

Um algoritmo Eficiente de Solução de Modelos com Aplicação a Redes Multimídia

Morganna Carmem Diniz^{1*}, Edmundo A. de Souza e Silva^{2†}

¹ morganna@ibmecrj.br

Faculdades IBMEC-RJ

Av. Rio Branco 108, Centro, Rio de Janeiro - RJ

² edmundo@land.ufrj.br

COPPE/Sistemas e Dept. de Computação - UFRJ

Caixa Postal 68.511, Rio de Janeiro - RJ

Resumo

Este trabalho apresenta um novo método de solução aproximado para resolver modelos de sistemas de redes de comunicação de dados que possuem uma determinada estrutura especial comum em vários modelos. Em particular uma classe onde as probabilidades de transição entre estados possuem valores bem distintos. Este método é baseado no algoritmo GTH e mantém as suas principais vantagens, e ainda é capaz de solucionar modelos de sistemas complexos com um número de estados e estrutura matricial que torna a solução pelo GTH inviável em termo de custo computacional. Um limite para o erro introduzido pelo novo algoritmo na solução dos modelos também é obtido. A utilidade do algoritmo proposto é mostrada através de exemplos provenientes de redes multimídia.

Abstract

This work presents a new approximate solution technique to solve models with a special structure that is common in many computer network models. In particular, the class of models we address is that in which the resulting state transition matrix has entries with values that differ in orders of magnitude. This method is based on the GTH algorithm and maintains its main advantages. Furthermore, it is capable of solving complex system models with state space cardinalities and transition matrix structures that would be too costly to solve using GTH. Error bounds are also obtained for the new algorithm. The applicability of the technique is illustrated by solving multimedia system models.

*Morganna Carmem Diniz foi apoiada em parte por bolsas do CNPq e CAPES

†Edmundo de Souza e Silva é apoiado em parte por projetos do CNPq/ProTeM, CNPq, PRONEX e FAPERJ

1 Introdução

A modelagem de sistemas de computação/comunicação e em particular redes de computadores é uma das tarefas mais importantes no processo de análise e desenvolvimento de novas tecnologias. Por este motivo, é fundamental o desenvolvimento de modelos que aproximem o comportamento de sistemas reais e de métodos de solução eficientes que possibilitem a obtenção de medidas de interesse destes sistemas.

A estimativa do desempenho de redes de telecomunicação e dos paradigmas distribuídos que são inerentes a esses sistemas tem sido tradicionalmente obtidos por dois métodos: análise e simulação. A simulação tem sido bastante usada uma vez que é muito fácil emular o comportamento dinâmico de sistemas simples em um computador. Redes modernas de comunicação tem entretanto se tornado bastante difícil de simular devido ao aumento da complexidade dos seus algoritmos e, mais importante ainda, devido a larga diferença de escala de tempo entre eventos do sistema. Sendo assim, o número de eventos a ser simulado para transmitir um número modesto de informação e ainda observar alguns eventos chave pode ser enorme, o que implica num tempo excessivamente grande de simulação. Além disso, a obtenção com precisão, das medidas estatísticas de interesse pode ser extremamente difícil nestes casos.

Os métodos analíticos, por outro lado, são mais difíceis de serem usados do que simulação, uma vez que é crucial a escolha do nível apropriado de detalhe e o formalismo para a representação do comportamento do sistema, e essas escolhas não são em geral triviais. Modelos Markovianos tem sido muito usados, e soluções numéricas destes modelos tem ganho uma atenção especial na literatura uma vez que soluções em forma fechada são raras.

Quando soluções numéricas são utilizadas, o parâmetro principal que afeta a complexidade computacional da solução é a cardinalidade do espaço de estados do modelo. A maioria dos modelos de sistemas reais possui cardinalidade demasiadamente grande, e a única forma de se obter uma solução é explorar estruturas especiais que são comumente presentes nestes sistemas [1, 2]. Os métodos numéricos de solução usados para obter soluções em estado estacionário são divididos em diretos e iterativos [1]. Um método é dito direto quando fornece as probabilidades exatas dos estados após um número finito de passos, e um método é dito iterativo quando fornece uma solução que converge para a solução exata a cada iteração do algoritmo. Neste artigo tratamos de métodos diretos.

Um dos métodos diretos mais importantes na solução de cadeia de Markov é o algoritmo proposto por Grassmann, Taksar e Heyman [3], e conhecido por método GTH. Um problema do algoritmo GTH é o custo computacional elevado para matrizes que não possuem estruturas especiais. Isto se deve ao fato do GTH introduzir modificações na matriz de transição de estados, e portanto estruturas especiais podem ser destruídas durante a execução do método. (Como exemplo simples, uma matriz esparsa pode se tornar densa, aumentando drasticamente o custo computacional de solução.)

Neste trabalho propomos uma aproximação, baseada no algoritmo GTH, de forma a resolver uma classe de modelos cujo custo de solução seria considerado elevado se usado o GTH. A classe de modelos estudada é aquela onde existem transições da matriz de probabilidade de transições de estado com valores muito inferiores a outros da matriz. Além disso essa classe **não** pertence a dos modelos quase que completamente particionáveis, para a qual existe uma teoria própria desenvolvida. Este algoritmo faz uso da teoria de dois outros métodos que se prestam a resolver outra classe de problemas (por exemplo modelos onde as matrizes são quase-aglutináveis). Apresentamos também exemplos de aplicação deste método em modelos proveniente do estudo de redes de comunicação de dados, de forma a mostrar a utilidade do método proposto. É importante ressaltar que o método pode também ser usado como parte da

soluções de uma classe modelos não-Markovianos.

Na seção 2 são apresentados alguns conceitos relacionados a métodos de solução e de aproximação utilizados neste trabalho e o algoritmo GTH é discutido com mais detalhes. Em 3 descrevemos o método proposto. Em 4 mostramos a solução de alguns modelos Markovianos e não Markovianos usando o algoritmo GTH e o algoritmo de aproximação aqui proposto. A seção 5 conclui o artigo.

2 Antecedentes

Esta seção apresenta uma revisão de algoritmos e conceitos relacionados a métodos de solução que serão úteis na próxima seção, uma vez que o método proposto neste trabalho baseia-se em resultados destes algoritmos. Inicialmente descrevemos o algoritmo GTH e o algoritmo proposto em [7] para solução de modelos não markovianos. Em seguida discutimos dois métodos de aproximação encontrados na literatura para a solução de modelos markovianos e que servem de base para o novo algoritmo que propomos.

2.1 Método de Solução GTH

Seja $\mathcal{Z} = \{Z_k : k = 0, 1, \dots\}$ uma cadeia de Markov homogênea de tempo discreto com espaço de estados $\mathcal{S} = \{s_i : i = 1, \dots, M\}$ e matriz de probabilidades de transição \mathbf{P} . Considere que os estados estão divididos em dois grupos. O primeiro grupo (G_1) é formado pelos estados s_1, \dots, s_{M-1} , e o segundo grupo (G_2) é formado apenas pelo estado s_M . Note que a soma das probabilidades estacionárias dos estados s_1, \dots, s_{M-1} corresponde à fração do tempo que o sistema passa em G_1 , enquanto que a probabilidade estacionária de s_M corresponde à fração de tempo que o sistema passa em G_2 . (Como a cadeia é de tempo discreto supomos que cada visita a um estado corresponde a uma unidade de tempo) O algoritmo GTH elimina a cada passo um ou mais estados do modelo. As probabilidades estacionárias dos estados não eliminados correspondem às probabilidades de estado do sistema condicionadas a que este esteja nos estados não eliminados.

