Pós-Graduação em Ciência da Computação

"ISPN: modelagem e avaliação estocástica intervalar"

Por

Sérgio Mário Lins Galdino

Tese de doutorado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, ABRIL/2009

Universidade Federal de Pernambuco
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Sérgio Mário LinsGaldino

"ISPN: modelagem e avaliação estocástica intervalar"

Este trabalho foi apresentado à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

ORIENTADOR: ***Prof. Dr. Paulo Romero Martins Maciel***

RECIFE,ABRIL/2009

Galdino, Sérgio Mário Lins

ISPN: Modelagem e avaliação estocástica intervalar / Sérgio Mário Lins Galdino - Recife : O Autor, 2009.

xxx, 149 p. : il., fig., tab.

Tese (doutorado) – Universidade Federal de Pernambuco. CIn. Ciência da Computação, 2009.

Inclui bibliografia e apêndice.

1. Avaliação de desempenho. 2. Redes de Petri. 3. Análise intervalar. I. Título.

004.029 CDD (22. ed.) MEI2009- 081

Tese de Doutorado apresentada por **Sérgio Mário Lins Galdino** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título **"ISPN: Modelagem e Avaliação Estocástica Intervalar"**, orientada pelo **Prof. Paulo Romero Martins Maciel** e aprovada pela Banca Examinadora formada pelos professores:

Prof. Paulo Roberto Freire Cunha
Centro de Informática / UFPE

Prof. Nelson Souto Rosa
Centro de Informática / UFPE

Prof. Ricardo Massa Ferreira Lima
Centro de Informática / UFPE

Prof. Virgílio Augusto Fernandes Almeida
Departamento de Ciência da Computação / UFMG

Prof. Avelino Francisco Zorzo
Faculdade de Informática / PUC-RS

Visto e permitida a impressão.
Recife, 3 de abril de 2009.

Prof. FRANCISCO DE ASSIS TENÓRIO DE CARVALHO
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Célia (Esposa)
Lília & Alice (Filhas)
Félix & Gertrudes (Pais)
Suely & Sandra (Irmãs)

Agradecimentos

Paulo Maciel, *père scientifique*, pela dedicação e apoio que permitiu o desenvolvimento deste trabalho.

Avelino Zorzo, Nelson Rosas & Paulo Cunha, como membros da Banca de Qualificação e da Proposta de Tese, pelas questões levantadas, que foram contempladas, contribuindo para a documento de defesa da tese. Aos membros membros da banca: Paulo Cunha, Virgílio Almeida, Ricardo Massa, Avelino Zorzo & Nelson Rosas pela contribuições para a forma final do documento.

Grupo MODCS : **http://www.modcs.org/**, como fonte de intercâmbio proporcionado.

Equipe dos Projetos HP-Gcap e HP-Desdac, pelo profissionalismo do trabalho coletivo vivenciado.

CIN-UFPE, pela excelência acadêmica e suporte logístico oferecido por seu *staff*.

When the moon is in the Seventh House
And Jupiter aligns with Mars
Then peace will guide the planets
And love will steer the stars

This is the dawning of the age of Aquarius
The age of Aquarius
Aquarius!
Aquarius!

Harmony and understanding
Sympathy and trust abounding
No more falsehoods or derisions
Golden living dreams of visions
Mystic crystal revelation
And the mind's true liberation
Aquarius!
Aquarius!

When the moon is in the Seventh House
And Jupiter aligns with Mars
Then peace will guide the planets
And love will steer the stars

This is the dawning of the age of Aquarius
The age of Aquarius
Aquarius!
Aquarius!

—LYRICS BY JAMES RADO AND GEROME RAGNI AND MUSIC BY GALT MACDERMOT (off-Broadway *début* in October 1967)

Quando a lua estiver na Sétima Casa
E Júpiter alinhar-se com o Marte
Então a paz guiará os planetas
E o amor dirigirá as estrelas

Este é traçado da época de Aquários
A época de Aquários
Aquários!
Aquários!

Harmonia e compreensão
Simpatia e confiança abundando
Nenhuma mais falsidade ou derrisões
Vivendo sonhos dourados de visões
Revelação de cristais místicos
E a liberação verdadeira da mente
Aquários!
Aquários!
Quando a lua estiver na Sétima Casa
E Júpiter alinhar-se com o Marte
Então a paz guiará os planetas
E o amor dirigirá as estrelas

Este é traçado da época de Aquários
A época de Aquários
Aquários!
Aquários!

—LETRAS POR JAMES RADO E GEROME RAGNI E MÚSICA POR GALT MACDERMOT ˙*début* fora da Broadway em outubro de 1967

Resumo

O estudo de sistemas através de modelos é baseado em abstrações do mundo real. Portanto, os cálculos resultantes devem ser interpretados com cautela. Quando incertezas ou variabilidades estão associadas com os parâmetros do sistema, a caracterização pontual dos parâmetros pode ser inadequada. As incertezas podem estar associadas a parâmetros que não são conhecidos antecipadamente, especialmente em estágios iniciais de projetos de sistemas. Um dos objetivos deste trabalho é a concepção do formalismo ISPN para análise quantitativa desses sistemas. Para alcançar este objetivo, desenvolveu-se a fundamentação teórica ISPN e foram adaptados algoritmos apropriados para análise dos modelos. A análise intervalar foi usada como método de análise do estado estacionário dos modelos ISPN, nos quais as taxas das transições exponenciais e os pesos das transições imediatas são intervalos. A ISPN é aplicada principalmente para modelar as situações em que os dados de entrada estão num determinado nível da exatidão. As incertezas das taxas são especificadas através de intervalos. Este ambiente de modelagem fornece uma maneira para formalizar e estudar os problemas relacionados à presença das incertezas. Tais incertezas incluem os erros dos dados que ocorrem durante os processos de medida e os erros de arredondamento gerados durante cálculos. O modelo proposto e o método de análise relacionado permitem que a análise de desempenho seja realizada, considerando variações simultâneas nos parâmetros. Os métodos intervalares foram aplicados na estimativa exterior do conjunto solução para sistemas de equações lineares intervalares resolvidas pela aritmética intervalar clássica e pela aritmética de Kaucher dentro do ambiente do MATLAB *toolbox* INTLAB. Usamos ISPN como uma ferramenta de alto nível para modelagem e análise. O poder de modelagem ISPN foi avaliado em diversos estudos de caso apresentados.

Palavras-chave: Modelagem Estocástica; Redes de Petri Estocásticas; Análise Intervalar

Abstract

In this work interval analysis has been used as an approach to steady state analysis of ISPN (*Interval Stochastic Petri Net*) models in which the exponential rates and immediate transition weights belong to a number of given intervals. The ISPN is mainly applied to model situations in which input data is known with a certain level of accuracy. The uncertainty rates have been subjectively specified as intervals. This framework provides a way to formalize and study problems related to the presence of uncertainties. Such uncertainties include data errors occurring during data measurements and rounding errors generated during calculations. The proposed model and the related method of analysis allow performance analysis to be carried out, considering simultaneous variations in parameters. Interval methods have been applied to the outer estimation of solution sets for real interval linear systems of equations solved by the classical interval arithmetic and by Kaucher arithmetic within MATLAB toolbox INTLAB framework. We use ISPN as a high-level modeling tool for modeling and analysis. The ISPN modeling power was verified in several presented case studies.

Keywords: Stochastic Modeling, Stochastic Petri Nets, Interval Analysis.

Sumário

1 Introdução **1**
1.1 Objetivo e Contribuições 2
1.2 Apresentação da Tese 2

2 Conceitos Básicos **5**
2.1 Análise Intervalar 5
2.1.1 Notação 5
2.1.2 Aritméticas intervalares 6
2.1.2.1 Aritmética intervalar clássica 6
2.1.2.2 Aritmética intervalar completa de Kaucher 7
2.1.3 Métodos de Solução para Sistemas de Equações Intervalares 10
2.1.3.1 *Conjunto solução unida* ***USS*** 11
2.2 Redes de Petri 12
2.2.1 Elementos básicos de uma rede de Petri 13
2.2.2 Definições nas PN 14
2.2.2.1 Regras de disparo de uma rede de Petri 16
2.2.3 Redes de Petri e modelagem de sistemas 17
2.2.3.1 Concorrência (ou Paralelismo) 18
2.2.3.2 Sincronização 18
2.2.4 Recursos limitados 19
2.2.4.1 Problema produtor/consumidor 19
2.2.4.2 Exclusão mútua 21
2.2.5 Propriedades das Redes de Petri 21
2.2.6 Análise de alcançabilidade 22
2.2.6.1 Geração do grafo de alcançabilidade 22
2.2.7 Extensões 25
2.2.7.1 Multiplicidade do arco 25
2.2.7.2 Arco inibidor 25
2.2.7.3 Arco inibidor múltiplo 26
2.2.7.4 Níveis de prioridade 26
2.2.7.5 Funções de habilitação 27
2.3 Redes de Petri Estocásticas 27
2.3.1 Transições Temporizadas Distribuídas Genericamente − Redes de Petri Estocásticas $(GDTT - SPN)$ 28
2.3.2 O Modelo Básico SPN 30

2.3.3 SPN Generalizada (GSPN) 34
2.3.3.1 Análise do estado estacionário 36
2.4 Análise GSPN 37
2.4.1 Análise de Desempenho de Sistema de Manufatura Automático 37
2.4.1.1 Análise GSPN de uma única máquina 37
2.4.1.2 Análise GSPN de uma linha de montagem com duas máquinas e um *buffer* 37
2.4.2 Análise GSPN para o Desempenho de um Sistema Paralelo 39
2.5 Considerações finais 42

3 Trabalhos relacionados 45
3.1 Sistemas nebulosos 45
3.1.1 Aritmética nebulosa 45
3.1.2 Cortes-alfa e a aritmética intervalar 46
3.1.3 Comparação dos métodos 47
3.2 Modelos intervalares em avaliação de desempenho 48
3.3 Considerações Finais 50

4 Modelo ISPN 51
4.1 Formalismo ISPN 51
4.2 Análise Estacionária da ISPN 53
4.2.1 Geração do IERG 53
4.2.2 Eliminação das Marcações Voláteis 54
4.2.3 Análise intervalar do estado estacionário 55
4.2.4 Índices intervalares de desempenho 56
4.3 Exemplos de modelos ISPN 57
4.3.1 **Exemplo 1: Análise de ISPN de uma Única Máquina** 57
4.3.2 **Exemplo 2: Análise do modelo ISPN de Duas máquinas e um *buffer* em uma Linha de Montagem** 61
4.4 Considerações Finais 67

5 Algoritmos para Análise ISPN do Estado Estacionário 69
5.1 Eliminação gaussiana intervalar 69
5.2 Algoritmo de Grassmann intervalar 71
5.3 O Método de Hansen-Bliek-Rohn-Ning-Kearfott-Neumaier 73
5.4 Método de Krawczyk 75
5.5 A função **verifylss** 77
5.6 As funções **verenclinthull** e **verinthull** 77
5.7 Método subdiferencial de Newton desenvolvido por Shary 78
5.7.1 Códigos de computador 81
5.8 Considerações finais 82

6 Estudo de Casos **83**
6.1 **Estudo de caso 1:** Modelo ISPN de um Sistema Paralelo 83
6.2 **Estudo de caso 2:** Cálculo de Dependabilidade com ISPN 85
6.3 **Estudo de caso 3:** Métodos para obtenção do vetor *p.m.f.* intervalar. 91
6.3.1 **Duas máquinas e um *buffer* em uma Linha de Montagem** 92
6.3.2 **Sistema Paralelo** 93
6.4 **Estudo de caso 4:** A aritmética Kaucher na análise ISPN 95
6.4.1 **"Sistema paralelo"** 97
6.5 Considerações finais 101

7 Conclusão e Trabalhos Futuros **103**
7.1 Observações para a conclusão e trabalhos futuros 105

A Modelos **119**
Não matemáticos 120
Matemáticos 120
A.1 Modelagem e Incertezas 120
A.2 Modelos Intervalares 121
A.3 Observações 122

B Processos estocásticos **123**
B.1 Cadeias de Markov 125
B.1.1 Cadeias de Markov de tempo contínuo 125
B.2 Observações 126

C Função de Inclusão Intervalar **127**
C.1 Computação de funções intervalares racionais 129
C.2 Observações 131

D Terminologia da probabilidade intervalar **133**
D.1 Observações 134

E ICTMC **135**
E.1 Observação 137

F Listagem de arquivos-M **139**
F.1 Observação 143

G Repositórios **145**

H Ferramentas de Software na Análise Intervalar **147**

Lista de Figuras

2.1 Soluções aproximadas da **USS** de um sistema intervalar de equações lineares. 12
2.2 Exemplo de uma Rede de Petri. 13
2.3 Exemplo de uma Rede de Petri. 14
2.4 Modificação do número de marcas no lugar p_i subsequente ao disparo da transição t_k. 16
2.5 Um modelo de PN com funções da entrada e de saída. 17
2.6 PN que modela duas atividades paralelas. 18
2.7 PN que modela duas atividades paralelas com sincronização. 18
2.8 Diagrama de bloco e PN de um *buffer* com tamanho finito. 19
2.9 O problema produtor/consumidor com *buffer* ilimitado. 20
2.10 O problema produtor/consumidor com *buffer* de tamanho finito. 20
2.11 O problema da exclusão mútua: duas tarefas paralelas com um recurso comum. 21
2.12 O grafo de alcançabilidade $\mathscr{G}_R(M_0)$ da PN da Figura 2.5. 23
2.13 O grafo de alcançabilidade $\mathscr{G}_R(M_0)$ da PN da Figura 2.11. 24
2.14 Geração da árvore de alcançabilidade para a PN da Figura 4.9 com buffer ilimitado. 24
2.15 Arco inibidor. 25
2.16 O problema da exclusão mútua da Figura 2.11, com atribuição de prioridade. 26
2.17 Arco inibidor múltiplo. 26
2.18 Modelo SPN de um abajur com a lâmpada que está desligada no estado atual. 32
2.19 Diagrama de transição de estado do modelo SPN de um abajur. 33
2.20 Construção do gráfico de alcançabilidade tangível 35
2.21 Modelo GSPN de uma única máquina 38
2.22 Modelo GSPN de linha de montagem com duas máquinas e um *buffer* na TimeNET 39
2.23 Modelo GSPN de um Sistema Paralelo na TimeNET 40

3.1 Função de pertinência para um número nebuloso triangular com suporte $A[0] = [-2;\ 6]$ e vértice em $x = 2$. A pertinência de -1 é 0,25. 46

4.1 O módulo da máquina simples 58
4.2 Grafo de Alcançabilidade e a cadeia de Markov intervalar embutida 59
4.3 Cadeia de Markov embutida reduzida intervalar 59
4.4 Modelo ISPN de Duas máquinas e um *buffer* em uma Linha de Montagem $(k = 3)$ 61

6.1 A descrição ISPN de um sistema paralelo 84

6.2 O modelo ISPN do multiplexador ADM. 87
6.3 O modelo ISPN do multiplexador SDH 88
6.4 Modelo ISPN de Duas máquinas e um *buffer* $(k = 3)$ 92

B.1 Caminhos de amostragem típicos de processos estocásticos 124

Lista de Tabelas

2.1 Tabela da multiplicação no $I(\mathbb{R})$ 7
2.2 Tabela da multiplicação sobre $\mathbb{KR}$ 9
2.3 Resumo dos mecanismos de memória. 29
2.4 Espaço de Estado do Modelo SPN de um abajur. 32
2.5 Representação do gerador infinitesimal do modelo SPN de um abajur. 32
2.6 Interpretação das marcas nos lugares e dos tempos das transições para o modelo GSPN de uma única máquina 38
2.7 Tempos de retardo do disparo das transições para o modelo GSPN de uma única máquina 38
2.8 Interpretação das marcas nos lugares e dos tempos das transições para o modelo GSPN de linha de montagem com duas máquinas e um *buffer* 39
2.9 Taxas/tempo de disparo de transição do modelo GSPN de duas máquinas e um *buffer* 40
2.10 Especificações dos parâmetros das transições da GSPN da Figura 2.23 41
2.11 Avaliação de Desempenho, com uso da ferramenta TimeNET, para vários modelos GSPN (fazendo variar o número de marcas no lugar $P1$ da Figura 2.23). A notação de T+V é o número de marcações tangíveis = T + voláteis = V. 42

4.1 Taxas intervalares (intervalos degenerados) de disparo das transições para uma Única Máquina e um *buffer* de uma Linha de Montagem. 58
4.2 Conjunto de alcançabilidade e distribuição de marcas da ISPN da Figura 4.1. 58
4.3 Taxas intervalares (intervalos degenerados) de disparo das transições do modelo ISPN de Duas máquinas e um *buffer* em uma Linha de Montagem. 61
4.4 Conjunto de alcançabilidade e distribuição de marcas da ISPN da Figura 4.4. 62
4.5 Descrição literal do grafo de alcançabilidade da ISPN da Figura 4.4. 62
4.6 Experimento $[TP]$ (Taxa de Produção) $= [\mu_1]$ para o modelo ISPN de Duas máquinas e um *buffer* de uma Linha de Montagem. Resultados obtidos com a ferramenta protótipo ISPN da utilização intervalar média das máquinas. 66
4.7 Utilização intervalar média das máquinas (intervalos resultantes exibidos como ponto médio/raio) para o Modelo Duas máquinas e um *buffer* numa Linha Montagem para três intervalos associados a $[\mu_1]$. 67

6.1 Especificações dos parâmetros das transições intervalares da ISPN da Figura 6.1 84

6.2 Avaliação de Desempenho e cardinalidade do espaço de estado para vários modelos ISPN do sistema paralelo de Figura 6.1. N é o número de marcas do lugar $P_1 = [\mathrm{E}]\{\#P_1\}$. A notação T+V denota o número de marcações *Tangíveis = T* e *Voláteis = V*. Métricas: $\mathbf{NP}_1$ = número médio de marcas no lugar P_1, **System throughput** $= [\mathrm{prob}]\{\#\mathrm{P}_9 > 0\} \cdot [\lambda]_{I/O}$. 85
6.3 **P3 933 MHz e 256MB de RAM / Linux:** tempo total de execução e tempos para execução de tarefas críticas na análise ISPN do sistema paralelo da Figura 6.1. 85
6.4 **P3 933 MHz 256 MB RAM / Windows XP** tempo total de execução e tempos para execução de tarefas críticas na análise ISPN do sistema paralelo da Figura 6.1. 86
6.5 **P4 2.4 GHz 256 MB RAM / Windows XP** tempo total de execução e tempos para execução de tarefas críticas na análise ISPN do sistema paralelo da Figura 6.1. 86
6.6 **P4 2.4 GHz 2 GB RAM / Windows XP** tempo total de execução e tempos para execução de tarefas críticas na análise ISPN do sistema paralelo da Figura 6.1. 86
6.7 **P4 2.4 GHz 256 MB RAM / Windows XP** tempo total de execução e tempos para execução de tarefas críticas na análise ISPN do sistema paralelo da Figura 6.1. 87
6.8 Expressões de guarda para o modelo agregado MUX ADM. 88
6.9 Expressões de guarda para o modelo agregado MUX SDH GPT com Pswitch. 89
6.10 Tempos das transições exponenciais para o MUX ADM. 89
6.11 Tempos das transições exponenciais para o MUX SDH GPT com Pswitch. 90
6.12 Disponibilidade do MUX ADM (Os intervalos finos (degenerados) correspondem ao ponto central dos intervalos espessos). 91
6.13 Disponibilidade do MUX SDH GPT com Pswitch (Os intervalos finos (degenerados) correspondem ao ponto central dos intervalos espessos). 91
6.14 Taxas intervalares de disparo das transições do modelo ISPN de Duas máquinas e um *buffer* em uma Linha de Montagem. 93
6.15 Tempos médios para solução dos sistemas de equações lineares intervalares e os intervalos relativos às utilizações (estas medidas foram obtidas através de um P4 2.4 GHz 256 MB RAM / Windows XP). O modelo ISPN de duas máquinas e um *buffer* em uma Linha de Montagem (Figura 6.4) tem 16 equações lineares intervalares. 94
6.16 A cardinalidade do espaço de estado do modelo ISPN do "sistema paralelo" para marcações iniciais diferentes do lugar p_1 (Figura 6.1). O T+V denota o número de marcações *tangíveis* $= T$ mais *voláteis* $= V$. N é o número inicial de marcas no lugar p_1. 94
6.17 Tempos médios para solução dos sistemas de equações lineares intervalares (estas medidas foram obtidas através de um P4 2.4 GHz 256 MB RAM / Windows XP). A família de sistemas paralelos do modelo ISPN tem um número de equações lineares intervalares que variam de 20 a 825. 95

6.18 Avaliação de Desempenho para o sistema paralelo com 196 e 825 marcações tangíveis. Métricas: $\mathbf{NP}_1$ = número médio de marcas no lugar P_1, **System throughput** = $Prob\left\{(m(P_9 > 0) \cdot t_{I/O}\right\}$. 96
6.19 Tempos médios para solução dos sistemas de equações lineares intervalares (estas medidas foram obtidas através de um P4 2.4 GHz 256 MB RAM / Windows XP). A família de sistemas paralelos do modelo ISPN tem um número de equações lineares intervalares que variam de 20 a 825. 97
6.20 Avaliação de Desempenho para o sistema paralelo com marcações tangíveis variando de 20 a 825. Métricas: $\mathbf{NP}_1$ = número médio de marcas no lugar P_1 = $[\mathrm{E}]\{\#P_1\}$, **System throughput** = = $[\mathrm{prob}]\{\#\mathrm{P}_9 > 0\} \cdot [\lambda]_{I/O}$. 99
6.21 Sistemas de equações lineares intervalares resolvidas por algoritmos usando $\mathbb{KR}$ e $I(\mathbb{R})$. Intervalos da média de espera por processamento e da produtividade do sistema para o modelo ISPN do "sistema paralelo" com marcações iniciais diferentes no lugar p_1 (Figura 6.1). O **T** é o número de *marcações tangíveis = número das equações*. 100

B.1 Processo Estocástico 124

Lista de Siglas ou Abreviaturas

COPRIN - (*Contraintes, OPtimisation, Résolution par INtervalles*) [21]

CSS - (*Control Solution Set*) Conjunto Solução Controle

CTMC - (*Continuous-Time Markov Chain*) Cadeia de Markov de Tempo Contínuo

GDTT-SPN - (*Generally Distributed Timed Transition - Stochastic Petri Nets*) Transições Temporizadas Distribuídas Genericamente - Redes de Petri Estocásticas

IBGE - Instituto Brasileiro de Geografia e Estatística

ICTMC - (*Interval Continuous-Time Markov Chain*) Cadeia de Markov Intervalar de Tempo Contínuo

IERG - (*Interval Extended Reachability Graph*) Grafo de Alcançabilidade Estendido Intervalar

IGE - (*Interval Gaussian Elimination*) Eliminação Gaussiana Intervalar

IGS - (*Interval Gauss-Seidel*) Gauss-Seidel intervalar

INRIA - *(Institut National de Recherche en Informatique et Automatique)* Instituto Nacional de Pesquisa em Informática e Automação (França)

INPC - Índice Nacional de Preço ao Consumidor - IBGE

MVA - (*Mean Value Analysis*) Análise de Valores Médios

ISPN - (*Interval Stochastic Petri Net*) Rede de Petri Estocástica Intervalar

p.d.f. - (*probability density function*) função densidade de probabilidade

p.m.f. - (*probability mass function*) função massa de probabilidade

PN - (*Petri Nets*) Redes de Petri

TSS - (*Tolerance Solution Set*) Conjunto Solução Tolerância

USS - (*United Solution Set*) Conjunto Solução Unida

Lista de Símbolos

$\mathbb{N} = \{0;\ 1;\ 2;\ \cdots\}$ - Conjunto dos números Naturais (incluído o zero)

$\mathbb{R}$ - Conjunto dos números reais

$\mathbb{KR}$ - Conjunto de todos os pares de números em $\mathbb{R}$

$I(\mathbb{R})$ - Conjunto de todos os intervalos próprios em $\mathbb{R}$, $[\underline{x};\ \overline{x}] \mid \underline{x},\ \overline{x} \in \mathbb{R},\ \underline{x} \leq \overline{x}$

$\overline{I(\mathbb{R})}$ - Conjunto de todos os intervalos impróprios com intervalos ordenados decrescentemente em $\mathbb{R}$, $\{[\underline{x};\ \overline{x}] \mid \underline{x},\ \overline{x} \in \mathbb{R},\ \underline{x} \geq \overline{x}\}$

$I(\mathbb{F})$ - Conjunto de todos intervalos próprios em $\mathbb{F}$

$\triangleq$ - igual por definição

$\triangle$ - arredondamento para cima

$\triangledown$ - arredondamento para baixo

$[x] = [\underline{x};\ \overline{x}]$ - intervalo escalar

$[\mathbf{x}] = [\underline{\mathbf{x}};\ \overline{\mathbf{x}}]$ - intervalo vetorial (ou caixa)

$[\mathbf{A}] = [\underline{\mathbf{A}};\ \overline{\mathbf{A}}]$ - intervalo matricial

$inf([x])$ - limite inferior de $[x]$

$sup([x])$ - limite superior de $[x]$

$mid([x]) = [\check{x}]$ centro de $[x]$

$w([x])$ largura de $[x]$

$rad([x])$ raio de $[x]$

$P = \{p_1; p_2; \cdots p_n\}$ - Conjunto de lugares

$T = \{t_1; t_2; \cdots t_m\}$ - Conjunto de transições

I - (*Input matrix*) Matriz de entradas

O - (*Output matrix*) Matriz de saídas

H - é uma matriz das multiplicidades, dependentes da marcação, dos arcos inibidores

Π - é um vetor de prioridades que associa a menor prioridade(0) para as transições temporizadas e prioridades superiores (≥ 1) para as transições imediatas

G - vetor que atribui uma condição de guarda relacionada as marcações dos lugares para uma dada transição

M_0 - vetor que atribui uma a marcação inicial dos lugares (o estado inicial)

$Atts$ - conjunto dos atributos para as transições

$[Dist]$ - função de distribuição intervalar do disparo, com possível dependência da marcação

$\mathscr{F}$ - conjunto de funções de distribuição intervalares positivas, isto é, com um suporte no $[0, \infty)$

$[W]$ - peso intervalar com possível dependência da marcação

CAPÍTULO 1

Introdução

Os sistemas de computação, as linhas de produção e os sistemas de comunicação processam requisições de tarefas complexas com serviços de demanda que são geralmente de natureza aleatória. Os projetistas de sistemas precisam de métodos e técnicas para quantificar fatores que os auxiliam no projeto do sistema, tais como desempenho, confiabilidade e disponibilidade.

As medidas de desempenho são geralmente realizadas no sistema em que está sendo avaliado. Contudo, durante a fase de projeto, o sistema não está disponível para tais experimentos, e índices de desempenho de um dado projeto precisam ser previstos para verificar se eles atendem os requisitos do projeto para se tomar as decisões necessárias. Os modelos abstratos são usados por projetistas para estimar desempenho. Uma classe de modelos muito utilizada para os sistemas com serviços de demanda aleatória é a classe de modelos de eventos discretos [10].

As cadeias de Markov são usadas em análise de desempenho desde a década de 1950 [10]. Os avanços em técnicas de solução numéricas, geração de espaço de estados automático e a disponibilidade de softwares conduzem à ampla aplicação das cadeias de Markov em análise de desempenho e dependabilidade (confiabilidade e disponibilidade combinadas). Os estados de uma cadeia de Markov representam várias situações de interesse para o sistema em estudo. Através dos estados é possível avaliar se os recursos que estão sendo utilizados falharam, se as tarefas estão esperando para usar um recurso, o número de tarefas concorrentes, e assim por diante. As transições entre estados são rotuladas com probabilidades de transição para cadeias de Markov de tempo discreto ou taxas de transição para cadeias de Markov de tempo contínuo.

O comportamento estacionário [10, 12] é normalmente estudado através da solução de sistema de equações lineares com o número de equações igual ao número de estados. O comportamento dependente do tempo, chamado transiente [10, 12], de uma cadeia de Markov de tempo contínuo é estudado através de um sistema de equações diferencias lineares de 1ª ordem. As soluções das equações fornecem as probabilidades nos estados da cadeia de Markov, dos quais as medidas de desempenho são calculadas. O número de estados de um sistema pode ser muito grande, portanto, gerações automáticas de espaço de estados e métodos numéricos de soluções são necessários.

A disponibilidade atual de ferramentas de software para a avaliação de desempenho permite que se escondam as questões técnicas para o usuário final. Os usuários especificam seus modelos usando alguma linguagem de alto nível, suportada por ferramentas como PEPSY-QNS [126], TimeNET 4.0 [157], SPNP 6.0 [158], SHARPE [158], GreatSPN 2.0 [50], ou PEPA[125], em que o modelo matemático subjacente é gerado automaticamente e analisado.

A avaliação de desempenho de sistemas através de modelos é baseado em abstrações do mundo real. Portanto, os cálculos resultantes devem ser interpretados com cautela. Quando incertezas ou variabilidades estão associadas com os parâmetros do sistema, a caracterização

pontual dos parâmetros pode ser inadequada. As incertezas podem estar associadas a parâmetros que não são conhecidos antecipadamente, especialmente em estágios iniciais de projetos de sistemas. Contudo, projetistas podem ter uma boa ideia sobre o intervalo de valores associados com estes parâmetros, através de experiencias anteriores com sistemas semelhantes. As variabilidades dos parâmetros do sistema podem representar diferentes demandas para um dado componente do sistema no decorrer do tempo. Entretanto, quando parâmetros não são conhecidos precisamente, mas se sabe que variam dentro de um certo intervalo, modelos intervalares podem ser usados e algoritmos implementados usando aritméticas intervalares.

Adotou-se o uso dos intervalos para representar as incertezas relacionadas aos parâmetros nos modelos ISPN (*Interval Stochastic Petri Net*,i.e. , Rede de Petri Estocástica Intervalar) [33, 34, 35, 36, 37]. Consequentemente, o conjunto dos métodos considerados para a geração da cadeia de Markov teve que ser adaptado, usando métodos de análise aritmética intervalar. Nas ISPN [35], as taxas das transições exponenciais e os pesos das transições imediatas são representados por intervalos.

1.1 Objetivo e Contribuições

O objetivo deste trabalho é a concepção do formalismo ISPN para análise quantitativa de sistemas. ISPN (*Interval Stochastic Petri Net*) é definida como um espaço de modelos de Rede de Petri Estocásticas (modelos intervalares), na qual o tempo (taxas) e pesos são representados por intervalos. Para alcançar este objetivo, desenvolveu-se:

1. A fundamentação teórica para modelos markovianos intervalares subjacentes a ISPN;

2. A adaptação dos algoritmos na para geração das cadeias de Markov intervalares subjacentes aos modelos ISPN;

3. Aplicação do modelo proposto e de métodos de análise intervalar para avaliar o desempenho e dependabilidade de sistemas.

A contribuição original contida neste trabalho de tese inclui os seguintes tópicos: proposição e descrição do formalismo ISPN (*Interval Stochastic Petri Net*) como uma extensão intervalar do formalismo GSPN (*Generalized Stochastic Petri Net*) [99]; análise quantitativa dos modelos markovianos intervalares subjacentes aos modelos ISPN, aplicação de diferentes métodos de solução e aritméticas intervalares no cálculo do vetor de intervalos de probabilidades; desenvolvimento de uma ferramenta-protótipo para analisar modelos ISPN, que implementa a geração do espaço de estado, executa a eliminação das marcações voláteis e calcula as medidas intervalares de avaliação de desempenho e dependabilidade.

1.2 Apresentação da Tese

Apresenta-se agora o que será tratado nos capítulos seguintes. O Capítulo 2, **Conceitos Básicos**, é dedicado a noções básicas sobre análise intervalar e Redes de Petri. É uma introdução aos

conceitos e métodos para análise do formalismo ISPN proposto. A aritmética intervalar clássica e a aritmética completa de Kaucher são apresentadas. Os tipos e os métodos de solução dos sistemas de equações lineares intervalares são definidos. Outros tópicos relacionados a conceitos e a análise intervalar da proposta ISPN são aprofundados posteriormente em capítulos e nos apêndices. Adicionalmente, é dada uma introdução de Redes de Petri com definições básicas que conduzem ao ISPN, um membro da família de formalismos de Redes de Petri. No final, é feita a análise GSPN, com o uso da ferramenta TimeNET, de modelos que serão explorados pela análise ISPN proposta. O Capítulo 3, **Trabalhos relacionados**, apresenta contribuições anteriores. Os trabalhos que envolvam tratamento de variabilidades intervalares em modelos markovianos usam o formalismo de redes de fila. Embora os modelos markovianos sejam subjacentes a redes GSPN, os algoritmos utilizados na análise GSPN são, em geral, distintos, ressaltando-se o poder de modelagem do formalismo GSPN. O Capítulo 4, **Modelo ISPN**, define formalmente o modelo ISPN e descreve todas as etapas da análise do estado estacionário que envolve algoritmos intervalares desenhados para geração do IERG (*Interval Extended Reachability Graph* Grafo de Alcançabilidade Estendido Intervalar); eliminação das marcações voláteis; obtenção do vetor de probabilidades intervalares do estado estacionário, culminando com o cálculo das métricas intervalares. O Capítulo 5, **Algoritmos para Análise ISPN do Estado Estacionário**, descreve os algoritmos que foram considerados na obtenção do vetor de probabilidades intervalares do estado estacionário. O Capítulo 6, **Estudo de Casos**, apresenta quatro estudos de casos com o objetivo, entre outros, de aplicar a análise ISPN e assim avaliar a ferramenta protótipo, que utiliza o *framework* MATLAB *toolbox* INTLAB, e os algoritmos usados. O Capítulo 7, **Conclusão e Trabalhos Futuros**, conclui e aponta para futuros trabalhos. Os apêndices complementam a fundamentação, conceituação e métodos relacionados para o estudo ISPN.

CAPÍTULO 2

Conceitos Básicos

Este capítulo apresenta uma breve introdução às Aritméticas Intervalares e às Redes de Petri. A Seção 2.1 apresenta os conceitos básicos das aritméticas intervalares e métodos para solução de sistemas de equações lineares com incertezas nos parâmetros (erros de dados) e cálculos computacionais (erros de arredondamento). Estes conceitos são essenciais para solução dos problemas da análise numérica considerados neste trabalho. Neste contexto,"solucionar um problema" corresponde à obtenção de contornos com limites estreitos (com superestimação reduzida). A Seção 2.2 é dedicada à introdução de redes de Petri com ênfase particular na modelagem de dependabilidade.

2.1 Análise Intervalar

Esta seção introduz alguns conceitos básicos necessários para compreender como as aritméticas intervalares são usadas na análise intervalar dos modelos ISPN. São introduzidos inicialmente alguns conceitos das aritméticas intervalares e os métodos de análise.

2.1.1 Notação

Neste documento todas as matrizes são denotadas por letras maiúsculas em negrito ($\mathbf{A}$), vetores por letras minúsculas em negrito ($\mathbf{a}$), e variáveis escalares por letras minúsculas (a). As variáveis intervalares são delimitadas por colchetes ($[\mathbf{A}]$, $[\mathbf{a}]$, $[a]$). O sublinhado e a barra superior denotam limites inferior e superior, respectivamente. Colchetes angulares $\langle\ ;\ \rangle$ são usados por intervalos definidos por um ponto médio e um raio.

Um intervalo real $[x]$ é um conjunto não vazio de números reais

$$[x] = [\underline{x};\overline{x}] = \{x \in \mathbb{R} : \underline{x} \leq x \leq \overline{x}\} \tag{2.1}$$

onde $\underline{x}$ e $\overline{x}$ são chamados de *infimum (inf)* e *supremum (sup)*, respectivamente.

O conjunto de todos os intervalos $\mathbb{R}$ é denotado por $I(\mathbb{R})$, onde

$$I(\mathbb{R}) = \{[\underline{x};\overline{x}] : \underline{x},\overline{x} \in \mathbb{R} : \underline{x} \leq \overline{x}\}\,. \tag{2.2}$$

O ponto médio de $[x]$ é definido como

$$mid([x]) = \check{x} = \frac{1}{2}(\underline{x}+\overline{x}) \tag{2.3}$$

e o raio de $[x]$ é definido como

$$rad([x]) = \Delta x = \frac{1}{2}(\overline{x} - \underline{x}). \tag{2.4}$$

O raio pode ser usado para definir um intervalo $[x] \in I(\mathbb{R})$. Então, $\langle m;\, r\rangle$ denota um intervalo com ponto médio m e raio r.

$[x;\, x] \equiv x$ é chamado intervalo pontual ou intervalo degenerado. Um intervalo pontual ou degenerado tem raio zero, enquanto um intervalo não degenerado tem um raio maior que zero.

2.1.2 Aritméticas intervalares

As chamadas aritméticas intervalares surgem como ferramentas básicas para a análise intervalar. A aritmética intervalar clássica e algumas generalizações bem conhecidas - aritmética de Kahan [79], aritmética de Hansen (veja [59]), multi-intervalar (isto é, a união finita dos intervalos e as semi-linhas infinitas [167]) - foram propostas para soluções de problemas específicos (como, por exemplo, otimização global). Esta seção primeiramente introduz brevemente a aritmética intervalar clássica e apresenta, logo após, uma descrição da aritmética intervalar completa de Kaucher. Estas aritméticas intervalares são usadas para encontrar estimativas exteriores para os conjuntos de solução de sistemas intervalares de equações lineares na análise do estado estacionário dos modelos ISPN.

2.1.2.1 Aritmética intervalar clássica

Uma forma da aritmética intervalar apareceu, primeiramente, em 1924 e 1931 [16, 168] e, a seguir, posteriormente, em [154]. O desenvolvimento moderno da aritmética intervalar começou com a dissertação de R.E. Moore [113] como um método para determinar erros absolutos de um algoritmo, considerando todos os erros dos dados e arredondamento, após R.E. Moore introduziu a análise intervalar [112]. A aritmética intervalar é uma aritmética definida por conjuntos de intervalos, em vez de conjuntos de números reais. O poder da aritmética intervalar está na sua implantação em computadores. Em particular, computações com arredondamento exterior (dirigidos para $-\infty$ no *infimum* e para $+\infty$ no *supremum* de cada intervalo) permitem cercos rigorosos para as operações e avaliação das funções intervalares.

As operações elementares com intervalos (aritmética intervalar) $[a] = [\underline{a};\, \overline{a}]$ e $[b] = \left[\underline{b};\, \overline{b}\right]$ em $I(\mathbb{F})$ são calculadas explicitamente como:

1. $[a] + [b] = \left[\underline{a} \triangledown \underline{b};\, \overline{a} \triangle \overline{b}\right]$;

2. $[a] - [b] = \left[\underline{a} \triangledown \overline{b};\, \overline{a} \triangle \underline{b}\right]$;

3. $[a] \cdot [b] = \left[min\left(\underline{a} \triangledown \underline{b};\, \underline{a} \triangledown \overline{b};\, \overline{a} \triangledown \underline{b};\, \overline{a} \triangledown \overline{b}\right),\, max\left(\underline{a} \triangle \underline{b};\, \underline{a} \triangle \overline{b};\, \overline{a} \triangle \underline{b};\, \overline{a} \triangle \overline{b}\right)\right]$;

4. $[b]^{-1} = \left[1 \triangledown \overline{b};\, 1 \triangle \underline{b}\right]$ $se\ 0 \notin \left[\underline{b};\, \overline{b}\right]$;

5. $[a]\,/\,[b] = [a]\cdot[b]^{-1}$; onde

$\triangledown$ e $\triangle$ denotam arredondamento dirigido para baixo $(x \to -\infty)$ e para cima $(x \to +\infty)$, respectivamente. Assim, $\underline{a} \triangledown \underline{b}$ significa que é calculada a soma dos limites inferiores e, então, feito o arrendondamento para baixo. Além destas operações, as funções intervalares (exp, log, sin, cos, ...) retornam os limites superiores e inferiores. As operações da aritmética intervalar são definidas para o cálculo exato [112]. As computações da máquina são afetadas por erros de arredondamento. Consequentemente, as fórmulas foram modificadas, a fim de considerar o chamado arredondamento dirigido [75].

