNANDES, P. H. L. *Methodes Numeriques pour la Solution de Systemes Markoviens a Grand Espace d'Etats*. PhD thesis, Institut National Polytechnique de Grenoble, INPG, 1998. Em francês.
- [47] FIGUEIREDO, D. R., E DE SOUZA E SILVA, E. Efficient Mechanisms for Recovering Voice Packets in the Internet. In *Globecom'99* (1999).
- [48] FILHO, F. J. S. Previsão de estatísticas de perdas de pacotes usando modelos de Markov ocultos. Tese de Mestrado, COPPE/UFRJ, 2006.
- [49] FILHO, F. S., E DE SOUZA E SILVA, E. Um Método de Previsão de Perdas de Pacotes com Aplicações à Transmissão de Mídia Contínua. In *XXIV Simpósio Brasileiro de Redes de Computadores* (Brasil, Maio 2006).
- [50] FILHO, F. S., WATANABE, E. H., E DE SOUZA E SILVA, E. Adaptive forward error correction for interactive streaming over the Internet. In *Proceedings of the IEEE Globecom'06* (November 2006).

- [51] Ekiga (GnomeMeeting). <http://www.gnomemeeting.org>.
- [52] GAIL, H. R., HANTLER, S., E TAYLOR, B. Non-skip-free M/G/1 and G/M/1 Type Markov Chains. *Adv. in Appl. Probab.* 29 (1997), 733–758.
- [53] GELENBE, E. Random Neural Networks with Negative and Positive Signals and Product Form Solution. *Neural Computation* 1, 4 (1989), 502–511.
- [54] GELENBE, E. Stability of the Random Neural Network Model. In *Proc. of Neural Computation Workshop* (Berlin, West Germany, fevereiro de 1990), pp. 56–68.
- [55] GELENBE, E. Learning in the Recurrent Random Neural Network. *Neural Computation* 5, 1 (1993), 154–511.
- [56] GELENBE, E. G-Networks: New Queueing Models with Additional Control Capabilities. In *Proceedings of the 1995 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems* (Ottawa, Ontario, Canada, 1995), pp. 58–59.
- [57] GELENBE, E., E FOURNEAU, J. Random Neural Networks with multiple classes of Signals. *Neural Computation* 11, 3 (1999), 953–963.
- [58] GELENBE, E., E HUSSAIN, K. Learning in the Multiple Class Random Neural Network. *IEEE Trans. on Neural Networks* 13, 6 (2002), 1257–1267.
- [59] GELENBE, E., MAO, Z.-H., E LI, Y.-D. Function Approximation with Spiked Random Networks. *IEEE Trans. on Neural Networks* 10, 1 (1999), 3–9.
- [60] GILBERT, E. Capacity of a burst-loss channel. *Bell Systems Technical Journal* 5, 39 (setembro de 1960).
- [61] GRASSMANN, W. K., E STANFORD, D. A. Matrix Analytic Methods. In *Computational Probability*, W. Grassmann, Ed. Kluwer, 2000, pp. 153–204.
- [62] HAJEK, B. Birth-Death Processes on Integers with Phases and General Boundaries. *J. Appl. Prob.* 19 (1982), 488–499.
- [63] HALL, T. A. Objective Speech Quality Measures for Internet Telephony. In *Voice over IP (VoIP) Technology, Proceedings of SPIE* (Denver, CO, USA, agosto de 2001), vol. 4522, pp. 128–136.
- [64] HAVERKORT, B. R., MARIE, R., RUBINO, G., E TRIVEDI, K. S. *Performability Modelling: Techniques and Tools*. Wiley, 2001.