Seja $p_{i,j}^{(M)}$ a probabilidade de transição do estado s_i para o estado s_j quando nenhum estado foi ainda eliminado. Portanto, para $i = 1, \dots, M - 1$, $p_{i,M}^{(M)}$ é a probabilidade do sistema sair de G_1 para G_2 onde o último estado visitado é s_i , e $p_{M,i}^{(M)}$ é a probabilidade do sistema visitar inicialmente o estado s_i após uma visita a G_2 .

Suponha agora que o tempo de permanência do sistema em G_2 seja nulo. Como a probabilidade de saída de G_2 é $1 - p_{M,M}^{(M)}$, podemos afirmar que a probabilidade do sistema reentrar em G_1 através do estado s_j dado que saiu de G_1 pelo estado s_i é $p_{i,M}^{(M)} \times (p_{M,j}^{(M)} / (1 - p_{M,M}^{(M)}))$. Logo, podemos eliminar o estado s_M e definir uma nova cadeia de Markov com $M - 1$ estados, onde as probabilidades de transição são definidas por

$$p_{i,j}^{(M-1)} = p_{i,j}^{(M)} + p_{i,M}^{(M)} \times \frac{p_{M,j}^{(M)}}{1 - p_{M,M}^{(M)}}, \quad 1 \leq i, j, \leq M - 1, \quad (1)$$

Podemos então repetir as operações discutidas acima para eliminarmos o estado s_{M-1} e obtermos uma cadeia com $M - 2$ estados. E assim sucessivamente. Após $M - 1$ passos, a cadeia de Markov tem apenas um estado e portanto, a probabilidade estacionária desse estado é 1, ou seja, $\pi_1 = 1$.

Na segunda parte do algoritmo GTH é feita a operação inversa. A cada passo do algoritmo um estado é adicionado e é calculada a probabilidade condicional deste estado. As probabilidades condicionais (normalizadas) encontradas na última parte do algoritmo correspondem às probabilidades estacionárias da cadeia de Markov original.

A primeira parte do algoritmo GTH tem um custo da $O(n^3)$, enquanto que a segunda parte do algoritmo tem custo da $O(n^2)$. Sendo então o custo total igual a $O(n^3)$.

2.2 Solução de Modelos Não Markovianos

Quando o tempo entre eventos de um modelo não tem distribuição exponencial, o modelo resultante não é markoviano. A seguir, vamos resumir o algoritmo introduzido em [7] e implementado na ferramenta TANGRAM-II [8, 9] para a solução de uma classe de modelos não markovianos onde no máximo 1 evento com duração não exponencial pode estar habilitado num instante qualquer. Abaixo particularizamos a descrição para modelos com eventos determinísticos ou exponenciais.

Seja φ_j um evento determinístico do modelo, i.e., o intervalo de tempo entre a ativação do evento e o disparo é determinístico. Este evento está habilitado durante o mini-intervalo Δ_j e poderá executar após transcorrido um intervalo de tempo T_j . Entretanto, φ_j pode ser desabilitado pela ocorrência de outros eventos. No caso em que o evento φ_j não é desabilitado no intervalo Δ_j , o comprimento δ_j deste intervalo é igual a T_j . Caso contrário, $\delta_j < T_j$. Δ_j corresponde então ao intervalo entre pontos embutidos onde φ_j está habilitado. Por conveniência de notação, chamamos de Δ_0 o intervalo onde nenhum evento determinístico está habilitado.

Entre dois pontos embutidos o comportamento dos sistemas descritos pode ser modelado por uma cadeia de Markov. Isto porque, todos os eventos entre pontos embutidos são exponenciais. A matriz de probabilidades de transição da cadeia embutida é então obtida através da análise transiente da cadeia de Markov associada aos intervalos Δ_j . Quando o intervalo não tem evento determinístico habilitado (intervalo Δ_0), as probabilidades de transição de estado são facilmente calculadas construindo-se uma cadeia de Markov associada a Δ_0 e mais um conjunto de estados, um para cada evento determinístico que pode ser habilitado ao término do intervalo.

A partir da obtenção da matriz de probabilidades de transição da cadeia embutida (chamada em [7] de matriz \mathcal{H}), o vetor de probabilidades estacionárias pode então ser calculado usando, por exemplo, o algoritmo GTH. É importante observar que o número de estados da matriz \mathcal{H} pode ser menor que o número de estados do modelo. Isto se deve ao fato que nem toda transição de estado da cadeia original representa um ponto embutido. É também importante notar que o vetor de probabilidades encontrado a partir da matriz \mathcal{H} representa as probabilidades estacionárias dos pontos embutidos e não dos estados do modelo. Veja [7] para mais detalhes do algoritmo de solução.

2.3 Cadeias de Markov ε -quasi-lumpable

Um dos métodos usados para a solução de cadeias de Markov com grande número de estados é a aglutinação (*lumping*). Com este método é possível reduzir o espaço de estados de um modelo markoviano preservando informações suficientes de forma a obter as medidas de interesse desejadas. O problema é que a aglutinação pode ser apenas aplicada a uma classe pequena de modelos, aqueles ditos aglutináveis (*lumpable*). Em [5], Franceschinis e Muntz propõem uma técnica que permite ampliar o conjunto de modelos que podem fazer uso do método de aglutinação, fornecendo valores aproximados para as medidas de interesse desejadas. Este

método é aplicável a uma cadeia de Markov não aglutinável se a cadeia puder ser transformada em aglutinável com uma *pequena* perturbação nas taxas de transição.

Dada uma partição $\mathcal{S} = \{S_1, \dots, S_M\}$ de uma cadeia de Markov \mathcal{X} de tempo discreto com matriz de probabilidades de transição \mathbf{P} , dizemos que \mathcal{X} é aglutinável para a partição \mathcal{S} se a probabilidade de transição do estado $s \in S_i$, $i = 1, \dots, M$, para um determinado estado da partição S_j , $j = 1, \dots, M$ e $j \neq i$, sempre tem um mesmo valor, qualquer que seja o estado s [1]. Uma cadeia de Markov é dita ε -quasi-lumpable se a sua matriz de probabilidade \mathbf{P} pode ser escrita como $\mathbf{P} = \mathbf{P}^- + \mathbf{P}^\varepsilon$, onde $\mathbf{P}^- + \text{diag}(\mathbf{P}^\varepsilon e^T)$ é aglutinável, \mathbf{P}^ε é uma matriz quase nula e e^T é a matriz coluna com valores 1. A solução de $\mathbf{P}^- + \text{diag}(\mathbf{P}^\varepsilon e^T)$ fornece uma boa aproximação para o modelo se os valores ε forem bem pequenos. O problema é a impossibilidade de se obter limites para o erro da solução quando esta aproximação é usada.

A idéia apresentada em [5] consiste em transformar a matriz de transição ε -quasi-lumpable \mathbf{P} com n estados em uma matriz de transição aglutinável \mathbf{P}_s com $n+1$ estados. Para isso, todas as transições ε da matriz \mathbf{P}^ε são direcionadas para um novo estado ao invés de serem colocadas na diagonal. O objetivo é definir a matriz \mathbf{P}_s de forma que, as probabilidades condicionais dos n primeiros estados de \mathbf{P}_s sejam iguais às probabilidades estacionárias dos estados de \mathbf{P} . Em [5] mostrou-se que é possível achar x de forma que as probabilidades condicionais dos estados em \mathbf{P}^- sejam iguais às probabilidades de \mathbf{P} , mas isso implica em resolver a matriz original. Entretanto, podemos solucionar aproximadamente a matriz \mathbf{P}_s usando os resultados de Courtois e Semal[10, 11]. Para isto, basta fazermos separadamente cada elemento de x igual a 1 e encontrar a solução do modelo correspondente. O maior e o menor valor encontrados para uma determinada medida de interesse fornecem, respectivamente, um limite superior e um limite inferior para a medida de interesse desejada [5].