Através dos testes de sinal, nos pontos extremos de $[a]$ e $[b]$ do item (3) da definição, são produzidos nove casos para gerar os pontos extremos de $[a]\cdot[b]$. Em oito destes casos, somente um produto real é necessário para cada ponto das extremidades $[a]\cdot[b]$. O nono caso requer dois produtos reais para cada extremidade de $[a]\cdot[b]$. A Tabela 2.1 indica os resultados de $[a]\cdot[b]$.

Tabela 2.1 Tabela da multiplicação no $I(\mathbb{R})$

	$[b] < 0^1$	$0 \in [b]^2$	$[b] > 0^3$
$[a] < 0$	$[\overline{a}\cdot\overline{b};\ \underline{a}\cdot\underline{b}]$	$[\underline{a}\cdot\overline{b};\ \underline{a}\cdot\underline{b}]$	$[\underline{a}\cdot\overline{b};\ \overline{a}\cdot\underline{b}]$
$0 \in [a]$	$[\overline{a}\cdot\underline{b};\ \underline{a}\cdot\underline{b}]$	$[min(\underline{a}\cdot\overline{b};\ \overline{a}\cdot\underline{b});\ max(\underline{a}\cdot\underline{b};\ \overline{a}\cdot\overline{b})]$	$[\underline{a}\cdot\overline{b};\ \overline{a}\cdot\overline{b}]$
$[a] > 0$	$[\overline{a}\cdot\underline{b};\ \underline{a}\cdot\overline{b}]$	$[\overline{a}\cdot\underline{b};\ \overline{a}\cdot\overline{b}]$	$[\underline{a}\cdot\underline{b};\ \overline{a}\cdot\overline{b}]$

1- $[b] < 0$ significa que $\underline{b} < 0$ e $\overline{b} < 0$.
2- $0 \in [b]$ significa que $\underline{b} < 0$ e $\overline{b} > 0$.
3- $[b] > 0$ significa que $\underline{b} > 0$ e $\overline{b} > 0$.

A referência [67] apresenta uma tabela expandida da divisão no $I(\mathbb{F})$. A entrada da tabela para a divisão por um intervalo que contém zero usa a notação $[+NaN; -NaN]$[1] (intervalo impróprio) como codificação especial do intervalo vazio. Desde que o resultado de uma operação intervalar é suposto sempre para ser um único intervalo novamente, o resultado que consiste da união de dois intervalos é fornecido e representado como o intervalo impróprio $[\overline{a}\triangledown\underline{b};\ \overline{a}\triangle\overline{b}]$ *e* $[\underline{a}\triangledown\overline{b};\ \underline{a}\triangle\underline{b}]$ (com codificação especial para o resultado $(-\infty; \overline{[a]/[b]}] \cap [\underline{[a]/[b]}; +\infty)$. Nestes casos particulares, o limite esquerdo é mais elevado que o limite direito.

O teorema a seguir fornece a propriedade da inclusão como importante resultado. Rall [129, 55] dá o nome de teorema fundamental da análise intervalar:

Theorem 2.1.1 (Teorema fundamental)**.** *Se a função* $f([x]_1;[x]_2;[x]_3;\cdots[x]_n)$ *é uma expressão com um número finito de intervalos* $[x]_1,[x]_2,[x]_3,\cdots[x]_n \in I(\mathbb{R})$ *e operações intervalares* $(+;\ -;\ \times;\ \div)$ *, e se* $[w]_1 \subseteq [x]_1, [w]_2 \subseteq [x]_2, [w]_3 \subseteq [x]_3,\ \cdots,\ [w]_n \subseteq [x]_n$ *então:* $f([w]_1;[w]_2;[w]_3;\ \cdots;[w]_n) \subseteq f([x]_1;[x]_2;[x]_3;\ \cdots;[x]_n)$.

2.1.2.2 Aritmética intervalar completa de Kaucher

A falta de completude nas estruturas algébricas e na ordem estimulou, naturalmente, diversas tentativas de melhorar a aritmética intervalar clássica, para criar uma aritmética intervalar

[1] *NaN* - Not a Number

"nova", tomando com base $I(\mathbb{R})$ ao manter a inclusão de monotonicidade. Uma delas foi bem estabelecida nos anos 70 por E. Kaucher [124, 64, 62, 63], resultando em um sistema algébrico que é chamada de aritmética intervalar completa no $\mathbb{KR}$, ou aritmética intervalar de Kaucher. Mais tarde, E. Gardeñes, M. Sainz e seus colegas de trabalho estudaram essa aritmética (que eles chamaram aritmética intervalar modal) e estabeleceram algumas de suas propriedades úteis tendo também implantado um sistema computacional de trabalho [38, 39, 40, 41, 42].

Os intervalos generalizados são os intervalos nos quais os limites não são restritos para estarem ordenados, $\underline{x} \leq \overline{x}$. Os elementos de $\mathbb{KR}$ são os pares de números reais e são divididos em dois subconjuntos:

- O conjunto de *intervalos próprios*, com limites ordenados crescentemente. Estes intervalos são identificados como intervalos clássicos: $I(\mathbb{R}) = \{[\underline{x};\, \overline{x}] \mid \underline{x},\, \overline{x} \in \mathbb{R},\, \underline{x} \leq \overline{x}\}$.
- O conjunto de *intervalos impróprios* com intervalos ordenados em ordem decrescente. Denota-se através de $\overline{I(\mathbb{R})} = \{[\underline{x};\, \overline{x}] \mid \underline{x},\, \overline{x} \in \mathbb{R},\, \underline{x} \geq \overline{x}\}$.

O conjunto de intervalos degenerados $\mathbb{R} = \{[\underline{x};\, \overline{x}] \mid \underline{x},\, \overline{x} \in \mathbb{R},\, \underline{x} = \overline{x}\} = I(\mathbb{R}) \cap \overline{I(\mathbb{R})}$. Os intervalos degenerados são identificados como reais.

Os intervalos próprios e impróprios, as duas "partes" de $\mathbb{KR}$, trocam de lugares como o resultado da dualização pelo mapeamento dual: $\mathbb{KR} \to \mathbb{KR}$, tal que:

$$dual[\underline{x};\, \overline{x}] = [\overline{x};\, \underline{x}].$$

Será útil passar de intervalos próprios para impróprios utilizando as seguintes operações:

- a projeção própria: $prop[\underline{x};\, \overline{x}] = [min\{\underline{x};\, \overline{x}\}\,;\, max\{\underline{x};\, \overline{x}\}] \in I(\mathbb{R})$;
- a projeção imprópria: $imp[\underline{x};\, \overline{x}] = [max\{\underline{x};\, \overline{x}\}\,;\, min\{\underline{x};\, \overline{x}\}] \in I(\mathbb{R})$.

Os intervalos generalizados são parcialmente ordenados por uma inclusão que estende a inclusão de intervalos clássicos. Como na aritmética intervalar clássica, dados dois intervalos generalizados, $[x] = [\underline{x};\, \overline{x}]$ e $[y] = [\underline{y};\, \overline{y}]$, a ordem de inclusão sobre $\mathbb{KR}$ é definida como segue:

$$[x] \subseteq [y] \quad \overset{def}{\Longleftrightarrow} \quad \underline{x} \geq \underline{y} \;\wedge\; \overline{x} \leq \overline{y} \tag{2.5}$$

Por exemplo, $[-2;1] \subseteq [-2,1;1,1]$ (isto corresponde à inclusão de conjunto), $[1,1;-2,1] \subseteq [1,1;-2,1]$ (a inclusão entre os conjuntos subjacentes de real é invertida para intervalos impróprios) e $[3;1] \subseteq [-1;2]$. Também $[3;\ -5] \subseteq [2;\ -3] \subseteq [-2;\ -2] \subseteq [-5;\ 1] \subseteq [-6;\ 2]$. É inútil interpretar esta inclusão. Servirá unicamente como um meio cômodo para tratar três casos diferentes, simultaneamente. Sendo assim, consideremos $[x] \in \mathbb{KR}$ e $[y] \in \mathbb{KR}$.

Provar a inclusão seguinte

$$[x] \subseteq [y]$$

equivale, por definição, a provar as três implicações seguintes:

$$[x] \in I(\mathbb{R}) \longrightarrow ([y] \in I(\mathbb{R})\ e\ [x] \subseteq [y]); \tag{2.6}$$

Tabela 2.2 Tabela da multiplicação sobre $\mathbb{KR}$

	$[b] \in -\mathscr{P}$	$[b] \in \mathscr{Z}$	$[b] \in \mathscr{P}$	$[a] \in dual\ \mathscr{Z}$
$[a] \in -\mathscr{P}$	$[\overline{a}\cdot\overline{b};\ \underline{a}\cdot\underline{b}]$	$[\underline{a}\cdot\overline{b};\ \underline{a}\cdot\underline{b}]$	$[\underline{a}\cdot\overline{b};\ \overline{a}\cdot\underline{b}]$	$[\overline{a}\cdot\overline{b};\ \overline{a}\cdot\underline{b}]$
$[a] \in \mathscr{Z}$	$[\overline{a}\cdot\underline{b};\ \underline{a}\cdot\underline{b}]$	$[min(\underline{a}\cdot\overline{b};\ \overline{a}\cdot\underline{b}),$ $max(\underline{a}\cdot\underline{b};\ \overline{a}\cdot\overline{b})]$	$[\underline{a}\cdot\overline{b};\ \overline{a}\cdot\overline{b}]$	0
$[a] \in \mathscr{P}$	$[\overline{a}\cdot\underline{b};\ \underline{a}\cdot\overline{b}]$	$[\overline{a}\cdot\underline{b};\ \overline{a}\cdot\overline{b}]$	$[\underline{a}\cdot\underline{b};\ \overline{a}\cdot\overline{b}]$	$[\underline{a}\cdot\underline{b};\ \underline{a}\cdot\overline{b}]$
$[a] \in dual\ \mathscr{Z}$	$[\overline{a}\cdot\overline{b};\ \underline{a}\cdot\overline{b}]$	0	$[\underline{a}\cdot\underline{b};\ \overline{a}\cdot\underline{b}]$	$[max(\underline{a}\cdot\underline{b};\ \overline{a}\cdot\overline{b}),$ $min(\underline{a}\cdot\overline{b};\ \overline{a}\cdot\underline{b})]$

$$[y] \in \overline{I(\mathbb{R})} \Longrightarrow ([x] \in \overline{I(\mathbb{R})}\ e\ prop[y] \subseteq prop[x]); \tag{2.7}$$

$$[x] \in \overline{I(\mathbb{R})}\ e\ [y] \in I(\mathbb{R}) \Longrightarrow [y] \cap prop[x] \neq \emptyset). \tag{2.8}$$

A operação de adição e a operação multiplicação por números reais são definidas sobre $\mathbb{KR}$ por:

$$\begin{gathered} [x]+[y] := [\underline{x}+\underline{y};\ \overline{x}+\overline{y}], \\ \lambda\cdot x := \begin{cases} [\lambda\cdot\underline{x};\ \lambda\cdot\overline{x}], & se\ \lambda \in \mathbb{R}^+ \\ [\lambda\cdot\overline{x};\ \lambda\cdot\underline{x}], & caso\ contrário. \end{cases} \end{gathered} \tag{2.9}$$

A multiplicação da aritmética Kaucher definida sobre $\mathbb{KR}$ pode ser definido através da chamada tabela de Cayley (usada para descrever a estrutura de um grupo finito), *i.e.*, uma tabela quadrada que lista todas as combinações possíveis dos operandos em entradas e nos resultados correspondentes mostrados na Tabela 2.2 [63]. Separemos sobre $\mathbb{KR}$ os seguintes subconjuntos:

$$\begin{aligned} \mathscr{P} &:= \{[x] \in \mathbb{KR} \mid (\underline{x} \geq 0)\ \&\ (\overline{x} \geq 0)\}, && -intervalos\ positivos; \\ \mathscr{Z} &:= \{[x] \in \mathbb{KR} \mid (\underline{x} \leq 0 \leq \overline{x})\}, && -intervalos\ contendo\ zero; \\ -\mathscr{P} &:= \{[x] \in \mathbb{KR} \mid -[x] \in \mathscr{P}\}, && -intervalos\ negativos; \\ dual\ \mathscr{Z} &:= \{[x] \in \mathbb{KR} \mid dual\ [x] \in \mathscr{Z}\}, && -intervalos\ contendo\ zero. \end{aligned}$$

O conjunto completo $\mathbb{KR} = \mathscr{P} \cap -\mathscr{P} \cap \mathscr{Z} \cap dual\ \mathscr{Z}$.

Cada elemento $[x]$ de $\mathbb{KR}$ tem um único elemento algébrico oposto original, denotado por "$opp\ [x]$", e

$$[x] + opp\ [x] = 0 \ \Rightarrow\ opp\ [\underline{x};\ \overline{x}] = opp\ [-\overline{x};\ -\underline{x}].$$

Por exemplo, $opp\ [-1;3] = [3;-1]$, i.e., um intervalo impróprio. Então seguem as propriedades que são satisfeitas:

$$\begin{aligned} &opp\ (opp\ [x]) = [x]; \\ &opp\ (\lambda[x]) = \lambda opp\ [x],\ for\ \lambda \in \mathbb{R}; \\ &dual([x]+[y]) = dual[x] + dual[y]; \\ &opp\ ([x]+[y]) = opp\ [x] + opp\ [y]. \end{aligned}$$

Também,

$$\begin{aligned}
&[x] \subseteq [y] \Rightarrow -[x] \subseteq -[y];\\
&dual\ [x] \supseteq dual\ [y];\\
&opp\ [x] \supseteq opp\ [y];\\
&\lambda[x] \subseteq \lambda[y],\ for\ \lambda \in \mathbb{R}.
\end{aligned}$$

A operação inversa para a adição é diferença na aritmética de Kaucher, e denota-se por $\ominus$, tal que

$$[x] \ominus [y] := [x] + opp\ [y].$$

A adição (2.9) é comutativa e associativa. Com respeito à adição, aritmética intervalar completa $\mathbb{KR}$ torna-se um grupo comutativo, que é isomorfo ao grupo aditivo do espaço linear padrão $\mathbb{R}^2$. Como consequência, pode-se rearranjar termos de um lado ao outro lado "com o sinal oposto" em uma equação intervalar (desigualdade, inclusão).

A multiplicação na aritmética de Kaucher permite resultados zero não triviais, por exemplo, $[-3;2] \cdot [3;-1] = 0$. É comutativa e associativa, como em $I(\mathbb{R})$, mas nem todo elemento $[x]$ de $\mathbb{KR}$ tem seu inverso $[x]^{-1}$. Isto é, o grupo com respeito à operação " $\cdot$ " é formado em $\mathbb{KR}$ somente para intervalos $[x]$ com $\underline{x} \cdot \overline{x} > 0$ [62]. Por exemplo, $[2;5]^{-1} = [1/2;\ 1/5]$, enquanto $[-1;\ 4]^{-1}$ não existe.

A operação inversa para a multiplicação é divisão na aritmética de Kaucher, e ela é denotada por $\oslash$, então,

$$[x] \oslash [y] := [x] \cdot [y]^{-1}$$

e,

$$\begin{aligned}
&dual\ [x] \cdot dual\ [y] = dual\ ([x] \cdot [y]),\\
&opp\ [x] \cdot opp\ [y] = dual\ ([x] \cdot [y]),\\
&\lambda([x] \cdot [y]) = (\lambda[x]) \cdot [y] = [x] \cdot (\lambda\ [y])\ para\ \lambda \in \mathbb{R}.
\end{aligned}$$

As operações aritméticas com os vetores e as matrizes são formadas pelos elementos de $\mathbb{KR}$, definidos similarmente aos da aritmética intervalar clássica [60, 119, 114].

2.1.3 Métodos de Solução para Sistemas de Equações Intervalares

A solução de sistemas de equações lineares é sujeita a erros devido à precisão finita da aritmética da máquina e a propagação de erros relacionados aos dados iniciais. Se o dado inicial é conhecido como pertencente a intervalos de variações especificadas, então a aritmética intervalar estabelece o cálculo de intervalos que contém o valor exato [52].

Um sistema de equações lineares é da forma:

$$[\mathbf{A}][\mathbf{x}] = [\mathbf{b}], \tag{2.10}$$

onde $[\mathbf{A}] \in I(\mathbb{R})^{n \times n}$ e $[\mathbf{b}] \in I(\mathbb{R})^n$.

Quando a matrix-$n \times n$, $[\mathbf{A}] = \left[\underline{\mathbf{A}};\overline{\mathbf{A}}\right]$, e o vetor-$n$, $[\mathbf{b}] = \left[\underline{\mathbf{b}};\overline{\mathbf{b}}\right]$, no sistema $[\mathbf{A}] \cdot [\mathbf{x}] = [\mathbf{b}]$ são intervalos, este é usualmente ainda chamado de um sistema de equações lineares intervalar [119], [138]. Mais precisamente, eles não são usualmente considerados um sistema

de equações. O espaço de um intervalo real não é um espaço linear, mas o sistema tem a forma de uma expressão linear. O uso da palavra *equação* é somente justificado quando se considera a solução algébrica do sistema, o que raramente acontece. Em muitos casos, outras definições são consideradas, usualmente, conjuntos de vetores reais (não necessariamente intervalos), definidos como segue (ver e.g. [144]):

*Conjunto solução unida **USS**(United Solution Set)* $= \Sigma_{\exists\exists}([\boldsymbol{A}];[\boldsymbol{b}])$*:*

$$\mathbf{USS} = \left\{\widetilde{\mathbf{x}} \in \mathbb{R}^n | (\exists \widetilde{\mathbf{A}} \in [\mathbf{A}])(\exists \widetilde{\mathbf{b}} \in [\mathbf{b}])(\widetilde{\mathbf{A}} \cdot \widetilde{\mathbf{x}} = \widetilde{\mathbf{b}})\right\} \tag{2.11}$$

Este conjunto solução inclui todas as soluções de cada sistema pontual $\widetilde{\mathbf{A}} \cdot \widetilde{\mathbf{x}} = \widetilde{\mathbf{b}}$, onde **=** é um símbolo para $\widetilde{\mathbf{A}} \cdot \widetilde{\mathbf{x}} \cap \widetilde{\mathbf{b}} \neq \emptyset$ com $\widetilde{\mathbf{A}} \in [\mathbf{A}]$ e $\widetilde{\mathbf{b}} \in [\mathbf{b}]$, isto é, calculando $min\,\{\widetilde{x}_k | \widetilde{\mathbf{x}} \in \Sigma_{\exists\exists}([\mathbf{A}];[\mathbf{b}])\}$ e $max\,\{\widetilde{x}_k | \widetilde{\mathbf{x}} \in \Sigma_{\exists\exists}([\mathbf{A}];[\mathbf{b}])\}$, $k = 1, 2, \ldots, n$.

*Conjunto solução controle **CSS**(Control Solution Set)* $= \Sigma_{\exists\forall}([\boldsymbol{A}];[\boldsymbol{b}])$*:*

$$\mathbf{CSS} = \left\{\widetilde{\mathbf{x}} \in \mathbb{R}^n | (\forall \widetilde{\mathbf{b}} \in [\mathbf{b}])(\exists \widetilde{\mathbf{A}} \in [\mathbf{A}])(\widetilde{\mathbf{A}} \cdot \widetilde{\mathbf{x}} = \widetilde{\mathbf{b}})\right\}. \tag{2.12}$$

*Conjunto solução tolerância **TSS**(Tolerance Solution Set)* $= \Sigma_{\forall\exists}([\boldsymbol{A}];[\boldsymbol{b}])$*:*

$$\mathbf{TSS} = \left\{\widetilde{\mathbf{x}} \in \mathbb{R}^n | (\forall \widetilde{\mathbf{A}} \in [\mathbf{A}])(\exists \widetilde{\mathbf{b}} \in [\mathbf{b}])(\widetilde{\mathbf{A}} \cdot \widetilde{\mathbf{x}} = \widetilde{\mathbf{b}})\right\}. \tag{2.13}$$

Estes conjuntos soluções generalizados originam-se na existência de muitas aplicações significativas e interessantes. As soluções acima não são soluções do sistema original. Elas são conjuntos de soluções reais de expressões relacionais intervalares, com diferentes relações colocadas no lugar do sinal de igualdade. Usando esta convenção, a equação $[\mathbf{A}] \cdot [\mathbf{x}] = [\mathbf{b}]$ tem um conjunto solução $\Sigma_{=}([\mathbf{A}];[\mathbf{b}])$ igual a $\Sigma_{\exists\forall}([\mathbf{A}];[\mathbf{b}]) \cap \Sigma_{\forall\exists}([\mathbf{A}];[\mathbf{b}])$, diferente da solução algébrica. Destas definições segue também que $\Sigma_{\forall\exists}([\mathbf{A}];[\mathbf{b}]) \subseteq \Sigma_{\exists\exists}([\mathbf{A}];[\mathbf{b}])$ e $\Sigma_{\exists\forall}([\mathbf{A}];[\mathbf{b}]) \subseteq \Sigma_{\exists\exists}([\mathbf{A}];[\mathbf{b}])$.

Não existe um nome abreviado para este tipo de fórmula na qual, não somente uma igualdade, mas qualquer outra relação, pode estar de acordo como um conectivo principal. O termo "desigualdade" não é apropriado, pois não se quer excluir igualdade. Kulpa [77] usou a frase "expressão relacional", ou, algumas vezes, simplesmente "relação". Contudo, será mantido aqui o uso da frase sistema de equações lineares intervalares.

2.1.3.1 *Conjunto solução unida **USS***

O conjunto solução unida USS forma um politopo convexo com $2(n^2 - n + 1)$ vértices. O cálculo do intervalo envoltório ("*hull interval*") necessita de um algoritmo "*NP-hard*" [72]. Soluções aproximadas requerem usualmente algoritmos com complexidade polinomial apenas. (ver Figura 2.1) [119, 52, 55].

A forma da solução de um sistema linear intervalar pode ser bastante complexa, mas desde que a aritmética intervalar é usada, só é possível calcular um vetor intervalar que contém a solução verdadeira. O objetivo é conseguir um vetor intervalar solução que é próximo ao menor vetor intervalar contendo a solução verdadeira.

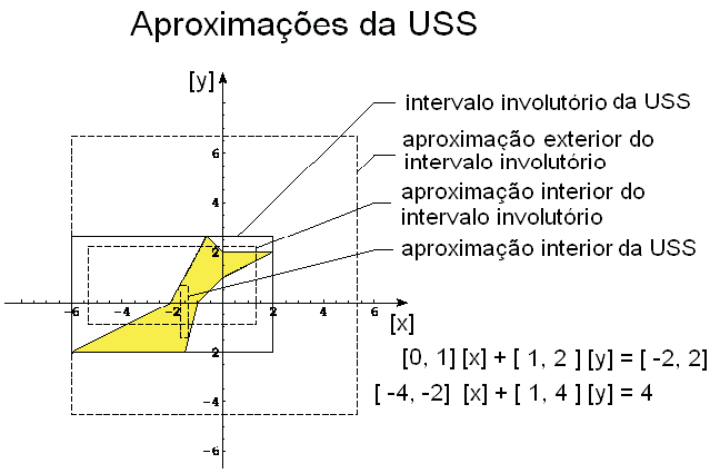


Figura 2.1 Soluções aproximadas da **USS** de um sistema intervalar de equações lineares.

A solução ótima de um Sistema Linear de Equações Algébricas Intervalares é tema constante em pesquisas recentes [91]. O chamado *problema externo (outer problem) para um sistema de equações algébricas intervalares* [146, 145]: *encontrar um vetor intervalar com largura mínima contendo o conjunto* $\Sigma([\mathbf{A}];[\mathbf{b}])$.

O vetor intervalar $[\mathbf{x}]^H = ([x]_1^H;[x]_2^H;\cdots;[x]_n^H)^T$ de largura mínima contendo $\Sigma([\mathbf{A}];[\mathbf{b}])$ é chamado intervalo envoltório (*the interval hull*) do conjunto $\Sigma([\mathbf{A}];[\mathbf{b}])$ ou a solução intervalar ótima:

$$[x]_i^H = [\underline{x}_i^H;\overline{x}_i^H], \qquad \begin{array}{l} \underline{x}_i^H = min\,\{x_i \mid x \in \Sigma([\mathbf{A}];[\mathbf{b}])\} \\ \overline{x}_i^H = min\,\{x_i \mid x \in \Sigma([\mathbf{A}];[\mathbf{b}])\} \\ \quad i = 1,2,\ldots,n. \end{array}$$

Algumas fontes importantes dos algoritmos para resolver as equações lineares do intervalo que usam a aritmética intervalar clássica são apresentadas em [119, 52, 22].

2.2 Redes de Petri

Redes de Petri (PN - *Petri Nets*) é uma família de paradigmas gráficos para a descrição formal das interações lógicas entre partes ou de fluxo de atividades em sistemas complexos. Em relação a outras técnicas da representação gráfica do sistema (como diagramas de bloco, diagramas de transições para Cadeias de Markov ou diagrama de árvores de falhas), a PN é particularmente adequada para representar as interações lógicas, de uma maneira natural, entre as peças ou atividades em um sistema. As situações típicas que podem ser modeladas pelas PN são concorrência e conflito, sequência, desvio condicional e iterações (*loop*), sincronização, compartilhamento de recursos limitados e exclusão mútua.

A teoria de PN teve sua origem na tese de doutorado de C.A. Petri, em 1962[128]. As redes de Petri (PN) [127, 128, 135] foram originalmente desenvolvidas e usadas para o estudo das propriedades qualitativas de sistemas, exibindo características de concorrência e sincronização. A PN original não tinha qualquer noção de tempo. Para análise de desempenho e de depend-

abilidade é necessário introduzir a duração dos eventos associados à transições, com a finalidade de usar o formalismo das PN. O modelo temporizado tem sido explorado extensivamente, seguindo duas linhas principais:

- Durações aleatórias $\longrightarrow$ PN Estocástica (SPN - Stochastic Petri Nets);
- Durações determinísticas ou intervalos $\longrightarrow$ PN Temporizada (TPN - Timed PN).

2.2.1 Elementos básicos de uma rede de Petri

Uma rede de Petri (PN) é um grafo dirigido bipartido, consistindo de dois tipos de nós, lugar e transição, e arcos que permitem conectar somente nós de classes diferentes. Os lugares representam tipicamente condições dentro do sistema modelado. As transições representam eventos que, ocorrendo no sistema, podem causar mudança na sua condição. Os arcos conectam lugares a transições e transições a lugares.

Do ponto de vista formal, uma PN marcada é uma quíntupla $(P;\ T;\ I;\ O;\ M)$, onde [127]:

- $P = (p_1;\ p_2;\ \cdots;\ p_n)$ é o conjunto dos n lugares da PN (traçados como círculos na reapresentação gráfica);
- $T = (t_1;\ t_2;\ \cdots;\ t_m)$ é o conjunto das m transições da PN (traçados como barras na reapresentação gráfica);
- I é a relação de entrada de transição e é representada por meio de arcos dirigidos de lugares para transições;
- O é a relação de saída de transição e é representada por meio de arcos dirigidos de transições para lugares;
- $M = (m_1;\ m_2;\ \cdots;\ m_n)$ é a marcação. A entrada genérica m_i é o número de marcas (traçados como discos) no lugar p_i na marcação M.

Os arcos múltiplos são possíveis na definição das relações de I e de O [127]. A Figura 2.2 é uma PN onde a marcação exibida representa a situação em que o recurso está inativo. Na Figura 2.3 a PN tem o lugar p_3 representando usuário e a marcação representa 4 usuários em espera para utilizar o recurso que está inativo.

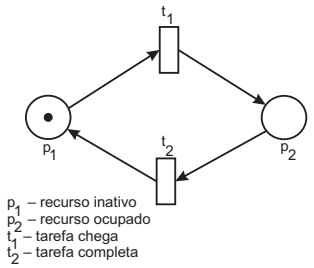


Figura 2.2 Exemplo de uma Rede de Petri.

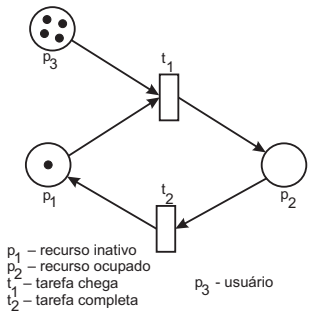
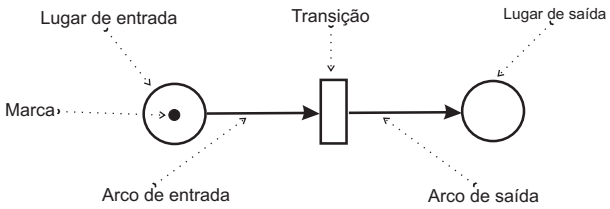


Figura 2.3 Exemplo de uma Rede de Petri.

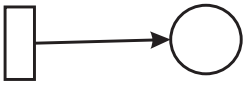
2.2.2 Definições nas PN

- Componentes Básicos de PN:

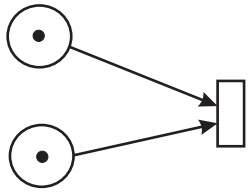


- Arcos de entrada são arcos dirigidos traçados de lugares para transições, representando as condições necessárias que precisam ser satisfeitas para o evento ser ativado.

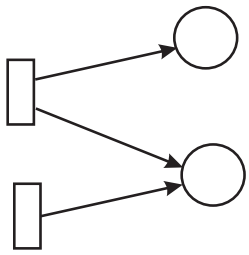
- Arcos de saída são arcos dirigidos traçados de transições para lugares, representando as condições resultantes da ocorrência do evento.



- Lugares de entrada de uma transição são o conjunto de lugares que estão conectados à transição através de arcos de entradas.



- Lugares de saída de uma transição são o conjunto de lugares para o qual existem arcos vindos da transição.

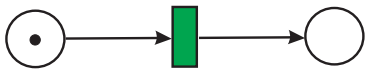


- Marcas são pontos (ou inteiros) associados com lugares; um lugar contendo marca indica que a condição correspondente está satisfeita:

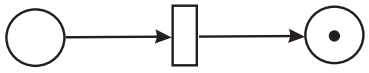
- A marcação de uma rede de Petri é um vetor listando o número de marcas em cada lugar da rede:

$$(m_1;\, m_2;\, \cdots;\, m_P) \quad ; \quad P = (\#de\ lugares)\,.$$

- Quando os lugares de entrada de uma transição têm o número de marcas necessárias, a transição está habilitada.



- Uma transição habilitada pode disparar (o evento ocorre), removendo uma marca de cada lugar de entrada e depositando uma marca em cada um de seus lugares de saída.



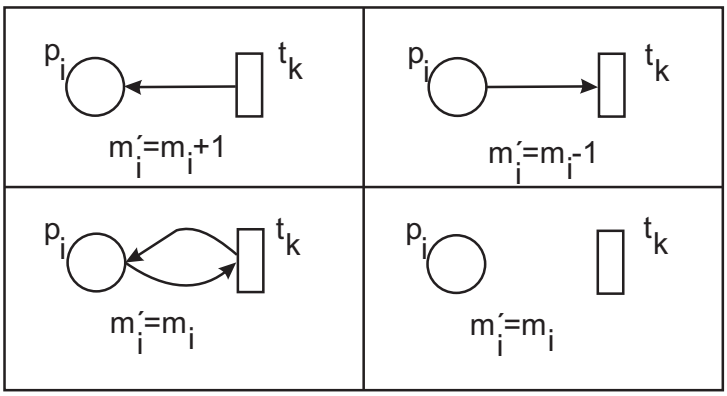


Figura 2.4 Modificação do número de marcas no lugar p_i subsequente ao disparo da transição t_k.

2.2.2.1 Regras de disparo de uma rede de Petri

A dinâmica de uma PN é obtida pelo movimento das marcas pelos lugares por meio das seguintes regras da execução:

- uma transição está habilitada numa marcação M se todos seus lugares de entrada possuem pelo menos uma marca;
- uma transição habilitada dispara removendo uma marca por arco de cada lugar da entrada e adicionando uma marca por arco a cada lugar da saída.

Dada uma marcação inicial M_0, o conjunto de alcançabilidade $\mathscr{R}(M_0)$ é o conjunto de todas as marcações que podem ser obtidas pela aplicação repetida das regras acima. De modo mais formal podemos dizer que a t_k está habilitada numa marcação M se:

$$\text{para todo } p_i \in I(t_k), \quad m_i \geq 1.$$

A marcação M', obtida de M pelo disparo de t_k, é dita ser imediatamente alcançável de M, e a operação do disparo é denotada por $(M\,|t_k\rangle\,M')$. A contagem de marcas em M' é representada de maneira pictórica na Figura 2.4, e dada pelas seguintes relações:

$$M'(p_i) = \begin{cases} M(p_i)+1 & se \;\; p_i \in O(t_k), \;\; p_i \notin I(t_k) \\ M(p_i)-1 & se \;\; p_i \notin O(t_k), \;\; p_i \in I(t_k) \\ M(p_i) & caso\ contrário \end{cases}$$

Examinaremos a geração do conjunto de alcançabilidade da PN da Figura 2.5 dada a marcação inicial $M_0 = (1;0;0;0;0)$. Em M_0, a única transição habilitada é t_1; o disparo de t_1 remove uma marca de p_1 e põe uma marca em p_2 e outra em p_3, produzindo uma nova marcação $M_1 = (0;1;1;0;0)$. Em M_1, as transições t_2 e t_3 estão habilitadas e podem disparar concorrentemente (ou em paralelo, dependendo da semântica de disparo, mas não considerada aqui). O disparo de t_2 conduz a $M_2 = (0;0;1;1;0)$, enquanto o disparo de t_3 conduz a $M_3 = (0;1;0;0;1)$. Em M_2, a única transição habilitada é t_3; o disparo de t_3 remove uma marca

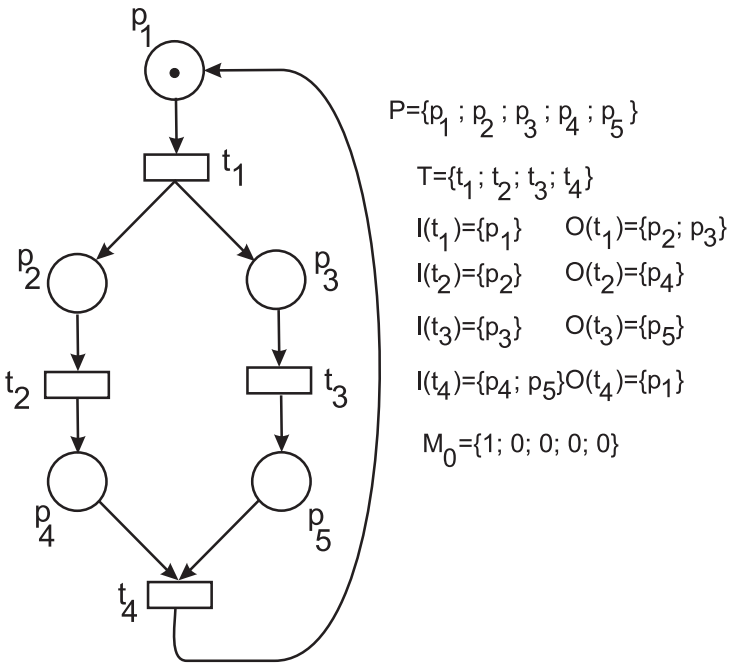


Figura 2.5 Um modelo de PN com funções da entrada e de saída.

de p_3 e põe uma marca em p_5, produzindo uma nova marcação $M_4 = (0;0;0;1;1)$. Em M_3, a única transição habilitada é t_2; o disparo de t_2 remove uma marca de p_2 e põe uma marca em p_4, produzindo também a marcação $M_4 = (0;0;0;1;1)$. Em M_4, a única transição habilitada é t_4; o disparo de t_4 remove uma marca de p_4 e outra de p_5 e põe uma marca em p_1, produzindo marcação inicial $M_0 = (1;0;0;0;0)$. Com todas as possíveis sequências de disparo tendo sido examinadas, o conjunto de alcançabilidade da rede da Figura 2.5 contém 5 elementos $\mathscr{R}(M_0) = M_0;M_1;M_2;M_3;M_4$, obtidas pelas operações de disparos:

Partindo de $M_0 = \{1;\ 0;\ 0;\ 0;\ 0\}$:

$$
\begin{aligned}
&M_0\,|t_1\rangle\,M_1 = \{0;\ 1;\ 1;\ 0;\ 0\}\\
&M_1\,|t_2\rangle\,M_2 = \{0;\ 0;\ 1;\ 1;\ 0\}\\
&M_1\,|t_3\rangle\,M_3 = \{0;\ 1;\ 0;\ 0;\ 1\}\\
&M_2\,|t_3\rangle\,M_4 = \{0;\ 0;\ 0;\ 1;\ 1\}\\
&M_3\,|t_2\rangle\,M_4\\
&M_4\,|t_4\rangle\,M_0
\end{aligned}
$$

2.2.3 Redes de Petri e modelagem de sistemas

Quando PN são usadas para modelar sistemas reais são, às vezes, chamadas redes de Condição/Evento. Lugares identificam as condições das partes do sistema (trabalhando, ocioso, em fila de espera, em estado de falha), e as transições descrevem a passagem de uma condição para outra (fim de uma tarefa, falha, reparo...). Um evento acontece (uma transição dispara) quando todas as condições são satisfeitas (lugares de entrada são marcados) e permitem a ocorrência

ao evento. A ocorrência do evento modifica em todo ou em parte o estado das condições (marcações). O número de marcas em um lugar pode ser usado para identificar o número de recursos que se encontra na condição denotada por aquele lugar. Os exemplos seguintes ilustram situações típicas de interação de atividades que surgem em modelagem dos sistemas.

2.2.3.1 Concorrência (ou Paralelismo)

Na PN da Figura 2.6 as transições t_1 e t_2 estão habilitadas simultaneamente; o disparo de uma delas não modifica o estado da outra. As atividades modeladas pelas duas transições são executadas simultaneamente. Na modelagem de confiabilidade, a PN da Figura 2.6 pode representar dois componentes, C_1 e C_2, com redundância paralela; neste caso, os lugares p_1 e p_3 representam a situação de funcionamento , p_2 e p_4, a situação de falha, e, t_1 e t_2, são eventos das falhas dos componentes C1 e C2.

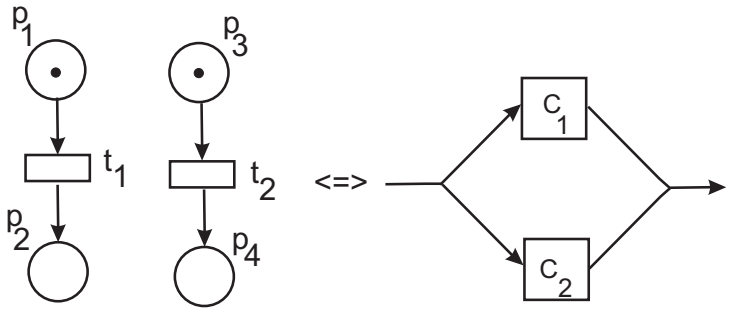


Figura 2.6 PN que modela duas atividades paralelas.