- [65] HEIDELBERGER, P., MUPPALA, J. K., E TRIVEDI, K. S. Accelerating Mean Time to Failure Computations. *Performance Evaluation* 28 (1996), pp. 627–645.
- [66] HOOPER, J., E RUSSELL, M. Objective Quality Analysis of a Voice Over Internet Protocol system. *IEEE Electronics Letters* 36, 22 (2000), 1900–1901.
- [67] INCERA, J., MARIE, R., ROSS, D., E RUBINO, G. FluidSim: A Tool to Simulate Fluid Models of High Speed Networks. *Performance Evaluation* 44 (2001), 25–49.
- [68] ITU-T RECOMMENDATION BT.500-10. Methodology for the Subjective Assessment of the quality of Television Pictures. In *International Telecommunication Union* (março de 2000).
- [69] ITU-T RECOMMENDATION G.107. The E-model, a Computational Model for Use in Transmission Planning.
- [70] ITU-T RECOMMENDATION G.114. One-way Transmission Time, 2003.
- [71] ITU-T RECOMMENDATION P.800. Methods for Subjective Determination of Transmission Quality, 1996.
- [72] ITU-T RECOMMENDATION P.862. Perceptual Evaluation of Speech Quality (Pesq), an Objective Method for End-To-End Speech Quality Assessment of Narrowband Telephone Networks and Speech Codecs, 2001.
- [73] ITU-T RECOMMENDATION P.920. Interactive Test Methods for Audiovisual Communications, 2000.
- [74] JAMES, J., BING, C., E GARRISON, L. Implementing VoIP: A Voice Transmission Performance Progress Report. *IEEE Communications Magazine* 42, 7 (julho de 2004).
- [75] JENSEN, A. Markoff chains as an aid in the study of markoff processes. *Skandinavisk Aktuarietidskrift* 36 (1953), 87–91.
- [76] JIANG, W., E SCHULZRINNE, H. Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia Service Quality. In *Proceedings of NOSSDAV* (June 2000).
- [77] Asterisk: The Open Source PBX. <http://www.asterisk.org>.

- [78] KARANDIKAR, R. L., E KULKARNI, V. G. Second-order Fluid Flow Models: Reflected Brownian Motion in a Random Environment. *Operations Research* 43, 1 (1995), 77–88.
- [79] KEMENY, J. G., E SNELL, J. L. *Finite Markov Chains*. Van Nostrand, 1976.
- [80] KITAWAKI, N., E ITOH, K. Pure delay effects on speech quality in telecommunications. *IEEE Journal on selected Areas in Communications* 9, 4 (maio de 1991), 586–593.
- [81] KLEINROCK, L. *Queueing Systems*. Wiley-Interscience Publication, 1975.
- [82] KOSTEN, L. Stochastic Theory of a Multi-Entry Buffer (1). Relatório Técnico 1, Delft Progress Report, 1974.
- [83] Linphone: La Téléphonie sous Linux. <http://www.linphone.org>.
- [84] LAND/COPPE/UFRJ. Vivavoz website. <http://www.land.ufrj.br>.
- [85] LATOUCHE, G. Algorithms for Infinite Markov Chains with Repeating Columns. In *Linear Algebra, Markov Chains and Queueing Models*, C. D. Meyer e R. J. Plemmons, Eds. Springer-Verlag, 1993, pp. 231–265.
- [86] LATOUCHE, G., E RAMASWAMI, V. A Logarithm Reduction Algorithm for Quasi-Birth-Death Processes. *J. Appl. Prob.* 30 (1993), 650–674.
- [87] LATOUCHE, G., E RAMASWAMI, V. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. SIAM, 1999.
- [88] LEÃO, R. M., DE SOUZA E SILVA, E., E DINIZ, M. C. Traffic Engineering using Reward Models. In *Proc. of the International Teletraffic Congress - ITC17* (December 2001), Elsevier, Ed., vol. 4, pp. 1101–1112.
- [89] LIU, B., FIGUEIREDO, D. R., GUO, Y., KUROSE, J., E TOWSLEY, D. A Study of Networks Simulation Efficiency: Fluid Simulation vs. Packet-level Simulation. In *Proceedings of IEEE Infocom* (2001).
- [90] MAHEVAS, S. *Modeles Markoviens de Grande Taille: Calculs de Borne*. PhD thesis, Universite de Rennes I, 1997. Em francês.
- [91] MAHEVAS, S., E RUBINO, G. Bound Computation of Dependability and Performance Measures. *IEEE Trans. Computers* 50 (2001), 399–413.