2.4 Solução para um Sub-conjunto de Estados de um Modelo Markoviano

Em [6], de Souza e Silva e Ochoa propõem um método de aproximação para calcular as medidas de interesse usando apenas um sub-conjunto de estados da cadeia de Markov correspondente. A idéia é que apenas os estados considerados *importantes* façam parte do grupo de estados selecionados. A importância de um estado é definida pela sua contribuição no cálculo da medida de interesse desejada.

Seja \mathcal{X} uma cadeia de Markov homogênea de tempo contínuo com espaço de estados $S = \{s_i : i = 1, \dots, M\}$ e matriz de taxas de transição \mathbf{Q} . Para $\{s_i, s_j\} \in S$, dizemos que o estado s_j está a k passos do estado s_i , se o menor caminho do estado s_j até o estado s_i é composto por k transições.

Seja $\mathcal{S} = \{S_0, \dots, S_{k-1}, S_k, S_f\}$ uma partição de \mathcal{X} onde S_i corresponde ao conjunto de estados que estão a i passos da partição S_0 , $i = 1, \dots, k$, e S_f corresponde ao resto dos estados da cadeia de Markov. Em [6], S_0 é uma partição com um único estado, onde o estado escolhido é considerado *importante* no cálculo da medida de interesse desejada.

Podemos então definir 3 grupos de estados no modelo: $G_0 = \{S_0\}$, $G_1 = \{S_1, \dots, S_k\}$ e $G_2 = \{S_f\}$. Usando a idéia apresentada em [14] podemos construir uma cadeia de Markov \mathcal{X}' com 4 grupos de estados (G'_0 , G'_{1a} , G'_{1b} e G'_2) a partir de \mathcal{X} da seguinte forma: $G'_0 = G_0$, $G'_{1a} = G'_{1b} = G_1$ e $G'_2 = G_2$. A Figura 1 mostra as transições entre os grupos do novo processo \mathcal{X}' . A solução de \mathcal{X} é igual à solução de \mathcal{X}' [6], ou seja, $\pi_{G_0} = \pi_{G'_0}$, $\pi_{G_1} = \pi_{G'_{1a}} + \pi_{G'_{1b}}$ e $\pi_{G_2} = \pi_{G'_2}$, onde π_G corresponde ao vetor de probabilidades estacionárias dos estados do grupo G .

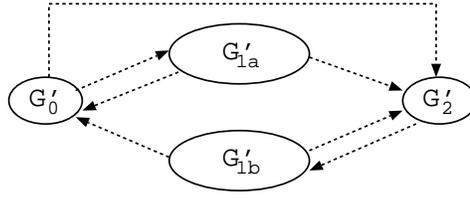


Figura 1: Novo Processo \mathcal{X}' .

Suponha que os estados de cada sub-conjunto S_i são agregados em um único estado f_i para $i = 1, \dots, k$, e que os estados do subconjunto S_f são agregados no estado f_{k+1} . Como os estados de G'_{1b} são cópias dos estados de G'_{1a} , dizemos que f_i é um *clone* da partição S_i , $i = 1, \dots, k$. O estado f_{k+1} é apenas a agregação dos estados de S_f . Vamos chamar este novo processo de \mathcal{X}'' . Como fizemos apenas a agregação exata de estados no processo \mathcal{X}' , a solução de \mathcal{X}'' é igual à solução de \mathcal{X} . O problema é que para fazermos a agregação desses estados precisamos solucionar \mathcal{X}' . Em [6], prova-se que é possível encontrar um limite inferior para a solução dos estados de G'_0 e G'_{1a} definindo as taxas de transição dos estados *clones*.

3 Um novo método de aproximação para o algoritmo GTH

Seja a equação 1. Convém observar dois casos no cálculo de $p_{i,j}^{(M-1)}$. No primeiro caso, já existe uma transição do estado s_i para o estado s_j quando o modelo tem M estados, ou seja, $p_{i,j}^{(M)} > 0$. Portanto, eliminar o estado s_M significa aumentar a probabilidade de transição entre estes dois estados se $p_{i,M}^{(M)} > 0$ e $p_{M,j}^{(M)} > 0$. No segundo caso, não existe uma probabilidade de transição do estado s_i para o estado s_j , ou seja, $p_{i,j}^{(M)} = 0$. Portanto, eliminar o estado s_M significa criar uma transição entre os estados s_i e s_j para a cadeia com $M - 1$ estados se $p_{i,M}^{(M)} > 0$ e $p_{M,j}^{(M)} > 0$. A partir da observação do segundo caso, podemos ver que o algoritmo GTH pode alterar a estrutura da matriz de probabilidades de transição do modelo original.

Sabemos que $1 - p_{M,M}^{(M)} = \sum_{j=1}^{M-1} p_{M,j}$, logo temos $0 \leq (p_{M,j}^{(M)}) / (1 - p_{M,M}^{(M)}) \leq 1$. Suponha que $(p_{M,j}^{(M)}) / (1 - p_{M,M}^{(M)}) = \varepsilon$ ou $p_{i,M}^{(M)} = \varepsilon$ na equação 1, onde ε é um valor pequeno (em relação ao erro admitido para a solução). Podemos então escrever a equação 1 como

$$p_{i,j}^{(M-1)} \leq p_{i,j}^{(M)} + \varepsilon. \quad (2)$$

Note que $p_{i,j}^{(M-1)} - p_{i,j}^{(M)} \leq \varepsilon$ para qualquer valor de $p_{i,M}^{(M)}$, $i = 1, \dots, M - 1$. Portanto, o algoritmo adiciona ao elemento (i, j) da matriz \mathbf{P} um valor que é menor ou igual a ε se $p_{i,M}^{(M)} > 0$. Para o caso em que $p_{i,M}^{(M)} = \varepsilon$, o algoritmo adiciona ε ao j -ésimo elemento da linha i se $p_{M,j}^{(M)} > 0$, $j = 1, \dots, M - 1$.

Podemos então concluir que uma matriz esparsa pode ser preenchida à medida em que os estados são eliminados na primeira parte do GTH. Se temos transições com valores muito pequenos ε , parte da matriz \mathbf{P} será preenchida com valores menores ou iguais a ε . A idéia básica do algoritmo proposto é diminuir o custo computacional do algoritmo GTH eliminando as operações que calculam valores menores ou iguais a ε . Uma maneira de atingir esse objetivo é redirecionar as transição de saída de um determinado estado que tem valor ε para a diagonal. Em outras palavras, algumas transições da matriz \mathbf{P} , por terem valores muito pequenos, podem ser eliminadas (ou redirecionadas) sem alterar significativamente os valores obtidos com a

solução exata do modelo. O problema com esta solução é a impossibilidade de se obter limites para o erro do resultado final.

A seguir vamos apresentar um algoritmo que calcula o erro introduzido na solução do modelo devido à eliminação de transições com valores ε . Antes, porém, vamos discutir quais transições com valores ε podem ser eliminadas.

3.1 Transições com valores ε

Lema 1. Seja \mathcal{X} uma cadeia de Markov de tempo discreto com espaço de estados $S = \{s_i : i = 1, \dots, M\}$. Suponha que \mathcal{X} não possui estados absorventes. Seja \mathcal{X}' uma cadeia de Markov idêntica à cadeia \mathcal{X} com a seguinte modificação: a transição p_{ij} é eliminada, onde p_{ij} corresponde à probabilidade de transição do estado s_i para o estado s_j , $\{s_i, s_j\} \in S$ e $s_i \neq s_j$. A cadeia \mathcal{X}' não tem estados absorventes se $p_{ij}/(1 - p_{ii}) < 1$.