2.2.3.2 Sincronização

Na Figura 2.7 as atividades modeladas por t_1 e t_2 são executadas concorrentemente; entretanto, se elas representarem tarefas de um algoritmo paralelo, ambos devem ser terminados antes que a execução possa prosseguir. A atividade de sincronização é modelada na rede da Figura 2.7 por meio da transição t_3, cujo disparo requer uma marca em p_2 e em p_4, simultaneamente.

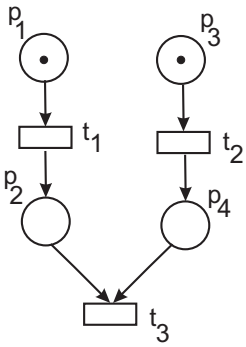


Figura 2.7 PN que modela duas atividades paralelas com sincronização.

2.2.4 Recursos limitados

Um fator típico que influi no desempenho de sistemas distribuídos (sistemas com multiprocessadores, sistemas de manufatura flexíveis e assim por diante) é o número limitado de recursos disponíveis. A estagnação dos recursos impede as atividades de prosseguir e bloqueia o sistema. A modelagem e análise de sistemas com bloqueios é uma tarefa difícil em quase todos os ambientes de modelagem [19, 165]. A representação PN de um *buffer* com tamanho limitado é mostrada na Figura 2.8(b) (A Figura 2.8(a) exibe a representação de diagrama bloco correspondente). O lugar p_3 modela o número de posições livres no *buffer*, enquanto p_2, o número de posições ocupadas; observe que a soma de marcas em p_2 e p_3 é constante e modela o número total de posições disponíveis no *buffer* (três posições na Figura 2.8(b)). A transição t_2 modela o preenchimento de uma posição do *buffer* e pode disparar se existe uma posição livre (pelo menos uma marca em p_3) e uma tarefa está disponível para ser armazenada (uma marca em p_1). A transição t_3 é habilitada quando pelo menos uma posição no *buffer* for preenchida, e o disparo de t_3 move uma marca de p_2 para p_3.

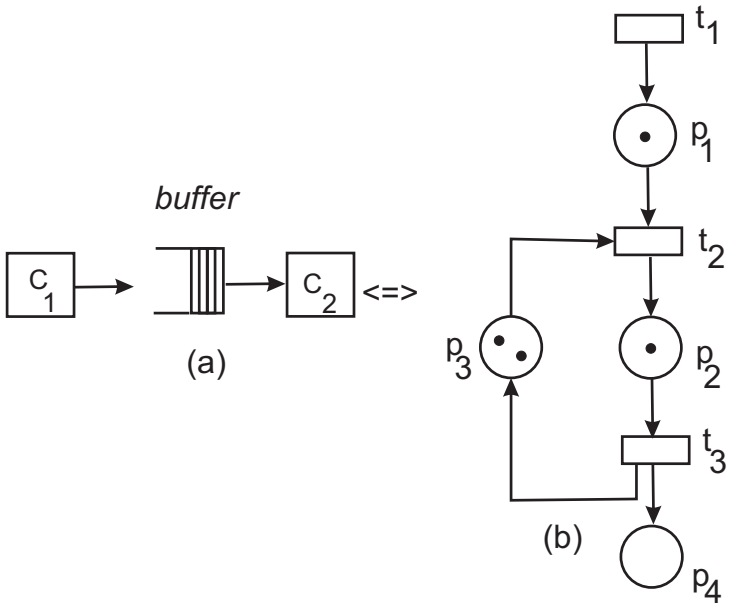


Figura 2.8 Diagrama de bloco e PN de um *buffer* com tamanho finito.

2.2.4.1 Problema produtor/consumidor

Um produtor produz itens que são postos em um *buffer* do qual podem ser removidos e adquiridos por um consumidor. O processo que consome deve ser em sequência em relação ao processo de produção. A solução de PN para este problema é mostrada em Figura 2.9. Uma marca em p1 significa que o produtor está pronto a produzir. Pelo disparo de t1 e t2, que um item é produzido (uma marca é colocada no *buffer* p5) e o produtor está novamente apto a produzir um novo item. Se o consumidor está pronto a consumir (marca em p3) e um item está no *buffer*, a transição t3 pode disparar e remover uma marca de p5.

Na PN da Figura 2.9 a produção e o acúmulo dos itens no *buffer* é ilimitado. Uma situação mais realística é obtida considerando um *buffer* de capacidade limitada (como na Seção 2.2.4). A PN correspondente é exibida na Figura 2.10. O lugar p6 modela as posições livres do *buffer* e o lugar p5 as posições preenchidas do *buffer*; o número de marcas em p5 e em p6 é constante e representa o total das posições disponíveis do *buffer*. Se uma única marca estiver em p6 na marcação inicial, o modelo representa a situação em que o produtor não pode mais produzir até que o consumidor adquira um item no *buffer* (uma ordenação estritamente sequencial das atividades).

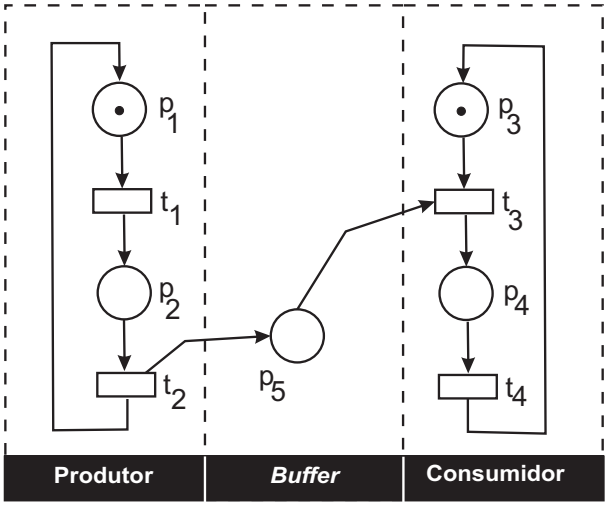


Figura 2.9 O problema produtor/consumidor com *buffer* ilimitado.

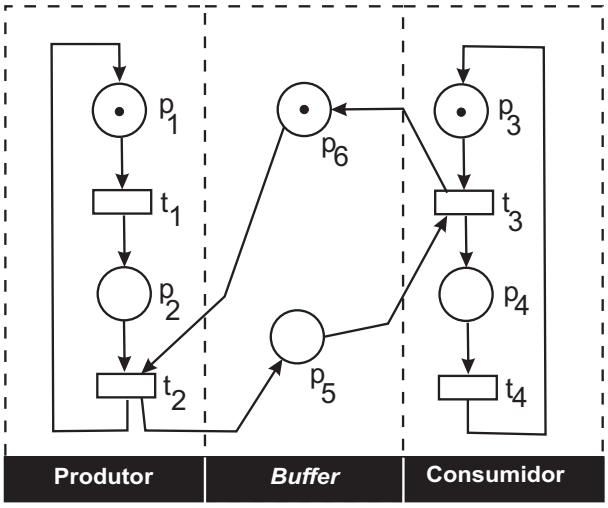


Figura 2.10 O problema produtor/consumidor com *buffer* de tamanho finito.

2.2.4.2 Exclusão mútua

A exclusão mútua acontece quando dois recursos C_1 e C_2 podem trabalhar em paralelo, mas estão conectados a um recurso compartilhado C_s que não pode ser acessado por C_1 e C_2, simultaneamente (diagrama de bloco na Figura 2.11(a). A PN correspondente está na Figura 2.12(b). Os lugares p_1 e p_5 representam C_1 e C_2, que trabalham independentemente; p_2 e p_6 representam C_1 e C_2, que requisitam o acesso ao C_s; p_3 e p_7 representam o C_s ocupado com o C_1 e o C_2, respectivamente. O lugar p_4 determina que o recurso pode realmente acessar o C_s, e impede os lugares p_3 e p_7 de serem marcados ao mesmo tempo; de fato, quando p_2 e p_6 estão ambos marcados, as transições t_2 e t_5 estão em conflito. O disparo de uma delas desabilita a outra. O disparo de t_3 ou de t_6 modela a liberação do recurso comum (retorno da marca para p_4) e o retorno à condição de funcionamento.

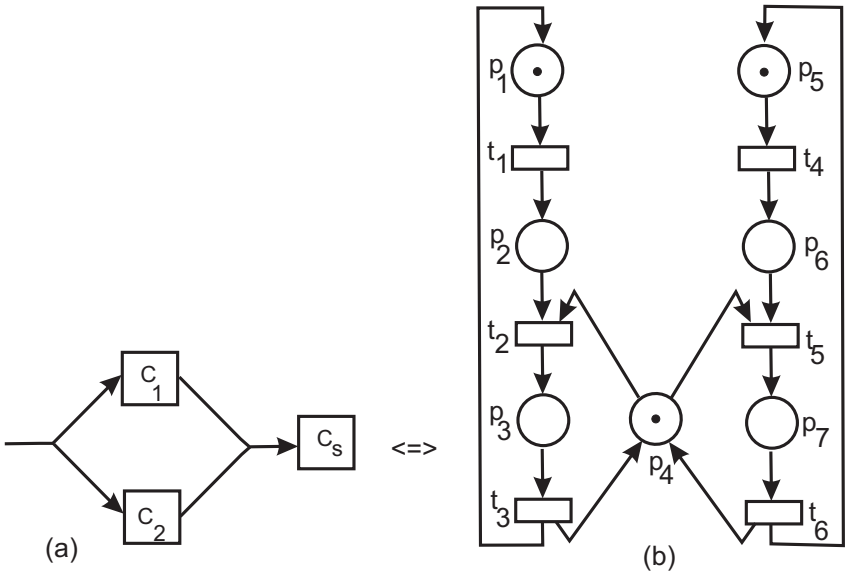


Figura 2.11 O problema da exclusão mútua: duas tarefas paralelas com um recurso comum.

2.2.5 Propriedades das Redes de Petri

O que torna as redes de Petri interessante é que elas fornecem um balanço entre poder de modelagem e a análise: muitas propriedades que se quer saber sobre sistemas concorrentes podem ser determinados automaticamente através de redes de Petri, embora algumas propriedades sejam muito caras de determinar no caso geral. Diversas subclasses das redes de Petri podem modelar classes interessantes de sistemas concorrentes, quando estes problemas se tornarem mais fáceis.

Os problemas de decisão, com resultados de decidibilidade e complexidade para redes de Petri e algumas subclasses, podem encontrar as propriedades das PN que permitem a classificação dos seus elementos primitivos ou da PN como um todo.

Dois tipos de propriedades podem ser estudadas num modelo de Rede de Petri [115]: aquelas que dependem da marcação inicial e aquelas que são independentes da marcação inicial. As

do primeiro tipo são chamadas como dependentes da marcação ou propriedades comportamentais, enquanto as do segundo tipo são chamadas de propriedades estruturais. Segue algumas propriedades estruturais básicas.

- **Viva** - Uma transição é potencialmente disparável em M se existir uma sequência de disparos de transições que conduzam a uma marcação na qual a transição está habilitada. Uma transição é *viva* se for potencialmente disparável em qualquer marcação de $\mathscr{R}(M_0)$. Uma transição é *morta* em M se não for potencialmente disparável; se a PN entrar na marcação M a transição morta não pode mais disparar.

- **Segurança** - Um lugar é seguro se a contagem das marcas não exceder 1 em qualquer marcação do $\mathscr{R}(M_0)$. Uma PN é segura se cada lugar for seguro.

- **Limitada** - Uma generalização simples de segurança é o conceito de limitada. Um lugar é limitado com limite k, se a contagem de marcas não excede k em qualquer marcação de $\mathscr{R}(M_0)$. A PN é $k-limitada$ se cada lugar é $k-limitado$.

- **Conservativa** - Uma PN é estritamente conservativa se o número total de marcas for constante em cada marcação de $\mathscr{R}(M_0)$. Um subconjunto de lugares forma um lugar-invariante se for estritamente conservativo.

2.2.6 Análise de alcançabilidade

Uma marcação é alcançável vindo de outra marcação se existe uma sequência de disparo de transições partindo daquela marcação e resulta nesta marcação. O conjunto de alcançabilidade de uma PN é o conjunto de todas marcações alcançadas da sua marcação inicial. Um grafo de alcançabilidade é um grafo dirigido, cujos nós são marcações no grafo no conjunto de alcançabilidade, com arcos dirigidos entre as marcações, representando transições de marcação-a-marcação. Os arcos dirigidos são rotulados com a transição correspondente, cujo disparo resulta em uma mudança da marcação, da marcação original para a nova marcação.

2.2.6.1 Geração do grafo de alcançabilidade

O conjunto de alcançabilidade $\mathscr{R}(M_0)$ de uma PN é gerado por meio da árvore de alcançabilidade. A marcação inicial M_0 é a raiz da árvore de alcançabilidade. A partir da raiz procura-se por todas as transições habilitadas; o disparo de uma transição habilitada produz uma marcação nova, representada como uma folha nova na árvore da qual o procedimento é iterado. Identificando os nós de fronteira da árvore corretamente, a geração da árvore de alcançabilidade envolve um número finito de passos [127], até mesmo se a PN é ilimitada, introduzindo três tipos de nós de fronteira:

- nó terminal (morto) : sem transição habilitada;
- nó duplicado: já gerado;
- nó infinitamente reprodutível.

A marcação M'' é um nó infinitamente reprodutível se $M'' \geq M' \left(m_i^{''} \geq m_i^{'}, i = 1,\ 2,\ \cdots,\ n_p \right)$ para alguma M' já gerada na árvore. Por causa da relação assumida, a sequência de transição da qual M'' foi gerada a partir de M', a sequência $M' \rightarrow M''$ pode ser reproduzida infinitamente, de forma que a contagem de marcas nos lugares para qual $m_i^{''} \geq m_i^{'}$ pode aumentar indefinidamente. O número arbitrariamente grande de marcas resultantes de nós infinitamente reprodutíveis é representado através de um símbolo especial ω com as propriedades seguintes:

$$\begin{aligned} &\omega + a = \omega \\ &\omega - a = \omega \\ &a < \omega \end{aligned}$$

para qualquer constante positiva a.

Permitindo que ω seja um símbolo legal na especificação da árvore de alcançabilidade, pode ser mostrado para que a geração da árvore de alcançabilidade sempre envolva um algoritmo de busca finito [127]. Se a geração da árvore de alcançabilidade termina sem chegar a nó infinitamente reprodutível, a PN é limitada. Neste caso, o conjunto de alcançabilidade é finito e pode ser representado num grafo dirigido rotulado, cujos vértices são os elementos de $\mathscr{R}(M_0)$ e tal que, para cada possível disparo de transição $M_i | t_k \rangle M_j$, existe um arco $(i; j)$ com rótulo k. O grafo de alcançabilidade associado a um conjunto de alcançabilidade $\mathscr{R}(M_0)$ será denotado por $\mathscr{G}_R(M_0)$.

A Figura 2.12 mostra o gráfico de alcançabilidade da PN da Figura 2.5 com marcação inicial $M_0 = (1;0;0;0;0)$, como discutido na Seção 2.2.6. A Figura 2.13 mostra o gráfico de alcançabilidade para o problema da exclusão mútuo de Figura 2.11, com marcação inicial $M_0 = (1;0;0;1;1;0;0)$.

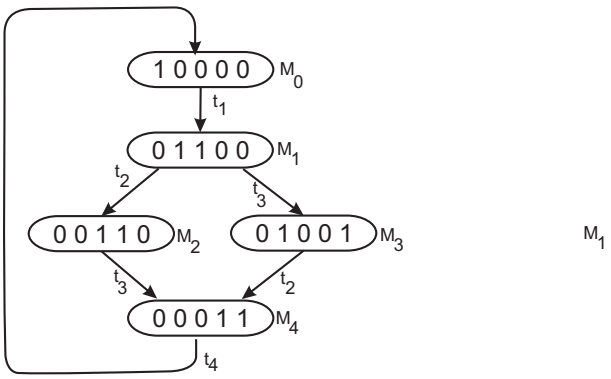


Figura 2.12 O grafo de alcançabilidade $\mathscr{G}_R(M_0)$ da PN da Figura 2.5.

Na Figura 2.14 é exibida a árvore de alcançabilidade da PN da Figura 2.9; desde que a rede é ilimitada, a fim de manter o algoritmo da geração finito, o símbolo ω foi utilizado. Se uma PN tiver um $\mathscr{R}(M1)$ finito, todas as propriedades da rede (segurança, viva, etc...) podem ser analisadas pela inspeção do grafo de alcançabilidade. Se a rede for ilimitada, a representação da árvore de alcançabilidade finita, por meio do símbolo ω, pode ser uma descrição imperfeita da rede (é possível encontrar PNs com propriedades e comportamentos diferentes que não podem ser distinguidas através da árvore de alcançabilidade, devido à informação incompleta fornecida pelo ω [127]).

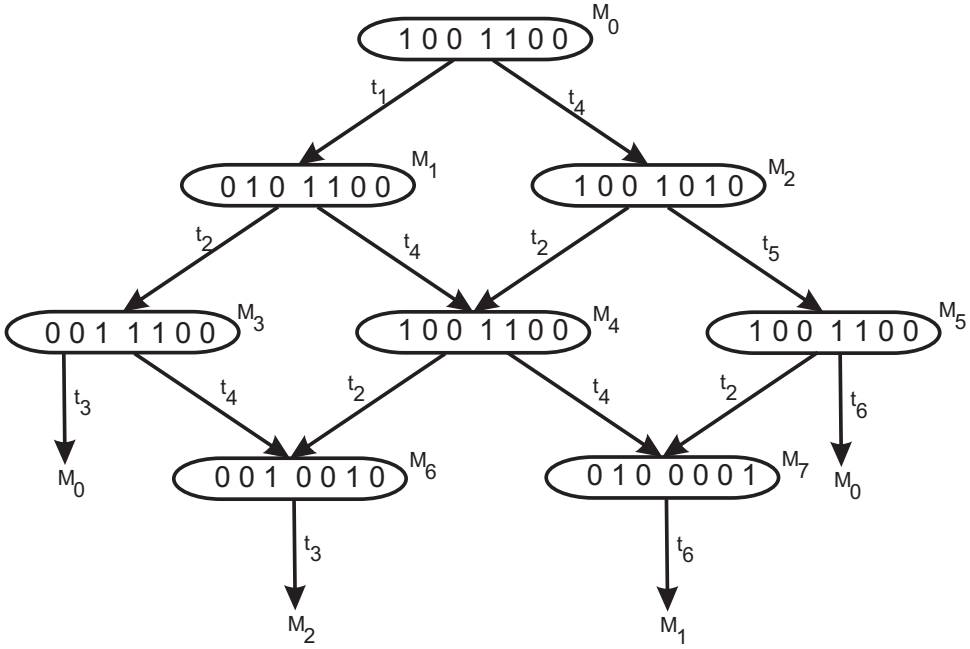


Figura 2.13 O grafo de alcançabilidade $\mathscr{G}_R(M_0)$ da PN da Figura 2.11.

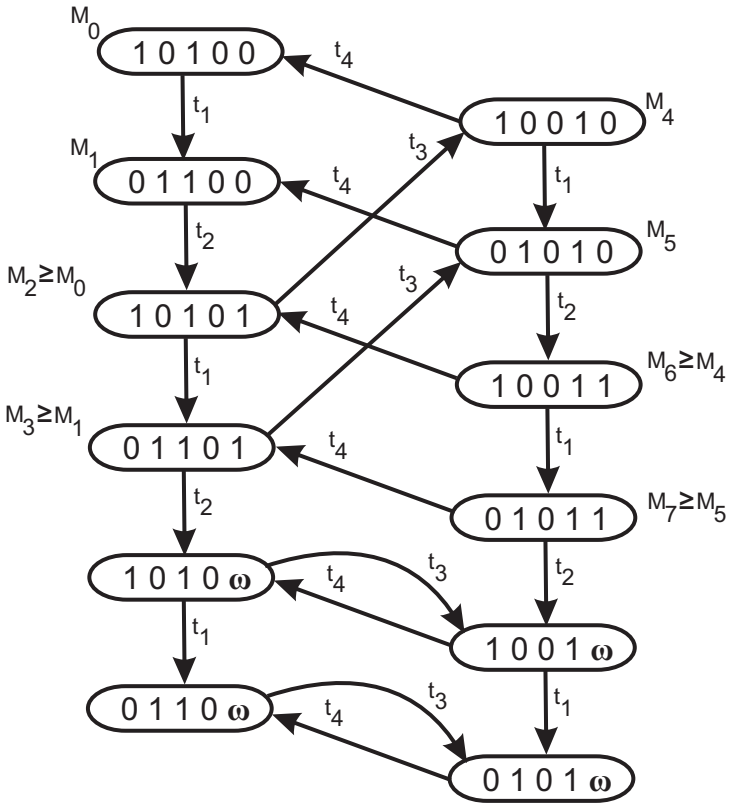


Figura 2.14 Geração da árvore de alcançabilidade para a PN da Figura 4.9 com buffer ilimitado.

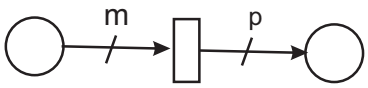
2.2.7 Extensões

No uso de PN para modelar sistemas reais é conveniente introduzir construções especiais para tornar a representação do modelo mais compacta em uma aplicação dada ou para estender o poder modelador do formalismo da PN. As extensões mais frequentemente encontradas na literatura (e que será usado na continuação), foram propostas em resposta a dificuldades em modelar disciplinas de prioridade através das PN. Todas as extensões mencionadas na continuação são equivalentes do ponto da vista do poder modelador. Assim, seu uso depende da facilidade ou da conveniência da implantação.

2.2.7.1 Multiplicidade do arco

Uma cardinalidade do arco (ou multiplicidade) pode ser associada com os arcos de entrada e de saída [10], permitindo a habilitação e regras de disparo, como segue:

- Cada lugar de entrada deve conter pelo menos tantas marcas quanto for a cardinalidade do arco correspondente da entrada.



- Quando a transição dispara, ela remove tantas marcas de cada lugar da entrada quanto for a cardinalidade do arco da entrada correspondente, e depositam tantas marcas em cada lugar de saída quanto for a cardinalidade do arco da saída correspondente.

2.2.7.2 Arco inibidor

Um arco inibidor do lugar p_j para a transição t_k modifica as regras de habilitação, no sentido que a transição pode disparar somente se o lugar p_j não contiver marcas. A função de inibição é representada graficamente como um arco terminado em um círculo pequeno na transição em vez de ponta-de-flecha, como na Figura 2.15, onde a transição t_k pode disparar se, e somente se, p_i contém pelo menos uma marca, mas nenhuma marca está, ao mesmo tempo, em p_j.

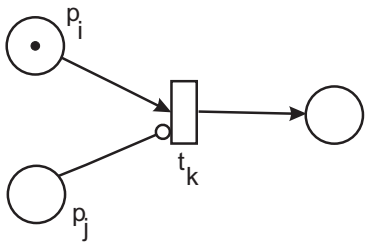


Figura 2.15 Arco inibidor.

No problema de exclusão mútua da Figura 2.11, a linguagem padrão da PN não fornece nenhum meio para estabelecer regras de precedência no caso onde os recursos C_1 e C_2 estejam simultaneamente requisitando acesso do recurso comum C_s (lugares p_2 e p_5 simultaneamente

marcados). Com a inserção de um arco do inibidor do lugar p_2 à transição t_5 (Figura 2.16), modela-se a situação em que, quando um conflito surge entre os recursos C_1 e C_2, C_1 tem sempre a precedência, e bloqueia (inibe) C_2 até que o recurso comum esteja liberado. Com respeito ao grafo de alcançabilidade $\mathscr{G}_R(M_0)$ da PN original da Figura 2.11 (exibida na Figura 2.13), o grafo de alcançabilidade da PN modificada da Figura 2.16 é tal que, da marcação M_4, somente a transição t_2 pode disparar, enquanto t_5 é inibida.

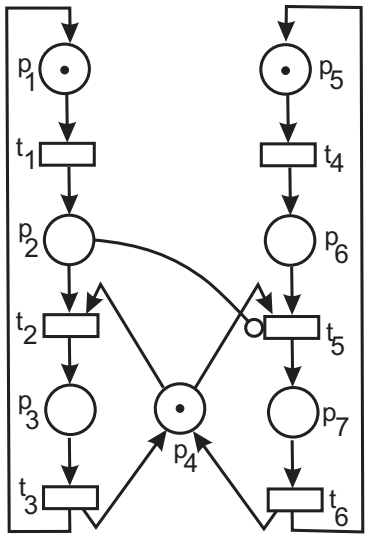


Figura 2.16 O problema da exclusão mútua da Figura 2.11, com atribuição de prioridade.

2.2.7.3 Arco inibidor múltiplo

Um arco inibidor múltiplo traçado de um lugar para uma transição significa que a transição não pode disparar se o lugar inibidor correspondente contiver pelo menos tantas marcas quanto for a cardinalidade do arco inibidor correspondente

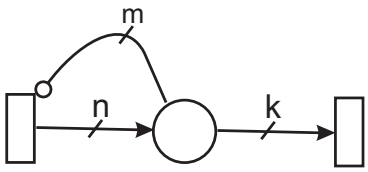


Figura 2.17 Arco inibidor múltiplo.

2.2.7.4 Níveis de prioridade

Um nível de prioridade pode ser atribuído a cada transição da PN como alternativa, mas sendo uma maneira equivalente para modelar as mesmas características consideradas com a introdução do arco inibidor. As regras de execução padrão são modificadas no sentido que, entre todas as transições habilitadas numa dada marcação, somente aquelas com nível de prioridade

mais elevada associado são permitidas disparar. Na Figura 2.16 exatamente a mesma política da precedência pode ser modelada, atribuindo à transição t_2 um nível da prioridade maior do que ao atribuído a t_5. Na marcação M_4 (veja Figura 2.13), onde ambas as transições estão habilitadas, somente t_2 pode disparar.

2.2.7.5 Funções de habilitação

As interações lógicas mais complexas entre elementos primitivos de um PN podem ser consideradas introduzindo as funções de habilitação (ou expressões de guarda). Dada uma marcação M, uma transição da PN está habilitada se, ao lado das exigências normais de habilitação permitidas (incluindo arcos inibidores e prioridades), a função de habilitação for verdadeira. As funções de habilitação podem ser muito eficientes em reduzir a complexidade gráfica de uma PN.

2.3 Redes de Petri Estocásticas

O uso de técnicas baseadas em redes de Petri para a análise quantitativa de sistemas requer a introdução da especificação temporal dentro do modelo básico não temporizado. Este fato foi reconhecido há cerca de 40 anos e conduz a várias propostas diferentes para a introdução da especificação temporal nas Redes de Petri. As principais alternativas que caracterizam as diferentes propostas tratam:

- os elementos de rede de Petri associados com o tempo (normalmente lugares ou transições, mas alguns autores também definem arcos temporizados ou marcas);
- a semântica de disparo no caso de transições temporizadas (disparo atômico ou disparo em três fases);
- a natureza da especificação temporal (determinística ou probabilística);
- a política de resolução de conflitos.

Os modelos SPN (*Stochastic Petri Nets* / Redes de Petri Estocásticas) foram propostos por pesquisadores do campo de modelagem estocástica aplicada, com o objetivo de desenvolver uma ferramenta que permita a integração da descrição formal, prova de correção e avaliação de desempenho. Pelas preocupações com o último aspecto, as propostas visavam uma equivalência entre modelos SPN e CTMC (*Continuous-Time homogeneous Markov Chain* / Cadeia de Markov homogênea de Tempo Contínuo), enquanto para as duas primeiras foi escolhido introduzir o tempo de tal maneira a não alterar o comportamento do sistema sem tempo.

De maneira a obter uma equivalência entre PN e CTMC, é necessário que a determinação dos próximos estados alcançáveis dependa somente do estado presente (verdadeiro por definição em redes de Petri), e que os tempos de residência nas marcações sejam variáveis aleatórias com *p.d.f. (probability density function)* exponencial negativa.

O primeiros trabalhos na área de PN temporizadas foram desenvolvidos por Ramchandani [130, 131], Merlin & Faber [108] e por Noe & Nut [117]. As redes de Petri estocásticas foram

introduzidas em 1980 [155, 116, 110] como formalismo para a descrição de DEDS (*Discrete Events Dynamic Systems* / Sistemas Dinâmicos de Eventos Discretos) cujo comportamento dinâmico é representado por meio da CMTC. A proposta original das SPN assumiu disparos atômicos, tempos de disparos exponencialmente distribuídos, e uma política de execução de corrida (*race execution policy*), i.e., quando transições múltiplas estão habilitadas simultaneamente, a política de corrida seleciona a transição com o mínimo retardo estocástico para o disparo.

Com o objetivo de estender o poder de modelagem das redes de Petri estocásticas, surgiu a proposta GSPN (*Generalized SPN*) [99]. Uma GSPN inclui duas classes de transições: transições temporizadas exponencialmente distribuídas e transições imediatas. Existem outras generalizações do formalismo SPN básico que estão relacionados a GSPN, a Rede de Petri Estocástica Estendida [32] e a Rede de Atividade Estocástica [109]. GSPN é um dos formalismos SPN mais comumente usados para análise de problemas importantes, e um esforço considerável tem sido dedicado para seu desenvolvimento desde sua proposição.

2.3.1 Transições Temporizadas Distribuídas Genericamente – Redes de Petri Estocásticas $(GDTT - SPN)$

Para a tradução de um modelo GDTT–SPN (*Generally Distributed Timed Transition - Stochastic Petri Nets*) [2] em seu processo estocástico subjacente, é necessário associar ao modelo uma "política de execução", compreendendo duas especificações: uma regra para escolher a próxima transição que vai disparar em qualquer marcação (uma política de disparo), e um critério para considerar a história passada sempre que uma transição é disparada (a política de memória) [2].

Em relação à política de disparo, duas alternativas são possíveis: usar o retardo associado à transições para decidir qual será o próximo disparo ou adicionar uma métrica específica para esta finalidade [103]. Os formalismos GDTT-SPN que serão considerados adotam a primeira opção, que corresponde à "política de corrida" (*race police*); a transição com retardo mínimo restante é a que dispara primeiro. Em relação à política de memória, novamente, existem duas alternativas básicas possíveis em cada mudança de marcação:

- **Continuar:** os temporizadores[3] associados com transições mantêm seus valores presentes e posteriormente irão continuar sendo decrementados.

- **Reiniciar:** os temporizadores da transição são reiniciados, *i.e.*, seus valores atuais serão descartados, e novos valores serão gerados sempre que for necessário.

A política de memória é implementada sempre que uma transição dispara. A política de memória afeta transições que disparam, as que deixam de estar habilitadas devido a mudanças

[2]Em redes de Petri (não temporizadas), a próxima transição para disparar é escolhida não deterministicamente e não existe necessidade de registrar "a história passada", que está capturada no estado atual (uma suposição Markoviana intrínseca).

[3]Pode-se descrever a evolução de uma GDTT-SPN associando um temporizador a cada transição: os temporizadores são decrementados a velocidade constante enquanto transições estiverem habilitadas, e quando um temporizador vai para zero, a transição correspondente dispara.

de marcação e as que continuam habilitadas na nova marcação. A memória da transição que dispara é irrelevante, desde que, neste caso, um novo retardo seja sempre gerado. A memória das transições que não disparam é frequentemente dos seguintes tipos:

- **Reamostragem** (*resampling*): o temporizador da transição é reinicializado (*reset*) para um novo valor em qualquer mudança de marcação. O novo valor é obtido (*sampled*) da *p.d.f.* do retardo (*delay*) associado à transição.
- **Memória habilitada** (*enabling memory*): se na nova marcação a transição ainda está habilitada, o valor do temporizador é mantido; ele é, em vez disso, reiniciado para um novo valor se a transição não está habilitada.
- **Memória de Idade** (*age memory*): o temporizador é mantido, mesmo se a transição não está habilitada na nova marcação.

A Tabela 2.3 resume as possíveis combinações de memória, dependendo da transição que está habilitada na nova marcação.

Tabela 2.3 Resumo dos mecanismos de memória.

Política de memória	**Na nova marcação:**	
	a transição permanece habilitada	**a transição perde habilitação**
Reamostragem	reinicia	reinicia
Memória habilitada	continua	reinicia
Memória de idade	continua	continua

Uma GDTT-SPN é uma óctupla,

$$\text{GDTT-SPN} = (P;T;I;O;H;M_0;W;\varepsilon),$$

onde $(P;T;I;O;H;M_0)$ é o sistema de rede de Petri subjacente, o qual usualmente compreende:

- um conjunto de lugares $P = \{p_1;p_2;p_3;p_4;\cdots;p_m\}$;
- um conjunto de transições $T = \{t_1;t_2;t_3;t_4;\cdots;t_m\}$;
- as funções de entrada, saída e inibidoras $I,O,H : T \rightarrow \mathbb{N}$;
- uma marcação inicial $M_0 = (m_{01};m_{02};\cdots;m_{0m})$;

para o qual é necessário adicionar:

- uma função de distribuição $W : T \rightarrow \{pdf\}$, que atribui a cada transição uma variável aleatória com uma *p.d.f.* (*probability density function* / função densidade de probabilidade) especificada;

- uma função política de execução $\varepsilon : T \rightarrow \{reamostragem, habilitação, idade\}$, que atribui a política de execução para cada transição.

O processo estocástico correspondente à evolução da GDTT-SPN sobre seus espaços de estado (ou conjunto de alcançabilidade) é chamado de grafo de marcação.

Deve ser observado que se W identifica somente funções contínuas, então a probabilidade que duas transições estejam agendadas para disparar exatamente no mesmo instante é zero. Em tal caso, um modelo GDTT-SPN envolve disparos de transições uma após outra.

Um modelo GDTT-SPN pode corresponder a um processo estocástico bastante complexo, cuja definição, sem mencionar sua solução, não é uma solução trivial. Foram definidas subclasses de GDTT-SPN, correspondendo a classes mais simples de processos estocásticos. Redes de Petri Estocásticas (*SPN - Stochastic Petri Nets*) [102, 155, 116, 110, 111, 115], Rede de Petri Estocástica Generalizada (*GSPN - Generalized Stochastic Petri Nets*) [99], Semi-Markov SPN [3, 116, 32], Tipo Fase SPN (PHSPN - *Phase Type SPN*) [24, 120], SPN Determinística (*DSPN - Deterministic SPN*) [101, 81, 82, 23, 17, 20], SPN Markov Regenerativa (MRSPN - *Markov Regenerative SPN*) [18, 7, 8, 134] são integrantes da taxonomia das GDTT-SPN. A GSPN será discutida a seguir, pois é deste modelo que estabelecemos uma extensão intervalar.

2.3.2 O Modelo Básico SPN

Uma SPN é uma GDTT-SPN na qual a função W atribui a cada transição uma *p.d.f.* exponencial negativa. Desde que a distribuição exponencial é completamente caracterizada pelo seu valor médio (ou por sua inversa, a taxa), e a sua característica de falta de memória torna não essencial, como iremos explicar posteriormente, a definição de uma política de execução para transições, então a definição de uma SPN é mais simples do que a de uma GDTT_SPN, isto é:

$$SPN = (P;\ T;\ I\ ;\ O;\ H;\ M_0;\ W)$$

onde:

- $(\mathrm{P;T;I;O;H;M_0})$ é a PN subjacente da GDTT_SPN;
- $W : T \rightarrow \mathbb{R}$ é uma função peso; $w(t)$ é a taxa da distribuição exponencial associada com a transição t. $w(t)$ é também chamada de taxa de disparo da transição t.

A função peso pode depender da marcação atual: neste caso escrevemos $w(t,\ M)$ para expressar a função peso da transição t na marcação M. Tipos comuns de dependência de marcação são dependências emprestadas da teoria de filas, isto é, simples, múltipla e servidor infinito, mas, enquanto em sistemas de fila, a dependência da taxa de serviço normalmente é sobre o número de usuários na fila, em SPN a dependência é no grau de habilitação da transição [107].

O retardo de disparo médio da transição t na marcação M é, então, $[w(t;M)]^{-1}$.

Com a política de corrida (*race policy*) que está sendo considerada, a transição com retardo mínimo é a transição que dispara primeiro. O retardo de disparo calculado para cada transição na entrada de uma nova marcação é o novo valor amostrado da distribuição exponencial associada com a transição (para transição que "reinicia"), ou o tempo residual do temporizador (para a

transição que "continua"). Entretanto, devido à propriedade de falta de memória da distribuição exponencial, a distribuição do tempo total de disparo é idêntica à distribuição do tempo de disparo residual; então, a corrida é sempre entre variáveis exponencialmente distribuídas, portanto torna-se desnecessária a definição da política de execução nos modelos SPN.

O conjunto de alcançabilidade de uma SPN é idêntico ao conjunto de alcançabilidade da PN sem tempo subjacente, com semântica de intercalação (*interleaving*), devido ao suporte ilimitado ($t \in [0;+\infty[$) e à continuidade da distribuição exponencial. O diagrama de taxa de transição de estado da CTMC correspondente a SPN é obtida pela construção do grafo de alcançabilidade, e pela rotulação dos arcos com a taxa de transição, cujo disparo produz a mudança de marcação.

A solução do estado estacionário do modelo [10] é obtida pela solução do sistema de equações lineares

$$\vec{\pi} \cdot \mathbf{Q} = \vec{0}$$
$$\textstyle\sum_{M \in \mathrm{RS}} \pi_M = 1$$

$\vec{\pi}$ é o vetor *p.m.f.* (*probability mass function* / função de probabilidade) de equilíbrio sobre as marcações alcançáveis, e escrevemos π_M para a probabilidade do estado estacionário de uma dada marcação M. $\mathbf{Q}$ é o gerador infinitesimal da CTMC associado a GSPN.

A solução transiente do modelo é de fato obtida resolvendo-se um conjunto de equações diferenciais

$$\frac{d\vec{\pi}(t)}{dt} = \mathbf{Q} \cdot \vec{\pi}(t)$$

onde $\pi_M(t)$ é a probabilidade do sistema estar no estado M no tempo t. $\mathbf{Q}$ é o gerador infinitesimal da CTMC associado a GSPN. Em [151, 10] é dado um tratamento detalhado do problema posto pela solução das equações acima.

A interpretação na modelagem das probabilidades do estado estacionário é a seguinte: π_M é a probabilidade conforme a qual um observador aleatório encontra o sistema no equilíbrio na marcação M (estado), ou equivalentemente à porcentagem de tempo que o sistema gasta na marcação M quando em condições de equilíbrio.
O custo de solução do sistema linear $\vec{\pi} \cdot \mathbf{Q} = \vec{0}$, *com* $\sum_i \pi_i = 1$ é polinomial no número de marcações. As técnicas iterativas são frequentemente aplicadas, cujo custo por iteração é da ordem do número de elementos não nulos em $\mathbf{Q}$, *i.e.*, da ordem do número de arcos no grafo de alcançabilidade.
Da *p.m.f.* do estado estacionário é possível derivar muitos parâmetros de interesse para quantificar a performance do sistema.

Como um exemplo, consideremos um abajur equipado com uma lâmpada incandescente. O abajur pode ser ligado e desligado, e a lâmpada incandescente pode queimar enquanto o abajur está ligado (assumindo que, com o abajur desligado, a lâmpada não queima). As lâmpadas queimadas são trocadas por lâmpadas novas, e antes que a operação de troca seja realizada, a chave da lâmpada é colocada na posição desligada. O modelo SPN descrevendo o sistema considerado no exemplo é descrito na Figura 2.18. Ao se comparar o modelo SPN, na Figura 2.18, e o diagrama de taxa de transição de estados, na Figura 2.19, observa-se a similaridade na

topologia. Isto é devido ao fato de que a PN subjacente ao nosso modelo SPN é uma máquina de estados [135].