- [92] MARKOPOULOU, A. P., TOBAGI, F. A., E KARAM, M. J. Assessing the Quality of Voice Communications Over Internet Backbones. *IEEE/ACM Transactions on Networking* 11, 5 (2003).
- [93] MARKOV, A. A. Rasprostranenie Zakona Bol'shish Chiesel na Velichiny, Zavisqaschie Drug ot Druga. *Izvestiya Fiziko-matematicheskogo obschestva pri Kazanskom universitete* 15, 4 (1906), 135–156.
- [94] MENASCHE, D. S., FIGUEIREDO, D. R., E DE SOUZA E SILVA, E. An Evolutionary Game-Theoretic Approach to Congestion Control. *Performance Evaluation* 62 (2005), 295–312.
- [95] MEYER, C. D. Stochastic Complementation, Uncoupling Markov Chains and the Theory of Nearly Reducible Systems. *Siam Review* 31 (1989), 240–272.
- [96] MEYER, J. F., E SANDERS, W. H. Specification and Construction of Performance Models. In *Proceedings of the Second International Workshop on Performance Modeling of Computer and Communication Systems* (1993).
- [97] MIRCEA AGAPIE AND KHOSROW SOHRABY. Algorithmic Solution to Second-order Fluid Flow. In *INFOCOM* (2001), pp. 1261–1270.
- [98] MITRA, D. Stochastic Theory of a Fluid Model of Producers and Consumers Coupled by a Buffer. *Adv. Appl. Prob* 20 (1988), 646–676.
- [99] M.MEO, E SILVA, E. S., E MARSAN, M. A. Efficient Solution for a Class of Markov Chain Models of Telecommunication Systems. *Performance Evaluation* 27&28 (Outubro 1996), 603–625.
- [100] MOGENS BLADT, BEATRICE MEINI, M. F. N., E SERICOLA, B. Distributions of Reward Functions on Continuous-Time Markov Chains. In *Matrix-analytic methods* (2002), G. Latouche e P.Taylor, Eds., World Sci. Publishing, pp. 39–62.
- [101] MOHAMED, S., RUBINO, G., E VARELA, M. Performance Evaluation of real-time speech through a packet network: a random neural networks-based approach. *Performance Evaluation* 57 (2004), 141–161.
- [102] MOHAMED, S. A. *Automatic Evaluation of Real-Time Multimedia Quality: a Neural Network Approach*. PhD thesis, INRIA/IRISA, univ. Rennes I, Rennes, France, jan 2003.

- [103] MOLER, C., E LOAN, C. V. Nineteen Dubious Ways to Compute the Exponential of a Matrix. *Siam Review* 20, 4 (October 1978), 801–836.
- [104] MUNTZ, R. R., DE SOUZA E SILVA, E., E GOYAL, A. Bounding Availability of Repairable Computer Systems. *IEEE Trans. Computers* 38 (1989), 1714–1723.
- [105] NABLI, H. Asymptotic Solution of Stochastic Fluid Models. *Performance Evaluation* 57 (2004), 121–140.
- [106] NABLI, H., E SERICOLA, B. Performability Analysis: a New Algorithm. *IEEE Trans. Comput.* 45 (1996), 491–494.
- [107] NEUTS, M. F. Markov Chains with Applications in Queueing Theory, wich have a Matrix-Geometric Invariant Vector. *Adv. in Appl. Probability* 10 (1978), 445–454.
- [108] NEUTS, M. F. *Matrix-Geometric Solutions in Stochastic Models. An Algorithmic Approach*. The Johns Hopkins University Press, 1981.
- [109] NEUTS, M. F. Explicit Steady-State Solutions to Some Elementary Queueing Systems. *Oper. Research* 30 (1982), 480–489.
- [110] NEUTS, M. F. The Caudal Characteristic Curve of Queues. *Advances in Applied Probability* 18 (1986).
- [111] NEUTS, M. F. *Structured Stochastic Matrices of M/G/I Type and Their Applications*. Marcel Dekker, 1989.
- [112] NLANR. Passive Measurement and Analysis (PMA), 2005. <http://pma.nlanr.net/>.
- [113] OPTICOM. Opera Quality Assessment Suite.
- [114] P.563, I.-T. R. Single Ended Method for Objective Speech Quality Assessment in Narrow-Band Telephony Applications, 2004.
- [115] PAREKH, A. K., E GALLAGER, R. G. A Generalized Processor-Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case. *IEEE/ACM Trans. on Networking* 3, 1 (June 1993).
- [116] POURRET, O. *The Slow-Fast Approximation for Markov Processes in Reliability*. PhD thesis, University of Paris-Sud Orsay, 1998. Em francês.
- [117] PSYTECHNICS. PSM – Psytechnics Speech Monitor.