Prova. Veja [12].

O Lema 1 define a condição necessária e suficiente para que uma transição de saída do estado s_i seja eliminada sem que s_i se torne um estado absorvente: é preciso que o estado s_i possua outras transições de saída. O problema é que todas as transições de saída de um determinado estado podem ter valor ε . Neste caso as transições ε podem não ser desprezíveis. A idéia é que a probabilidade de transição p_{ij} do estado s_i para o estado s_j seja considerada *eliminável* se e somente se, $(p_{ij})/(1 - p_{ii}) \leq \varepsilon$, para $i \neq j$. Portanto, a transição do estado s_i para o estado s_j só será eliminada quando o peso da probabilidade p_{ij} em relação à probabilidade total de saída do estado s_i for bem pequeno (ε).

Precisamos agora definir como as transições ε são eliminadas (ou redirecionadas) durante a execução do algoritmo GTH. Considere novamente a cadeia de Markov com espaço de estados $S = \{s_i : i = 1, \dots, M\}$. Seja p_{ij} a probabilidade de transição do estado s_i para o estado s_j , $\{s_i, s_j\} \in S$. Queremos eliminar o estado S_M e calcular as novas probabilidades de transição dos estados restantes. Para $i = 1, \dots, M$, de acordo com o que foi discutido acima: a transição $p_{i,M}$ é eliminada se e somente se $p_{i,M}/(1 - p_{i,i}) \leq \varepsilon$; e a transição $p_{M,i}$ é eliminada se e somente se $p_{M,i}/(1 - p_{M,M}) \leq \varepsilon$. Note que só precisamos testar as transições de entrada e de saída do estado s_M , pois apenas estas transições são usadas pelo GTH no passo que elimina o estado S_M .

Como observação final, note que as regras para eliminação de uma transição ε discutidas acima não garantem que a matriz resultante seja ergódica. Elas apenas garantem que a nova matriz não terá estados absorventes e que a eliminação de uma transição não altera significativamente a solução do modelo. O algoritmo identifica que a matriz não é ergódica quando o estado a ser eliminado tem na matriz de transição a coluna acima da diagonal zerada (não existem transições de entrada para o estado). Neste caso, o algoritmo informa ao usuário da ocorrência desse caso (a probabilidade estacionária do estado é zero) e continua a execução.

3.2 Novo algoritmo de aproximação com cálculo do erro

De acordo com a discussão acima, podemos imaginar que para obtermos uma aproximação da solução do modelo, basta adicionarmos um novo estado ao modelo e redirecionarmos as transições ε para este estado (veja [5]). A probabilidade estacionária do novo estado neste caso corresponde ao erro introduzido na solução. Infelizmente, temos um problema nesta proposta: não sabemos o valor das transições do novo estado para os estados originais do modelo, e obter o valor destas transições pode sair mais caro que obter a solução do modelo original uma vez que a matriz resultante não é aglutinável como em [5]. Entretanto, podemos substituir este

novo estado por um conjunto de estados *clones* dos estados do modelo original (veja [6]) e redirecionar as transições ε para estes estados. A diferença é que agora podemos calcular o valor das transições dos estados *clones* para os estados originais do modelo. O problema é saber se o custo desta aproximação é menor que o custo da solução do modelo original e se é possível manter o erro dentro de um limite tolerável. A seguir introduzimos formalmente este método de aproximação e discutimos essas e outras questões com mais detalhes.

Seja \mathbf{P} uma matriz de probabilidades de transição com M estados. De acordo com o método de [5] podemos transformar a matriz \mathbf{P} de M estados em uma matriz de transição \mathbf{P}_s com $M+1$ estados como mostra a Figura 2. Suponha que o espaço de estados de \mathbf{P} é $S = \{s_1, \dots, s_M\}$ e que o espaço de estados de \mathbf{P}_s é $S' = \{s_\varepsilon\} + S$. Como discutido em [5], temos $\mathbf{P}^- = \mathbf{P} - \mathbf{P}^\varepsilon$, $y^T = \mathbf{P}^\varepsilon e^T$ e $x e^T = 1$, onde \mathbf{P}^ε possui todas as transições ε . Isto significa que o vetor y^T representa as transições ε redirecionadas para estado s_ε , e o vetor x representa as probabilidades de transição do estado s_ε para os outros estados da cadeia. O vetor x é definido de forma que o vetor das probabilidades estacionárias de \mathbf{P} seja igual ao vetor das probabilidades condicionais dos M últimos estados de \mathbf{P}_s . **Lema 2.** Seja \mathbf{P} a matriz de probabilidades de transição de

0	x
y^T	\mathbf{P}^-

Figura 2: Nova matriz \mathbf{P}_s .

uma cadeia de Markov com M estados. Seja \mathbf{P}^- um limite inferior para \mathbf{P} , ou seja, $\mathbf{P}^- \leq \mathbf{P}$. Seja \mathbf{P}_s a matriz estocástica da Figura 2, onde $y^T = \mathbf{P}^\varepsilon e^T$ e $x e^T = 1$. Então existe um vetor x tal que as probabilidades condicionais dos M últimos estados de \mathbf{P}_s é igual às probabilidades estacionárias dos estados de \mathbf{P} .

Prova. Veja [5].

O problema é que para calcular x precisamos da solução da matriz \mathbf{P} . Entretanto, podemos usar os resultados de Courtois e Semal [10, 11] para obter uma aproximação para a solução de \mathbf{P}_s . Para isto, precisamos fazer cada elemento do vetor x igual a 1 e solucionar a cadeia correspondente. O menor e o maior valores obtidos para as probabilidades estacionárias dos M últimos estados de \mathbf{P}_s fornecem um limite inferior e superior para as probabilidades estacionárias dos estados de \mathbf{P} . Infelizmente, esta solução pode sair mais cara que a solução do modelo original, pois é necessário solucionar \mathbf{P}_s várias vezes. Além disso, é possível que novas transições ε sejam geradas a cada passo da primeira parte do GTH e portanto, estas transições também devem ser redirecionadas para o estado s_ε . Isto significa que pode ser necessário utilizar os resultados de Courtois e Semal [10, 11] em cada passo da primeira parte do GTH.

Resumindo, o redirecionamento das transições ε para o estado s_ε fornece uma solução exata do modelo se as transições de saída de s_ε são conhecidas. Infelizmente, para calcular os valores dessas transições precisamos da solução do modelo original. Portanto, nosso objetivo é tentar obter um limite para o erro da solução do modelo aproximado e ainda conseguir ganhos no cálculo desta solução. A idéia é substituir o estado s_ε por um conjunto de estados *clones* dos estados da cadeia como definido no método de [6].

Seja \mathcal{X} uma cadeia de Markov de tempo contínuo com espaço de estados $S = \{s_1, \dots, s_M\}$. Suponha que existam transições com valores ε entre alguns dos estados de \mathcal{X} . Podemos trans-

formar \mathcal{X} na cadeia de Markov \mathcal{X}' de tempo contínuo e espaço de estados $S' = \{s_1, \dots, s_M, s_\varepsilon\}$, onde todas as transições ε dos M primeiros estados de \mathcal{X} são redirecionadas para o estado s_ε em \mathcal{X}' como discutido anteriormente.