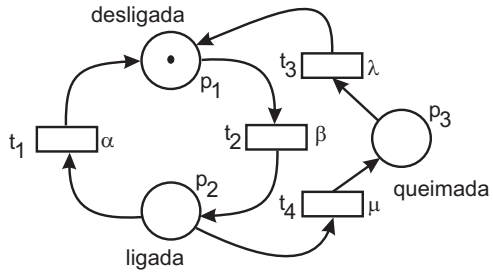


Figura 2.18 Modelo SPN de um abajur com a lâmpada que está desligada no estado atual.

O espaço de estado desta SPN é formado por três estados: 1) desligado ($p_1 = 1,\ p2 = 0,\ p3 = 0$); 2) ligado ($p_1 = 0,\ p2 = 1,\ p3 = 0$); e 3) lâmpada queimada ($p_1 = 0,\ p2 = 0,\ p3 = 1$) é mostrado na Tabela 2.4, enquanto o gerador infinitesimal é dado na Tabela 2.5, e seu diagrama de transição é mostrado na Figura 2.19. O gerador infinitesimal é

$$\mathbf{Q} = \begin{pmatrix} -\beta & \beta & 0 \\ \alpha & -(\alpha+\mu) & \mu \\ \lambda & 0 & -\lambda \end{pmatrix}$$

Tabela 2.4 Espaço de Estado do Modelo SPN de um abajur.

M_1	$p_1 = 1,\ p2 = 0,\ p3 = 0$
M_2	$p_1 = 0,\ p2 = 1,\ p3 = 0$
M_3	$p_1 = 0,\ p2 = 0,\ p3 = 1$

Tabela 2.5 Representação do gerador infinitesimal do modelo SPN de um abajur.

	M_1	M_2	M_3
M_1		$w(t_2) = \beta$	
M_2	$w(t_1) = \alpha$		$w(t_4) = \mu$
M_3	$w(t_3) = \lambda$		

A distribuição do estado estacionário é facilmente calculada pela solução do sistema de equações lineares $\vec{\pi} \cdot \mathbf{Q} = 0,\ com\ \sum_i \pi_i = 1$. Do sistema seguinte:

$$\begin{aligned} &\beta\pi_1 = \alpha\pi_2 + \lambda\pi_3 \\ &(\alpha+\mu)\pi_2 = \beta\pi_1 \\ &\lambda\pi_3 = \mu\pi_2 \\ &\pi_1 + \pi_2 + \pi_3 = 1 \end{aligned}$$

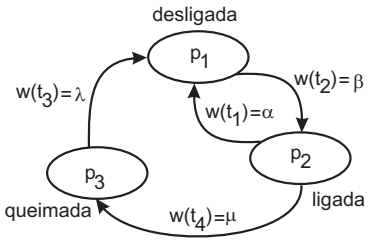


Figura 2.19 Diagrama de transição de estado do modelo SPN de um abajur.

obtém-se

$$\vec{\pi} = \frac{1}{\lambda(\alpha+\beta)+\mu(\lambda+\beta)}(\lambda(\alpha+\mu);\ \lambda\beta;\ \beta\mu)$$

Observamos que as três primeiras equações do sistema linear podem ser interpretadas como igualdades do fluxo de entrada e saída de um dado estado, onde o fluxo de probabilidade sobre um arco é o produto da probabilidade do estado estacionário do estado do qual o arco origina-se vezes o rótulo do arco. Então, por exemplo, no caso do estado M_1, temos:

$$\begin{aligned} &fluxo\ de\ saída = \beta\pi_1 \\ &fluxo\ de\ entrada = \alpha\pi_2 + \lambda\pi_3 \end{aligned}$$

Isto também implica a dependência linear das equações; contudo, a solução é única quando se considera o fato de que probabilidades devem somar 1. A partir de $\vec{\pi}$ é possível computar vários índices de desempenho do estado estacionário:

- π_2 é a fração do tempo no qual o abajur está ligado;
- π_3 é a fração do tempo no qual a lâmpada está queimada;
- $[\lambda\pi_3] = [\mu\pi_2]$ é o número médio de falhas em unidades de tempo (o *throughput* das falhas);
- $[\lambda\pi_3]^{-1} = [\mu\pi_2]^{-1}$ é o tempo médio entre duas falhas consecutivas;
- $[(\alpha+\mu)\pi_2]^{-1} = [\beta\pi_1]^{-1}$ é o tempo médio entre dois instantes consecutivos, no qual o abajur é ligado.

Da solução transiente podemos derivar outros índices de desempenho interessantes:

- $\pi_2(t)$ é a probabilidade de o abajur estar ligado no tempo t;
- $\pi_3(t)$ é a probabilidade de a lâmpada estar queimada no tempo t.

Se o reparo não é possível no sistema (i.e., o arco entre os estados M_3 e M_1 no modelo CTMC tem taxa 0, ou é removido), pode-se calcular a probabilidade de a lâmpada não ter ainda falhado no tempo t como $1-\pi_3(t)$.

2.3.3 SPN Generalizada (GSPN)

O fator-chave que limita a aplicabilidade das SPN é a complexidade de sua análise. O número muito grande de possíveis marcações alcançáveis é de longe um dos fatores mais críticos. A presença em um modelo de atividade que acontece em uma escala de tempo muito rápida (ou muito lenta) do que uma relacionada a eventos que têm papel crítico no desempenho geral do sistema modelado. Quando eles resultam em sistemas de equações lineares densos, *i.e.*, difíceis de resolver com um grau de precisão aceitável por meio de técnicas numéricas usuais [103]. Por outro lado, negligenciar as atividades "rápidas" (ou "lentas") pode produzir modelos logicamente incorretos. Também pode acontecer de, na construção da topologia de um modelo SPN, o analista inserir transições que correspondam a aspectos puramente lógicos do comportamento do sistema para facilitar a descrição das mudanças complexas de marcação, tal que nenhum temporizador possa ser razoavelmente associado com ele.

SPN Generalizada (GSPN) foi proposta originalmente para tratar estes problemas [104]. A definição foi posteriormente aprimorada para explorar melhor as estruturas da ferramenta de modelagem [100, 106]. Em [107] é apresentado em detalhe o formalismo GSPN com vários exemplos de aplicação.

Modelos GSPN compreendem dois tipos de transições:

- **Transições temporizadas:** são associadas retardos de disparos aleatórios distribuídos exponencialmente, como numa SPN, e
- **Transições imediatas:** disparam em tempo zero, com prioridade superior a transições temporizadas.

Além disso, diferentes níveis de prioridade das transições imediatas podem ser usadas, e pesos estão associados com transições imediatas.

Uma GSPN é uma óctupla:

$$GSPN = (P;\ T;\ I;\ O;\ H;\ M_0;\ \Pi;\ W),$$

onde $(P;\ T;\ I;\ O;\ H;\ M_0)$ é a PN não temporizada subjacente, que compreende:

- um conjunto de lugares P;
- um conjunto de transições T;
- as funções de entrada, saída e inibidoras $I, O, H : T \rightarrow \mathbb{N}$;
- uma marcação inicial M_0;
- a função de prioridade $\Pi : T \rightarrow \mathbb{N}$, a qual associa a menor prioridade ($= 0$) com transições temporizadas, e prioridades mais alta (≥ 1) para transições imediatas:

$$\Pi(t) = \begin{cases} 0 \ if\ T\ é\ temporizada \\ \geq 1\ if\ T\ é\ imediata; \end{cases} \tag{2.14}$$

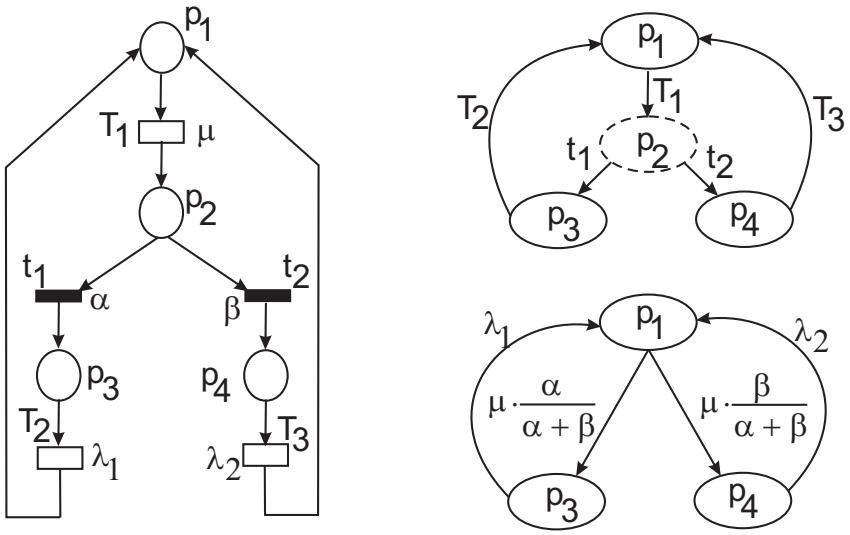


Figura 2.20 Construção do gráfico de alcançabilidade tangível

- finalmente, o último item da definição GSPN é a função $W : T \rightarrow \mathbb{R}$, que associa um valor real com transições, $w(t)$ é:

  √ o parâmetro da *p.d.f.* exponencial negativa das transições com retardo de disparos, se t é uma transição temporizada;

  √ um peso usado para o cálculo das probabilidades de disparo das transições imediatas, se t é uma transição imediata.

Na representação gráfica da GSPN, transições imediatas são traçadas como caixas retangulares pretas (ou barras), e transições exponenciais, como caixas retangulares, como mostra a rede de Petri desenhada na esquerda da Figura 2.20.

O modelo subjacente não temporizado de uma GSPN é a rede P/T (*Place / Transition*) com arcos inibidores e prioridades globais: a adição de prioridades a um sistema P/T pode reduzir o número de estados atingíveis, e pode eventualmente destruir propriedades tipo *liveness* e *home states*, enquanto todas as propriedades de segurança (*safety*) ficam mantidas [103].

Quando se atinge uma marcação, é primeiramente necessário determinar se ela habilita transições temporizadas somente, ou, no mínimo, uma transição imediata. A marcação do tipo que compreende somente transições temporizadas são chamadas tangíveis, enquanto as que envolvem, no mínimo, uma transição imediata são chamadas voláteis (que desaparece/*vanishing*).

A semântica de um modelo GSPN sempre assume que transições são disparadas uma após outra, mesmo em uma marcação volátil formada por transições imediatas não conflitantes.

Um exemplo da construção do gráfico de alcançabilidade tangível é apresentado na Figura 2.20, que descreve uma GSPN muito simples, na qual existe um conflito entre as duas transições imediatas. Seu grafo de alcançabilidade é mostrado na porção superior direita. Linhas pontilhadas para o estado p_2 indicam que o estado é volátil: de fato, quando a transição T_1 dispara, o sistema entra na marcação p_2, na qual duas transições imediatas estão habilitadas, e a marcação muda em tempo zero para p_3 ou p_4, com probabilidades $\frac{\alpha}{\alpha+\beta}$ e $\frac{\beta}{\alpha+\beta}$, respectivamente. O grafo

de alcançabilidade tangível na porção direita inferior é obtido eliminando-se a marcação volátil p_2. A taxa na qual o sistema se move de p_1 até $p_3(p_4)$ é obtida pela multiplicação da taxa μ da transição do estado p_1 para p_2, com a probabilidade de ir do estado p_2 para o estado $p_3(p_4)$. A solução do estado estacionário do modelo CTMC subjacente da GSPN é obtida pela solução do sistema de equações lineares:

$$\begin{cases} \vec{\pi} \cdot \mathbf{Q} = \vec{0} \\ \sum_{M \in RS} \pi_M = 1 \end{cases} \tag{2.15}$$

$\vec{\pi}$ é a *p.m.f.* (função massa de probabilidade ou função de probabilidade - *probability mass function*) sobre uma marcação tangível, e escreve-se π_M para a probabilidade de estado-estacionário de uma dada marcação M.

A solução transiente do modelo é de fato obtida, resolvendo-se um conjunto de equações diferenciais

$$\frac{d\vec{\pi}(t)}{dt} = \mathbf{Q}\vec{\pi}(t), \tag{2.16}$$

onde

$\vec{\pi}(t)$ é o vetor de probabilidade de estado no tempo t;

Q é o gerador infinitesimal da CTMC associado à GSPN.

2.3.3.1 Análise do estado estacionário

A análise de um modelo GSPN requer a solução de um sistema de equações lineares com tantas equações quanto o número de marcações tangíveis. O gerador infinitesimal CTMC associado com um modelo GSPN é derivado da construção de um grafo rotulado com as taxas ou pesos das transições associado à mudança de marcação. Duas fontes didáticas de algoritmos com técnicas para solução estado-estacionário são os livros de Stewart [151] e de Bolch, Greiner & Trivedi [10].

Para determinar a probabilidade do estado estacionário de cadeias de Markov finitas, três diferentes métodos têm sido considerados: métodos numéricos diretos, métodos numéricos iterativos e técnicas que fornecem soluções exatas (algébricas). As soluções exatas são altamente desejadas, mas elas só podem ser obtidas para uma pequena classe de modelos que têm alguma estrutura especial em sua matriz.

A eliminação gaussiana (GE - *Gaussian elimination*) e o algoritmo variante de Grassmann são métodos diretos comumente aplicados [10]. O GE sofre algumas vezes de erros de cancelamento [10] e o algoritmo de Grassmann evita subtrações, sendo menos sensível a erros de arredondamento e cancelamento causado pela subtração de números praticamente iguais.

O método da potência, o método de Jacobi, o método de Gauss-Seidel e o método da sobre-relaxação sucessiva são métodos iterativos [10]. O método da potência trabalha com a matriz de transição de probabilidade **P**. Dada a matriz geradora $\mathbf{Q}\left[q_{i,j}\right]$ de uma CTMC, a matriz transição de probabilidade pode ser definida por:

$$\mathbf{P} = \mathbf{Q}/q + \mathbf{I} \tag{2.17}$$

onde q é escolhida, tal que $q > max|q_{ii}|$. A matriz **P** resultante é usada para determinar o vetor de probabilidade do estado estacionário $\vec{\pi} = \vec{\nu}$, resolvendo $\vec{\nu} = \vec{\nu} \cdot \mathbf{P}$ *com* $\sum_i \nu_i = 1 (\vec{\nu} \cdot \vec{1} = 1)$.

Por definição, linhas de **P** e elementos de $\vec{\pi}$ ambos somam 1. Este método reduz a CTMC a uma DTMC e é chamado aleatorização ou uniformização.

Como regras gerais para escolha entre as alternativas de métodos diretos e iterativos as seguintes sugestões podem ser seguidas [10]:

- sistemas pequenos são resolvidos por métodos diretos;
- grandes sistemas, cuja matriz tem uma estrutura esparsa, são solucionados por métodos iterativos (que preservam a esparsividade).

2.4 Análise GSPN

Nesta seção são dados exemplos de diversos modelos GSPN que são analisados através da ferramenta TimeNET e são explorados no modelo ISPN proposto nesta tese.

2.4.1 Análise de Desempenho de Sistema de Manufatura Automático

As GSPN modelam as dependências lógicas e causais em sistemas de manufatura, como também permitem a análise quantitativa de desempenho. Além disto, as GSPN podem modelar características *nonproductform* [27], tais como bloqueio, sincronização, disciplina de prioridades de filas, e operações de aquisição e uso de recursos múltiplos. A modelagem de sistemas de manufatura automáticos conduz a modelos com um grande número de estados. As PN podem ser usadas para tratar esta complexidade. O uso de linha de montagens permite o desenvolvimento de uma metodologia modular que conduz à construção direta de modelos [27].

2.4.1.1 Análise GSPN de uma única máquina

O modelo descrito na Figura 2.21 representa uma máquina propensa à falha e *buffer* de capacidade finita [27]. A Tabela 2.6 a seguir interpreta os lugares e as transições do modelo. A Tabela 2.7 mostra o "tempo de retardo do disparo = *delay*" das transições temporizadas em unidades de tempo, onde ν representa produção de um item, λ representa falha, e μ representa reparo. A taxa de produção da máquina pode ser calculada usando a fórmula TimeNET $P = P\{\#P0 > 0\} * Tp$, onde $Tp = 1/T0$ é a taxa de disparo da transição $T0$. Executa-se a análise do estado estacionário para obter-se a métrica calculada. A taxa de produção da máquina é $P = 6,25$.

2.4.1.2 Análise GSPN de uma linha de montagem com duas máquinas e um *buffer*

Consideremos o Modelo GSPN de duas máquinas e um *buffer* de linha de montagem da Figura 2.22 [27]. A Tabela 2.8 interpreta os lugares e as transições do modelo. A Tabela 2.9 mostra as taxas/tempo de disparo das transições temporizadas (por unidade de tempo/em unidades de tempo), onde ν representa produção, λ representa falha, e μ representa reparo. As informações das probabilidades do estado estacionário podem ser então combinadas para produzir métricas.

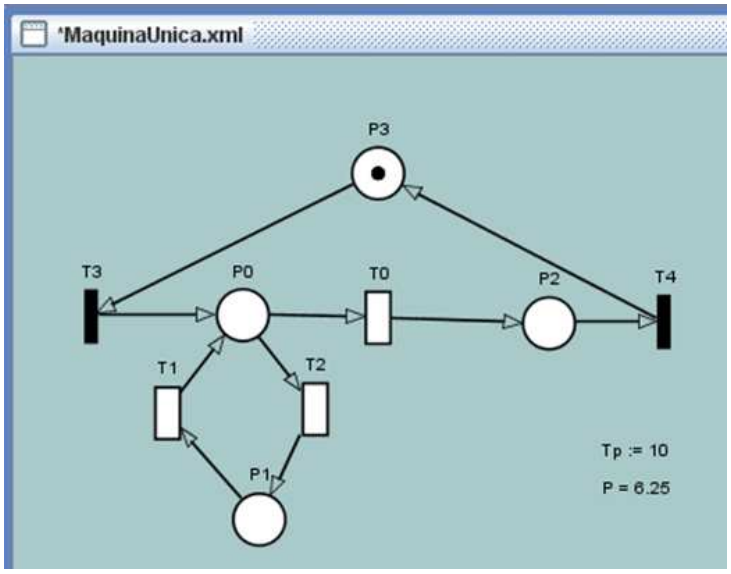


Figura 2.21 Modelo GSPN de uma única máquina

Tabela 2.6 Interpretação das marcas nos lugares e dos tempos das transições para o modelo GSPN de uma única máquina

Nome	Interpretação
P0	A máquina está fabricando um item.
P1	A máquina está quebrada (em reparo).
P2	A máquina fabricou um item que está no *buffer*.
P3	A máquina está disponível.
T0	Tempo médio de processamento na máquina.
T1	Tempo médio para reparo na máquina.
T2	Tempo médio para falha na máquina.

Tabela 2.7 Tempos de retardo do disparo das transições para o modelo GSPN de uma única máquina

Transição	Taxa $[t]^{-1}$	Tempo $[t]$	Símbolo
$T0$	10	0,1	ν
$T1$	5	0,2	μ
$T2$	3	0,33333333	λ

A utilização média das máquinas podem ser calculadas como:

$$UM1 = P\{\#P1 = 1\} \ e \ UM2 = P\{\#P5 = 1\}$$

Executa-se a análise do estado estacionário para obter-se as métricas calculadas. A utilização média das máquinas $UM1 = 0,119467$ e $UM2 = 0,597335$.

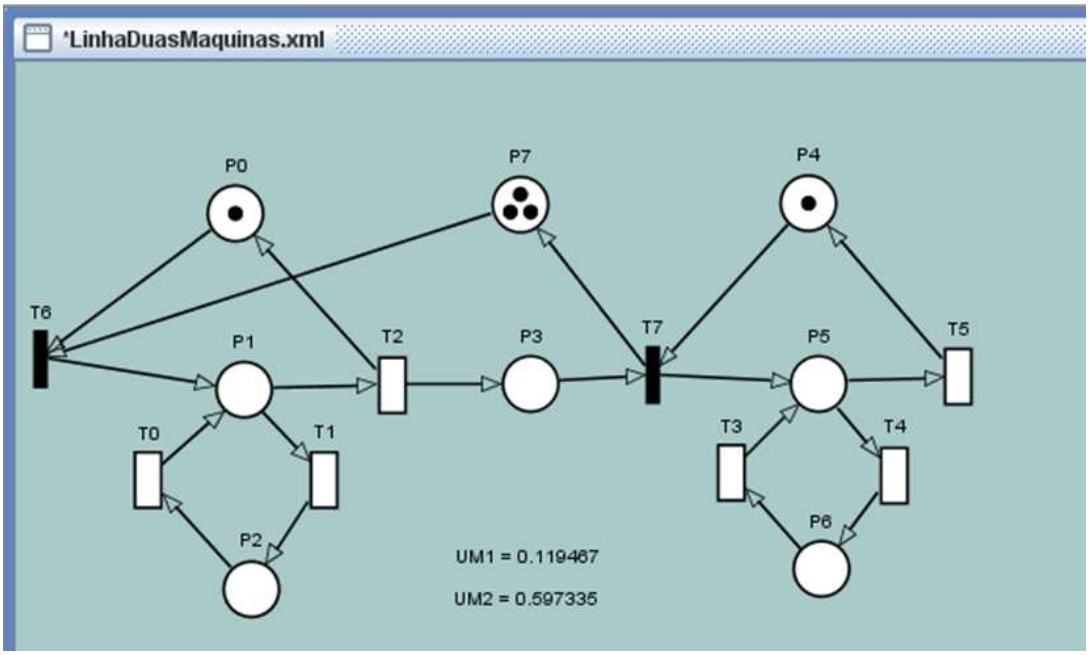


Figura 2.22 Modelo GSPN de linha de montagem com duas máquinas e um *buffer* na TimeNET

Tabela 2.8 Interpretação das marcas nos lugares e dos tempos das transições para o modelo GSPN de linha de montagem com duas máquinas e um *buffer*

Nome	Interpretação
P0	Máquina 1 funcionando e livre.
P1	Máquina 1 funcionando e ocupada.
P2	Máquina 1 parada (em reparo).
P3	*Buffer* de capacidade k (=3).
P4	Máquina 2 funcionando e livre.
P5	Máquina 2 funcionando e ocupada.
P6	Máquina 2 parada (em reparo).
P7	Capacidade do *buffer* k=3.
T0	Tempo médio para reparo na máquina 1.
T1	Tempo médio para falha na máquina 1.
T2	Tempo de processamento médio na máquina 1.
T3	Tempo médio para reparo na máquina 2.
T4	Tempo médio para falha na máquina 2.
T5	Tempo de processamento médio na máquina 2.

2.4.2 Análise GSPN para o Desempenho de um Sistema Paralelo

O modelo GSPN da Figura 2.23 é uma descrição de um sistema paralelo simples [107]. O GSPN inclui nove lugares e nove transições. As características das transições deste modelo

Tabela 2.9 Taxas/tempo de disparo de transição do modelo GSPN de duas máquinas e um *buffer*

Transição	**Taxa** $[t]^{-1}$	**Tempo** $[t]$	**Símbolo**
$T0$	5	0,2	μ_1
$T1$	3	0,33333333	λ_1
$T2$	10	0,1	ν_1
$T3$	6	0,16666666	μ_2
$T4$	4	0,25	λ_2
$T5$	2	0,5	ν_2

são resumidas na Tabela 2.10. Uma marcação inicial $(P1 = N,\ N = 2)$ *e* $P2 = P3 = \ldots P9 = 0$ é exibida na Figura 2.23. Fazendo variar o número de marcas no lugar $P1$ terremos uma família de modelos com espaço de estado crescendo de 38 (N=2) até 3685 (N=8) no estudo apresentado.

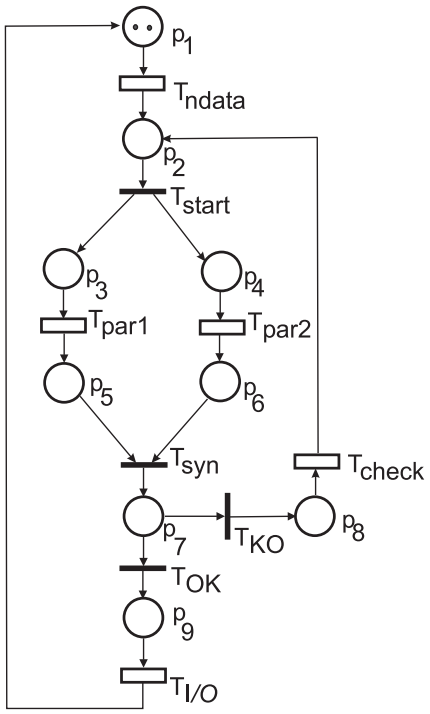


Figura 2.23 Modelo GSPN de um Sistema Paralelo na TimeNET

As operações do sistema paralelo (computação paralela simples) são modeladas como descrito a seguir. Um conjunto de novos dados é lido (disparo da transição $Tndata$), e dois processos são iniciados em paralelo com o mesmo conjunto de dados (operação de bifurcação - disparo de $Tstart$). Quando ambos os processos terminam (disparo de $Tpar1$ e $Tpar2$, respectivamente), a sincronização acontece (com a operação de junção - disparo da transição $Tsyn$). A consistência dos dois resultados é então controlada, e uma das duas transições, Tok ou Tko, dispara, indicando se os resultados são aceitáveis (ou não). Se os resultados são inconsistentes,

Tabela 2.10 Especificações dos parâmetros das transições da GSPN da Figura 2.23

Transição	**Tempo** $[t]$	**Semântica**
$Tndata$	$5,0$	Servidor-Infinito
$Tpar1$	$0,5$	Servidor-Simples
$Tpar2$	$1,0$	Servidor-Simples
$Tcheck$	$10,0$	Servidor-Simples
Ti/o	$0,2$	Servidor-Simples

Transição	**Peso**	**Prioridade**
t_{start}	1	1
t_{syn}	1	1
t_{KO}	$0,01$	1
t_{OK}	$0,99$	1

a computação completa é repetida sobre os mesmos dados, após um controle adicional (disparo de $Tcheck$); caso contrário, teremos a saída dos resultados (disparo da transição Ti/o), e um novo conjunto de dados é considerado para dar início a um novo ciclo de operações do sistema.

As duas transições Tok e Tko, juntas com o lugar de entrada, modelam uma escolha e formam o que em termos da PN é chamado de conflito de escolha livre (*free-choice conflict*). O nome conflito indica a presença de uma escolha, e o fato de que, se o lugar comum a Tok e Tko contém somente uma marca, o disparo de uma das duas transições proíbe a outra de fazer o mesmo. O conflito é de escolha livre porque as duas transições são sempre habilitadas ao mesmo tempo: a escolha de qual dispara não depende da marcação da rede.

Para dar uma especificação temporal significativa ao modelo, a transição $Tndata$, que descreve a atividade de leitura de novos dados, é associada a um tempo de disparo aleatório, com *p.d.f* exponencial, cuja média corresponde ao tempo médio para a operação de leitura dos novos dados. Se o tempo médio observado para a leitura dos novos dados é de 5 unidades de tempo, a *p.d.f.* exponencial tem taxa 0.2. Note que a observação do sistema poderá revelar que a leitura de dados da atividade real de leitura está distante de ter uma duração distribuída exponencialmente. Entretanto o paradigma de modelagem $GSPN \supseteq SPN$ não tem flexibilidade no tipo de *p.d.f.* para a descrição da duração da atividade, e portanto, somente é possível ver que corresponde a duração observada média. A taxa de transição é o único parâmetro necessário para a especificação da *p.d.f.* do tempo de disparo.

De maneira similar para as transições temporizadas $Tpar1$, $Tpar2$, Ti/o, *e* $Tcheck$ do modelo temos:

- a transição $Tpar1$, que descreve o primeiro processo paralelo, está associada com um tempo de disparo aleatório, com *p.d.f.* exponencial, cuja média corresponde ao tempo médio para a execução do primeiro processo;

- a transição $Tpar2$, que descreve o segundo processo paralelo, está associada com um tempo de disparo aleatório, com *p.d.f.* exponencial, cuja média corresponde ao tempo

médio para a execução do segundo processo;

- a transição Ti/o, que descreve a saída de resultados, está associada com um tempo de disparo aleatório, com $p.d.f.$ exponencial, cuja média corresponde ao tempo médio para a operação de saída;

- A transição $Tcheck$, que descreve o controle dos resultados, está associada com um tempo de disparo aleatório, com $p.d.f.$ exponencial, cuja média corresponde ao tempo médio para a operação de controle.

As transições $Tstart, Tsyn, Tok, eTko$ são imediatas no modelo GSPN. Deve ser observado que os valores dos pesos associados com $Tstart$ e $Tsyn$ são irrelevantes, porque estas transições nunca habilitam um conflito com outras transições imediatas. Muito pelo contrário, a escolha dos pesos associados com Tok e Tko é importante para o modelo, uma vez que eles definem a probabilidade de resultados inconsistentes após a execução paralela.

A Tabela 2.11 resume os resultados dos cálculos para a produtividade (*System Throughput*, avaliado através da expressão TimeNET $P\{\#P9 > 0\}/0.2$) e para o processamento médio (NP_1, i.e., o número médio de marcas no lugar $P1$, avaliado através da expressão TimeNET $E\{\#P0\}$ do sistema paralelo). Os resultados que estão relatados na Tabela 2.11 cobrem resultados numéricos nos quais a dimensão do problema permitiu o uso da ferramenta protótipo ISPN, que usa análise intervalar, como será visto.

Tabela 2.11 Avaliação de Desempenho, com uso da ferramenta TimeNET, para vários modelos GSPN (fazendo variar o número de marcas no lugar $P1$ da Figura 2.23). A notação de T+V é o número de marcações tangíveis = T + voláteis = V.

N	**T+V**	NP_1	***System throughput***
2	$20+18$	$1,50555067213383$	$0,301110134424231$
3	$50+60$	$2,191698215923789$	$0,438339643669379$
4	$105+150$	$2,818244201068655$	$0,56364884902282$
5	$196+315$	$3,3721863476126$	$0,674437269685144$
6	$336+588$	$3,841255710038225$	$0,768251142007472$
7	$540+1008$	$4,216923215186429$	$0,843384643062969$
8	$825+1620$	$4,497756850783655$	$0,899551370521738$
9	$1210+2475$	$4,691392885702292$	$0,93827857972694$

2.5 Considerações finais

A aritmética intervalar tem um vasto campo de aplicação, principalmente onde existe aritmética intervalar implantada (*softwares* disponíveis) [55, 66]. Um destes campos é a otimização global [55]. O cálculo da imagem de uma função em um espaço de parâmetros é um problema de otimização global com restrições. Contudo, do ponto de vista da aritmética intervalar ele é

um problema de otimização global sem restrições. Outros campos são robótica, processamento de imagens, análise espectral, astrofísica, geologia, computação gráfica, etc., campos onde dados são imprecisos ou obtidos por medidas indiretas. Em [66] estão descritas aplicações da aritmética intervalar em algoritmos de otimização global, solução de sistemas lineares, lógica nebulosa (*fuzzy logic*), modelos econômicos, controle de qualidade, sistemas médicos especialistas, etc.

Este capítulo introduziu a aritmética intervalar, que tem sido implantada em muitas plataformas e linguagens de programação, e alguns *softwares* são comercializados. A breve introdução sobre aritméticas intervalares é complementada nos capítulos seguintes. O Capítulo 5 é dedicado à solução de sistemas de equações lineares intervalares. O Apêndice A coloca os modelos intervalares como membro nas famílias de modelos. No Apêndice B o processo estocástico é ampliado para abrigar as cadeias de Markov com probabilidades/taxas intervalares. A avaliação de funções intervalares, apresentada no Apêndice C, é o complemento da análise intervalar que possibilita a computação das métricas intervalares dos modelos ISPN

Após a parte inicial dedicada à análise intervalar a maior parte deste capítulo foi dedicada a Redes de Petri. Tratou-se de conceitos e definições básicas para formalizar a definição das Redes de Petri Estocásticas. Apresentou-se alguns modelos que foram analisados com ajuda da ferramenta TimeNET. O material contido neste capítulo fornece a conceituação e fundamentação essencial para o desenvolvimento do formalismo ISPN apresentado no Capítulo 4.

CAPÍTULO 3

Trabalhos relacionados

Este Capítulo mostra um sumário de trabalhos relacionados. Ele está dividido em duas seções principais: (1) Sistemas Nebulosos e(2) Modelos intervalares em avaliação de desempenho. Os modelos intervalares aqui considerados surgem quando parâmetros intervalares estão associados aos modelos e, além disso, tratados matematicamente através da análise intervalar.

3.1 Sistemas nebulosos

Parte significativa dos sistemas reais são muito complexos e mal-definidos para serem modelados por técnicas analíticas. Estes sistemas tratam informações e características vagas, imprecisas, qualitativas, linguísticas ou até mesmo incompletas. O conceito nebuloso emula o pensamento humano, capaz de lidar com incertezas, ambiguidades e contradições, através de técnicas fundamentadas em conjuntos nebulosos e lógica nebulosa [73, 147, 132]. Os conjuntos nebulosos e a lógica nebulosa são aplicados a campos básicos, tais como aritmética, topologia, teoria dos grafos nebulosos e teoria de probabilidade nebulosa.

3.1.1 Aritmética nebulosa

A aritmética nebulosa lida com adição, subtração, multiplicação e divisão de números nebulosos. Existem dois métodos de realizar a aritmética nebulosa [147]: (1) através do princípio da extensão, e (2) usando a aritmética intervalar.

Princípio da Extensão é uma ferramenta utilizada para tornar nebulosas as operações, equações, funções, e assim por diante. $\odot$ denota a adição, subtração, multiplicação, ou divisão de números nebulosos. Para calcular $P = M \odot N$ com números nebulosos é usado o princípio da extensão seguinte:

$$P(z) = sup_{x,y}\{min(M(x);N(y))|x \odot y = z\}. \tag{3.1}$$

Quando $\odot$ é divisão, 0 (zero) não pode ser a pertinência positiva do divisor ($M \div N$ assume $N(0) = 0$). Esboçando a adição dos números nebulosos M e N para produzir o número nebuloso P, considerando, para isto, que $P(z = 7)$, então, para cada par $(x;y)$ calcula-se $M(x)$ e $N(y)$, obtém-se o mínimo de $M(x)$ e $N(y)$, e, finalmente, obtém-se o máximo de todos os mínimos como valor para $P(7)$.

Quando a Equação (3.1) pode ser calculada analiticamente, tem-se um método para generalizar operadores precisos (*crisp operators* [147]) em operadores nebulosos (*fuzzy operators* [147]), mas este procedimento tem - geralmente - grande complexidade computacional. Em

seguida será apresentado o segundo procedimento, que é implementado mais facilmente em programas de computadores.

3.1.2 Cortes-alfa e a aritmética intervalar

Se $\alpha \in [0;\ 1]$, o corte-alfa de $\mathscr{A}$, um subconjunto nebuloso do conjunto universal $\mathscr{X}$, escrito como $\mathscr{A}[\alpha]$,é definido como sendo o conjunto preciso $\{x \mid \mathscr{A}(x) \geq \alpha\}$. Este conjunto é uma coleção de todo $x \in \mathscr{X}$, cujo valor de pertinência é no mínimo alfa. $\mathscr{A}(0)$ é definido separadamente como a base do número nebuloso. Por exemplo, [147] $\mathscr{A}[0] = [a;\ c]$ para números nebulosos triangulares. A função relacionando números membros a seu grau de pertinência é chamada de função de pertinência, assumindo que o número nebuloso triangular parte subindo de zero em $x = a$, atinge um máximo em $x = b$, e desce até zero em $x = c$. Esta é a função de pertinência $\mu(x)$ de um número nebuloso triangular, dado por

$$\mu(x) = \begin{cases} 0 & \text{se } x \leq a \\ (x-a)/(b-a) & \text{se } a < x \leq b \\ (c-x)/(c-b) & \text{se } b < x \leq c \\ 0 & \text{se } x > c \end{cases}$$

A função de pertinência é melhor visualizada através de um gráfico como na Figura 3.1 para o número nebuloso triangular.

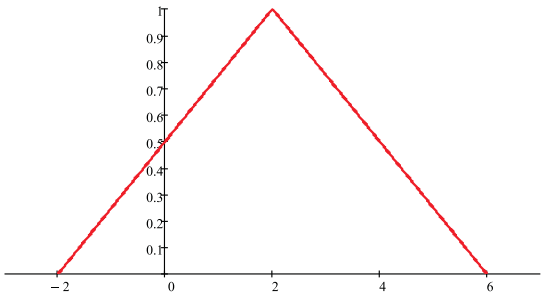


Figura 3.1 Função de pertinência para um número nebuloso triangular com suporte $A[0] = [-2;\ 6]$ e vértice em $x = 2$. A pertinência de -1 é 0,25.

O corte-alfa de um número nebuloso é sempre um intervalo fechado limitado. O núcleo de um número nebuloso é o conjunto $A[1]$ e o suporte é o intervalo $A[0]$. Escreve-se $A[\alpha] = [a_1(\alpha);\ a_2(\alpha)]$ para um número nebuloso A, onde $a_i(\alpha)$ representa os pontos extremos dos intervalos, que são, em geral, função de alfa. Por exemplo, se A é o número nebuloso da Figura 3.1 com intervalo base $[-2;\ 6]$ e vértice em $x = 2$, então $A[\alpha] = [-2+4\alpha;\ 6-4\alpha]$ para $\alpha \in [0;\ 1]$.

Para os números nebulosos A e B consideremos:

$$A[\alpha] = [a_1(\alpha);\ a_2(\alpha)]\ e$$
$$B[\alpha] = [b_1(\alpha);\ b_2(\alpha)]\,.$$

Usando o corte-alfa e a aritmética intervalar (clássica), calcula-se $P = A + B$ como $P[\alpha] = A[\alpha] + B[\alpha] = [a_1(\alpha) + b_1(\alpha);\ a_2(\alpha) + b_2(\alpha)]$ para obter-se o corte-alfa da soma P. O mesmo raciocínio dever ser seguido para subtração, multiplicação e quando 0 (zero) não pertence a B. Então, $P = A \div B$ é definido e cortes-alfa de P são calculados como $P[\alpha] = [a_1(\alpha);\ a_2(\alpha)] \times [1 \div b_2(\alpha);\ 1 \div b_1(\alpha)]$.

3.1.3 Comparação dos métodos

O procedimento corte-alfa com a aritmética intervalar é facilmente incorporado em programas de computadores, desde que seja calculado em valores discretos e somente calculado para, digamos, α igual a 0; 0,1; ...1.

Outros procedimentos calculam os mesmos valores para $A + B$, $A - B$, $A \times B$ e $A \div B$ para números nebulosos A e B. Porém isto não é verdadeiro para cálculos de expressões nebulosas em geral. Por exemplo, $P = (A + B) \div B$ pode produzir resultados diferentes pelos dois métodos.

Deve ser ressaltado que o uso da aritmética intervalar com cortes-alfa como método para realizar a aritmética nebulosa não significa que sistemas nebulosos usando cortes-alfa possam ser entendidos como modelos intervalares. Esta compreensão deve ser, geralmente, desconsiderada.

O editorial de Lodwick & Jamison [83] menciona 4 pontos que demonstram a relação existente entre teoria de conjuntos nebulosos e análise intervalar. O primeiro ponto é aritmética nebulosa definida pelo principio da estensão (arimética nebulosa através da aritmética intervalar sobre cortes-alfa). O segundo ponto de contato, e talvez mais básico que o primeiro, é o próprio princípio de extensão. O princípio da extensão na análise intervalar é chamado a extensão unida. O princípio da extensão de Zadeh [122, 166] e a extensão unida de Moore são relacionados fundamentalmente pelo significado de conjunto de funções [152, 153].