- [118] QURESHI, M. A., E SANDERS, W. H. A New Methodology for Calculating Distributions of Reward Accumulated During a Finite Interval. In *Proceedings of the 26th Annual International Symposium on Fault-Tolerant Computing* (1996), pp. 116–125.
- [119] RABINER, L. R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceeding of the IEEE* 77, 2 (1989).
- [120] RAMASWAMI, V. Matrix Analytic Methods for Stochastic Fluid Flows. In *Proceedings ITC 16* (1999), pp. 1019–1030.
- [121] RISKÁ, A., E SMIRNI, E. M/G/1-type processes: A Tutorial. In *Performance Evaluation of Complex Computer Systems; Techniques and Tools*, M. Calzarossa e S. Tucc, Eds. Springer-Verlag, 2002, pp. 36–63.
- [122] RIX, A. W. Perceptual Speech Quality Assessment - A Review. In *In Proceedings of ICASSP'04* (May 2004), vol. III, pp. 1056–1059.
- [123] ROSS, S. M. *Introduction to Probability Models*. Academic Press, 1989.
- [124] RUBINO, G., E VARELA, M. A New Approach for the Prediction of end-to-end Performance of Multimedia Streams. In *Proceedings of the First International Conference on Quantitative Evaluation of Systems (QEST'04)* (September 2004).
- [125] SAHNER, R., E TRIVEDI, K. S. Reliability Modeling using SHARPE. *IEEE Trans. Reliability* 36, 2 (1987), pp. 186–193.
- [126] SALAMATIAN, K., E VATON, S. Hidden Markov modeling for network communication channels. In *Proceedings of the ACM SIGMETRICS* (June 2001), pp. 92–101.
- [127] SANNECK, H., CARLE, G., E KOODLI, R. A Framework Model for Packet Loss Metrics Based on Loss Runlengths. In *Proceedings of the SPIA/ACM SIGMM Multimedia Computing and Networking Conference* (San Jose, CA, janeiro de 2000), pp. 177–187.
- [128] SERICOLA, B. Transient Analysis of Stochastic Fluid Models. Relatório Técnico 3152, INRIA, 1997.
- [129] S.HEMMINGER. Netem website. <http://developer.osdl.org/shemminger/netem/>.

- [130] SIMON, H., E ANDO, A. Aggregation of Variables in Dynamic Systems. *Econometrica* 29 (1961), 111–138.
- [131] SKELLY, P., SCHWARTZ, M., E DIXIT, S. A Histogram-Based Model for Video Traffic Behavior in an ATM Multiplexer. *IEEE/ACM Transactions on Networking* 1, 4 (August 1993), 445–459.
- [132] STEWART, W., Ed. *Numerical Solution of Markov Chains*. Marcel Dekker, 1991.
- [133] TRIVEDI, K. S. *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. Prentice-Hall, 2002.
- [134] Vqmon. <http://www.telechemy.com/vqmonep.html>.
- [135] VALIN, J.-M. Speex website. <http://www.speex.org>.
- [136] VARELA, M. *Évaluation Pseudo-Subjective de la Qualité d'un Flux Multimédia et ses Applications au Contrôle*. PhD thesis, Université de Rennes I, 2005. Em inglês.
- [137] VORAN, S. Estimation of Perceived Speech Quality Using Measuring Normalizing Blocks. In *IEEE Workshop on Speech Coding For Telecommunications Proceeding* (Pocono Manor, PA, USA, setembro de 1997), pp. 83–84.
- [138] WEI, W., WANG, B., E TOWSLEY, D. Performability Analysis: a New Algorithm. *Performance Evaluation* 49 (2002), 129–146.
- [139] W.K.GRASSMANN, M.I.TAKSAR, E D.P.HEYMAN. Regenerative Analysis and Steady State Distributions for Markov Chains. *Operation Research* 33 (1985), 1107–1116.
- [140] YAJNIK, M., MOON, S., KUROSE, J., E TOWSLEY, D. Measurement and Modeling of the Temporal Dependence. In *IEEE/Infocom* (1999), pp. 345–352.
- [141] YANG, W. *Enhanced Modified Bark Spectral Distortion (EMBSD): an Objective Speech Quality Measure Based on Audible Distortion and Cognition Model*. PhD thesis, Temple University Graduate Board, maio de 1999.
- [142] YE, J., E LI, S. Q. Folding Algorithm: A Computational Method for Finite QBD Processes with Level- Dependent Transitions. *IEEE Trans. Commun.* 42, 2 (1994), pp. 625–639.