Seja $s_{inicial}$ um estado de \mathcal{X}' tal que $s_{inicial} \in \{s_1, \dots, s_M\}$. Assuma que K é a maior distância de um estado de \mathcal{X}' para o estado $s_{inicial}$, ou seja, o menor caminho entre um estado de \mathcal{X}' e o estado $s_{inicial}$ nunca é maior que K transições. Considere a partição $\mathcal{S} = \{S_0, S_1, \dots, S_K\}$ da cadeia \mathcal{X} , onde S_i corresponde ao conjunto de estados de \mathcal{X}' que estão a i passos de $s_{inicial}$, $i = 1, \dots, K$. Podemos gerar a cadeia de Markov \mathcal{X}'' de tempo contínuo a partir da cadeia \mathcal{X}' usando o método de [6]. A Figura 3 mostra o novo processo markoviano \mathcal{X}'' , onde definimos $S^* = \{S_1, \dots, S_K\}$. Note que as transições ε de S_0 e de qualquer estado de S^* são redirecionadas para f_K . **Lema 3.** Seja \mathcal{R} (\mathcal{R}'') a recompensa média acumulada

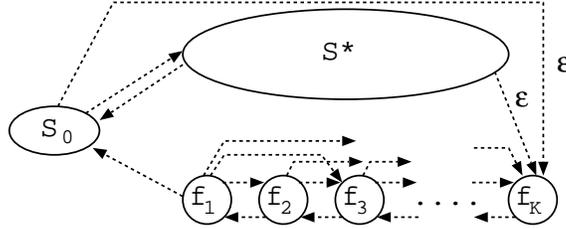


Figura 3: Processo markoviano descrito por \mathcal{X}'' .

por \mathcal{X} (\mathcal{X}'') durante o intervalo $(0, t)$, quando $t \rightarrow \infty$. Suponha que r_{lb} é a menor recompensa associada a um estado da cadeia \mathcal{X} e que r_{lb} é associada a cada estado f_i para todo $1 \leq i \leq K$. Então, $\mathcal{R}'' \leq \mathcal{R}$.

Prova. Veja [6].

O Lema 3 nos permite fazer a seguinte afirmação sobre as soluções de \mathcal{X} e \mathcal{X}'' : as probabilidades estacionárias dos estados $\{s_1, \dots, s_M\}$ de \mathcal{X}'' são limites inferiores para as probabilidades estacionárias dos estados $\{s_1, \dots, s_M\}$ de \mathcal{X} . Isto é verdade pois, pelas regras acima, as transições entre estados f_i são escolhidas de forma a aumentar o tempo de permanência nestes estados clones o que consequentemente diminui o tempo relativo nos estados de S_0 e S^* (ver [6] para maiores detalhes).

Vimos que a partição \mathcal{S} de \mathcal{X}'' depende da escolha do estado $s_{inicial}$. O método de [6] apenas supõe que $s_{inicial}$ é um estado *importante* para a medida de interesse a ser calculada para o modelo. A escolha de $s_{inicial}$ pode não ser trivial. Sabemos que quanto menor o tempo de permanência nos estados *clones*, menor o erro introduzido na solução do modelo. Devemos então escolher os estados de S_0 de forma que as taxas de transição dos estados *clones* em direção a S_0 seja maior que as taxas de transição dos estados *clones* em direção ao estado *clone* f_k . Portanto, a escolha dos estados de S_0 pode ser feita através da análise da matriz de transição do modelo. Por isso, usamos a ferramenta TANGRAM-II [8] [9] para estudar os modelos apresentados neste trabalho e escolher o conjunto S_0 .

Caso a partição S_0 contenha mais de um estado resta ainda determinar as taxas de transição de f_1 para cada estado em S_0 a partir da taxa de transição de f_1 para o conjunto S_0 .

Lema 4. Seja q a taxa de transição do estado f_1 para a partição S_0 na cadeia \mathcal{X}'' . Faça a transição de f_1 para um dos estados de S_0 igual a q e encontre a solução do modelo. Repita esta operação para cada estado de S_0 . A solução de \mathcal{X}'' corresponde aos menores valores obtidos com cada solução. As probabilidades estacionárias dos estados $\{s_1, \dots, s_M\}$ de \mathcal{X}'' é um limite inferior para as probabilidades dos estados $\{s_1, \dots, s_M\}$ de \mathcal{X} .

Prova. Veja [12].

O Lema 4 mostra que se o número de estados da partição S_0 for m , teremos que solucionar o modelo m vezes, onde a taxa de transição de f_1 para um dos estados de S_0 é igual à taxa total de transição de f_1 para S_0 . Para reduzir o custo computacional decorrente do Lema 4, suponha que a numeração dos estados de \mathcal{X}'' é feita na seguinte ordem: estados clones, estados de S_0 e o resto dos estados do modelo (veja a Figura 4). Além disso, suponha que $S'' = \{e_1, \dots, e_{M+k}\}$, isto é, o i -ésimo estado de S'' é denominado e_i , portanto $e_1 = f_1, \dots, e_k = f_k, e_{k+1} = s_1, \dots, e_{M+k} = s_M$. É interessante aqui lembrarmos de dois fatos. Em primeiro lugar, os estados são eliminados pelo algoritmo na seguinte ordem: $e_{M+k}, e_{M+k-1}, \dots, e_2$. Em segundo lugar, não conhecemos os valores das transições de e_1 (f_1) para os estados de S_0 . **Lema 5.** Suponha

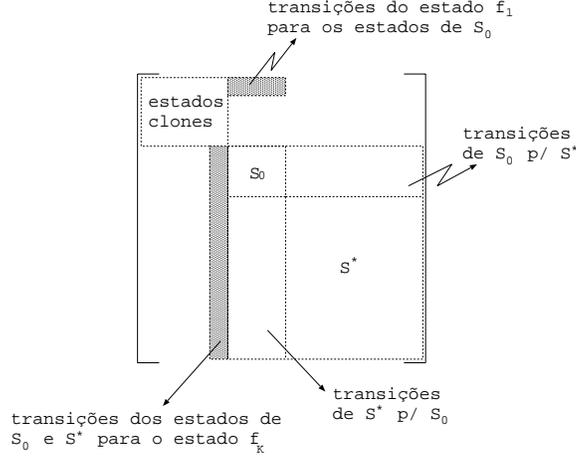


Figura 4: Estrutura da matriz de \mathcal{X}'' .

que $S_0 = \{s_{k+1}, \dots, s_{m+k}\}$, isto é, S_0 é formado pelos m primeiros estados da cadeia após os estados clones. Seja ${}^l\pi_i$ a probabilidade estacionária (não normalizada) do estado $e_i \in S''$ quando a transição de e_1 para S_0 é feita somente através do estado e_{k+l} , $1 \leq l \leq m$. Sejam

$$\pi_i^{min} = \begin{cases} \min_{1 \leq j \leq m} \{ {}^j\pi_i \} & \text{para } 1 \leq i \leq m+k \\ \sum_{j=1}^{i-1} \frac{p_{ji}^{(i)}}{1 - p_{ii}^{(i)}} \pi_j^{min} & \text{para } m+k+1 \leq i \leq M+k \end{cases} \quad (3)$$

e

$$\pi_i^{max} = \begin{cases} \max_{1 \leq j \leq m} \{ {}^j\pi_i \} & \text{para } 1 \leq i \leq m+k \\ \sum_{j=1}^{i-1} \frac{p_{ji}^{(i)}}{1 - p_{ii}^{(i)}} \pi_j^{max} & \text{para } m+k+1 \leq i \leq M+k. \end{cases} \quad (4)$$

Para $i = 1, \dots, M$, temos $\frac{\pi_{i+k}^{min}}{\sum_{j=1}^{M+k} \pi_j^{max}} \leq \pi_i$ e $\text{Erro} = 1 - \frac{\sum_{i=1}^M \pi_{i+k}^{min}}{\sum_{j=1}^{M+k} \pi_j^{max}}$, onde $p_{ji}^{(i)}$ é a probabilidade de transição do estado s_j para o estado s_i da matriz \mathbf{A}_i (a cadeia de Markov nesse passo do GTH tem i estados) e π_i é a probabilidade estacionária (normalizada) do estado s_i na solução exata do modelo.