O terceiro ponto de contato é talvez mais explícito e ele é a análise intervalar nebulosa [30, 28, 98]. Neste caso o uso da análise intervalar na teoria de conjuntos nebulosos é direta. Um intervalo $[a;b]$ é considerado como um número na análise intervalar. Contudo, um intervalo pode ser considerado como um conjunto. Como um conjunto, um intervalo é também um conjunto nebuloso com função de pertinência retangular:

$$\mu(x) = \begin{cases} 1 & \text{se } x \in [a;\ b] \\ 0 & \text{caso contrário.} \end{cases}$$

Desta maneira a análise intervalar é vista como um subconjunto da teoria de conjuntos nebulosos. Isto produz um quarto ponto de contato de análise de incerteza com a lógica. A teoria de conjuntos nebulosos surgiu da necessidade de um modelo matemático de incertezas. A análise

de intervalo surgiu da necessidade de modelar erros e incertezas em cálculos com computadores digitais para desenvolver ferramentas para a análise automática de erros na solução de problemas contínuos resolvidos no computador digital.

3.2 Modelos intervalares em avaliação de desempenho

Em contribuições onde intervalos foram propostos como meio para modelar incertezas dos parâmetros os algoritmos existentes na solução do modelo podem ser adaptados aos parâmetros intervalares e substituindo a aritmética convencional pela aritmética intervalar. Entretanto, o chamado problema da dependência pode causar intervalos extremamente largos para as medidas de desempenho computadas. A divisão intervalar foi proposta como uma técnica para superar este problema. No trabalho [89], é dada uma visão geral de algoritmos de divisão existentes e proposto o uso de um método de divisão seletivo que reduz significativamente a complexidade computacional das avaliações intervalares. Além disso, a exploração das propriedades parciais da monotonicidade para diminuir ainda mais a complexidade computacional é também discutida. Os métodos propostos são ilustrados através de dois exemplos: um modelo de desempenho pequeno do acesso múltiplo, com prevenção da colisão pelo protocolo (MACABI) convite para redes móveis sem fio ad hoc, e um modelo mais complexo de uma implantação de um servidor *Enterprise Java Beans* (EJB).

Uma visão geral das aplicações da aritmética intervalar está apresentada em [95]. Aplicações anteriores da aritmética intervalar para avaliação de desempenho incluem o cálculo de limites robustos de processamento para redes de filas gerais com múltiplas classes [96, 92]. Em [90] é discutida a análise de valores médios (*MVA - Mean Value Analysis*) com variabilidades representadas por intervalos. Os limites da avaliação do desempenho para redes de fila com uma única classe, caracterizado por incertezas e variabilidades, são apresentados em [86]. Uma extensão de técnicas de análise de gargalo convencional é considerado em [87] (O recurso com a utilização mais elevada é chamado de gargalo. O desempenho do sistema é limitado pelo desempenho do componente do gargalo). A aritmética intervalar é também usada para o cálculo de limites na avaliação de desempenho de concorrência (*softwares*) [93].

Os modelos analíticos usam parâmetros de valores simples (tal como demanda média do recurso) e produzem um único ponto médio para cada índice de desempenho de interesse (tal como tempo médio de resposta e utilização média do recurso). Contudo, a caracterização pontual dos parâmetros é inadequada quando incertezas e variabilidades são associadas aos parâmetros do sistema. A avaliação de desempenho aplicada na engenharia de software integra modelagem de desempenho em várias fases do projeto do software e também na fases de implementação e testes [47, 149]. As incertezas e variabilidades associadas a um ou mais parâmetros do sistema são comuns aos estágios primários do projeto de um sistema. O projetista experiente normalmente tem uma boa ideia sobre intervalos de valores associados com os parâmetros do modelos em fase de projeto devido a experiências anteriores com sistemas similares. Quando os parâmetros são obtidos através de medidas, os intervalos de confiança, obtidos pela análise estatística, são uma ferramenta importante para aumentar a confiabilidade dos resultados. Os

parâmetros intervalares podem surgir de situações nas quais a análise de limites é usada em um nível de um modelo hierárquico que produz parâmetros de entradas intervalares para outro nível. Outros empregos dos parâmetros intervalares são na análise de pior/melhor caso e nos estudos de caso de sensibilidade.

Na avaliação de desempenho que utiliza modelagem hierárquica, as técnicas para estabelecer limites usados em uma camada do modelo hierárquico produzem os intervalos de entrada para a próxima camada [25]. As técnicas para estabelecer limites para tempo de execução [54] que estão integradas na ferramenta de modelagem PEPP [25] são exemplos. Os limites para o tempo médio de execução de programas paralelos são também fornecidos na análise serial usada na predição de desempenho com PAMELA [45]. Os intervalos de entrada são utilizados por sistemas representados usando técnicas de modelagem nebulosas [147, 73, 65] através dos chamados cortes-α ($\alpha - cuts$) dos parâmetros representados por números nebulosos [85].

As variabilidades nos parâmetros ocorrem em diferentes fases de operação de sistemas. Como exemplo, consideremos um sistema cliente-servidor onde diferentes demandas médias para um dado dispositivo podem ocorrer durante vários períodos de tempo. Outro bom exemplo, presente no cotidiano de quem faz compras (clientes), é a variabilidade exibida num caixa de supermercado (servidor), em que taxas de serviços diferentes ocorrem em diferentes períodos do dia. As variabilidades nas demandas dos serviços ocorrem também indiretamente; demandas diferentes de serviços foram observadas em um sistema de banco de dados descrito em [11] durante períodos de tempo, quando uma quantidade de memória menor estava disponível para processamento das transações e um grande número de operações de I/O foi observado. Mesmo considerando o sistema como estando em estado-estacionário durante cada fase de operação, um modelo convencional de redes de filas com uma única classe ou um modelo de redes de filas com múltiplas classes não foram adequadas para o cálculo da avaliação de desempenho [11]. Os modelos de redes de filas com uma classe (análise de valores médios) fornecem estimativas de medidas de desempenho, e não simplesmente limites. Uma única classe pode ser usada para: a) próximo passo para modelos mais detalhados; b) uma única carga de trabalho de interesse; c) cargas homogêneas. Uma única classe pode não ser apropriada para: a) cargas de trabalhos múltiplas e distintas (*batch* x *timesharing*, *CPU-bound* x *I/O bound*); b) entradas/saídas do modelo dependentes da classe (neste caso, deve-se usar modelos com múltiplas classes). As variabilidades nos parâmetros do sistema surgem também de outras situações, conduzindo a grandes erros nos cálculos das métricas quando um único valor médio é usado no modelo para cada parâmetro de interesse [11]. A associação de números nebulosos para parâmetros de modelos com incertezas é proposto em [88].

Com foco na caracterização de incertezas e variabilidades nos parâmetros do modelo, Majumdar *et al* [94] usaram intervalos, histogramas e adaptações de técnicas analíticas existentes para manusear parâmetros de valores intervalares. Vários modelos analíticos existentes foram considerados, incluindo modelos de redes de filas [80] usados extensivamente em vários estudos de sistemas de avaliação de desempenho computacionais. Os modelos de redes de filas podem ser também modelados por redes de Petri, mas o alto-nível na modelagem das redes de

Petri é superior aos das redes de filas. As redes de filas são geralmente usadas como primeira aproximação na busca por modelos mais refinados. Para modelagem de sistemas concorrentes, os modelos apresentado por Thomassian & Bay [156] são mais aplicáveis que modelos de redes de filas.

3.3 Considerações Finais

A procura por trabalhos relacionados com a tese tem sido constante. É possível ampliar um pouco o espectro com a inclusão de trabalhos relacionados que se utilizam das mesmas técnicas de solução no estudo de modelos intervalares. No nosso trabalho ficamos restritos à análise do estado-estacionário dos modelos ISPN. No caso, é necessário determinar o conjunto solução de um sistema de equações lineares intervalares. Por exemplo, modelos intervalares aplicados a engenharia estrutural [51] e análise de circuitos [69]. Podemos encontrar, também, trabalhos que poderão servir, direta ou indiretamente, para aprofundar os estudos ISPN. O trabalho de Sakalna [148] traz um modelo intervalar para uma treliça, e os trabalhos recentes de Kolev [70, 71] também seguem a mesma linha de raciocínio: poderão contribuir para futuros trabalhos ISPN. Neste casos, o método de solução para sistemas de equações intervalares lineares poderá ser utilizado na análise de estado estacionário ISPN. É sempre útil ter métodos de solução disponíveis para serem aplicados. O projeto COPRIN [21] é dedicado ao estudo e à aplicação da análise intervalar [14, 15, 48, 49].

Foi dada prioridade aos trabalhos relacionados à utilização de modelos intervalares a sistemas de eventos discretos. Buscou-se, em particular, por formalismos tais como redes de filas, álgebra de processos ou autômatos temporizados que tenham sido estendidos para abordar modelos intervalares. Podemos afirmar que não foi detectada, após o começo das pesquisas para desenvolvimento formal do ISPN em 2003, uma formalização para modelos intervalares usando redes de filas, álgebra de processos ou autômatos temporizados. A escolha desses formalismos é devido ao fato de serem eles utilizados na modelagem de sistemas de eventos discretos para aplicações de avaliação de desempenho, confiabilidade, planejamento de capacidade, etc. É possível considerar que a formalização ISPN é inédita no domínio de aplicações de modelagem estocástica de alto nível.

CAPÍTULO 4

Modelo ISPN

O Capítulo 2 foi em grande parte dedicado à introdução do formalismo de Rede de Petri com ênfase na conceituação e aplicação da metodologia na área de modelagem de desempenho, confiabilidade e análise dos sistemas. A análise quantitativa do sistema considera as variáveis temporizadas associadas a PN como variáveis aleatórias. Com a introdução do tempo como variável aleatória temos uma classe de modelos chamados SPN (*Stochastic Petri Net* - Rede de Petri Estocástica). Neste capítulo é definido formalmente o modelo ISPN, uma extensão intervalar do modelo GSPN, com cadeias de Markov intervalares subjacentes. A estratégia geral para análise, as definições das métricas intervalares e alguns exemplos são dados com o objetivo de introduzir a análise ISPN.

4.1 Formalismo ISPN

ISPN é uma extensão do modelo GSPN (*Generalized Stochastic Petri Nets*) de maneira a introduzir a análise intervalar [33] e tem um grande potencial para aplicações. Como parte da contribuição no desenvolvimento do presente trabalho, foram feitas aplicações na avaliação de desempenho e dependabilidade (*dependability*) [33, 34, 35]. A GSPN é uma PN (*Petri Net*) temporizada particular que incorpora transições estocásticas temporizadas (traçadas como caixas brancas) e transições imediatas (traçadas como caixas pretas ou barras pretas finas). As transições temporizadas têm tempo de disparo exponencialmente distribuído, e transições imediatas disparam em tempo zero. GSPN foi originalmente definida em [99] e reformulada posteriormente [9].

Ao considerar os efeitos das variabilidades e incertezas com a introdução de intervalos para especificar os parâmetros das distribuições exponenciais e dos pesos das transições imediatas, é possível abordar situações em estágios iniciais de projetos de sistemas, em que os parâmetros não estão bem definidos, mas sabe-se pertencer a determinados intervalos. Esta situação também é comum em sistemas com demandas médias diferentes no decorrer de tempo. Os resultados das métricas intervalares, provenientes da análise intervalar, incluem os efeitos das variabilidades simultâneas de todos os parâmetros no modelo intervalar.

O formalismo ISPN exige o desenvolvimento de conceituação e interpretação relativos a cadeias de Markov intervalares, distribuição de probabilidades intervalares e uso de algoritmos e metodologias de análise intervalar. Uma definição formal de modelos ISPN é fornecida a seguir. Esta definição mantém diversos componentes da definição de SPN apresentada em [46] inalterados, mas considera intervalos reais atribuídos como retardos de disparo e pesos das transições, em vez de adotar valores reais únicos.

Considere $ISPN = (P;T;I;O;H;\Pi;G;M_0;Atts)$ seja a tupla que define a rede de Petri estocástica intervalar, onde:

- $P = \{p_1;p_2;\cdots p_n\}$ é o conjunto de lugares;
- $T = \{t_1;t_2;\cdots t_m\}$ é o conjunto de transições;
- $I \in (\mathbb{N}^n \rightarrow \mathbb{N})^{n\times m}$ é uma matriz das multiplicidades, dependentes da marcação, dos arcos de entrada, onde a entrada i_{jk} de I corresponde à multiplicidade dos arcos da entrada de um lugar p_j para a transição t_k

$$[A \subset (P \times T) \cup (T \times P) - \text{ conjunto de arcos}];$$

- $O \in (\mathbb{N}^n \rightarrow \mathbb{N})^{n\times m}$ é uma matriz das multiplicidades, dependentes da marcação, dos arcos de saída, onde a entrada o_{jk} de O corresponde à multiplicidade dos arcos da saída de uma transição t_j para o lugar p_k;
- $H \in (\mathbb{N}^n \rightarrow \mathbb{N})^{n\times m}$ é uma matriz das multiplicidades dos arcos inibidores, onde a entrada h_{jk} de H dá a possível multiplicidade, dependente da marcação, dos arcos inibidores de um lugar p_j para a transição t_k;
- $\Pi \in (\mathbb{N}^m \rightarrow \mathbb{N})^m$ é um vetor de prioridades que associa a menor prioridade(0) para as transições temporizadas e prioridades superiores (≥ 1) para as transições imediatas;
- $G \in (N^n \rightarrow \{true;false\})^m$ é um vetor que atribui uma condição de guarda relacionada às marcações dos lugares para uma dada transição;
- $M_0 \in \mathbb{N}^n$ é um vetor que atribui uma marcação inicial dos lugares (o estado inicial);
- $Atts = ([Dist];[W];Markdep;Policy;Concurrency)^m$ abrange o conjunto dos atributos para as transições, onde
  - $[Dist] \in \mathbb{N}^m \rightarrow [\mathscr{F}]$ é uma função de distribuição intervalar do disparo, com possível dependência da marcação ($\mathscr{F}$ é o conjunto de funções de distribuição intervalares positivas, isto é, com um suporte no $[0,\infty)$);
  - $[W] \in \mathbb{N}^m \rightarrow I(\mathbb{R}^+)$ é um peso intervalar com possível dependência da marcação,
  - $Markdep \in \{constant;enabdep\}$, onde a distribuição intervalar do tempo de disparo da transição poderá ser independente da marcação $(constant)$ ou com dependência na habilitação (onde $enabdep$ indica que distribuição intervalar depende da condição de habilitação atual);
  - $Policy \in \{prd;prs\}$ é a política de preempção (*prd- preemptive repeat different - repetição preemptiva diferente*, significa que quando uma transição que ficou em preempção é habilitada novamente o tempo de disparo prévio decorrido é perdido; *prs- preemptive resume - preemptiva retomada*, na qual o tempo de disparo relacionado a uma transição que ficou em preempção é retomada quando a transição se torna habilitada outra vez);

– *Concurrency* $\in \{ss; is\}$ é o grau de concorrência das transições, onde *ss* representa a semântica do servidor único e *is* descreve a semântica do servidor infinito.

Vale destacar que quando somente intervalos degenerados (pontos) são atribuídos aos retardos de disparos e aos pesos das transições imediatas, o modelo ISPN é reduzido a um GSPN equivalente. Neste caso a análise ISPN fornece resultados GSPN verificados (garantidos).

4.2 Análise Estacionária da ISPN

Uma ISPN é uma extensão da GSPN, na qual parâmetros intervalares devem ser considerados para obtenção do gerador infinitesimal intervalar do ICTMC (Apêndice E). Os algoritmos clássicos encontrados na literatura [10] foram adaptados para considerar os coeficientes intervalares do modelo de ISPN. Esta é uma contribuição para análise de modelos ISPN que será apresentada a seguir. A análise de desempenho ISPN é realizada através de quatro sub-tarefas:

- geração do IERG (*Interval Extended Reachability Graph* - Grafo de alcançabilidade estendido intervalar);
- eliminação das marcações voláteis e as correspondentes transições de estado;
- análise do estado estacionário intervalar;
- computação das métricas. Métricas tais quais o número médio de marcas em cada lugar ou conjunto de lugares, e a taxa de disparo (*throughput*) das transições temporizadas é calculada pela aritmética intervalar.

As ISPNs contêm transições imediatas e transições com tempos de disparo exponencialmente distribuídos (com parâmetros intervalares possíveis), em que pelo menos um intervalo não degenerado é considerado em uma transição exponencial ou imediata. Isto resulta em várias possibilidades ao determinar quais transições são habilitadas. As marcações nas quais pelo menos uma transição imediata é habilitada são chamadas marcações voláteis. Por outro lado, marcações nas quais estão habilitadas só transições exponenciais são chamadas marcações tangíveis, da mesma forma que nas GSPN. Uma ISPN que contém marcações tangíveis e voláteis é equivalente a uma ICTMC (Apêndice E). Neste caso, é chamada uma cadeia de Markov intervalar embutida. Se removermos as marcações voláteis, então teremos uma cadeia de Markov intervalar reduzida embutida. Essa cadeia que só contém marcações tangíveis é usada para computar as probabilidades intervalares de estado estacionário de um lugar que é marcado ou não.

4.2.1 Geração do IERG

A primeira das quatro etapas da análise ISPN é a geração do IERG (*interval extended reachability graph*). Com a geração do IERG é o conjunto de marcações $\mathscr{M} = \mathscr{V} \cup \mathscr{T}$ divididas nos conjuntos de marcações tangíveis $\mathscr{T}$ e voláteis $\mathscr{V}$. Através da eliminação de marcações

voláteis, discutida a seguir, usando métodos da análise intervalar, obtém-se a matriz geradora infinitesimal $[Q]$ da ICTMC subjacente ao modelo ISPN.

Dada uma ISPN, um grafo de alcançabilidade estendido intervalar IERG é gerado contendo marcações como nós e informações estocásticas anotadas aos arcos relacionando às marcações com as demais. O grafo de alcançabilidade gerado de uma ISPN é um grafo dirigido $RG(ISPN) = (V;E)$ (RG = *Reachability Graph* Grafo de Alcançabilidade), onde $V = RS(ISPN)$ (RS = *Reachability Set* Conjunto de Alcançabilidade) e $E = \left\{ \langle \mathbf{m}; t; \mathbf{m}' \rangle \mid \mathbf{m}, \mathbf{m}' \in RS(ISPN) \; e \; \mathbf{m} \xrightarrow{t} \mathbf{m}', \; t \in T \right\}$ são o conjunto de nós e arestas, respectivamente.

Se a rede ISPN é limitada, o $RG(ISPN)$ é finito e pode ser construído, por exemplo, baseado no **Algoritmo 1** [47]. A atividade definida no Passo 2.1 assegura que nenhuma marcação é visitada mais que uma vez. Cada marcação visitada é etiquetada (Passo 2.1), e o Passo 2.2.3 assegura que somente as únicas marcações adicionadas a V são aquelas que não foram previamente adicionadas. Quando uma marcação é visitada, somente aquelas arestas que representam o disparo de uma transição habilitada são adicionadas ao conjunto E (Passo 2.2.4).

Algoritmo 1

==========

(Geração do IERG **)**

Entrada - Um modelo ISPN.
Saída - Um grafo dirigido $RG(ISPN) = (V;E)$ de um sistema de rede limitado.

1. Inicializar $RG(ISPN) = (\{\mathbf{m}_0\}, \emptyset)$; $\mathbf{m}_0$ está não etiquetada.
2. **enquanto** existe um nó não etiquetado **m** em V **faça**
   2.1 Selecione um nó não etiquetado $\mathbf{m} \in V$ e etiquete-o
   2.2 **para** cada transição habilitada **t** em **m faça**
      2.2.1 Calcule $\mathbf{m}'$ tal que $\mathbf{m} \xrightarrow{t} \mathbf{m}'$;
      2.2.2 **se** existe $\mathbf{m}'' \in V$ tal que $\mathbf{m}'' \xrightarrow{\sigma} \mathbf{m}'$ e **m**" $\leq$ **m**'
            então o algoritmo falha e termina;
                (a condição de não limitação foi detectada).
      2.2.3 **se** não há nenhum $\mathbf{m}'' \in V$ tal que $\mathbf{m}'' = \mathbf{m}'$
            então $V := V \cup \{\mathbf{m}'\}$; ($\mathbf{m}'$ é um nó não etiquetado).
      2.2.4 $E := E \cup \{\langle \mathbf{m}; t; \mathbf{m}' \rangle\}$
3. O algoritmo tem sucesso e RG(ISPN) é o grafo de alcançabilidade estendido intervalar.

==========

4.2.2 Eliminação das Marcações Voláteis

A segunda das quatro etapas da análise ISPN é a eliminação das marcações voláteis, que é o passo para geração da ICTMC de um determinado modelo ISPN. Uma vez que o IERG foi gerado, gera-se a ICTMC com uso de algoritmos matriciais eficientes [10].

As marcações $\mathscr{M} = \mathscr{V} \cup \mathscr{T}$ do conjunto de alcançabilidade de uma ISPN é dividido em dois cunjuntos, as marcações voláteis $\mathscr{V}$e as marcações tangíveis $\mathscr{T}$. Considere:

$$[\mathbf{P}]^{\mathscr{V}} = [\mathbf{P}]^{\mathscr{V}\mathscr{V}} \mid [\mathbf{P}]^{\mathscr{V}\mathscr{T}} \tag{4.1}$$

Denote a matriz intervalar, onde:

- $[\mathbf{P}]^{\mathscr{V}\mathscr{V}}$ - probabilidades intervalares de transições entre marcações voláteis;
- $[\mathbf{P}]^{\mathscr{V}\mathscr{T}}$ - probabilidades intervalares de transições de marcações voláteis para marcações tangíveis.

Além disso, considere:

$$[\mathbf{U}]^{\mathscr{T}} = [\mathbf{U}]^{\mathscr{T}\mathscr{V}} \mid [\mathbf{U}]^{\mathscr{T}\mathscr{T}} \tag{4.2}$$

Denote uma matriz intervalar, onde:

- $[\mathbf{U}]^{\mathscr{T}\mathscr{V}}$ - taxa intervalar das transições de marcações tangíveis para marcações voláteis;
- $[\mathbf{U}]^{\mathscr{T}\mathscr{T}}$ - taxa intervalar das transições entre marcações tangíveis.

A mesma informação, como contida no IERG, é fornecida por $[\mathbf{P}]^{\mathscr{V}}$ junto com $[\mathbf{U}]^{\mathscr{T}}$.

A matriz de taxa intervalar $[\mathbf{U}]$ tem dimensão $|\mathscr{T}| \times |\mathscr{T}|$, onde $\mathscr{T}$ denota o conjunto de marcações tangíveis:

$$[\mathbf{U}] = [\mathbf{U}]^{\mathscr{T}\mathscr{T}} + [\mathbf{U}]^{\mathscr{T}\mathscr{V}}(1 - [\mathbf{P}]^{\mathscr{V}\mathscr{V}})^{-1}[\mathbf{P}]^{\mathscr{V}\mathscr{T}}. \tag{4.3}$$

A matriz geradora infinitesimal intervalar é $[\mathbf{Q}] = [q]_{ij}$, onde suas entradas são determinadas por:

$$[q]_{ij} = \begin{cases} [u]_{ij} \;\; se\; i \neq j \\ \\ -\sum\limits_{\substack{k \in \mathscr{T} \\ k \neq i}} [u]_{ik} \;\; se\; i = j \end{cases} \tag{4.4}$$

e $\mathscr{T}$ denota o conjunto de marcações tangíveis. Observa-se que $[\mathbf{U}]$ pode conter entradas diferentes de zero na sua diagonal. As entradas diferentes de zero são ignoradas e consideradas como informações redundantes, quando é criada a matriz geradora infinitesimal intervalar $[\mathbf{Q}]$, como mostra a Equação 4.4.

4.2.3 Análise intervalar do estado estacionário

Agora será descrita a terceira das quatro etapas da análise ISPN. A solução do estado estacionário do modelo ICTMC subjacente ao modelo ISPN é obtida pela solução do sistema linear de equações intervalares com o número de equações igual ao número de marcações tangíveis:

$$\begin{cases} [\vec{\pi}] \cdot [\mathbf{Q}] = \vec{0} \\ \sum_{M \in \mathscr{T}} [\pi](M) = 1 \end{cases} \tag{4.5}$$

$[\vec{\pi}]$ é o vetor intervalar para a *p.m.f.* (*probability mass function* - função de probabilidade) do equilíbrio, considerando as marcações tangíveis alcançáveis, e escrevemos $[\pi](M)$ para representar a probabilidade intervalar do estado estacionário em uma marcação tangível M.

Uma vez que a matriz geradora intervalar $[\mathbf{Q}]$ da ICTMC associada com um modelo da ISPN foi derivada, a probabilidade do estado estacionário é calculada de modo que as métricas de interesse possam ser computadas subsequentemente.

4.2.4 Índices intervalares de desempenho

A computação dos índices de desempenho (métricas) de interesse representa a quarta, e última, etapa na análise ISPN. No caso da análise ISPN do estado estacionário, onde a p.m.f. intervalar já foi obtida, os índices são calculados através da avaliação de funções intervalares. A avaliação de funções intervalares é o assunto explorado no Apêndice C. Segue, então, a descrição de alguns índices intervalares de desempenho [105].

- A fração de tempo de permanência esperada em um subconjunto de marcações $\mathscr{M}$ onde $M \in \mathscr{M}$, no intervalo $t \in [0,\ t]$, pode ser calculada como:

$$[\psi](\mathscr{M}; t) = \frac{1}{t} \sum_{M \in \mathscr{M}} \int_0^t [\pi]_M(z) dz;$$

  $[\pi]_M(t)$ é a probabilidade intervalar associada à marcação M no instante t. No estado estacionário, $[\psi](\mathscr{M}; t)$ é igual à probabilidade intervalar:

$$[\pi](\mathscr{M}) = \sum_{M \in \mathscr{M}} [\pi]_M.$$

- A probabilidade intervalar associada à uma condição definida através das marcações de lugares pode ser computada pela adição das probabilidades intervalares de todas as marcações, na qual a condição correspondente ao evento permanece válida. Então, por exemplo, a probabilidade intervalar do estado estacionário de uma condição $\mathscr{A}$ associada às marcações $(M \in \mathscr{M})$ é obtida como:

$$[P](\mathscr{A}) = \sum_{M \in \mathscr{M}} [\pi]_M;$$

  enquanto a probabilidade da condição $\mathscr{A}$ no instante t é:

$$[P](\mathscr{A}; t) = \sum_{M \in \mathscr{M}} [\pi]_M(t).$$

  Esta fórmula também pode ser usada para computar a probabilidade intervalar de uma dada condição que é satisfeita *pela primeira vez* no instante t.

- A *p.m.f.* intervalar do número de marcas do estado estacionário num lugar, digamos p, pode ser obtida pela computação das probabilidades intervalares individuais da *p.m.f.* intervalar como probabilidades intervalares das condições "lugar p contém k marcas". - A *p.m.f.* do número de marcas num lugar, no instante t, pode ser obtida similarmente, usando a condição "o lugar p contém k marcas no instante t".

- O número médio de marcas num lugar (no instante t) pode ser calculado da *p.m.f.* intervalar do número de marcas naquele lugar (no instante t).

- O número intervalar esperado de disparos de uma transição t_k no intervalo de tempo $[0;t]$, $[f]_{t_k}(t)$, pode ser calculado, integrando sobre um dado intervalo de tempo a taxa intervalar de disparo da transição t_k, expressa como a soma sobre todos os estados M que habilitam t_k, da taxa intervalar de disparo t_k em M, ponderada pela probabilidade intervalar de estar em M no instante t:

$$[f]_{t_k}(t) = \int_0^t \sum_{M:t_k \in E(M)} [w](t_k;M) \cdot [\pi]_M(z)dz;$$

  onde $E(M)$ é o conjunto de transições habilitadas em m, e $[w](t_k;M)$ é a taxa de disparo de t_k em M. A soma e a integral pode ser mudada para resultar em:

$$[f]_{t_k}(t) = \sum_{M:t_k \in E(M)} \int_0^t [w](t_k;M) \cdot [\pi]_M(z)dz.$$

- A *frequência intervalar de disparo de uma transição* (*throughput interval*=taxa intervalar de disparo), *i.e.*, o número médio de vezes que a transição t_k dispara numa unidade de tempo, pode ser calculada como a soma ponderada da taxa intervalar de disparo da transição:

$$[f]_{t_k}(t) = \sum_{M:t_k \in E(M)} [w](t_k;M) \cdot [\pi]_M.$$

4.3 Exemplos de modelos ISPN

O objetivo desta seção é de apresentar de uma forma clara todas as fases da análise ISPN. Para isto usou-se dois exemplos. Um é muito pequeno e pode ser seguido e ter cálculos realizados até sem uso de um computador. O segundo caso, contudo, é necessário usar um software com uma biblioteca da aritmética intervalar para que possa ser seguido e acompanhado por todos os seus cálculos. Os exemplos considerados já foram usados na Seção 2.4.1.1 com o objetivo de introduzir a análise GSPN com o uso da ferramenta TimeNET. O Exemplo 1 tem apenas duas marcações tangíveis e duas marcações voláteis e o Exemplo 2 tem dezesseis marcações tangíveis e doze marcações voláteis. As análises de desempenho são realizadas no *framework* MATLAB com a caixa de ferramentas INTLAB (MATLAB *toolbox* INTLAB). A análise de modelos ISPN considerando apenas intervalos degenerados (pontos) nos leva ao modelo GSPN clássico com resultados verificados.

4.3.1 Exemplo 1: Análise de ISPN de uma Única Máquina

O modelo descrito na Figura 4.1 (veja também a Figura 2.21) representa uma máquina susceptível a falha e um *buffer* de capacidade finita. A Tabela 4.1 apresenta as taxas intervalares

(degeneradas) de disparo das transições temporizadas por unidade de tempo, onde $[\nu]$ representa a taxa de produção intervalar, $[\lambda]$ representa a taxa de falha intervalar, e $[\mu]$ representa a taxa de reparação intervalar. Neste caso, temos um modelo equivalente ao modelo GSPN, porque só há parâmetros intervalares degenerados.

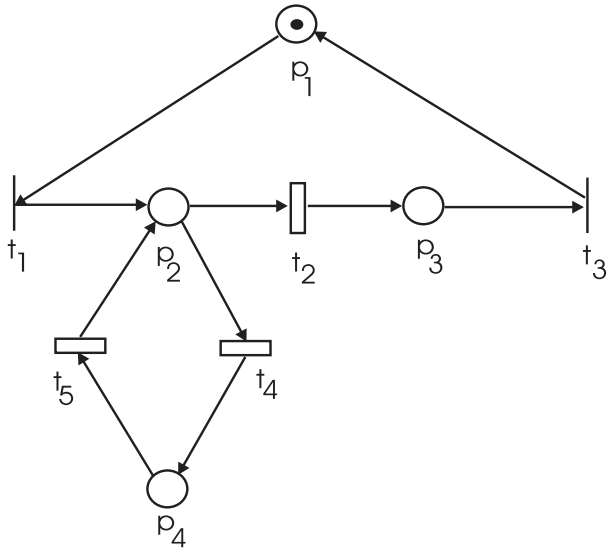


Figura 4.1 O módulo da máquina simples

Tabela 4.1 Taxas intervalares (intervalos degenerados) de disparo das transições para uma Única Máquina e um *buffer* de uma Linha de Montagem.

Transição	**Taxa** $([t]^{-1})$	**Símbolo**
$[t_2]$	[10; 10]	$[\nu]$
$[t_4]$	[3; 3]	$[\mu]$
$[t_5]$	[5; 5]	$[\lambda]$

Como produto da primeira etapa do processo de análise obtém-se o conjunto de alcançabilidade (na Tabela 4.2) e o grafo de alcançabilidade (na Figura 4.2). Finalmente, obtêm-se as matrizes $[\mathbf{P}]^{\mathscr{V}\mathscr{V}}$, $[\mathbf{P}]^{\mathscr{V}\mathscr{T}}$, $[\mathbf{U}]^{\mathscr{T}\mathscr{V}}$ e $[\mathbf{U}]^{\mathscr{T}\mathscr{T}}$:

Tabela 4.2 Conjunto de alcançabilidade e distribuição de marcas da ISPN da Figura 4.1.

Estado	Marcação $(m_1; m_2; m_3; m_4)$
1	$M_0 = (1; 0; 0; 0)$
2	$M_1 = (0; 1; 0; 0)$
3	$M_2 = (0; 0; 1; 0)$
4	$M_3 = (0; 0; 0; 1)$

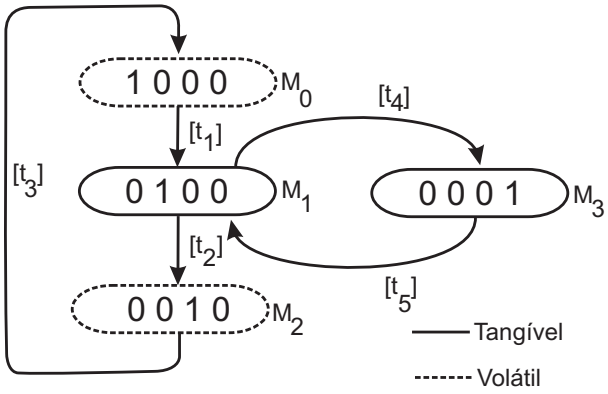


Figura 4.2 Grafo de Alcançabilidade e a cadeia de Markov intervalar embutida

$$[\mathbf{P}]^{\mathscr{V}\mathscr{V}} = \begin{pmatrix} [\,0;\ 0] & [\,0;\ 0] \\ [\,1;\ 1] & [\,0;\ 0] \end{pmatrix} \quad [\mathbf{P}]^{\mathscr{V}\mathscr{T}} = \begin{pmatrix} [\,1;\ 1] & [\,0;\ 0] \\ [\,0;\ 0] & [\,0;\ 0] \end{pmatrix}$$
$$[\mathbf{U}]^{\mathscr{T}\mathscr{T}} = \begin{pmatrix} [\,0;\ 0] & [\,5;\ 5] \\ [\,3;\ 3] & [\,0;\ 0] \end{pmatrix} \quad [\mathbf{U}]^{\mathscr{T}\mathscr{V}} = \begin{pmatrix} [\,0;\ 0] & [\,10;\ 10] \\ [\,0;\ 0] & [\,0;\ 0] \end{pmatrix}.$$

Posteriormente, eliminam-se as marcações voláteis (Equação 4.3) para obtenção da matriz intervalar de taxa $[U]$. A matriz de taxa representa a IREMC (*Interval Reduced Embedded Markov Chain* - cadeia de Markov embutida reduzida intervalar exibida na Figura 4.3):

$$[\mathbf{U}] = \begin{pmatrix} [\,10;\ 10] & [\,5;\ 5] \\ [\,3;\ 3] & [\,0;\ 0] \end{pmatrix}.$$

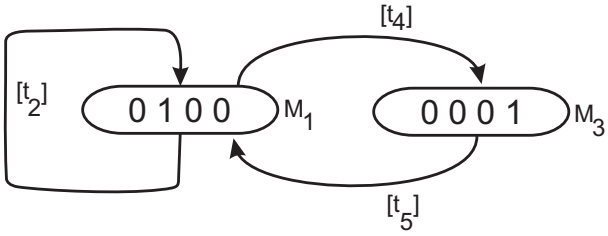


Figura 4.3 Cadeia de Markov embutida reduzida intervalar

Finalmente, através da Equação 4.4, obtém-se a matriz intervalar geradora infinitesimal:

$$[\mathbf{Q}] = \begin{pmatrix} [\,-5;\ -5] & [\,5;\ 5] \\ [\,3;\ 3] & [\,-3;\ -3] \end{pmatrix}.$$

A terceira etapa da análise ISPN resolve o sistema de equações lineares intervalares descrito pela Equação (4.5) (ver Capítulo 5). Os resultados são os intervalos verificados de cada probabilidade nos estados tangíveis:

$$[\pi]_{(1)} = [\,0,62499999999998;\ 6,25000000000001]$$

e

$$[\pi]_{(2)} = [\,0,37499999999999;\ 0,37500000000001].$$

Finalmente, pode-se efetuar a quarta e última etapa da análise ISPN, computação das métricas (ver Seção 4.2.4). A taxa de produção da máquina é

$$[P] = [\,6,24999999999998;\ 6,25000000000001]$$

(calculado pela fórmula $[P] = [\pi]_{(1)} \cdot [t_2]$). Estes resultados exibem a delimitação do valor exato obtido pela análise de GSPN convencional. Os resultados da análise ISPN nos dá resultado verificados, garantindo que o valor exato está dentro do mesmo. Pode-se, então, comparar este resultado intervalar com o valor $P = 6,25$ obtido com ferramenta TimeNET na Seção 2.4.1.1.

Considera-se agora uma situação em que os parâmetros são representados por intervalos, $[\mu] = \langle 3; 0,01 \rangle$e $[\lambda] = \langle 5; 0,01 \rangle$.

Como produto da primeira etapa da análise (subproduto do conjunto de alcançabilidade), obtêm-se as matrizes $[\mathbf{P}]^{\mathscr{V}\mathscr{V}}$, $[\mathbf{P}]^{\mathscr{V}\mathscr{T}}$, $[\mathbf{U}]^{\mathscr{T}\mathscr{V}}$ e $[\mathbf{U}]^{\mathscr{T}\mathscr{T}}$:

$$[\mathbf{P}]^{\mathscr{V}\mathscr{V}} = \begin{pmatrix} [\,0;\ 0] & [\,0;\ 0] \\ [\,1;\ 1] & [\,0;\ 0] \end{pmatrix} \qquad [\mathbf{P}]^{\mathscr{V}\mathscr{T}} = \begin{pmatrix} [\,1;\ 1] & [\,0;\ 0] \\ [\,0;\ 0] & [\,0;\ 0] \end{pmatrix}$$

$$[\mathbf{U}]^{\mathscr{T}\mathscr{T}} = \begin{pmatrix} [\,0;\ 0] & [\,4{,}99;\ 5{,}01] \\ [\,2{,}99;\ 3{,}01] & [\,0;\ 0] \end{pmatrix} \quad [\mathbf{U}]^{\mathscr{T}\mathscr{V}} = \begin{pmatrix} [\,0;\ 0] & [\,10;\ 10] \\ [\,0;\ 0] & [\,0;\ 0] \end{pmatrix}.$$

Após a eliminação das marcações voláteis (Equação 4.3)), obtém-se a matriz intervalar de taxa $[U]$:

$$[\mathbf{U}] = \begin{pmatrix} [\,10;\ 10] & [\,4{,}99;\ 5{,}01] \\ [\,2{,}99;\ 3{,}01] & [\,0;\ 0] \end{pmatrix}.$$

Finalmente, através da Equação 4.4, obtém-se a matriz intervalar geradora infinitesimal

$$[\mathbf{Q}] = \begin{pmatrix} [\,-5{;}01;\ -4{;}99] & [\,4{;}99;\ 5{;}01] \\ [\,2{;}99;\ 3{;}01] & [\,-3{;}01;\ -2{;}99] \end{pmatrix}.$$

Posteriormente, soluciona-se o sistema de equações lineares intervalares (Equação (4.5), os intervalos de cada probabilidade dos estados tangíveis:

$$[\pi]_{(1)} = [\,0,62374686716791;\ 0,62625313283209]$$

e

$$[\pi]_{(2)} = [\,0,37374686716791;\ 0,37625313283209].$$

Finalmente, calcula-se a taxa de produção da máquina:

$$[P] = [\,6,23123120300751;\ 6,26879385964913]$$

(calculado pela fórmula $[P] = [\pi]_{(1)} \cdot [\nu]$).

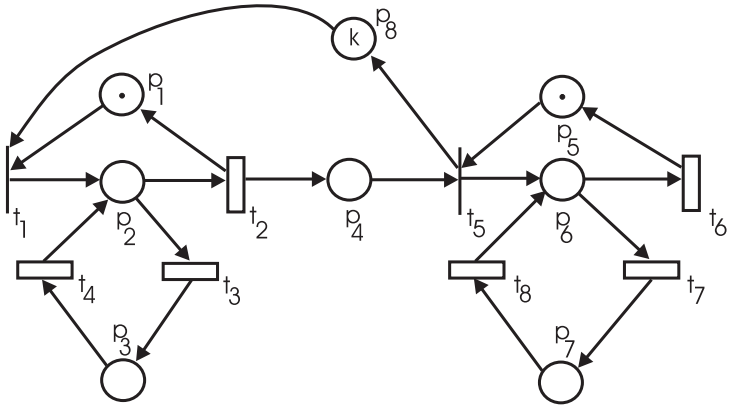


Figura 4.4 Modelo ISPN de Duas máquinas e um *buffer* em uma Linha de Montagem ($k = 3$)

4.3.2 Exemplo 2: Análise do modelo ISPN de Duas máquinas e um *buffer* em uma Linha de Montagem

Considere o modelo ISPN de Duas máquinas e um *buffer* de uma Linha Montagem da Figura 4.4. A Tabela 4.3 mostra as taxas intervalares (intervalos degenerados) de disparo das transições temporizadas, onde $[\nu]$ representa a taxa de produção, $[\lambda]$ representa a taxa de falha, e $[\mu]$ representa a taxa de reparo.

Tabela 4.3 Taxas intervalares (intervalos degenerados) de disparo das transições do modelo ISPN de Duas máquinas e um *buffer* em uma Linha de Montagem.

Transição	**Taxa** $([t]^{-1})$	**Símbolo**
$[t_2]$	$[1;\ 1]$	$[\nu_1]$
$[t_3]$	$[3;\ 3]$	$[\lambda_1]$
$[t_4]$	$[5;\ 5]$	$[\mu_1]$
$[t_6]$	$[2;\ 2]$	$[\nu_2]$
$[t_7]$	$[4;\ 4]$	$[\lambda_2]$
$[t_8]$	$[6;\ 6]$	$[\mu_2]$

O produto da primeira etapa da análise obtém-se o conjunto de alcançabilidade (Tabela 4.4) e o grafo de alcançabilidade (Tabela 4.5). As marcações que habilitam as transições t_1 e t_5 são voláteis, pois as transições habilitadas são imediatas. As transições imediatas t_1 e t_5 (observe o t minúsculo para representar literalmente transições imediatas). Identificam-se as doze marcações voláteis $M_0, M_2, M_4, M_5, M_7, M_{12}, M_{13}, M_{17}, M_{19}, M_{22}, M_{24}, M_{26}$ (disparo das transições imediatas t_1 e t_5) e as demais são as marcações tangíveis. Finalmente, obtêm-se as matrizes $[\mathbf{P}]^{\mathscr{V}\mathscr{V}}$, $[\mathbf{P}]^{\mathscr{V}\mathscr{T}}$, $[\mathbf{U}]^{\mathscr{T}\mathscr{V}}$ e $[\mathbf{U}]^{\mathscr{T}\mathscr{T}}$:

Tabela 4.4 Conjunto de alcançabilidade e distribuição de marcas da ISPN da Figura 4.4.

Estado	Marcação[1]	Estado	Marcação[1]
1	$M_0 = [1;0;0;0;1;0;0;3]$	15	$M_{14} = [0;0;1;1;0;1;0;1]$
2	$M_1 = [0;1;0;0;1;0;0;2]$	16	$M_{15} = [0;1;0;1;0;0;1;1]$
3	$M_2 = [1;0;0;1;1;0;0;2]$	17	$M_{16} = [0;1;0;2;0;1;0;0]$
4	$M_3 = [0;0;1;0;1;0;0;2]$	18	$M_{17} = [0;0;1;1;1;0;0;1]$
5	$M_4 = [0;1;0;1;1;0;0;1]$	19	$M_{18} = [0;0;1;1;0;0;1;1]$
6	$M_5 = [1;0;0;0;0;1;0;3]$	20	$M_{19} = [1;0;0;2;0;0;1;1]$
7	$M_6 = [0;1;0;0;0;1;0;2]$	21	$M_{20} = [1;0;0;3;0;1;0;0]$
8	$M_7 = [1;0;0;1;0;1;0;2]$	22	$M_{21} = [0;0;1;2;0;1;0;0]$
9	$M_8 = [0;0;1;0;0;1;0;2]$	23	$M_{22} = [0;1;0;2;1;0;0;0]$
10	$M_9 = [0;1;0;0;0;0;1;2]$	24	$M_{23} = [0;1;0;2;0;0;1;0]$
11	$M_{10} = [0;1;0;1;0;1;0;1]$	25	$M_{24} = [1;0;0;3;1;0;0;0]$
12	$M_{11} = [0;0;1;0;0;0;1;2]$	26	$M_{25} = [1;0;0;3;0;0;1;0]$
13	$M_{12} = [1;0;0;1;0;0;1;2]$	27	$M_{26} = [0;0;1;2;1;0;0;0]$
14	$M_{13} = [1;0;0;2;0;1;0;1]$	28	$M_{27} = [0;0;1;2;0;0;1;0]$

1- $Marcação = [\, m_1;m_2;m_3;m_4;m_5;m_6;m_7;m_8 \,]$

Tabela 4.5 Descrição literal do grafo de alcançabilidade da ISPN da Figura 4.4.

Marcação | Disparo da transição | Nova marcação				
$M_0 \,\vert t_1 \rangle\, M_1$	$M_1 \,\vert T_2 \rangle\, M_2$	$M_1 \,\vert T_3 \rangle\, M_3$	$M_2 \,\vert t_1 \rangle\, M_4$	$M_2 \,\vert t_5 \rangle\, M_5$
$M_3 \,\vert T_4 \rangle\, M_1$	$M_4 \,\vert t_5 \rangle\, M_6$	$M_5 \,\vert t_1 \rangle\, M_6$	$M_6 \,\vert T_2 \rangle\, M_7$	$M_6 \,\vert T_3 \rangle\, M_8$
$M_6 \,\vert T_6 \rangle\, M_1$	$M_6 \,\vert T_7 \rangle\, M_9$	$M_7 \,\vert t_1 \rangle\, M_{10}$	$M_8 \,\vert T_4 \rangle\, M_6$	$M_8 \,\vert T_6 \rangle\, M_3$
$M_8 \,\vert T_7 \rangle\, M_{11}$	$M_9 \,\vert T_2 \rangle\, M_{12}$	$M_9 \,\vert T_3 \rangle\, M_{11}$	$M_9 \,\vert T_8 \rangle\, M_6$	$M_{10} \,\vert T_2 \rangle\, M_{13}$
$M_{10} \,\vert T_3 \rangle\, M_{14}$	$M_{10} \,\vert T_6 \rangle\, M_4$	$M_{10} \,\vert T_7 \rangle\, M_{15}$	$M_{11} \,\vert T_4 \rangle\, M_9$	$M_{11} \,\vert T_8 \rangle\, M_8$
$M_{12} \,\vert t_1 \rangle\, M_{15}$	$M_{13} \,\vert t_1 \rangle\, M_{16}$	$M_{14} \,\vert T_4 \rangle\, M_{10}$	$M_{14} \,\vert T_6 \rangle\, M_{17}$	$M_{14} \,\vert T_7 \rangle\, M_{18}$
$M_{15} \,\vert T_2 \rangle\, M_{19}$	$M_{15} \,\vert T_3 \rangle\, M_{18}$	$M_{15} \,\vert T_8 \rangle\, M_{10}$	$M_{16} \,\vert T_2 \rangle\, M_{20}$	$M_{16} \,\vert T_3 \rangle\, M_{21}$
$M_{16} \,\vert T_6 \rangle\, M_{22}$	$M_{16} \,\vert T_7 \rangle\, M_{23}$	$M_{17} \,\vert t_5 \rangle\, M_8$	$M_{18} \,\vert T_4 \rangle\, M_{15}$	$M_{18} \,\vert T_8 \rangle\, M_{14}$
$M_{19} \,\vert t_1 \rangle\, M_{23}$	$M_{20} \,\vert T_6 \rangle\, M_{24}$	$M_{20} \,\vert T_7 \rangle\, M_{25}$	$M_{21} \,\vert T_4 \rangle\, M_{16}$	$M_{21} \,\vert T_6 \rangle\, M_{26}$
$M_{21} \,\vert T_7 \rangle\, M_{27}$	$M_{22} \,\vert t_5 \rangle\, M_{10}$	$M_{23} \,\vert T_2 \rangle\, M_{25}$	$M_{23} \,\vert T_3 \rangle\, M_{27}$	$M_{23} \,\vert T_8 \rangle\, M_{16}$
$M_{24} \,\vert t_5 \rangle\, M_{13}$	$M_{25} \,\vert T_8 \rangle\, M_{20}$	$M_{26} \,\vert t_5 \rangle\, M_{14}$	$M_{27} \,\vert T_4 \rangle\, M_{23}$	$M_{27} \,\vert T_8 \rangle\, M_{21}$

$$[\mathbf{P}]^{\mathscr{V}\mathscr{V}} = \begin{pmatrix}
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0{,}5;\ 0{,}5] & [0{,}5;\ 0{,}5] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0]
\end{pmatrix}$$

$$
[\mathbf{P}]^{\mathscr{V}\mathscr{T}} = \left(\begin{array}{cccccccccccccc}
[1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [1;\ 1]
\end{array}\right.
$$

$$
\left.\begin{array}{cccc}
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0]
\end{array}\right)
$$

$$
[\mathbf{U}]^{\mathscr{T}\mathscr{T}} = \left(\begin{array}{cccccccccccccc}
[0;\ 0] & [3;\ 3] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[5;\ 5] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[2;\ 2] & [0;\ 0] & [0;\ 0] & [3;\ 3] & [4;\ 4] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [2;\ 2] & [5;\ 5] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [4;\ 4] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [6;\ 6] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [3;\ 3] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [3;\ 3] & [4;\ 4] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [6;\ 6] & [5;\ 5] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [5;\ 5] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [4;\ 4] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [6;\ 6] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [3;\ 3] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [1;\ 1] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [6;\ 6] & [5;\ 5] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [5;\ 5] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [6;\ 6] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [6;\ 6] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0]
\end{array}\right.
$$

$$
\left.\begin{array}{cccc}
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[3;\ 3] & [4;\ 4] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [4;\ 4] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [4;\ 4] \\
[0;\ 0] & [0;\ 0] & [1;\ 1] & [3;\ 3] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[6;\ 6] & [5;\ 5] & [0;\ 0] & [0;\ 0]
\end{array}\right)
$$

$$
[\mathbf{U}]^{\mathscr{TV}} = \begin{pmatrix}
[0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [2;\ 2] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [2;\ 2] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [2;\ 2] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [2;\ 2] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [2;\ 2] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0]
\end{pmatrix}
$$

Em seguida, eliminam-se das marcações voláteis (Equação 4.3) para se obter a matriz intervalar de taxas $[U]$ que representa a IREMC:

$$
[\mathbf{U}] = \begin{pmatrix}
[0;\ 0] & [3;\ 3] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[5;\ 5] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[2;\ 2] & [0;\ 0] & [0;\ 0] & [3;\ 3] & [4;\ 4] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [2;\ 2] & [5;\ 5] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [4;\ 4] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [6;\ 6] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [3;\ 3] & [0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [2;\ 2] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [3;\ 3] & [4;\ 4] & [1;\ 1] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [6;\ 6] & [5;\ 5] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [2;\ 2] & [0;\ 0] & [5;\ 5] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [4;\ 4] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [6;\ 6] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [3;\ 3] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [2;\ 2] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [1;\ 1] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [6;\ 6] & [5;\ 5] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [2;\ 2] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [2;\ 2] & [0;\ 0] & [5;\ 5] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [6;\ 6] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [6;\ 6] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0]
\end{pmatrix}
$$

$$
\begin{matrix}
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [1;\ 1] & [0;\ 0] & [0;\ 0] \\
[3;\ 3] & [4;\ 4] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [4;\ 4] & [0;\ 0] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [4;\ 4] \\
[0;\ 0] & [0;\ 0] & [1;\ 1] & [3;\ 3] \\
[0;\ 0] & [0;\ 0] & [0;\ 0] & [0;\ 0] \\
[6;\ 6] & [5;\ 5] & [0;\ 0] & [0;\ 0]
\end{matrix}
\Bigg) .
$$

Finalmente, através da Equação 4.4, obtém-se a matriz intervalar geradora infinitesimal:

$$
[\mathbf{Q}] = \left(\begin{array}{cccccccccc}
[-4;4] & [3;3] & [1;1] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] \\
[5;5] & [-5;-5] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] \\
[2;2] & [0;0] & [-10;10] & [3;3] & [4;4] & [1;1] & [0;0] & [0;0] & [0;0] & [0;0] \\
[0;0] & [2;2] & [5;5] & [-11;-11] & [0;0] & [0;0] & [4;4] & [0;0] & [0;0] & [0;0] \\
[0;0] & [0;0] & [6;6] & [0;0] & [-10;10] & [0;0] & [3;3] & [0;0] & [1;1] & [0;0] \\
[0;0] & [0;0] & [2;2] & [0;0] & [0;0] & [-10;-10] & [0;0] & [3;3] & [4;4] & [1;1] \\
[0;0] & [0;0] & [0;0] & [6;6] & [5;5] & [0;0] & [-11;-11] & [0;0] & [0;0] & [0;0] \\
[0;0] & [0;0] & [0;0] & [2;2] & [0;0] & [5;5] & [0;0] & [-11;-11] & [0;0] & [0;0] \\
[0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [6;6] & [0;0] & [0;0] & [-10;-10] & [0;0] \\
[0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [2;2] & [0;0] & [0;0] & [0;0] & [-10;10] \\
[0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [6;6] & [5;5] & [0;0] \\
[0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [2;2] \\
[0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [2;2] & [0;0] & [5;5] \\
[0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [6;6] \\
[0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] \\
[0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0]
\end{array}\right.
$$

$$
\left.\begin{array}{cccccc}
[0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] \\
[0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] \\
[0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] \\
[0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] \\
[0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] \\
[0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] \\
[0;0] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] \\
[4;4] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] \\
[3;3] & [0;0] & [0;0] & [1;1] & [0;0] & [0;0] \\
[0;0] & [1;1] & [3;3] & [4;4] & [0;0] & [0;0] \\
[-11;-11] & [0;0] & [0;0] & [0;0] & [0;0] & [0;0] \\
[0;0] & [-6;-6] & [0;0] & [0;0] & [4;4] & [0;0] \\
[0;0] & [0;0] & [-11;-11] & [0;0] & [0;0] & [4;4] \\
[0;0] & [0;0] & [0;0] & [-10;-10] & [1;1] & [3;3] \\
[0;0] & [6;6] & [0;0] & [0;0] & [-6;-6] & [0;0] \\
[6;6] & [5;5] & [0;0] & [-11;-11] & [0;0] & [0;0]
\end{array}\right)
$$

A terceira etapa da análise ISPN resolve o sistema de equações lineares intervalares (Equação 4.5), e produz os limites intervalares verificados das probabilidades estacionárias dos estados tangíveis:

$$
[\vec{\pi}]^t = \left(\begin{array}{l}
[\,0{,}30162341059172;\ 0{,}30162341059173] \\
[\,0{,}20129241213850;\ 0{,}20129241213851] \\
[\,0{,}10001579083719;\ 0{,}10001579083720] \\
[\,0{,}05079591445866;\ 0{,}05079591445867] \\
[\,0{,}05701697985273;\ 0{,}05701697985274] \\
[\,0{,}05122652318522;\ 0{,}05122652318523] \\
[\,0{,}03402132703571;\ 0{,}03402132703572] \\
[\,0{,}02728986215974;\ 0{,}02728986215975] \\
[\,0{,}03607316886026;\ 0{,}03607316886027] \\
[\,0{,}02968055852738;\ 0{,}02968055852739] \\
[\,0{,}01976172320179;\ 0{,}01976172320180] \\
[\,0{,}02527256969829;\ 0{,}02527256969830] \\
[\,0{,}01396928749535;\ 0{,}01396928749536] \\
[\,0{,}02086458086920;\ 0{,}02086458086922] \\
[\,0{,}02032580994373;\ 0{,}02032580994374] \\
[\,0{,}01077008114445;\ 0{,}01077008114446]
\end{array}\right).
$$

Finalmente, calcula-se métrica de interesse, ou seja, a utilização média das máquinas, i.e., a probabilidade que uma máquina esteja processando um item através das equações:

$$[UM_1] = [prob](m(p_2) = 1) \;\; e \;\; [UM_2] = [prob](m(p_6) = 1).$$

O resultado da avaliação provê os seguintes valores:

$$[UM_1] = [0,59650101272372; 0,59650101272374] \;\; e$$

$$[UM_2] = [0,29825050636186; 0,29825050636187].$$

Estes resultados exibem a delimitação do valor exato obtido pela análise de GSPN convencional.

A Tabela 4.6 mostra a utilização intervalar média das máquinas, $[UM_1]$ e $[UM_2]$ para três intervalos de taxas $[\mu_1]$. Os resultados da análise ISPN, fornecidos pela ferramenta protótipo ISPN, são equivalentes aos resultados da GSPN convencional com limites intervalares verificados [52].

Tabela 4.6 Experimento $[TP]$ (Taxa de Produção) $= [\mu_1]$ para o modelo ISPN de Duas máquinas e um *buffer* de uma Linha de Montagem. Resultados obtidos com a ferramenta protótipo ISPN da utilização intervalar média das máquinas.

Taxa $[\mu_1]$	**Utilização intervalar média das máquinas**	
	$[UM_1]$	$[UM_2]$
[0,1E2; 0,1E2]	$[0,11946700722573;$ $0,11946700722574]$	$[0,59733503612865;$ $0,59733503612866]$
[0,1E1; 0,1E1]	$[0,59650101272372;$ $0,59650101272374]$	$[0,29825050636186;$ $0,29825050636187]$
[0,1E0; 0,1E0]	$[0,62499328409659;$ $0,62499328409660]$	$[0,03124966420482;$ $0,03124966420483]$
[0,1E-2; 0,1E-2]	$[0,62499999999992;$ $0,62499999999993]$	$[3,124999999998600e-004;$ $3,124999999998605e-004]$

A seguir, consideram-se variações nas taxas das transições exponenciais. Para se evitar redundância, não será apresentado o detalhamento do processo como nos exemplos anteriores. A Tabela 4.7 mostra a utilização intervalar média das máquinas, $[UM_1]$ e $[UM_2]$ para três intervalos de taxas $[\mu_1]$. As variações de todas as taxas exponenciais são de ± 1 como erro no 3^0 dígito significativo. Por exemplo, se $[\mu_1] = [0,099E1; 0,101E1]$, a utilização intervalar média da máquina 1 pode se situa entre 49.61 % e 69,69 %. Para $[\mu_1] = [0,099E2; 0,101E2]$, a utilização intervalar média da máquina 1 apresenta-se entre 11,39 % e 12,50 %.

Tabela 4.7 Utilização intervalar média das máquinas (intervalos resultantes exibidos como ponto médio/raio) para o Modelo Duas máquinas e um *buffer* numa Linha Montagem para três intervalos associados a $[\mu_1]$.

Taxa $[\mu_1]$	**Utilização intervalar média das máquinas**	
	$[UM_1]$	$[UM_2]$
[0,099E2; 0,101E2]	$\langle 0,1195;0,0056\rangle$	$\langle 0,5973;0,0178\rangle$
[0,099E1; 0,101E1]	$\langle 0,5965;0,1004\rangle$	$\langle 0,2983;0,0701\rangle$
[0,199E0; 0,201E0]	$\langle 0,6249;0,0805\rangle$	$\langle 0,0625;0,0391\rangle$

4.4 Considerações Finais

Os modelos ISPN tratam das incertezas do sistema considerando intervalos para representar o tempo e também os pesos atribuídos às transições do modelo. O modelo proposto e os métodos respectivos, adaptados considerando a análise intervalar, permitem a influência de variabilidades simultâneas dos parâmetros na computação das métricas serem consideradas, fornecendo desse modo limites intervalares rigorosos das métricas. Seria necessário uma quantidade muito grande de experimentos para se obter os efeitos da variação de todos os parâmetros, assim, na bordagem clássica, ficaria intratável do ponto de vista computacional. É também importante destacar que mesmo quando somente intervalos degenerados (pontos) estão presentes, será possível empregar o modelo proposto, desde que os erros de arredondamento e de truncamento sejam tratados na aritmética intervalar, de modo que fica garantido que os resultados das métricas obtidas pertencem aos intervalos intervalares computados.

Este capítulo apresentou a conceituação teórica e análise que foram desenvolvidas para este uso específico. Vários aspectos teóricos e da análise intervalar relacionados são apresentados em outras partes deste documento (Capítulos 2 e 5 e os Apêndices A,C,D e E). O Capítulo 5 apresenta e discute vários algoritmos que foram usados para solução do estado estacionário através do cálculo do vetor *p.m.f.* intervalar dos exemplos dados no Capítulo atual e nos estudos de casos apresentados no Capítulo 6.

CAPÍTULO 5

Algoritmos para Análise ISPN do Estado Estacionário

A análise do estado estacionário de um modelo ISPN requer a solução de um sistema de equações lineares intervalares com tantas equações quanto for o número de marcações tangíveis. Tão logo a matriz geradora intervalar $[\mathbf{Q}]$ da ICTMC seja produzida, o sistema linear intervalar respectivo será solucionado para produzir o vetor da *p.m.f.* intervalar do estado estacionário.

Aqui será discutido o conjunto solução unida $\sum_{\exists\exists}([\mathbf{A}];[\mathbf{b}])$ e como toda matriz degenerada pertencente ao conjunto solução é não-singular, sendo então, o conjunto solução limitado [22]. O principal, quando se trabalha com sistemas lineares intervalares limitados, é encontrar uma solução do intervalo envoltório de $\square\sum_{\exists\exists}([\mathbf{A}];[\mathbf{b}])$. O símbolo $\square\sum_{\exists\exists}([\mathbf{A}];[\mathbf{b}])$ denota o intervalo envoltório do conjunto $\sum_{\exists\exists}([\mathbf{A}];[\mathbf{b}])$.

Os métodos para inclusão da solução, obtidos em tempo polinomial, de um sistema linear intervalar $[\mathbf{A}]\cdot[\mathbf{x}]=[\mathbf{b}]$, são classificados em diretos e iterativos. No caso dos métodos iterativos usam-se critérios de parada. Normalmente, as iterações terminam quando os resultados produzidos por iterações sucessivas são suficientemente próximos.

5.1 Eliminação gaussiana intervalar

O algoritmo da eliminação gaussiana intervalar (IGE - *Interval Gaussian Elimination*) trata dos coeficientes intervalares. Um sistema triangular intervalar pode ser formado usando-se a aritmética intervalar. O cuidado usual tem que ser tomado com divisão por intervalos que inclui o zero. Pela propriedade da inclusão, Teorema 2.1.1, a solução deste sistema triangular intervalar dá uma inclusão do conjunto da solução USS. Uma implementação do algoritmo, escrita no MATLAB toolbox INTLAB, com pivotação usando mignitude, é dada pela função **INTGAUSS.m** em [52].

A mignitude de $[x]$ é definida como

$$mig([x]) = \begin{cases} min\left(|\underline{x}|\,;\ |\overline{x}|\right) & se\ 0 \notin [x] \\ 0 & caso\ contrário \end{cases}$$

A pivotação por coluna usando mignitude pode ser adotada para escolher um pivô com a maior mignitude. Apesar da eliminação gaussiana intervalar não ser eficaz, em geral ela é apropriada para determinadas classes de matrizes. Em particular, limites realísticos para o conjunto solução USS são obtidos para matrizes $[\mathbf{M}]$, matrizes $[\mathbf{H}]$, matrizes diagonalmente dominantes, matrizes tridiagonais e matrizes 2×2 [52]. No caso onde $[\mathbf{A}]$ são matrizes $[\mathbf{M}]$ o *hull* exato é

obtido para muitos **[b]**: Neumaier [119] mostra isso para **[b]** ≥ 0, **[b]** ≤ 0 ou $0 \in$ **[b]**.

Uma matriz $[\mathbf{M}]$ é uma matriz quadrada $[\mathbf{A}] \in I(\mathbb{R})^{n\times n}$, tal que $[A]_{ik} \leq 0$ para todo $i \neq k$, $[\mathbf{A}] \cdot \mathbf{u} > 0$, para algum vetor positivo $\mathbf{u} \in \mathbb{R}^n$.

Por exemplo, $[\mathbf{A}] := \begin{pmatrix} 2 & [-1,\ 0] \\ [-1,\ 0] & 2 \end{pmatrix}$,

tome $\mathbf{u} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

As matrizes $[\mathbf{H}]$ são obtidas pelo abondono da condição restitiva do sinal na definição das matrizes $[\mathbf{M}]$. Uma propriedade importante é que a inversa de uma matriz $[\mathbf{A}]$ matriz $[\mathbf{H}]$ pode ser limitada pela inversa da matriz relacionada, a chamada matriz comparação <[**A**]> de Ostrowski (seção 5.3).

Se $[\mathbf{A}]$ não é uma matriz apropriada (gerando superestimação excessiva ou divisão por zero no processo de eliminação) para a solução através da IGE, recomenda-se combinar a eliminação gaussiana intervalar com precondicionamento. O precondicionamento de um sistema é realizado pela multiplicação da matriz dos coeficientes $[\mathbf{A}]$ por uma matriz precondicionante $\mathbf{C} \in \mathbb{R}^{n\times n}$. Com o precondicionamento espera-se transformar o sistema original em um novo sistema, com uma matriz de coeficientes mais tratável [119]. Destaca-se que a escolha de $\mathbf{C}$ como a inversa da matriz do ponto médio de $[\mathbf{A}]$, que conduzem frequentemente a matriz $\mathbf{C}[\mathbf{A}]$ a ser uma matriz $[\mathbf{H}]$, é particularmente eficaz quando se trata dos métodos iterativos, e este é o caso para o método da IGE. Se $[\mathbf{A}]$ é fortemente regular, então a matriz do sistema precondicionado $\mathbf{C} \cdot [\mathbf{A}] \cdot [\mathbf{x}] = \mathbf{C} \cdot [\mathbf{b}]$ é uma matriz $\mathbf{H}$ cuja matriz central é a identidade. Pode-se mostrar que para tal matriz o IGE dá cercos para o *hull* exato, os quais são pelo menos tão bons quanto os cercos obtidos de um método de Gauss-Seidel intervalar (IGS - *Interval Gauss-Seidel*), isto é, o procedimento de precondicionamento da IGE passa a ser ainda mais eficaz do que o método iterativo IGS [119, 22].

A eliminação gaussiana intervalar é a versão da aritmética intervalar do bem conhecido algoritmo gaussiano para solução de sistemas lineares de equações. Começando com uma matriz intervalar $[\mathbf{A}]^{n\times n}$ e um vetor intervalar $[\mathbf{b}]^n$, ele produz um vetor intervalar $[\mathbf{x}]^G = IGE([\mathbf{A}], [\mathbf{b}])$ o qual contém o conjunto solução $\sum([\mathbf{A}]; [\mathbf{b}])$. O algoritmo IGE [22] é apresentado no topo da próxima página.

No **PASSO 1** é a entrada de dados da matriz dos coeficientes $[\mathbf{A}]$ e do vetor de termos independentes $[\mathbf{b}]$, no **PASSO 2** é realizado o processo de eliminação, que transforma o sistema original em um sistema triangular superior que inclui o conjunto solução do sistema original. No **PASSO 3** o sistema triangular superior é resolvido pelo processo de substituição retroativa. A complexidade do algoritmo de IGE é proporcional a n^3 operações ponto-flutuante intervalar.

PASSO 1

$[\mathbf{A}]^0 = [\mathbf{A}], [\mathbf{b}]^0 = [\mathbf{b}]$

PASSO 2

$$
\begin{aligned}
&\textit{para } k = 1,\ n-1 \\
&\quad \textit{para } i = k+1,\ n \\
&\qquad [m]_{ik} = \frac{[A]_{ik}^{k-1}}{[A]_{kk}^{k-1}} \\
&\qquad \textit{para } j = k+1,\ n \\
&\qquad\quad [A]_{ij}^{k} = [A]_{ij}^{k-1} - [m]_{ik} \cdot [A]_{kj}^{k-1} \\
&\qquad \textit{fim} \\
&\qquad [b]_{i}^{k} = [b]_{i}^{k-1} - [m]_{ik} \cdot [b]_{k}^{k-1} \\
&\quad \textit{fim} \\
&\textit{fim}
\end{aligned}
$$

PASSO 3

$$
\begin{aligned}
&[x]_n = \frac{[b]_n^{n-1}}{[A]_{nn}^{n-1}} \\
&\textit{para } i = n-1,\ 1,\ -1 \\
&\quad [x]_i = \left([b]_i^{n-1} - \sum_{j=i+1}^{n} [A]_{ij}^{n-1} \cdot [x]_j \right) / [A]_{ii}^{n-1} \\
&\textit{fim}
\end{aligned}
$$

5.2 Algoritmo de Grassmann intervalar

O algoritmo de Grassmann constitui uma variante numericamente estável do procedimento de eliminação gaussiana [10]. O algoritmo evita completamente subtrações, e é, consequentemente, muito menos sensível aos erros de arredondamento e de cancelamento. O algoritmo de Grassmann foi introduzido originalmente para a análise da cadeia de Markov ergódiga de tempo discreto. Aqui foi considerada a variante apresentada por Kumar *et al*, que permite uma interpretação direta nos termos das cadeias de Markov de tempo contínuo [78].

De modo usual, a eliminação é seguida por um processo de substituição para expressar as relações entre probabilidades do estado estacionário. Dado que a condição de normalização não é considerada no começo, ela deve ser aplicada para fornecer o vetor final da probabilidade intervalar do estado estacionário. A entrada de dados do algoritmo de Grassmann, tem sido apresentado por meio de uma matriz geradora da CTMC, e, consequentemente, na versão intervalar apresentada a seguir, a matriz dos coeficientes $[\mathbf{A}]$ deve ser definida como: $[\mathbf{Q}]$ para uma ICTMC e $[[\mathbf{P}] - \mathbf{I}]$ para uma IDTMC. O algoritmo de Grassmann é descrito abaixo.

PASSO 1

$$[\mathbf{A}] = \begin{cases} [\mathbf{Q}], & \textit{para uma ICTMC} \\ [[\mathbf{P}] - \mathbf{I}], & \textit{para uma IDTMC} \end{cases}$$

PASSO 2

$$\textit{para } l = n-1,\ n-2,\ \cdots, 1 \ \textit{faça}$$

$$[A]_{ji}^{n-l} = \begin{cases} \dfrac{[A]_{ji}^{n-l-1}}{\sum_{m=0}^{l-1} [A]_{lm}^{n-l-1}} & j < l, i = l, \\ [A]_{ji}^{n-l-1} + \dfrac{[A]_{jl}^{n-l-1} \cdot [A]_{li}^{n-l-1}}{\sum_{m=0}^{l-1} [A]_{lm}^{n-l-1}} & j \neq i, 1 \leq j, i \leq l-1, \\ -1, & j = i = l, \\ 0, & j = l, i < l. \end{cases}$$

Obs.: A colocação dos elementos da diagonal igual a -1 e abaixo da diagonal igual a 0 é somente por padronização. Essas atribuições podem ser omitidas numa implementação real por uma questão de eficiência.

PASSO 3

$$\textit{para } l = 1,\ 2,\ \cdots, n-1 \ \textit{faça}$$

$$[x]_l = \sum_{i=0}^{l-1} [x]_i \cdot [A]_{il}^{n-l}$$

PASSO 4

$$\textit{para } i = 0,\ 1,\ \cdots, n-1 \ \textit{faça}$$

$$\left.\begin{matrix} [\pi]_i \\ [\nu]_i \end{matrix}\right\} = \frac{[x]_i}{\sum_{j=0}^{n-1} [x]_j}$$

No **PASSO 1** é a entrada de dados da matriz dos coeficientes $[\mathbf{A}]$, no **PASSO 2** é realizado o processo de eliminação, que transforma o sistema original em um sistema triangular superior que inclui o conjunto solução do sistema original. Este processo aproveita as estrutura da ICTMC de modo que elimina as subtrações no processo de triangulação. No **PASSO 3** o sistema triangular superior é resolvido pelo processo de substituição retroativa. E, finalmente, no **PASSO 4** é feita a normalização para obtenção da solução. A complexidade do algoritmo de IGE é proporcional a n^3 operações ponto-flutuante intervalar. A complexidade computacional do algoritmo de Grassmann é, similar à do algoritmo de eliminação gaussiana, $(n^3/3)$ multiplicações ou divisões em ponto-flutuante intervalar, e o requisito de armazenamento é $O(n^2)$ números em ponto-flutuante intervalar.

Uma implementação do algoritmo de Grassmann intervalar é dada pela função **INTGRASSMANN.m** (escrita com MATLAB *toolbox* INTLAB). Ela é uma adaptação direta do algoritmo encontrado em [10]. Ela será usado na Seção 6.3.1.

5.3 O Método de Hansen-Bliek-Rohn-Ning-Kearfott-Neumaier

O método de Hansen-Bliek-Rohn-Ning-Kearfott-Neumaier foi introduzido por Hansen [56]. Este método fornece um limite para o conjunto da solução hull relacionado às equações lineares intervalares para o caso em que a matriz do ponto médio de $[\mathbf{A}]$ é igual à identidade. Bliek [6] obteve também este resultado, mas Rhon [139] forneceu uma prova rigorosa. Ning & Kearfott [121] generalizaram o resultado para o caso onde $[\mathbf{A}]$ é uma matriz $[\mathbf{H}]$. Finalmente, Neumaier [118] apresentou uma prova simplificada. Este método é baseado no seguinte teorema que usa a matriz comparação $\langle[\mathbf{A}]\rangle$, definida por

$$\langle[\mathbf{A}]\rangle_{ii} = min\{|\alpha| : \alpha \in [\mathbf{A}]_{ii}\},$$

$$\langle[\mathbf{A}]\rangle_{ik} = max\{|\alpha| : \alpha \in [\mathbf{A}]_{ik}\}.$$

Theorem 5.3.1. *Considere $[\boldsymbol{A}] \in I(\mathbb{R}^{n\times n})$ ser uma matriz $[\boldsymbol{H}]$, $[\boldsymbol{b}] \in I(\mathbb{R}^n)$ ser o vetor de termos independentes,*

$$\boldsymbol{u} = \langle[\boldsymbol{A}]\rangle^{-1} \cdot |[\boldsymbol{b}]|, \;\; d_i = \left(\langle[\boldsymbol{A}]\rangle^{-1}\right)_{ii}$$

e

$$\alpha_i = \langle[\boldsymbol{A}]\rangle_{ii} - 1/d_i, \quad \beta_i = u_i/d_i - |[\boldsymbol{b}]|_i$$

Então, $\square\Sigma_{\exists\exists}([\boldsymbol{A}];[\boldsymbol{b}]) \subset [\boldsymbol{x}]$ está contido num vetor $[\boldsymbol{x}]$ com componentes

$$[\boldsymbol{x}]_i = \frac{[\boldsymbol{b}]_i + [-\beta_i;\beta_i]}{[\boldsymbol{A}]_{ii} + [-\alpha_i;\alpha_i]}$$

Uma prova simplificada é feita por Neumaier [118].

Para estimar 0 cerco (cálculos realizados com a aritmética intervalar com arredondamento dirigido) do envoltório intervalar usando a aritmética do ponto flutuante, os limites superiores rigorosos para α_i e β_i são requeridos. Estes são obtidos se um limite rigoroso $\mathbf{B}$ para $\langle[\mathbf{A}]\rangle^{-1}$ é utilizado. A explicação seguinte de como isto é alcançado está baseada no que é dado em [118].

Uma propriedade de uma matriz $[\mathbf{A}]$ matriz $[\mathbf{H}]$ é que $\langle[\mathbf{A}]\rangle^{-1}$ é não negativa. Isto sugere que o limite superior $\mathbf{B}$ para $\langle[\mathbf{A}]\rangle^{-1}$ pode ser expresso em termos de $\tilde{\mathbf{B}}$, uma aproximação para que $\langle[\mathbf{A}]\rangle^{-1}$, e os vetores $\mathbf{v}$ e $\mathbf{w} \in \mathbb{R}^n$, satisfazendo $\mathbf{I} - \langle[\mathbf{A}]\rangle\,\tilde{\mathbf{B}} \leq \langle[\mathbf{A}]\rangle\,\mathbf{v}\,\mathbf{w}^T$, é fornecida por

$$\mathbf{B} = \tilde{\mathbf{B}} + \mathbf{v}\,\mathbf{w}^T. \tag{5.1}$$

Pela definição de uma matriz $[\mathbf{H}]$, existe um vetor $\mathbf{v} > 0$ tal que $\mathbf{u} = \langle[\mathbf{A}]\rangle\,\mathbf{v} > 0$. Este vetor $\mathbf{v}$ deve satisfazer a Equação 5.1, toma-se vetor $\mathbf{w}$ com componentes

$$w_k = \max_i \frac{-R_{ik}}{u_i},$$

onde

$$\mathbf{R} = \langle[\mathbf{A}]\rangle\,\tilde{\mathbf{B}} - \mathbf{I}$$

Com relação ao vetor $\mathbf{v}$. Assumindo que existe um vetor $\tilde{\mathbf{u}}$ que $\mathbf{v} = \tilde{\mathbf{B}}\tilde{\mathbf{u}} \approx \langle[\mathbf{A}]\rangle^{-1}\tilde{\mathbf{u}} > 0$ então $[\mathbf{A}]$ é uma matriz-$\mathbf{H}$, e se $\mathbf{u} = \langle[\mathbf{A}]\rangle\mathbf{v} \approx \tilde{\mathbf{u}}$ é positivo, então a aproximação $\tilde{\mathbf{B}}$ é suficiente. Desde que $\langle[\mathbf{A}]\rangle^{-1}$ é não negativa, o vetor $\tilde{\mathbf{u}} = (1;\cdots;1)$ é suficiente para produzir $\langle[\mathbf{A}]\rangle^{-1}\tilde{\mathbf{u}} > 0$.