Prova. Veja [12].

Resumindo, podemos dividir o método de aproximação aqui proposto em quatro passos. No primeiro passo, os estados de S_0 são escolhidos através da análise da matriz de transição do modelo. No segundo passo, o conjunto de estados *clones* é gerado e as taxas de transição

do novo modelo são calculadas de acordo com [6]. No terceiro passo, é executada a primeira parte do algoritmo GTH onde as transições ε são redirecionadas para o estado clone f_K . No quarto passo, é executada a segunda parte do algoritmo GTH e o Lema 5 é usado para calcular o erro introduzido pelo novo método.

4 Exemplos

Nesta seção são mostrados resultados numéricos de dois modelos de redes usando os algoritmos discutidos neste artigo. O primeiro é o modelo de *jitter* apresentado em [12] e o segundo é um servidor de disco modelado em [13].

4.1 Modelo 1: Jitter

Considere um tráfego de voz enviado através de uma rede IP. Nosso objetivo é dimensionar o *playout buffer* no receptor de forma a reduzir a valores aceitáveis do *jitter* [16] entre pacotes de voz.

O primeiro exemplo aqui discutido corresponde a um dos modelos de *jitter* apresentados em [12]. Neste exemplo escolhemos o desvio padrão para o *jitter* igual a 36, bem acima do valor típico de 20, de forma a super dimensionar o *playout buffer*. Neste caso, as probabilidades de transição entre estados possuem valores bem distintos, a menor probabilidade de transição é $1.8e-04$ e a maior é $9.1e-01$.

Neste modelo, o processo de entrega de pacotes para a aplicação só é inicializado após um tempo T_1 da chegada do primeiro pacote. Após o período de silêncio, a aplicação entrega pacotes a intervalos fixos de duração T_2 . O *playout buffer* armazena até B pacotes. A medida de interesse desejada para este modelo é a probabilidade do *jitter* ser maior que um determinado valor após o *playout buffer*.

Visando exemplificar o uso dos algoritmos discutidos neste trabalho, inicialmente iremos calcular a medida de interesse assumindo que o tempo de permanência em qualquer estado do modelo é exponencialmente distribuído. Em seguida, vamos calcular a medida de interesse para o modelo com transições determinísticas e exponenciais.

Para $B = 50$, a matriz de transição possui 2801 estados onde cada estado é representado por 4 variáveis de estado (veja [12] para mais detalhes): n_1 (número total de pacotes armazenados na fila), n_2 (número de pacotes armazenados na fila antes do último período de silêncio), f (estado do processo de chegada) e s (estado do servidor). Suponha que \mathbf{P} corresponde a matriz de probabilidades de transição de estados ordenados pelos valores de n_1, n_2, f e s .

Precisamos agora definir os estados de S_0 do novo algoritmo de aproximação. Uma maneira é agrupar os estados da cadeia de acordo com uma ou mais variáveis de estado do modelo. Para este exemplo escolhemos a variável de estado n_2 . Portanto, o estado clone f_i representa os estados que tem armazenado i pacotes dos períodos ativos anteriores, $i = 1, \dots, 50$. Vamos chamar a esta matriz reordenada de \mathbf{P}' . Os estados de S_0 de \mathbf{P}' possuem $n_2 = 0$ (são 201 estados neste exemplo)

A Tabela 1 apresenta três soluções para o exemplo, quando $\varepsilon = 1.0e - 04$ e $T_1 = 800$. A primeira solução corresponde à solução exata do modelo onde a matriz de probabilidades de transição é \mathbf{P} . A segunda solução corresponde à solução da matriz $\mathbf{P}^- + \text{diag}(\mathbf{P}^\varepsilon e^T)$, onde $\mathbf{P}' = \mathbf{P}^- + \mathbf{P}^\varepsilon$ e portanto, solução sem limites. E a terceira solução corresponde à solução encontrada pelo algoritmo proposto na seção 3 usando a matriz de probabilidades de transição \mathbf{P}' . Convém aqui observar que o motivo de usarmos a matriz \mathbf{P} no primeiro caso se deve ao fato

que a solução desta matriz tem um menor custo para o algoritmo do GTH (isto é, favorecendo a solução do GTH), enquanto que o motivo de usarmos a matriz \mathbf{P}' nas duas outras soluções se deve ao fato do novo algoritmo exigir que os estados de S_0 (grupo inicial) sejam os primeiros da matriz.

x	\mathbf{P}	Sem obteção de Limites (Alg. Proposto)	Limite Inferior (Alg. Proposto)
0	2.5092073088e-01	2.6131190857e-01	2.4587567667e-01
20	1.6094900813e-01	1.6761426285e-01	1.5771258112e-01
40	1.0324322287e-01	1.0751875328e-01	1.0116716685e-01
60	6.6226957407e-02	6.8969562317e-02	6.4895239257e-02
80	4.2482303103e-02	4.4241589316e-02	4.1628051960e-02
100	2.7250928438e-02	2.8379449709e-02	2.6702955869e-02

Tabela 1: $\Pr[\text{Jitter} > x]$ para $T_1 = 800$ e $B = 50$.

A Figura 5a mostra o número de operações requeridas para as três soluções quando o valor de T_1 varia de 100 a 800. Podemos fazer dois comentários a respeito da Figura: a) o número de operações requeridas pelos algoritmos praticamente se mantém constante quando variamos T_1 ; b) existe uma diferença de pelo menos uma ordem de magnitude entre o número de operações da solução exata e o número de operações das outras duas soluções (lembramos que usamos a matriz \mathbf{P} na solução exata e a matriz \mathbf{P}' nas outras soluções). É interessante notar em relação à segunda observação que os algoritmos de aproximação são mais baratos que o GTH, apesar da matriz \mathbf{P} possuir uma estrutura mais compacta que a matriz \mathbf{P}' .

A Figura 5b mostra o número de operações requeridas para as três soluções quando o valor de B varia de 50 a 150. O modelo tem 2801 estados quando $B = 50$ e 23401 estados quando $B = 150$. Por problemas de memória só foi possível usar o algoritmo GTH para B variando de 50 a 100. O número de operações do GTH para os outros valores de B foi calculado de acordo com a estrutura da matriz de transição. Para os outros dois algoritmos obtivemos as soluções do modelo facilmente.

Vejam agora o modelo de *jitter* com as transições determinísticas. Neste exemplo, temos dois tipos de evento determinístico: φ_1 que corresponde ao início da entrega dos pacotes (T_1 unidades de tempo após a chegada do primeiro pacote) e φ_2 que corresponde à entrega de um pacote ao destino. Portanto, o modelo agora não é markoviano. A matriz \mathcal{H} gerada a partir da matriz \mathbf{P} discutida acima possui 2701 estados. Considere que a matriz \mathcal{H}' é gerada a partir da reordenação dos estados de \mathcal{H} usando o mesmo critério de ordenação de \mathbf{P}' : n_2, n_1, f, s . Novamente, os estados de S_0 correspondem aos estados onde $n_2 = 0$.

Quando estudamos uma solução para modelos não markovianos em 2.2, vimos que a solução da Matriz \mathcal{H} nos fornece o vetor β das probabilidades estacionárias dos pontos embutidos e que precisamos ainda calcular o vetor π das probabilidades estacionárias dos estados do modelo. Como o vetor β é uma aproximação para a solução de \mathcal{H} , o vetor π não é a solução exata da matriz \mathbf{P} , mas uma aproximação para a solução de \mathbf{P} .