Os valores de $\mathbf{u}$ e $\mathbf{R}$ devem ser calculados com arredondamentos direcionados para baixo, $\mathbf{w}$ e $\mathbf{B}$ calculados com arredondamentos para cima, enquanto $\tilde{\mathbf{B}}$ e $\mathbf{v}$ podem ser calculados com arredondamento para o mais próximo. Observando as operações envolvidas, é fácil verificar que o programa MATLAB a seguir (adaptado para eliminar comandos obsoletos no toolbox INTLAB) produz os desejados limites rigorosos e, portanto, um cerco rigoroso.

```
01: function x = nsolve(A,b)
02: n = dim(A);                                          % dimensão de A
03: dA = diag(A);                  % salva as entradas da diagonal antes de sobre-escrever A
04: A = compmat(A);                                      % A contém agora a matriz comparação
05: B = inv(A);                                          % inversa aproximada da matriz comparação
06: v = abss(B*ones(n,1));                               % força a não negatividade de v
07: setround(-1)                                         % a sintaxe atual do comando SetRoundDown
08: u = A*v;                                             % é aproximadamente igual ao vetor de um
09: if  all(min(u)>0)                                    % confere a positividade de u
10:     x = midrad(0, inf+zeros(n,1));                   % A não é numericamente uma matriz-H
11: else
12:     dAc = diag(A);             % salva as entradas da diagonal antes de sobre-escrever A
13:     A = A*B-eye(n);                                  % A contém a matriz do resíduo
14:     setround(1)                                      % a sintaxe atual do comando SetRoundUp
15:     w = zeros(1,n);
16:     for i=1:n,
17:         w = max(w,(-A(i,:))/u(i));
18:     end;
19:     B = B+v*w;                                       % limites rigorosos para o B exato
20:     u = B*abss(b);
21:     d = diag(B);
22:     alpha = dAc+(-1)./d;                             % garante a arredondamento para cima
23:     beta = u./d-abss(b);
24:     x = (b+midrad(0,beta))./(dA+midrad(0,alpha));
25: end
```

Para calcular o cerco Ning-Kearfoot na maneira acima são necessárias $2n^3 + O(n^2)$ operações (contando adição e multiplicação separadamente), para a inversão aproximada, e uma mesma quantidade para o cálculo do resíduo, mais $O(n^2)$ operações para as operações restantes, dando um total de $4n^3 + O(n^2)$ operações. Usando uma análise de erro na execução do processo de inversão, limites sub-ótimos para o resíduo podem ser obtidos com $O(n^2)$ operações [150].

Uma implementação escrita usando MATLAB *toolbox* INTLAB para o método de Hansen-Bliek-Rohn-Ning-Kearfott-Neumaier é fornecida pela função **HSOLVE.m** in [52]. A função **HSOLVE.m** é, de uma maneira geral, quase igual à função **NSOLVE.m** apresentada anteriormente. As principais diferenças são:

- Entre a segunda e a terceira linha de código a função **HSOLVE.m** realiza o pré-condicionamento

dado por:

```
C = inv(mid(A));
A = C*A;
b = C*b;
```

- Entre a linha 18 e a linha 19 foram introduzido dois comandos na função **HSOLVE.m**:

```
dlow = v.*w'-diag(B);
dlow = -dlow;
```

- Na função **HSOLVE.m** o comando de atribuição da linha 23 foi modificado para:

```
beta = u./dlow-abss(b);
```

A função **HSOLVE.m** foi utilizada no Capítulo: estudos de casos.

5.4 Método de Krawczyk

O método de Krawczyk é um esquema iterativo que, dado um cerco intervalar inicial do conjunto solução *hull*, realiza refinamentos iterativos para melhorar a estimativa inicial. O sistema linear intervalar $[\mathbf{A}]\cdot[\mathbf{x}]=[\mathbf{b}]$ pode ser precondicionado, utilizando uma matriz $\mathbf{C}\in\mathbb{R}^{n\times n}$, que corresponde à inversa da matriz do ponto médio de $[\mathbf{A}]$, tal que:

Assumindo um vetor intervalar conhecido $[\mathbf{x}]^{(i)}$ onde $\square\Sigma_{\exists\exists}([\mathbf{A}];[\mathbf{b}])\subseteq[\mathbf{x}]^{(i)}$, então ,

$$\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{b}}=\mathbf{C}\tilde{\mathbf{b}}+(\mathbf{I}-\mathbf{C}\tilde{\mathbf{A}})\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{b}}\in\mathbf{C}[\mathbf{b}]+(\mathbf{I}-\mathbf{C}[\mathbf{A}])[\mathbf{x}]^{(i)}$$

é satisfeito para $\tilde{\mathbf{A}}\in[\mathbf{A}]$ e $\tilde{\mathbf{b}}\in[\mathbf{b}]$, tal que

$$\square\Sigma_{\exists\exists}([\mathbf{A}];[\mathbf{b}])\subseteq[\mathbf{x}]^{(i)}\Rightarrow\square\Sigma_{\exists\exists}([\mathbf{A}];[\mathbf{b}])\subseteq(\mathbf{C}[\mathbf{b}]+(\mathbf{I}-\mathbf{C}[\mathbf{A}])[\mathbf{x}]^{(i)})\cap[\mathbf{x}]^{(i)}. \tag{5.2}$$

Isto dá a fórmula de iteração de Krawczyk:

$$[\mathbf{x}]^{(i+1)}=(\mathbf{C}[\mathbf{b}]+(\mathbf{I}-\mathbf{C}[\mathbf{A}])[\mathbf{x}]^{(i)})\cap[\mathbf{x}]^{(i)}. \tag{5.3}$$

Para começar a iteração é necessário um vetor intervalar inicial $[\mathbf{x}]^{(0)}$, tal que a solução $\tilde{\mathbf{x}}\in[\mathbf{x}]^{(0)}$ e $[\mathbf{x}]^{(0)}\supseteq\square\Sigma_{\exists\exists}([\mathbf{A}];[\mathbf{b}])$. Na solução do sistema intervalar relacionado à análise no estado da ISPN, foi utilizado o vetor inicial intervalar como sendo

$$[\mathbf{x}]^{(0)}=([0;\ 1];\cdots;[0;\ 1])^T.$$

A soma dos raios pode ser computada após cada iteração e ser comparada com a soma precedente. O processo da iteração pára, se a soma dos raios das componentes de $[\mathbf{x}]^{(i)}$ não diminuem. Uma implementação pessoal deste algoritmo (escrito em MATALAB *toolbox* INTLAB) é dada pela função **mkraw.m**. Outra implementação do algoritmo é dada pela função **kraw.m** em [52]. O vetor intervalar inicial e o critério de parada são distintos em cada implementação. **kraw.m** define o vetor intervalar inicial como sendo

$$[\mathbf{x}]^{(0)} = ([-\alpha;\ \alpha];\cdots;[-\alpha;\ \alpha])^T$$

com

$$\alpha = \frac{\|\mathbf{C}[\mathbf{b}]\|_\infty}{1-\beta}$$

$$\beta = \left\|\mathbf{I} - \mathbf{C}\tilde{\mathbf{A}}\right\| < 1.$$

O processo iterativo para se $\frac{1+\beta}{2} \cdot \sum rad([x]_i)$ for menor que $\sum rad([x]_i)$ na computação precedente. O algoritmo de Krawczyk é exibido abaixo [22].

$[\mathbf{x}]^0 = [\mathbf{x}]$ (*vetor intervalar inicial*)
$k = 0$
$[\mathbf{e}] = \mathbf{C} \cdot [\mathbf{b}]$
$[\mathbf{F}] = \mathbf{C} \cdot [\mathbf{A}] - \mathbf{I}$
repita
  $[\mathbf{x}]^{k+1} = ([\mathbf{e}] - [\mathbf{F}] \cdot [\mathbf{x}]^k) \bigcap [\mathbf{x}]^k$
  $k = k + 1$
até (critério de parada)

O algoritmo de Krawczyk pode ser formulado em função do resíduo. Considere $\tilde{\mathbf{x}} \in \mathbb{R}^n$ ser um vetor intervalar inicial, $[\mathbf{r}]$ o resíduo $[\mathbf{b}] - [\mathbf{A}] \cdot \tilde{\mathbf{x}}$, e considere $[\mathbf{d}]$ ser um cerco para a solução do sistema intervalar $[\mathbf{A}] \cdot [\mathbf{x}] = [\mathbf{r}]$. A iteração de Krawczyk, em função do resíduo, é mostrada a seguir [22].

$[\mathbf{d}]^0 = [\mathbf{d}]$ (*vetor intervalar inicial*)
$k = 0$
$[\mathbf{e}] = \mathbf{C} \cdot [\mathbf{r}]$
$[\mathbf{F}] = \mathbf{C} \cdot [\mathbf{A}] - \mathbf{I}$
repita
  $[\mathbf{y}]^k = \tilde{\mathbf{x}} + [\mathbf{d}]^k$
  $[\mathbf{d}]^{k+1} = ([\mathbf{e}] - [\mathbf{F}] \cdot [\mathbf{d}]^k) \bigcap [\mathbf{d}]^k$
  $k = k + 1$
até (critério de parada)

Será provado na Seção 5.5 que se os vetores intervalares inicias $[\mathbf{x}]^0$ e $[\mathbf{d}]^0$ na iteração Krawczyk e na iteração Krawczyk em função do resíduo, respectivamente, são iguais, então $[\mathbf{x}]^k \subseteq [\mathbf{d}]^k \; \forall \; k > 0$. Em outras palavras, a iteração Krawczyk em função do resíduo nunca dá melhores resultados que a versão simples do método. Por outro lado, se os raios $rad([\mathbf{A}]$ e $rad([\mathbf{b}])$ são muito pequenos, o resíduo $[\mathbf{b}] - [\mathbf{A}] \cdot \tilde{\mathbf{x}}$ pode ter seu cerco com um menor erro de arredondamento do que $[\mathbf{b}]$, então, na prática, a versão em função do resíduo pode ser mais efetiva neste caso. Observa-se que, em ambas versões do método, o núcleo computacional principal de cada iteração é um produto de um vetor por uma matriz, portanto a complexidade é de ordem n^2 operações de ponto flutuante intervalar.

5.5 A função verifylss

O MATLAB *toolbox* INTLAB [142] fornece a função **verifylss.m** para solução de sistemas equações lineares intervalares ou para fornecer cercos verificados da solução de um sistema intervalar degenerado (pontual). Uma discussão sucinta da função é apresentada em [52]. Se qualquer dos elementos e $[\mathbf{A}]$ ou $[\mathbf{b}]$ são do tipo intervalar então o operador MATLAB " \ " (por exemplo, `x=a/b`, em `x` teremos a solução do sistema no MATLAB *toolbox* INTLAB) será sobrecarregado pela chamada da função **verifylss.m**. A função **verifylss.m** é também usada para solução de sistemas esparsos, para solução de sistemas sobredeterminados e para sistemas subdeterminados. Mas o foco dado aqui será no caso em que $[\mathbf{A}] \in I(\mathbb{R}^{n \times n})$ for denso.

Quando a função vai solucionar um sistema linear denso ela tenta primeiro resolver o sistema usando a interação de Krawczyk em função do resíduo (ver o algoritmo na Seção 5.4). Para a maioria das matrizes $[\mathbf{A}]$ um cerco será encontrado, contudo, se após 7 iterações, o processo iterativo não estiver convergindo, a função **verifylss.m** reverte para o método Hansen-Bliek-Rohn-Ning-Kearfott-Neumaier (descrito na Seção 5.3).

A complexidade da função está avaliada em [52]. A função requer $2n^3$ operações para calcular a inversa de uma matriz, $4n^3$ operações para multiplicação matricial intervalar e $O(n^2)$ para todos os outros cálculos. Isto resulta no termo dominante de $6n^3$, operações o qual, teoricamente deve produzir um fator de 9 vezes o tempo gasto para resolver um sistema degenerado, usando a função **verifylss.m** no lugar da solução do sistema sem usar aritmética intervalar no framework MATLAB toolbox INTLAB. O fator, geralmente, é menor na prática [52].

5.6 As funções verenclinthull e verinthull

As funções **verinthull.m** e **verenclinthull.m**, do MATLAB *toolbox* INTLAB, foram obtidas da *Home Page* de Jiri Rohn (25/08/2007) [137]. Estas funções serão usadas em experimentos numéricos do Capítulo 6. Infelizmente, naquele tempo, nenhum detalhe relacionado a ambas funções estavam disponíveis. Além dissso, cada uma dessas funções fazem chamadas a funções de arquivos-P, os arquivos-P são BINÁRIO puro obtidos pela codificação de arquivos-M, de modo que uma análise detalhada dos códigos fonte não foi possível.

É possível converter código-P $\rightarrow$ arquivo-M, pelo uso de engenharia reversa. Porém, na prática, tornou-se impossível: por não ser apoiado por MATLAB ou por qualquer outro produto

(software) de terceiros conhecido.

A *Home Page* de Jiri Rohn foi transferida para um novo "link". Em (15/12/2008) a função **verintervalhull.m** (**intervalhull.p**, sua função subordinada) tornou-se a única disponível para solução de sistemas lineares intervalares. A função **verinthull.m** (**intervalhull.p**, sua função subordinada) e a função **verenclinthull.m** (**verenclinthull.p**, sua função subordinada) já não estão mais disponíveis. No apêndice F está a listagem dos arquivos-M presentes à época.

Felizmente **verintervalhull.m** (15/12/2008) comenta alguns pontos do código de fonte (ver a listagem do arquivo-M no apêndice F), de modo que foi possível verificar que está baseado no algoritmo descrito em [136]. Logo foi possível determinar que **verenclinthull.m** implanta "*The Hansen-Bliek-Rohn enclosure algorithm*" (ver pag. 65, [136]) e **verinthull.m** implanta "*An algorithm for computing the hull*" (ver pag. 64, [136]). A complexidade computacional mencionada do método "*The Hansen-Bliek-Rohn enclosure algorithm*" é de tempo polinomial, e a do "*hull algorithm*" é um problema *NP-hard*.

5.7 Método subdiferencial de Newton desenvolvido por Shary

A proposta desta seção é descrever um método algébrico para encontrar uma aproximação do conjunto solução unido de um sistema linear intervalar. Para isso, a solução é realizada por meio de uma equação linear não intervalar no espaço euclidiano de dimensão dupla ($\mathbb{R}^{(}2n)$). O algoritmo é desenvolvido por meio da aritmética Kaucher definida previamente.

O conjunto solução do sistema intervalar

$$[\mathbf{A}] \cdot [\mathbf{x}] = [\mathbf{b}]$$

coincide com o conjunto solução do sistema intervalar [145]

$$[\mathbf{x}] = [\mathbf{C}] \cdot [\mathbf{x}] + [\mathbf{b}] \; se \; [\mathbf{C}] = I - [\mathbf{A}]. \tag{5.4}$$

A solução de um sistema intervalar de equações lineares é transformado no problema de encontrar uma solução algébrica da equação intervalar [145]

$$[\mathbf{C}] \cdot [\mathbf{x}] \ominus [\mathbf{x}] + [\mathbf{b}] = \mathbf{0},$$

onde $[\mathbf{A}] = \left[\underline{\mathbf{A}}; \overline{\mathbf{A}}\right]$ é a matriz $n \times n$ e $[\mathbf{b}] = \left[\underline{\mathbf{b}}; \overline{\mathbf{b}}\right]$ o vetor n de intervalos reais ($[\mathbf{A}] \in \mathbb{KR}^{n \times n}$, $[\mathbf{b}] \in \mathbb{KR}^{n}$).

Em uma primeira vista, parece ser o conhecido problema da solução do sistema linear de equações da análise numérica clássica e da álgebra linear. A diferença principal é o conjunto $\mathbb{KR}$, no qual a equação é considerada não é um espaço linear devido à falta da lei distributiva que viola o axioma,

$$(\alpha + \beta) \cdot [\mathbf{x}] = \alpha \cdot [\mathbf{x}] + \beta \cdot [\mathbf{x}],$$

para todos os vetores $[\mathbf{x}] \in \mathbb{KR}^n$, e quaisquer escalares α, $\beta \in \mathbb{R}$. Assim a maioria dos métodos computacionais não é aplicada diretamente.

O método subdiferencial de Newton desenvolvido por Shary parte da imersão $\sigma : \mathbb{KR}^n \to \mathbb{R}^{2n}$, como segue:

$$([x]_1;\ [x]_2;\ \cdots;\ [x]_n) \longmapsto (-\underline{x}_1; -\underline{x}_2; \cdots, \underline{x}_n;\ \overline{x}_1;\ \overline{x}_2;\ \cdots;\ \overline{x}_n), \tag{5.5}$$

ou seja, os pontos extremos esquerdos de $[\mathbf{x}]$ são tomados com sinais opostos das n primeiras componentes de um vetor real de dimensão $2n$, e os pontos extremos direitos de $[\mathbf{x}]$ vão da $(n+1)-$*ésima* até a $2n-$*ésima* componentes do vetor $2n$ real. Este processo é chamado *imersão padrão* do espaço intervalar $\mathbb{KR}^n$ no $\mathbb{R}^{2n}$. Para imersão $\sigma : \mathbb{KR}^n \to \mathbb{R}^{2n}$, segue que

$$\sigma([\mathbf{x}]) \neq \mathbf{0} \; no \; \mathbb{R}^{2n} \quad \Longleftrightarrow \quad [\mathbf{x}] \neq \mathbf{0} \; no \; \mathbb{KR}^n,$$

enquanto

$$\sigma(\mathbf{0} \; no \; \mathbb{KR}^n) = \mathbf{0} \; no \; \mathbb{R}^{2n}.$$

Pode-se, portanto transformar o "problema do sistema linear intervalar", ou seja, o de encontrar os zeros do mapeamento

$$\psi([\mathbf{x}]) = [\mathbf{C}] \cdot [\mathbf{x}] \ominus [\mathbf{x}] + [\mathbf{b}] \tag{5.6}$$

no problema da solução da equação induzida

$$\Psi(\mathbf{x}) = \mathbf{0} \tag{5.7}$$

no $\mathbb{R}^{2n}$, tal que $\Psi = \sigma \circ \psi \circ \sigma^{-1} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ (onde $\circ$ representa composição), ou seja,

$$\begin{aligned} \Psi(\mathbf{x}) &= \sigma([\mathbf{C}] \cdot \sigma^{-1}(\mathbf{x}) \ominus \sigma^{-1}(\mathbf{x}) + [\mathbf{b}]) \\ &= \sigma\left([\mathbf{C}] \cdot \sigma^{-1}(\mathbf{x})\right) - \mathbf{x} + \sigma([\mathbf{b}]). \end{aligned} \tag{5.8}$$

assim sendo, a equação original $\psi([\mathbf{x}]) = \mathbf{0}$ tem uma solução $[\mathbf{x}] \in \mathbb{KR}^n$ se e somente se a equação induzida $\Psi(\mathbf{x}) = \mathbf{0}$ tem uma solução $\mathbf{x}^* \in \mathbb{R}^{2n}$, e satisfaz

$$[\mathbf{x}]^* = \sigma^{-1}(\mathbf{x}^*).$$

Uma prova separada para a existência e a unicidade da solução das Equações (5.7)-(5.8) não foi feita, mas elas são garantidas [145].

Para uma função $f : \mathbb{R}^n \to \mathbb{R}$, a derivada unilateral em um ponto $\mathbf{x}$ com relação a uma direção $\mathbf{y} \in \mathbb{R}^n$, a derivada direcional, é definida como o limite

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{y}} = \lim_{\alpha \to 0} \frac{f(\mathbf{x} + \alpha \mathbf{y}) - f(\mathbf{x})}{\alpha},$$

contanto que o limite exista. Portanto temos os seguintes casos:

$$\frac{\partial \Psi_i(\mathbf{x})}{\partial x_j^-} = \lim_{\alpha \to 0} \frac{\Psi_i(x_1; \cdots x_{j-1}; x_j - \alpha; x_{j+1}; \cdots x_{2n}) - \Psi_i(x_1; \cdots x_{2n})}{\alpha}$$

e

$$\frac{\partial \Psi_i(\mathbf{x})}{\partial x_j^+} = \lim_{\alpha \to 0} \frac{\Psi_i(x_1; \cdots x_{j-1}; x_j + \alpha; x_{j+1}; \cdots x_{2n}) - \Psi_i(x_1; \cdots x_{2n})}{\alpha},$$

que são derivadas parciais unilaterais da componente Ψ_i em um ponto $\mathbf{x}$, da esquerda e da direita, com relação a direção da $j-$*ésima* coordenada.

A subdiferencial [1] $\partial\Psi(\mathbf{x})$ do mapeamento Ψ definido pela Equação 5.8 é a matriz $(2n \times 2n)$ intervalar (própria) :

$$\begin{pmatrix} \left[\frac{\partial\Psi_1(\mathbf{x})}{\partial x_1^-}; \frac{\partial\Psi_1(\mathbf{x})}{\partial x_1^+}\right] & \cdots & \left[\frac{\partial\Psi_1(\mathbf{x})}{\partial x_{2n}^-}; \frac{\partial\Psi_1(\mathbf{x})}{\partial x_{2n}^+}\right] \\ \vdots & \ddots & \vdots \\ \left[\frac{\partial\Psi_{2n}(\mathbf{x})}{\partial x_1^-}; \frac{\partial\Psi_{2n}(\mathbf{x})}{\partial x_1^+}\right] & \cdots & \left[\frac{\partial\Psi_{2n}(\mathbf{x})}{\partial x_{2n}^-}; \frac{\partial\Psi_{2n}(\mathbf{x})}{\partial x_{2n}^+}\right] \end{pmatrix}. \tag{5.9}$$

Consideremos Ψ diferenciável em $\mathbf{x}$,

$$\frac{\partial\Psi_i(\mathbf{x})}{\partial x_j^-} = \frac{\partial\Psi_i(\mathbf{x})}{\partial x_j^+} =$$

a derivada parcial comum

$$\frac{\partial\Psi_i(\mathbf{x})}{\partial x_j},$$

de maneira a simplificar a representação matricial da Expressão 5.9. Assim sendo, a subdiferencial $\partial\Psi(\mathbf{x})$ consiste no único elemento, a saber, da matriz de Jacobi:

$$\begin{pmatrix} \frac{\partial\Psi_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial\Psi_1(\mathbf{x})}{\partial x_{2n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial\Psi_{2n}(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial\Psi_{2n}(\mathbf{x})}{\partial x_{2n}} \end{pmatrix}$$

A *parte positiva* $[x]^+$ e a *parte negativa* $[x]^-$ de um intervalo próprio (ver Seção 2.1.2.2) $[x]$ são os seguintes intervalos:

$[x]^+ = \{[x]^+ \mid x \in [x]\} = \{\max(x;0) \mid x \in [x]\}$,
$[x]^- = \{[x]^- \mid x \in [x]\} = \{\max(-x;0) \mid x \in [x]\}$.

Por exemplo,

$[-3;1]^+ = [0;1], \quad [-2;1]^- = [0;2],$
$[3;5]^+ = [3;5], \quad [2;3]^- = [0;0].$

Para a subdiferencial $\partial\Psi(x)$ do mapeamento Ψ definido por(5.8), a seguinte estimativa é válida

$$\partial\Psi(\mathbf{x}) \subseteq \begin{pmatrix} \mathbf{C}^+ & \mathbf{C}^- \\ \mathbf{C}^- & \mathbf{C}^+ \end{pmatrix} - I \tag{5.10}$$

Como consequência da propriedade de inclusão apresentada, tem-se que

$$\mathbf{D}^{(k)} \in \begin{pmatrix} \mathbf{C}^+ & \mathbf{C}^- \\ \mathbf{C}^- & \mathbf{C}^+ \end{pmatrix} - I$$

é satisfeita para cada $\mathbf{D}^{(k)}$.

Para resolver as equações (5.7)-(5.8) no espaço no $\mathbb{R}^{2n}$, é proposto o algoritmo iterativo seguinte:

[1] nome utilizado por Shary em seus trabalhos

ALGORITMO [145]: Método subdiferencial de Newton.

Partiremos do vetor $\mathbf{x}^{(0)} \in \mathbb{R}^{2n}$, calcula-se a solução do sistema do "ponto médio"

$$I - (mid([\mathbf{C}])^{\sigma})\mathbf{x} = \sigma([\mathbf{b}]).$$

Se a $k-th$ aproximação $\mathbf{x}^{(k)}$, $k = 0, 1, \cdots$, já for conhecida, calcule qualquer $D^{(k)} \in \partial\Psi(\mathbf{x}^{(k)})$ e faça

$$\mathbf{x}^{(k+1)} := \mathbf{x}^{(k)} - \tau \left(D^{(k)}\right)^{-1} \left(\Psi(\mathbf{x}^{(k)})\right)$$

para alguma constante $\tau \in [0;\ 1]$.

Ao comparar o método algébrico com outros métodos para solução do "*outer problem*", Shary afirma que o método tem as propriedades de encontrar cerco, para o conjunto solução USS, dos esquemas iterativos intervalares, com complexidade computacional polinomial. De uma maneira geral, o método algébrico apropriadamente modificado e atualizado, pode servir como ferramenta para cálculo rápido de cercos razoavelmente bons do conjunto solução para sistemas intervalares de equações lineares gerais. Os códigos computacionais a seguir representam implementações do método.

5.7.1 Códigos de computador

A *home-page* de Sergey P. Shary (23/03/2008) [143] apresenta alguns programas para a solução de vários problemas relacionados a sistemas lineares de equações intervalares:

- computação das soluções formais dos sistemas lineares intervalares $[\mathbf{A}] \cdot [\mathbf{x}] = [\mathbf{b}]$ na aritmética Kaucher pelo método subdiferencial de Newton (arquivo fonte em C **subdiff.c**, código executável **subdiff.exe**),
- computação das soluções formais dos sistemas lineares intervalares $[\mathbf{A}] \cdot [\mathbf{x}] = [\mathbf{b}]$ na aritmética Kaucher por um método iterativo de passo simples estacionário, com divisão real modificada da matriz (arquivo fonte em C **re_split.c**, código executável **re_split.exe**),
- computação das soluções formais dos sistemas lineares intervalares $[\mathbf{x}] = [\mathbf{C}] \cdot [\mathbf{x}] + [\mathbf{b}]$ na aritmética Kaucher pelo método subdiferencial de Newton (arquivo fonte em C **alouter0.c**, código executável **alouter0.exe**),
- computação das soluções formais dos sistemas lineares intervalares $[\mathbf{G}] \cdot [\mathbf{x}] = ([\mathbf{G}] - [\mathbf{A}]) \cdot [\mathbf{x}] + [\mathbf{b}]$ na aritmética Kaucher pelo método subdiferencial de Newton (arquivo fonte em C **alouter2.c**, código executável **alouter2.exe**).

Os arquivos fonte em C (**subdiff.c**, **re_split.c**, **alouter0.c**, **alouter2.c**) foram modificados para permitir a interface com a ferramenta protótipo ISPN prototype (no *framework* MATLAB *toolbox* INTLAB), assim como para proporcionar medir o tempo de execução do programa. As versões modificadas dos programas foram compiladas em Dev-C++.

5.8 Considerações finais

Os sistemas lineares representam o núcleo da solução computacional de muitos modelos que surgem no campo social, econômico, técnico e cientifico. Tem-se, portanto, vários métodos e algoritmos (softwares) para solução de sistemas lineares. A aritmética com precisão finita dos computadores torna necessária a análise dos erros de arredondamento. A aritmética intervalar clássica foi desenvolvida para lidar com a solução de problemas de erros de arredondamento e truncamento em cálculos com computadores [113, 114], dado que um número em ponto flutuante representa um intervalo de números reais. Por outro lado, em muitas aplicações do mundo real os dados são afetados por incertezas e variabilidades, ou seja, sabe-se somente que eles pertencem a certos intervalos. Muitos modelos intervalares dependem da solução de um sistema de equações lineares intervalares. Este é o caso da solução do estado estacionário dos modelos ISPN para encontrar a *p.m.f.* intervalar. A solução de um sistema de equações lineares intervalares é um problema *NP-hard* [72]. No entanto, é possível se encontrar cercos (vetor intervalar contendo a solução exata relacionado ao USS) em tempo polinomial. Este capítulo apresentou o conjunto de métodos que foram explorados e implementados (até o momento) no protótipo computacional para análise ISPN do estado estacionário. Observa-se que nas implantações dos algoritmos é usado o pré-condicionamento com a inversa da matriz do ponto médio. O pré-condicionamento tem um custo computacional adicional proporcional a n^3 operações do ponto flutuante. Existem algoritmos para o cálculo do vetor intervalar envoltório exato, mas geralmente têm complexidade exponencial [119]. Muitos autores também encontram a solução do envoltório com algoritmos de complexidade sub-exponencial para alguns tipos de sistemas intervalares com propriedades específicas. A integração dos algoritmos a ferramenta protótipo ISPN foi uma contribuição dada para avaliar os resultados das métricas nos estudos de caso apresentados no capítulo 6. O versão intervalar o algoritmo de Grassmann foi uma contribuição de um método usado para análise de cadeias de Markov intervalar. A versão do método de Krawczyk (função **mkraw.m**) é outra contribuição de método usado especificamente para análise de cadeias de Markov intervalar.

CAPÍTULO 6

Estudo de Casos

Neste capítulo, serão tratados 4 estudos de casos com o objetivo geral de explorar a modelagem ISPN. Na Seção 6.1 o principal objetivo foi avaliar os gargalos na ferramenta protótipo ISPN, que utiliza o *framework* MATLAB *toolbox* INTLAB, ao mesmo tempo em que se fez a avaliação de desempenho do sistema usando a modelagem ISPN. Os gargalos são as tarefas críticas em relação ao tempo de execução. A Seção 6.2 é uma aplicação da modelagem intervalar ISPN para análise da disponibilidade. Neste caso, consideram-se alguns parâmetros que pertencem a intervalos e, então, realizaram-se cálculos das métricas intervalares. A Seção 6.3 avalia vários métodos para a solução dos sistemas de equações intervalares subjacentes a modelos ISPN para determinação do vetor *p.m.f.* intervalar do estado estacionário. Finalmente, a Seção 6.4 investiga a aplicação da aritmética Kaucher, através da ferramenta protótipo ISPN para avaliar o potencial da solução como ferramenta na análise ISPN.

6.1 Estudo de caso 1: Modelo ISPN de um Sistema Paralelo

Nesta seção vamos explorar a ferramenta protótipo com a análise do Modelo ISPN de um Sistema Paralelo. O modelo ISPN da Figura 6.1 é uma descrição de um sistema paralelo [107]. O modelo ISPN é composto por nove lugares e nove transições. As características das transições deste modelo são apresentadas na Tabela 6.1. As marcações iniciais são $(m(p_1) = N,\ N = 2, \cdots, 8)$.

Os resultados apresentados na Tabela 6.2 mostram a propagação dos efeitos dos parâmetros intervalares com o crescimento do espaço de estados. Quanto maior for o espaço de estado verificam-se intervalos mais espessos (métricas obtidas).

Testou-se a influência: a) das CPU utilizadas Pentium 3 (P3) de 933MHz e 256 MB de RAM e Pentium 4 de 2.4GHz com 256 MB e, também com 2GB de RAM; b) da quantidade de RAM do P4 (256 MB e 2GB de RAM)

As Tabelas 6.3 e 6.4 mostram tempos decorridos utilizando plataformas Linux/Windows XP. O aumento de memória de 256MB para 2GB no P4 não afetou os tempos decorridos (ver as Tabelas 6.5 e 6.6). A única melhoria no tempo decorrido, para execução das tarefas críticas examinadas, é observada em função da CPU usada.

Após a obtenção das informações apresentadas nas Tabelas 6.3, 6.4, 6.5 e 6.6 modificou-se a ferramenta protótipo visando melhorar o tempo de execução das tarefas críticas. A Tabela 6.7 apresenta os tempos associados a execução das tarefas críticas após as alterações efetuadas. A geração do espaço de estado/matriz [**Q**] permaneceu crítica.

Para avaliar o estudo de caso 4, a ferramenta teve que ser adaptada para poder exportar o

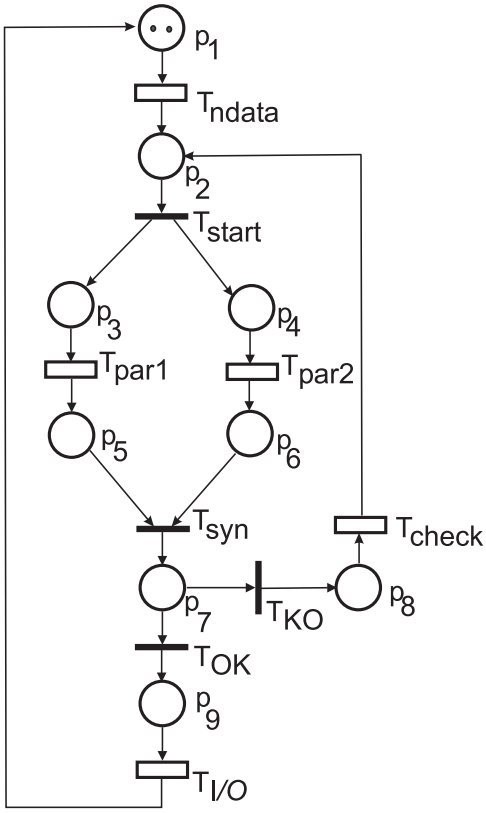


Figura 6.1 A descrição ISPN de um sistema paralelo

Tabela 6.1 Especificações dos parâmetros das transições intervalares da ISPN da Figura 6.1

Transição	**Taxa Intervalar** (h^{-1})	**Semântica**
t_{ndata}	$0,2 \pm 1e-4$	Servidor-Infinito
t_{par1}	$2,0 \pm 1e-3$	Servidor-Simples
t_{par2}	$1,0 \pm 1e-3$	Servidor-Simples
t_{check}	$0,1 \pm 1e-4$	Servidor-Simples
$t_{I/O}$	$5,0 \pm 1e-3$	Servidor-Simples

Transição	**Peso**	**Prioridade**
t_{start}	1	1
t_{syn}	1	1
t_{KO}	$0,01$	1
t_{OK}	$0,99$	1

sistema intervalar para aplicação da aritmética de Kaucher.

Tabela 6.2 Avaliação de Desempenho e cardinalidade do espaço de estado para vários modelos ISPN do sistema paralelo de Figura 6.1. N é o número de marcas do lugar $P_1 = [\mathrm{E}]\{\#P_1\}$. A notação T+V denota o número de marcações *Tangíveis* = *T* e *Voláteis* = *V*. Métricas: $\mathbf{NP}_1$ = número médio de marcas no lugar P_1, **System throughput** $= [\mathrm{prob}]\{\#\mathrm{P}_9 > 0\} \cdot [\lambda]_{I/O}$.

N	T+V	$\mathbf{NP}_1$	System throughput
2	$20+18$	$< 1,50555067214211;$ $0,00295064291426 >$	$< 0,30111056237629;$ $0,00219996135411 >$
3	$50+60$	$< 2,19169821603979;$ $0,00936230661157 >$	$< 0,43834084369878;$ $0,00609012205930 >$
4	$105+150$	$< 2,81824419811506;$ $0,02118586818893 >$	$< 0,56365121324037;$ $0,01198081656274 >$
5	$196+315$	$< 3,37218634785193;$ $0,04010364791493 >$	$< 0,67444122568113;$ $0,01991544118533 >$
6	$336+588$	$< 3,84125571003933;$ $0,06741766653902 >$	$< 0,76825707379898;$ $0,02981260579384 >$
7	$540+1008$	$< 4,21692321527027;$ $0,10394270993835 >$	$< 0,84339293965107;$ $0,04165166201173 >$
8	$825+1620$	$< 4,49775685335994;$ $0,14960261938225 >$	$< 0,89956237993883;$ $0,05522624449375 >$

Tabela 6.3 P3 933 MHz e 256MB de RAM / Linux: tempo total de execução e tempos para execução de tarefas críticas na análise ISPN do sistema paralelo da Figura 6.1.

N	Tempo decorrido hh:mm:ss	Geração do Espaço/ de Estado/ matriz Q (s)	Cálculo de $(1-[\mathbf{P}]^{\mathscr{V}\mathscr{V}})^{-1}$ (s)	Solução do sistema linear intervalar (s)
2	$00:00:08$	$0,710413$	$0,458641$	$0,257773$
3	$00:00:27$	$1,274040$	$0,116563$	$0,115428$
4	$00:03:39$	$6,148245$	$1,177881$	$0,208152$
5	$00:27:44$	$32,704917$	$4,096681$	$0,339843$
6	$03:49:00$	$160,516087$	$22,221522$	$1,069268$
7	$27:53:57$	$696,485623$	$122,496979$	$4,078872$

6.2 Estudo de caso 2: Cálculo de Dependabilidade com ISPN

Esta seção apresenta os resultados da análise de disponibilidade de dois dispositivos de rede extensamente usados em redes de computadores: um Multiplexador ADM (MADM) e um Multiplexador SDH (MSDH). Estes dispositivos são compostos por vários componentes (memória, fonte, *switch*, controlador de mux, tributador - *source power, switch, mux controller, tributary*) e podem possuir ou não mecanismos de redundância.

Os modelos ISPN destes dispositivos são mostrados nas Figuras 6.2 e 6.3. Cada com-

Tabela 6.4 P3 933 MHz 256 MB RAM / Windows XP tempo total de execução e tempos para execução de tarefas críticas na análise ISPN do sistema paralelo da Figura 6.1.

N	**Tempo decorrido hh:mm:ss**	**Geração do Espaço/ de Estado/ matriz Q (s)**	**Cálculo de** $(1-[\mathbf{P}]^{\mathscr{V}\mathscr{V}})^{-1}$ **(s)**	**Solução do sistema linear intervalar (s)**
2	00 : 00 : 02	0,251000	0,030000	0,040000
3	00 : 00 : 12	0,941000	0,080000	0,030000
4	00 : 01 : 56	5,037000	0,451000	0,090000
5	00 : 23 : 28	32,14600	3,835000	0,260000
6	04 : 18 : 20	$1,69594e+002$	22,282000	0,972000
7	32 : 37 : 41	$7,63558e+002$	$1,09027e+002$	3,775000

Tabela 6.5 P4 2.4 GHz 256 MB RAM / Windows XP tempo total de execução e tempos para execução de tarefas críticas na análise ISPN do sistema paralelo da Figura 6.1.

N	**Tempo decorrido hh:mm:ss**	**Geração do Espaço/ de Estado/ matriz Q (s)**	**Cálculo de** $(1-[\mathbf{P}]^{\mathscr{V}\mathscr{V}})^{-1}$ **(s)**	**Solução do sistema linear intervalar (s)**
2	00 : 00 : 01	0,157000	0,015000	0,016000
3	00 : 00 : 07	0,610000	0,047000	0,015000
4	00 : 00 : 41	2,297000	0,156000	0,016000
5	00 : 07 : 59	14,422000	1,422000	0,094000
6	01 : 37 : 16	63,047000	9,063000	0,359000
7	12 : 33 : 30	$2,73719e+002$	38,141000	1,359000

Tabela 6.6 P4 2.4 GHz 2 GB RAM / Windows XP tempo total de execução e tempos para execução de tarefas críticas na análise ISPN do sistema paralelo da Figura 6.1.

N	**Tempo decorrido hh:mm:ss**	**Geração do Espaço/ de Estado/ matriz Q (s)**	**Cálculo de** $(1-[\mathbf{P}]^{\mathscr{V}\mathscr{V}})^{-1}$ **(s)**	**Solução do sistema linear intervalar (s)**
2	00 : 00 : 01	0,172000	0,015000	0,046000
3	00 : 00 : 06	0,578000	0,016000	0,015000
4	00 : 00 : 41	2,313000	0,234000	0,062000
5	00 : 07 : 47	11,813000	1,437000	0,109000
6	01 : 34 : 57	61,141000	7,985000	0,375000
7	12 : 06 : 38	$2,73719e+002$	38,141000	1,359000

ponente tem dois parâmetros, *MTTF_X* (*Mean Time To Fail*) e *MTTR_X* (*Mean Time To Repair*) que representam os tempos associadas às transições *Failure_X* e *Repair_X*, respectiva-

Tabela 6.7 P4 2.4 GHz 256 MB RAM / Windows XP tempo total de execução e tempos para execução de tarefas críticas na análise ISPN do sistema paralelo da Figura 6.1.