A Tabela 2 apresenta três soluções para a matriz \mathcal{H} quando $\varepsilon = 1.0e - 04$ e $T_1 = 800$. Note que a aproximação da medida de interesse obtida usando o algoritmo proposto é melhor do que a aproximação obtida pela solução de $\mathcal{H}^- + \text{diag}(\mathcal{H}^\varepsilon e^T)$.

A Figura 5c mostra o número de operações requeridas para as três soluções quando o valor de T_1 varia de 100 a 800. Novamente temos uma diferença de pelo menos 1 ordem de magnitude entre a solução do algoritmo GTH e os outros dois algoritmos.

x	\mathcal{H}	Sem obtenção de Limites (Alg. Proposto)	Limite Inferior (Alg. Proposto)
0	2.5961166006e-01	2.6173145370e-01	2.5871515108e-01
20	1.6653190277e-01	1.6789167709e-01	1.6595681975e-01
40	1.0682445709e-01	1.0769670530e-01	1.0645556122e-01
60	6.8524195323e-02	6.9083712386e-02	6.8287561376e-02
80	4.4314812638e-02	4.3955901790e-02	4.3804109295e-02
100	2.8196191039e-02	2.8426419938e-02	2.8098821403e-02

Tabela 2: $\Pr[\text{Jitter} > x]$ para $T_1 = 800$ e $B = 50$.

4.2 Modelo 2: Servidores de Vídeo com Carga Mista

Nosso segundo exemplo é um modelo de um servidor com serviço *G-Gated*. Temos um único recurso, neste caso um disco, e vários pedidos de acesso às informações armazenadas no disco. Imagine que o disco armazena arquivos com características bem distintas como, por exemplo, filmes e dados. Precisamos, portanto, definir como estes pedidos vão ser atendidos de forma a atender a QoS dos pedidos de aplicação em tempo real. A seguir vamos discutir o modelo de um servidor de disco com política de serviço *g-gated* que foi apresentado em [13].

O servidor de disco [13] atende a duas classes de pedidos: *continuous media C* e *no continuous media NC*. Quando um pedido chega ao servidor, ele é armazenado na fila correspondente ao seu tipo e aguarda a sua vez. O tempo é dividido em ciclos de tamanho T . Em cada período T o servidor deve atender exatamente N_c pedidos C . O valor de T e N_c dependem do nível da QoS que o sistema fornece aos seus usuários. Assume-se que em cada ciclo chegam N_c pedidos C para serem servidos no próximo ciclo. O servidor utiliza o algoritmo *g-gated* para escalonar os pedidos a serem atendidos (veja [13]).

Os parâmetros para o exemplo são os mesmos utilizados em [13] para um disco com 2.25 GBytes e 5288 cilindros, uma taxa de transferência de 75 Mbps e tempo de latência máximo de 8.33 segundos: $T = 1.48754$ segundos; $N_c = 24$; $N_{mc} = 6$; o tempo de serviço de 24 pedidos C tem uma distribuição erlangiana com 6 estágios (o número de estágios depende da QoS escolhida, veja [13] para mais detalhes); e o tempo de serviço de um pedido NC é exponencialmente distribuído com média de 0.018407 segundos.

O limite T é constante, e portanto o modelo a ser resolvido não é markoviano. Os pontos embutidos deste exemplo correspondem ao início e ao fim de cada mini-ciclo. Para este exemplo a matriz de transição do modelo tem 4941 estados e a matriz \mathcal{H} dos pontos embutidos tem 3996 estados. A matriz \mathcal{H}' é obtida pela ordenação dos estados conforme definido em [13]. Os estados do grupo inicial usado no algoritmo proposto representam o sistema no N_{mc} -ésimo mini-ciclo (259 estados).

As Tabelas 3 e 4 mostram o tempo médio de resposta para um pedido NC para $\varepsilon = 1.0e-05$ e $\varepsilon = 1.0e-03$, respectivamente, quando a taxa de chegada de pedidos NC varia de 1 a 10.

A Figura 6 mostra o número de operações requeridas para os algoritmos quando a taxa de chegada de pedidos NC varia de 1 a 10 e ε é igual a $1.0e-05$ e a $1.0e-03$. Note que o número de operações é praticamente constante para os três algoritmos quando variamos a taxa de chegada, e que existe uma diferença de pelo menos uma ordem de magnitude entre o número de operações do algoritmo GTH e o número de operações dos outros dois algoritmos.

A partir da solução de H , as medidas de interesse de [13] podem ser calculadas sem maiores esforço computacional. Dentre estas medidas destacam-se probabilidade de *overflow* da classe

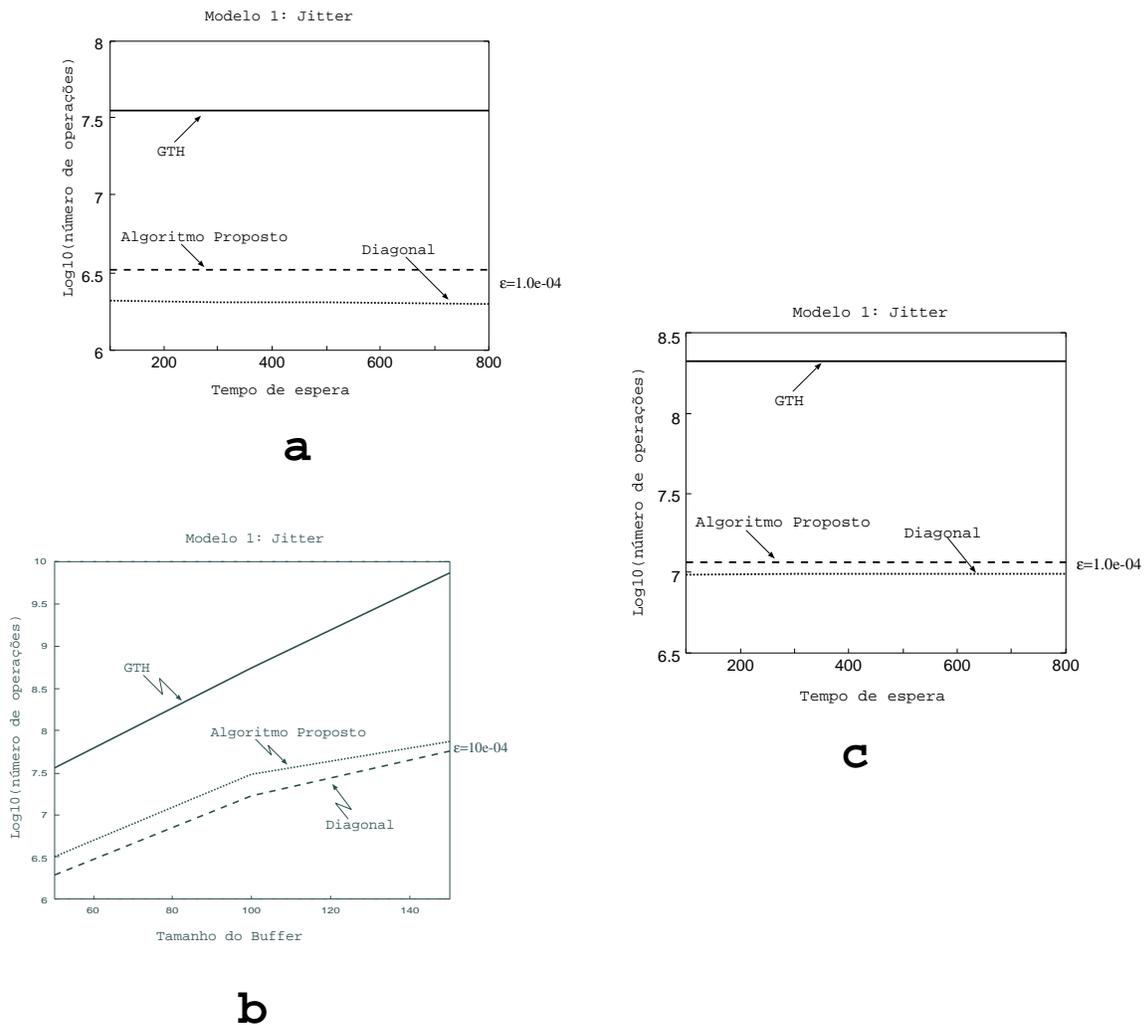


Figura 5: Número de operações requeridas na solução do modelo *jitter*.

C e tempo médio de resposta dos pedidos da classe NC , importantes para a avaliação de servidor multimídia.

5 Conclusões

Um dos métodos mais importantes na solução de cadeias de Markov é o método GTH [3]. Entretanto, o uso do GTH tem algumas restrições, a principal delas é o custo computacional da ordem de $O(N^3)$, onde N corresponde ao número de estados do sistema sendo modelado. Este custo pode ser reduzido se a matriz em questão possuir uma estrutura de banda, por exemplo.

Neste artigo, propomos uma aproximação para o método GTH de forma que ele possa ser usado na solução de modelos cuja cardinalidade do espaço de estados é muito grande, de tal forma que impossibilita o uso do GTH. O método tem como base dois métodos de aproximação existentes na literatura ([5] e [6]). O novo método é útil quando existem probabilidades de transição que diferem em ordens de magnitude. Uma perturbação na matriz original é feita de forma a ser possível obter limites para o erro introduzido com a modificação.

A utilidade do algoritmo é ilustrada usando modelos de redes multimídia. Os resultados apresentados mostram que o novo algoritmo fornece uma boa aproximação para a solução de

Taxa de Chegada	\mathcal{H}	Sem obtenção de Limites (Alg. Proposto)	Limite Inferior (Alg. Proposto)
1	8.5309255855e-02	8.5211824070e-02	8.5249021836e-02
2	8.8155120799e-02	8.8042486353e-02	8.8091853646e-02
4	9.4480540868e-02	9.4364984173e-02	9.4430481616e-02
6	1.0130366183e-01	1.0119635330e-01	1.0127005822e-01
8	1.0827843193e-01	1.0817052606e-01	1.0825465280e-01
10	1.1508465436e-01	1.1497640075e-01	1.1506956013e-01

Tabela 3: Tempo médio de resposta de um pedido NC quando $\varepsilon = 10e - 05$.

Taxa de Chegada	\mathcal{H}	Sem obtenção de Limites (Alg. Proposto)	Limite Inferior (Alg. Proposto)
1	8.5309255855e-02	8.2554738838e-02	8.4659286935e-02
2	8.8155120799e-02	8.4853090716e-02	8.7640481682e-02
4	9.4480540868e-02	8.9013311956e-02	9.3782893023e-02
6	1.0130366183e-01	9.3946296685e-02	1.0057588108e-01
8	1.0827843193e-01	9.8744536955e-02	1.0758563017e-01
10	1.1508465436e-01	1.0330094600e-01	1.1452575879e-01

Tabela 4: Tempo médio de resposta de um pedido NC quando $\varepsilon = 10e - 03$.

modelos Markovianos e de não Markovianos.

Referências

- [1] de Souza e Silva, E., Muntz, R. R., *Métodos Computacionais de Solução de Cadeias de Markov: Aplicações a Sistemas de Computação e Comunicação*. VIII Escola de Computação, 1992.
- [2] W.J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, 1994.
- [3] Grassmann, W., Taksar, M., Heyman, D., "Regenerative Analysis and Steady State Distributions for Markov Chains", *Operation Research*, V. 33, N.5, pp. 1107-1116, 1985.
- [4] Golub, G. H., Loan, C. F. V., *Matrix Computation*, 2 ed., The John Hopkins University Press, 1989.
- [5] Franceschinis, G., Muntz, R. R., "Bounds for Quasi-lumpable Markov Chains", *Performance Evaluation*, 20(1994), pp. 223-243.
- [6] de Souza e Silva, E., Ochoa, P. M., "State Space Exploration in Markov Models", *Performance Evaluation Review*, V. 20, N. 1, pp. 152-166, Junho 1992.
- [7] de Souza e Silva, E., Gail, H. R., Muntz, R. R., "Efficient Solutions for a Class of Non-Markovian Models". *Computations with Markov Chains*, 1 ed., Kluwer Academic Publishers, pp. 483-506, 1995.

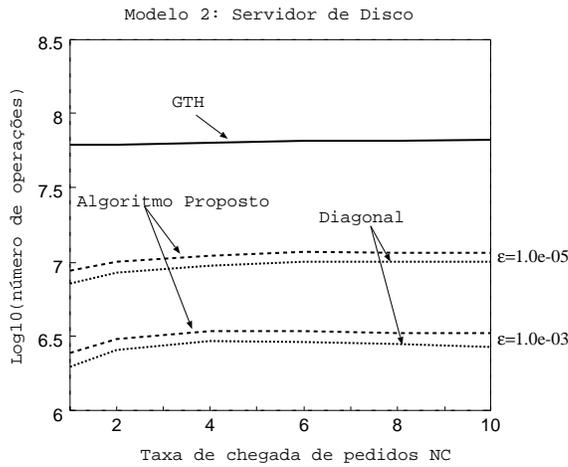


Figura 6: Número de operações requeridas na solução da matriz \mathcal{H} .

- [8] Carmo, R. M. L. R., Carvalho, L. R. de, de Souza e Silva, E., Diniz, M. C., Muntz, R. R., "Performance/availability modeling with the TANGRAM-II modeling Environment", *Performance Evaluation*, n.740, pp. 1-20, 1998.
- [9] de Souza e Silva, E., M.M.Leão, R. "Computer Performance Evaluation - Modelling Techniques and Tools", 11th *International Conference (TOOLS2000)*, Schaumburg, IL, USA, Março 2000, Lectures Note in Computer Science, volume 1786, Springer, pp. 366-369.
- [10] Courtois, P. J., Semal, P., "Bounds for the Positive Eigenvectors of Nonnegative Matrices and for their Approximations", *Journal of the ACM*, V.31, pp. 804-825, 1984.
- [11] Courtois, P. J., Semal, P., "Computable Bounds for Conditional Steady-state Probabilities in Large Markov Chains and Queueing Models", *IEEE Journal on Selected Areas in Communications*, SAC-4(6), pp. 926-937, 1986.
- [12] Diniz, M. C. "Técnicas de Solução para Modelos Provenientes de Redes Multimídias". Tese de Doutorado. COPPE/Sistemas - UFRJ. 2000.
- [13] de Souza e Silva, E., Gail, H. R., Golubchik, L., Lui, J. C. S., "Analytical Models for Mixed Workload Multimedia Storage Servers". *Performance Evaluation*, 36-37 (1999), 185-211.
- [14] Muntz, R. R., de Souza e Silva, E., Goyal, A., "Bounding Availability of Repairable Computer Systems". *IEEE Transactions on Computers*, V. 38, N. 12, pp. 1714-1723, Dezembro 1989.
- [15] Berson, S., de Souza e Silva, E., Muntz, R. R., *Numerical Solution of Markov Chains*, chapter: An Object Oriented Methodology for the Specification of Markov Models, pp. 11-36, Marcel Dekker, 1991.
- [16] Kurose, J., Ross, K., *Computer Networking: a top-down approach featuring the Internet*. Addison Wesley Longman, Inc. 2001.