N	**Tempo decorrido hh:mm:ss**	**Geração do Espaço/ de Estado/ matriz Q (s)**	**Cálculo de** $(1-[\mathbf{P}]^{\mathscr{V}\mathscr{V}})^{-1}$ **(s)**	**Solução do sistema linear intervalar (s)**
7	00 : 05 : 58	$2,69985e+002$	$53,781000$	$1,625000$
8	00 : 34 : 28	$1,627077e+003$	$3,45594e+002$	$4,219000$

mente. O rótulo "*X*" deve ser substituído pelo nome (abreviatura) do componente, por exemplo, "*MADM_Source_ON*" e "*MADM_Source _OFF*". Os lugares do modelo *X_ON* e *X_OFF* representam a atividade e inatividade de um determinado componente, respectivamente.

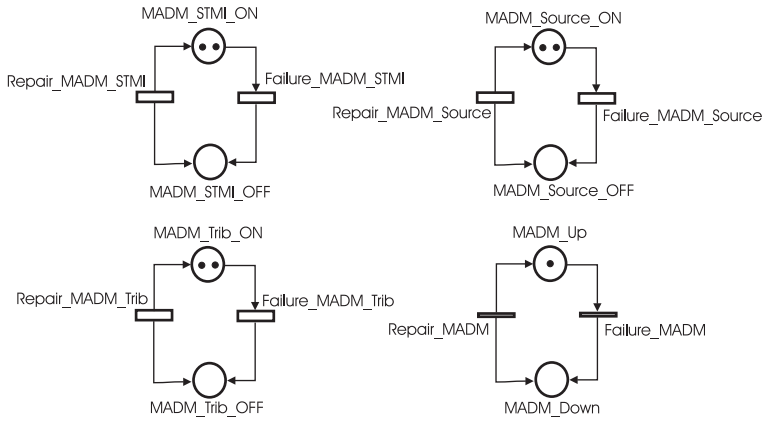


Figura 6.2 O modelo ISPN do multiplexador ADM.

Os modelos agregados (direita inferior da Figura 6.2 e esquerda inferior da Figura 6.3) têm apenas transições imediatas. Os modelos apresentados permitem a análise da confiabilidade e disponibilidade de sistemas. Uma marca nos lugares (*MADM_Up*, *MDH_PSW_Up*) e (*MADM_Down*, *MSDH_PSW_Down*) descrevem os estados operacional e de falha, respectivamente. Todas as transições do modelo agregado têm funções de habilitação (funções de guarda) anexadas. Estas funções são consideradas para combinar componentes que têm uma expressão específica, considerando uma estrutura particular. As Tabelas 6.8 e 6.9 apresentam as expressões de guarda associadas às funções de habilitação do modelo agregado.

As Tabelas 6.10 e 6.11 apresentam os tempos atribuídos às transições do modelo ISPN. Os intervalos de tempo ([7,4; 8,6] e [7,8; 8,2] em horas, atribuídos a MUX ADM e MUX SDH, respectivamente) foram associados a transições do tipo "*Repair_X*" de cada modelo. Estes intervalos de tempo são relacionados à política de manutenção adotada para prover um contrato serviço com qualidade assegurada. As tabelas também exibem a semântica do disparo das transições (IS = (*Infinity Server*) Infinitos Servidores, SS = (*Single Server*) Servidor Simples).

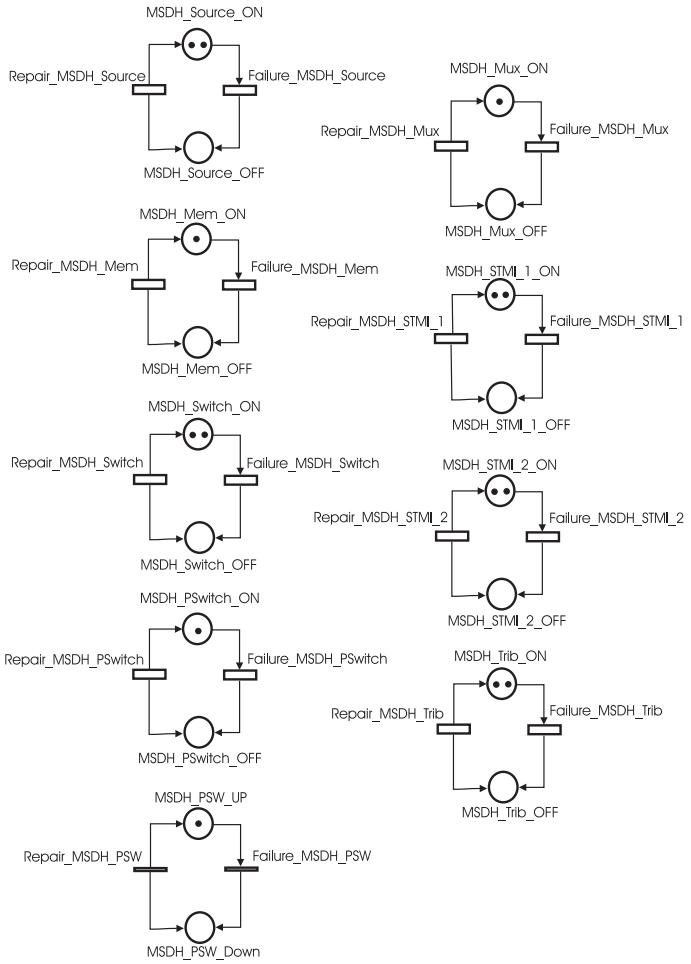


Figura 6.3 O modelo ISPN do multiplexador SDH

Tabela 6.8 Expressões de guarda para o modelo agregado MUX ADM.

Transição	Interpretação	Expressão de Guarda
Repair_MADM	*Reparo do MUX ADM*	$(\#MADM_STMI_ON > 0)$ *and* $(\#MADM_Source_ON > 0)$ *and* $(\#MADM_Trib_ON > 0)$
Failure_MADM	*Falha do MUX ADM*	$(\#MADM_STMI_ON = 0)$ *or* $(\#MADM_Source_ON = 0)$ *or* $(\#MADM_Trib_ON = 0)$

A análise da disponibilidade foi inicialmente conduzida considerando-se intervalos degenerados (pontuais), que corresponde à análise clássica de modelos GSPN, não obstante, os resultados obtidos sejam verificados; significando que o arredondamento e truncamento foram considerados consistentemente através do ambiente da aritmética intervalar. A análise de ISPN

Tabela 6.9 Expressões de guarda para o modelo agregado MUX SDH GPT com Pswitch.

Transição	Interpretação	Expressão de Guarda
Repair_MSDH_PSW	*Reparo do MUX SDH*	$(\#MSDH_Source_ON > 0)$ *and* $(\#MSDH_Mux_ON > 0)$ *and* $(\#MSDH_Mem_ON > 0)$ *and* $(\#MSDH_STM_1_ON > 0)$ *and* $(\#MSDH_Switch_ON > 0)$ *and* $(\#MSDH_STM_2_ON > 0)$ *and* $(\#MSDH_Pwitch_ON > 0)$ *and* $(\#MSDH_Trib_ON > 0)$
Failure_MSDH_PSW	*Falha do MUX SDH*	$(\#MSDH_Source_ON = 0)$ *or* $(\#MSDH_Mux_ON = 0)$ *or* $(\#MSDH_Mem_ON = 0)$ *or* $(\#MSDH_STM_1_ON = 0)$ *or* $(\#MSDH_Switch_ON = 0)$ *or* $(\#MSDH_STM_2_ON = 0)$ *or* $(\#MSDH_Pwitch_ON = 0)$ *or* $(\#MSDH_Trib_ON = 0)$

Tabela 6.10 Tempos das transições exponenciais para o MUX ADM.

Transição	Tempo (h)	Interpretação	Semântica
Repair_STMI	$[7,4; 8,6]$	Reparo do STMI	SS
Failure_STMI	248000	Falha do STMI	IS
Repair_MADM_Source	$[7,4; 8,6]$	Reparo da fonte	SS
Failure_MADM_Source	657000	Falha da fonte	IS
Repair_MADM_Trib	$[7,4; 8,6]$	Reparo do Tributador	SS
Failure_MADM_Trib	271000	Falha do Tributador	IS

(com intervalos espessos) descreve os efeitos das variabilidades nas taxas de transições exponenciais e parâmetros de transições imediatas para índices de dependabilidade.

As Tabelas 6.12 e 6.13 mostram os resultados de avaliação da disponibilidade relacionados aos *MUX ADM* e *MUX SDH GPT com Pswitch*, respectivamente. Se todo retardo relacionado a uma transição temporizada e pesos associados a uma transição imediata são todos os intervalos degenerados (finos), implica dizer que são equivalentes a valores pontuais. Os resultados mostram que são comparáveis aos obtidos por modelos GSPN. Porém, deve ser enfatizado mais uma vez que esses são verificados, dado que os arredondamentos e truncamentos foram tratados através da aritmética intervalar. Desta forma, é importante enfatizar que os resultados das métricas intervalares fornecem garantia de inclusão dos valores pontuais reais. As disponibilidades dos sistemas são

$$< 0,99999999587965; 0,00000000000001 >$$

e

$$< 0,99995446712728; 0,00000000000022 >,$$

isto é, o valor pontual pertencem a

$$[0,99999999587964; 0,99999999587965]$$

e

$$[0,99995446712706; 0,99995446712750],$$

Tabela 6.11 Tempos das transições exponenciais para o MUX SDH GPT com Pswitch.

Transição	Tempo (h)	Interpretação	Semântica
Repair_MSDH_Source	$[7,8; 8,2]$	Reparo da Fonte	SS
Failure_MSDH_Source	657000	Falha da Fonte	IS
Repair_MSDH_Mux	$[7,8; 8,2]$	Reparo do controlador Mux	SS
Failure_MSDH_Mux	350000	Falha do controlador Mux	IS
Repair_MSDH_Mem	$[7,8; 8,2]$	Reparo da Memória	SS
Failure_MSDH_Mem	438000	Falha da Memória	IS
Repair_MSDH_STMI_1	$[7,8; 8,2]$	Reparo do $STMI_1$	SS
Failure_MSDH_STMI_1	762000	Falha do $STMI_1$	IS
Repair_MSDH_Switch	$[7,8; 8,2]$	Reparo do *Switch*	SS
Failure_MSDH_Switch	657000	Falha do *Switch*	IS
Repair_MSDH_STMI_2	$[7,8; 8,2]$	Reparo do $STMI_2$	SS
Failure_MSDH_STMI_2	762000	Falha do $STMI_2$	IS
Repair_MSDH_PSwitch	$[7,8; 8,2]$	Reparo do PSwitch	SS
Failure_MSDH_PSwitch	1490000	Falha do PSwitch	IS
Repair_MSDH_Trib	$[7,8; 8,2]$	Reparo do Tributador	SS
Failure_MSDH_Trib	438000	Falha do Tributador	IS

respectivamente, dado que representam limites superiores e inferiores verificados.

Quando se considera que o tempo médio para reparo (MTTR) não é conhecido precisamente (embora se saiba a sua distribuição), mas se sabe pertencer aos intervalos [7,4; 8,6] e [7,8; 8,2] em horas, a disponibilidade do MUX ADM e MUX SDH é

$$< 0,99984112578944; 0,00015887421057 >$$

e

$$< 0,99993565053080; 0,00006434946921 >,$$

respectivamente. Portanto os valores relativos as disponibilidades dos dispositivos estão garantidamente dentro dos intervalos

$$[0,99968225157888; 1,00000000000000]$$

e

$$[0,99987130106160; 1,00000000000000],$$

respectivamente.

Os resultados obtidos exibem os efeitos da variabilidade (7,5% MUX ADM e 2,5% MUX SDH) devido os MTTRs na disponibilidade do sistema. Em tal caso, o leitor pode observar a redução esperada na disponibilidade do sistema.

Tabela 6.12 Disponibilidade do MUX ADM (Os intervalos finos (degenerados) correspondem ao ponto central dos intervalos espessos).

Tempo exponencial intervalar (tipo)	***Exibição do intervalo/Valor***
Fino	**mid/rad** $\langle 0,99999999587965; 0,00000000000001\rangle$ **inf/sup** $[0,99999999587964; 0,99999999587965]$
Espesso	**mid/rad** $\langle 0,99984112578944; 0,00015887421057\rangle$ **inf/sup** $[0,99968225157888; 1,00000000000000]$

Tabela 6.13 Disponibilidade do MUX SDH GPT com Pswitch (Os intervalos finos (degenerados) correspondem ao ponto central dos intervalos espessos).

Tempo exponencial intervalar (tipo)	***Exibição do intervalo/Valor***
Fino	**mid/rad** $\langle 0,99995446712728; 0,00000000000022\rangle$ **inf/sup** $[0,99995446712706; 0,99995446712750]$
Espesso	**mid/rad** $\langle 0,99993565053080; 0,00006434946921\rangle$ **inf/sup** $[0,99987130106160; 1,00000000000000]$

Uma vez mais, é importante enfatizar a compreensão apropriada do vetor intervalar resultante obtido pela solução de um "sistema intervalar de equações lineares". O número de marcações é 46 (27T+19V, T=*Tangível*, V=*Voláteis*) para o modelo do MUX ADM e 2152 (1944T+208V) marcações para o modelo do MUX SDH GPT com Pswitch. Assim, nós consideramos um sistema intervalar de equações lineares com dimensão igual ao número de marcações tangíveis para cada modelo. O vetor intervalar resultante mostra que os intervalos contêm o conjunto envoltório da solução exata [52]. O método de Krawczyk foi considerado, através da função **verifylss** do *INTLAB toolbox*, para resolver o sistema linear intervalar (E.12). Esta função fornece soluções limites exteriores.

6.3 Estudo de caso 3: Métodos para obtenção do vetor *p.m.f.* intervalar.

Nesta seção um conjunto de soluções numéricas de sistemas de equações lineares intervalares é avaliado. O estudo comparativo das funções do MATLAB *toolbox* INTLAB observa em dois aspectos importantes: 1) o tempo da solução do sistema usando uma função dada; 2) a qualidade das probabilidades intervalares resultantes. Este aspecto é avaliado através das espessuras dos intervalos dos índices do desempenho (intervalo mais estreito é o melhor).

Dois estudos de caso são apresentados. O primeiro diz respeito à solução de um sistema com as 16 equações lineares intervalares. Este exemplo foi considerado anteriormente (para análise GSPN no Capítulo 4 e para análise ISPN no Capítulo 5). É dada ênfase a avaliação

da qualidade dos índices intervalares calculados. Verificando-se o resultado da avaliação deste estudo de caso, dois algoritmos não foram considerados no segundo estudo de caso, desde que, mesmo para sistemas destas dimensões, as espessuras dos intervalos observadas eram demasiadas espessas em comparação com aquelas fornecidas pelas espessuras dos intervalos dos outros algoritmos.

O segundo estudo de caso trata de sistemas que variam de 20 a 825 equações lineares intervalares. O objetivo foi avaliar tempos da solução com o aumento na dimensão assim como para analisar a qualidade dos resultados que se relacionam à largura do intervalo das métricas obtidas usando cada algoritmo.

6.3.1 Duas máquinas e um *buffer* em uma Linha de Montagem

Este estudo de caso diz respeito a análise do estado estacionário do exemplo "Duas máquinas e um *buffer* em uma Linha de Montagem" [27] (veja a Figura 6.4 que é uma reprodução da Figura 4.4). O lugar p_8, possui inicialmente k marcas, que define a capacidade do *buffer* (os resultados foram obtidos para $k = 3$). O número de estados tangíveis é $4(k+1)$. Os intervalos das taxas de disparo das transições, por hora, são exibidos na Tabela 6.14 onde $[\mu]$ representa a taxa de produção inerente à máquina, $[f]$ representa a taxa de falha, e $[r]$ representa as taxa de reparo, respectivamente.

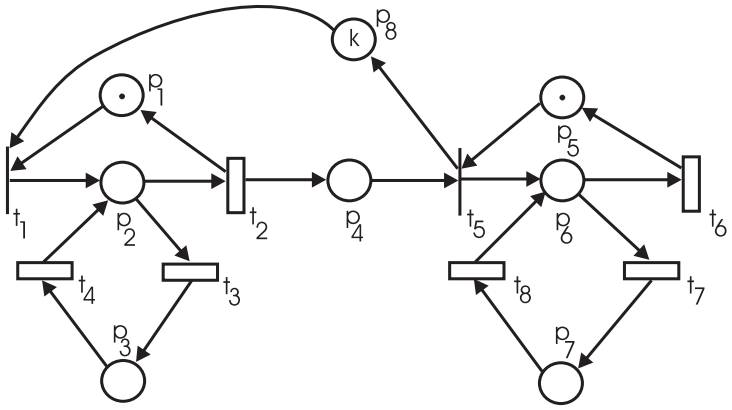


Figura 6.4 Modelo ISPN de Duas máquinas e um *buffer* $(k = 3)$

As métricas de interesse neste estudo são a utilização média das máquinas, isto é, a probabilidade intervalar que uma máquina esteja processando um item, que são dadas por

$$[UM]_1 = [prob(m(p_2) = 1)] \;\; e$$

$$[UM]_2 = [prob(m(p_6) = 1)]$$

A Tabela 6.15 exibe os intervalos da utilização média para cada máquina e o tempo médio para encontrar a solução, associado a cada função. Observam-se que os intervalos das métricas obtidos através das funções VERIFYLSS, VERENCLINTHULL, VERINTHULL, HSOLVE, MKRAW e INTGAUSSP, não são significativamente diferentes. As funções INTGAUSS e INTGRASSMANN produzem resultados que não diferem significativamente entre eles, mas

Tabela 6.14 Taxas intervalares de disparo das transições do modelo ISPN de Duas máquinas e um *buffer* em uma Linha de Montagem.

Transição	Valor	Símbolo
t_2	[9,9; 10,1]	$[\mu]_1$
t_3	[2,99; 3,01]	$[f]_1$
t_4	[4,99; 5,01]	$[r]_1$
t_6	[1,99; 2,01]	$[\mu]_2$
t_7	[3,99; 4,01]	$[f]_2$
t_8	[5,99; 6,01]	$[r]_2$

são maiores que aqueles obtidos através das demais funções. Estes, entretanto, são resultado s esperados, pois o pré-condicionamento não são usados nestas funções.

Os tempos médios para solução do sistema de equações lineares intervalares revelam diferenças significativas. As funções VERENCINTHULL, VERINTHULL, INTGRASSMANN, INTGAUSS e INTGAUSP têm um tempo médio maior do que as funções VERIFYLSS, HSOLVE e MKRAW para fornecer as respectivas soluções.

6.3.2 Sistema Paralelo

Este estudo de caso explora uma família de ICTMC ("Sistema Paralelo") subjacente aos diferentes espaços de estado, maior do que aquele fornecido pelo estudo de caso precedente,para avaliar o impacto das dimensões do espaço do estado na qualidade da métrica intervalar assim como no tempo de computação relacionado a cada algoritmo considerado para resolver o sistema de equações lineares intervalares. Esta família de modelos ICTMC foi usada no Estudo de Caso 1 para explorar a ferramenta protótipo ISPN e corresponde ao modelo da Figura 6.1

A Tabela 6.1 apresenta os parâmetros intervalares para o “sistema paralelo”. A marcação inicial para o lugar denotado por p_1 identifica os membros da família das ICTMCs. As outras marcações dos demais lugares permanecem fixas para cada marcação inicial.

A Tabela 6.16 indica a cardinalidade do espaço de estado da família do “sistema paralelo” do modelo ISPN em função da marcação inicial (veja Figura 6.1 com $m(p_1) = 2$).

A Tabela 6.17 apresenta o tempo médio de solução para cada função particular considerada na resolução do sistema de equações lineares intervalares de cada modelo específico da família do “sistema paralelo” ISPN. Com relação ao tempo da solução, os das funções VERENCLINTHULL, VERINTHULL e INTGAUSSP são maiores se comparados ao das funções VERIFYLSS, HSOLVE e MKRAM. Para a função VERINTHULL, a solução do sistema de equações lineares intervalares com dimensão 336 (número inicial do símbolo no lugar $m(p_1) = 6$) não foi obtido até 12 h com estas dimensões ou maiores não foram resolvidos usando esta função.

Pode-se também observar (ver na Tabela 6.17) que MKRAW é a função mais rápida para resolver o sistema com 825 equações lineares intervalares (número inicial de marcas no lugar $(p_1) = 8$). Esta observação está relacionada à qualidade da implementação e a qualidade de convergência dos algoritmos. Um exemplo é o caso da função HSOLVE e VEREN-

Tabela 6.15 Tempos médios para solução dos sistemas de equações lineares intervalares e os intervalos relativos às utilizações (estas medidas foram obtidas através de um P4 2.4 GHz 256 MB RAM / Windows XP). O modelo ISPN de duas máquinas e um *buffer* em uma Linha de Montagem (Figura 6.4) tem 16 equações lineares intervalares.

Função	**Tempo decorrido[5](s)**	**UM1 e UM2**
VERIFYLSS[1]	$0,074 \pm 0,093$	$< 0,11946700722573;\ 0,00547020800828 >$ $< 0,59733503612865;\ 0,01772832630210 >$
VERENCLINTHULL[2]	$0,188 \pm 0,046$	$< 0,11951738793535;\ 0,00540803591227 >$ $< 0,59745278049312;\ 0,01757586596544 >$
VERINTHULL[2]	$0,355 \pm 0,063$	$< 0,11957231217023;\ 0,00522988467944 >$ $< 0,59745647372415;\ 0,01708441822446 >$
HSOLVE[3]	$0,011 \pm 0,015$	$< 0,11951738793535;\ 0,00540803591227 >$ $< 0,59745278049312;\ 0,01757586596544 >$
MKRAW	$0,027 \pm 0,005$	$< 0,11946700722573;\ 0,00545841662189 >$ $< 0,59733503612865;\ 0,01769361032990 >$
INTGAUSS[3]	$0,536 \pm 0,035$	$< 0,12027536538932;\ 0,02139381256537 >$ $< 0,59792707137548;\ 0,06427884621044 >$
INTGRASSMANN	$3,456 \pm 0,060$	$< 0,11882731854876;\ 0,01849580153517 >$ $< 0,59265684102608;\ 0,06862696014744 >$
INTGAUSSP[4]	$0,517 \pm 0,008$	$< 0,11951696454462;\ 0,00540845930300 >$ $< 0,59745167709622;\ 0,01757696936233 >$

1 - ***Home page*** **de Siegfried M. Rump: [142]**.
2 - ***Home page*** **de Jiri Rohn: [137]**.
3 - ***Home page*** **de Gareth I. Hargreaves: [53]** .
4 - **Adaptado de (3) acima**.
5 - **Intervalo de Confiança de 95% (t-student)** .
Obs : As medidas foram realizadas no PC (10 amostras por resultado apresentado), *i.e.*, não foram controlados os processos embora a máquina não estivesse em rede.

Tabela 6.16 A cardinalidade do espaço de estado do modelo ISPN do "sistema paralelo" para marcações iniciais diferentes do lugar p_1 (Figura 6.1). O T+V denota o número de marcações *tangíveis* $= T$ mais *voláteis* $= V$. N é o número inicial de marcas no lugar p_1.

N	**2**	**3**	**4**	**5**	**6**	**7**	**8**
T	20	50	105	196	336	540	825
V	18	60	150	315	588	1008	1620
T+V	38	110	255	511	924	1548	2445

CLINTHULL, que parecem ser implantações do mesmo algoritmo. Infelizmente, os códigos fontes das funções VERENCLINTHULL e VERINTHULL não foram fornecidas (é uma mistura de arquivo-M e código-P do *framework* MATLAB *toolbox* INTLAB) para uma verificação mais detalhada.

Tabela 6.17 Tempos médios para solução dos sistemas de equações lineares intervalares (estas medidas foram obtidas através de um P4 2.4 GHz 256 MB RAM / Windows XP). A família de sistemas paralelos do modelo ISPN tem um número de equações lineares intervalares que variam de 20 a 825.

Função	**Tempo decorrido(s)**[5]				
	T = marcações tangíveis				
	20	**105**	**196**	**336**	**825**
VERIFYLSS[1]	$0,005\pm$ $0,005$	$0,031\pm$ $0,008$	$0,141\pm$ $0,042$	$0,419\pm$ $0,021$	$7,539\pm$ $1,853$
VERENCLINTHULL[2]	$0,181\pm$ $0,019$	$1,419\pm$ $0,054$	$6,656\pm$ $0,777$	$28,639\pm$ $0,730$	$4,223E+02\pm$ $1,03E+01$
VERINTHULL[2]	$0,414\pm$ $0,006$	$60,675\pm$ $1,027$	$1,939E+03\pm$ $5,19E+01$	$> 12\,h$	$> 12\,h$
HSOLVE[3]	$0,019\pm$ $0,039$	$0,027\pm$ $0,006$	$0,128\pm$ $0,027$	$0,444\pm$ $0,035$	$10,849\pm$ $1,522$
MKRAW	$0,017\pm$ $0,004$	$0,055\pm$ $0,008$	$0,160\pm$ $0,010$	$0,608\pm$ $0,030$	$5,152\pm$ $0,039$
INTGAUSSP[4]	$0,798\pm$ $0,006$	$25,246\pm$ $0,383$	$1,089E+02\pm$ $4,35E+00$	$5,502E+02\pm$ $1,17E+00$	$1,240E+04\pm$ $3,93E+01$

1 - ***Home page*** **de Siegfried M. Rump: [142]**.
2 - ***Home page*** **de Jiri Rohn: [137]**.
3 - ***Home page*** **de Gareth I. Hargreaves: [53]**.
4 - **Adaptado de (3) acima** .
5 - **Intervalo de Confiança de 95% (t-student)**.
Obs : As medidas foram realizadas no PC (10 amostras por resultado apresentado), *i.e.*, não foram controlados os processos embora a máquina não estivesse em rede.

A Tabela 6.18 descreve os efeitos dos parâmetros intervalares na medida em que o número das equações do sistema intervalar aumenta (igual ao número de marcações tangíveis do modelo ICTMC subjacente a ISPN). Os resultados (espessura dos intervalos) obtidos por cada função usada na solução dos sistemas intervalares foram muito similares para o grupo das funções usadas neste estudo de caso.

6.4 Estudo de caso 4: A aritmética Kaucher na análise ISPN

Este estudo de caso analisa os resultados de um conjunto de algoritmos que usam a aritmética de Kaucher e um algoritmo que usa aritmética intervalar clássica para resolver sistemas de equações lineares intervalares para a solução do estado estacionário. A análise leva em consideração os efeitos das variabilidades nas taxas das transições exponenciais e nos pesos das transições imediatas ao calcular índices do desempenho. Os exemplos deste estudo de caso foram conduzidos com uma comparação dos algoritmos.

Nesta seção um conjunto de códigos executáveis (SUBDIFF, RESPLIT, ALOUTER0 e ALOUTER2) [143] que usam a aritmética de Kaucher é avaliado. A função MKRAW do MATLAB **toolbox** INTLAB, que usa a aritmética intervalar clássica, é usada para analisar a

Tabela 6.18 Avaliação de Desempenho para o sistema paralelo com 196 e 825 marcações tangíveis. Métricas: **NP**$_1$ = número médio de marcas no lugar P_1, **System throughput** = $Prob\left\{(m(P_9 > 0) \cdot t_{I/O}\right\}$
.

Função	**NP$_1$ e System throughput**	
	T = marcações tangíveis	
	196	**825**
VERIFYLSS[1]	$< 3,37218634785193;$ $0,04010364791493 >$ $< 0,67444122568113;$ $0,01991544118533 >$	$< 4,49775685335994;$ $0,14960261938225 >$ $< 0,89956237993883;$ $0,05522624449375 >$
VERENCLINTHULL[2]	$< 3,37222978038842;$ $0,04000713228195 >$ $< 0,67444687057441;$ $0,01988753329495 >$	$< 4,49792321561811;$ $0,14895865922024 >$ $< 0,89958092824098;$ $0,05503739011690 >$
VERINTHULL[2]	$< 3,37218671189105;$ $0,03933690766144 >$ $< 0,67444101116486;$ $0,01960977054537 >$	-
HSOLVE[3]	$< 3,37222978038857;$ $0,04000713228667 >$ $< 0,67444687057445;$ $0,01988753330621 >$	$< 4,49792321561912;$ $0,14895865937647 >$ $< 0,89958092824128;$ $0,05503739036024 >$
MKRAW	$< 3,37245777796249;$ $0,03977913470763 >$ $< 0,67734694762011;$ $0,01698745624921 >$	$< 4,50117216864740;$ $0,14570970618966 >$ $< 0,90922513818705;$ $0,04539318017058 >$
INTGAUSSP[4]	$< 3,37222937457557;$ $0,04000753809456 >$ $< 0,67444686136312;$ $0,01988754250620 >$	$< 4,49792140100313;$ $0,14896047383395 >$ $< 0,89958086009464;$ $0,05503745826300 >$

1 - ***Home page*** **de Siegfried M. Rump: [142]**.
2 - ***Home page*** **de Jiri Rohn: [137]** .
3 - ***Home page*** **de Gareth I. Hargreaves: [53]** .
4 - **Adaptado de (3) acima**.

qualidade relativa do intervalo resultante das métricas. Os sistemas são os mesmos da análise do estado estacionário de modelo ISPN do "sistema paralelo" (ver Figura 6.1). O estudo comparativo observa dois aspectos importantes: 1) o tempo da solução do sistema usando uma dada função; 2) a qualidade das probabilidades intervalares resultantes. Este aspecto é avaliado através das espessuras dos intervalos dos índices do desempenho (intervalo mais estreito é o melhor).

O estudo apresentado diz respeito à solução dos sistemas que variam 20 a 825 equações lineares intervalares. O objetivo foi avaliar tempos de execução, a medida que a dimensão do sistema aumenta, assim como para analisar a qualidade dos resultados em função da largura do intervalo das métricas obtidas usando cada um dos algoritmos.

6.4.1 "Sistema paralelo"

O modelo ISPN denominado "sistema paralelo" (Figura 6.1) fornece uma família de ICTMC subjacente com espaços de estado diferentes, o que possibilita avaliar o impacto da dimensão do espaço de estado na qualidade métricas intervalares com também o tempo de computação relacionado a cada algoritmo considerado para resolver o sistema de equações lineares intervalares.

A Tabela 6.1 apresenta os parâmetros intervalares para o " sistema paralelo". A marcação inicial para o lugar denotado por p_1 identifica os membros da família de ICTMC. As outras marcações dos lugares remanescentes permanecem inalteradas para cada marcação inicial.

A Tabela 6.16 exibe a cardinalidade do espaço de estados da família de modelos em função da marcação inicial em p_1 (veja Figura 6.1).

A Tabela 6.19 apresenta o tempo médio de execução para o cálculo da solução para cada função particular que foi considerada para resolver o sistema de equações lineares intervalares de cada instância particular do modelo ISPN.

Tabela 6.19 Tempos médios para solução dos sistemas de equações lineares intervalares (estas medidas foram obtidas através de um P4 2.4 GHz 256 MB RAM / Windows XP). A família de sistemas paralelos do modelo ISPN tem um número de equações lineares intervalares que variam de 20 a 825.

	Tempo decorrido(s)[3]				
	Código executável / Função				
T	SUBDIFF[1]	RE_SPLIT[1]	ALOUTER0[1]	ALOUTER2[1]	MKRAW[2]
20	$(0,206\pm 0,005)\cdot 10^{-3}$	$(0,515\pm 0,009)\cdot 10^{-3}$	$(0,203\pm 0,011)\cdot 10^{-3}$	$(0,228\pm 0,006)\cdot 10^{-3}$	$(31,70\pm 0,830)\cdot 10^{-3}$
50	$(1,710\pm 0,008)\cdot 10^{-3}$	$(3,880\pm 0,129)\cdot 10^{-3}$	$(1,680\pm 0,018)\cdot 10^{-3}$	$(2,575\pm 0,007)\cdot 10^{-3}$	$(43,65\pm 0,386)\cdot 10^{-3}$
105	$(11,878\pm 0,064)\cdot 10^{-3}$	$(27,077\pm 0,085)\cdot 10^{-3}$	$(11,703\pm 0,071)\cdot 10^{-3}$	$(26,385\pm 0,016)\cdot 10^{-3}$	$(103,94\pm 1,745)\cdot 10^{-3}$
196	$(8,167\pm 0,039)\cdot 10^{-2}$	$(17,327\pm 0,016)\cdot 10^{-2}$	$(8,030\pm 0,124)\cdot 10^{-2}$	$(19,019\pm 0,013)\cdot 10^{-2}$	$(34,50\pm 0,773)\cdot 10^{-2}$
336	$(4,859\pm 0,009)\cdot 10^{-1}$	$(1,744\pm 0,009)\cdot 10^{-1}$	$(4,722\pm 0,024)\cdot 10^{-1}$	$(15,289\pm 0,012)\cdot 10^{-1}$	$(15,59\pm 0,236)\cdot 10^{-1}$
540	$5,683\pm 0,005$	$3,906\pm 0,034$	$5,467\pm 0,082$	$22,216\pm 0,152$	$4,84\pm 0,114$
825	$21,892\pm 0,024$	$14,781\pm 0,047$	$21,236\pm 0,048$	$95,283\pm 0,025$	$14,62\pm 0,973$

1 - ***Home page*** **de Sergey P. Shary: [143]**.
2 - **A implementação usou o MATLAB *toolbox* INTALAB** .
3 - **Intervalo de Confiança de 95% (t-student)** .
Obs : As medidas foram realizadas no PC (10 amostras por resultado apresentado), *i.e.*, não foram controlados os processos embora a máquina não estivesse em rede.

Os resultados das medidas de tempo devem ser analisados linha por linha porque as potencias 10^{-3}, 10^{-2}, 10^{-1} indicam que os resultados foram divididos por estes valores que correspondem a medidas realizadas por uma estrutura iterativa de $1000, 100, 10$ repetições, re-

spectivamente. Então, em cada linha, o mesmo *overhead* é esperado na medida dos tempos. No caso da função MKRAW, a observação sobre *overhead* não se aplica da mesma maneira porque ela é uma função do MATLAB *toolbox* INTLAB (as outras funções são códigos C++ compilados).

Os códigos executáveis SUBDIFF, RE_SPLIT, ALOUTER0 e ALOUTER2 têm o melhor tempo de solução para os sistemas menores que < 200 *equações*. Para 825 equações os tempos são melhores para a função MKRAW e para o código executável RE_SPLIT quando comparados aos demais.

Os códigos executáveis SUBDIFF, ALOUTER0 e ALOUTER2 (os algoritmos implementados pelos códigos SUBDIFF, RE_SPLIT, ALOUTER0 e ALOUTER2, resolvem um sistema de equações lineares em cada passo iterativo) usam a eliminação gaussiana para soluções do sistema em cada passo iterativo. Este fato reduz a aplicabilidade destes códigos a sistemas maiores. Já os códigos RE_SPLIT e MKRAW usam métodos iterativos em cada passo iterativo e, por isso, podem ser aplicados em soluções do sistema da dimensão maior.

O código RE_SPLIT (como também os demais) é um arquivo executável (gerado de um código fonte C++) e o MKRAW é interpretado pelo (*framework* do MATLAB *toolbox* INTLAB). O código RE_SPLIT adota a aritmética Kaucher no $\mathbb{KF}$ e a função MKRAW é implantada considerando a aritmética intervalar clássica no $I(\mathbb{F})$. As operações da multiplicação no $\mathbb{KF}$ usa uma tabela de 16 entradas e 9 entradas no $I(\mathbb{F})$. Este fator é desfavorável para o desempenho dos algoritmos em $\mathbb{KF}$ ao serem comparados com os algoritmos em $I(\mathbb{F})$.

As Tabelas 6.20 e 6.21 descrevem os efeitos da propagação, relacionados aos parâmetros intervalares, na medida em que o número das equações do sistema aumenta (igual ao número de marcações tangíveis de cada ICTMC subjacente ao modelo de ISPN). A qualidade relativa dos resultados fornecida por cada código executável/função é muito similar para o grupo de funções/código de cada tabela, respectivamente.

Os dados fornecidos para o fator de arraste[1] (recomendado ser igual ou perto de 1 [145]), para o defeito requerido da solução aproximada e para limitação no número da iteração foram o mesmo $(1.\ ,\ 1.E-9\ ,\ 100)$, respectivamente, para cada sistema intervalar resolvido com os códigos executáveis (SUBDIFF, RE_SPLIT, ALOUTER0, ALOUTER2).

Na Tabela 6.20 vemos que os resultados de SUBDIFF e de RE_SPLIT diferem somente, no pior caso, nos três últimos dígitos (acima do erro requerido da aproximação $= 1.E-9$). Além disto, todos os resultados apresentados na Tabela 6.20 têm espessuras muito próximas (dentro do erro de aproximação requerido).

A Tabela 6.21 apresenta os resultados do código executável ALOUTER2 e da função de MKRAW espessuras próximas. Deve ser destacado os intervalos muito menores produzidos por SUBDIFF, RE_SPLIT e por ALOUTER0 se comparado com aqueles produzidos por ALOUTER2 e por MKRAW para este estudo de caso variam de 20 a 825.

[1] $\tau \in [0;\ 1]$ nos algoritmo para o método subdiferencial.

Tabela 6.20 Avaliação de Desempenho para o sistema paralelo com marcações tangíveis variando de 20 a 825. Métricas: $\mathbf{NP}_1$ = número médio de marcas no lugar $P_1 = [\mathrm{E}]\{\#P_1\}$, **System throughput** = $= [\mathrm{prob}]\{\#\mathrm{P}_9 > 0\} \cdot [\lambda]_{I/O}$.

	$\mathbf{NP}_1$ e System throughput		
	Código Executável[1]		
T	**SUBDIFF**	**RE_SPLIT**	**ALOUTER0**
20	$< 1,50555047442717;$ $0,00070311561836 >$ $< 0,30111024206477;$ $0,00037177704207 >$	$< 1,50555047442753;$ $0,00070311561762 >$ $< 0,30111024206478;$ $0,00037177704202 >$	$< 1,50555047420520;$ $0,00070821825648 >$ $< 0,30111024279409;$ $0,00037281269421 >$
50	$< 2,19169792344089;$ $0,00170095235629 >$ $< 0,43833982492257;$ $0,00063348988506 >$	$< 2,19169792344126;$ $0,00170095235509 >$ $< 0,43833982492258;$ $0,00063348988501 >$	$< 2,19169792384633;$ $0,00168476918413 >$ $< 0,43833982289306;$ $0,00063016245381 >$
105	$< 2,81824382392545;$ $0,00313350326918 >$ $< 0,56364910495348;$ $0,00096552165487 >$	$< 2,81824382392544;$ $0,00313350326919 >$ $< 0,56364910495348;$ $0,00096552165487 >$	$< 2,81824382367298;$ $0,00317115191693 >$ $< 0,56364910913050;$ $0,00097335040774 >$
196	$< 3,37218591117944;$ $0,00505644796393 >$ $< 0,67444305375222;$ $0,00138446018923 >$	$< 3,37218591117944;$ $0,00505644796394 >$ $< 0,67444305375222;$ $0,00138446018923 >$	$< 3,37218591038768;$ $0,00497732734800 >$ $< 0,67444290622015;$ $0,00136798801967 >$
336	$< 3,84125524337818;$ $0,00726307152160 >$ $< 0,76826442361786;$ $0,00185752921308 >$	$< 3,84125524337819;$ $0,00726307152157 >$ $< 0,76826442361787;$ $0,00185752921307 >$	$< 3,84125524713486;$ $0,00741283109504 >$ $< 0,76826469525436;$ $0,00188896578188 >$
540	$< 4,21692556194614;$ $0,00981808126232 >$ $< 0,84341671994622;$ $0,00239360792137 >$	$< 4,21692556194614;$ $0,00981808126229 >$ $< 0,84341671994623;$ $0,00239360792136 >$	$< 4,21692547360049;$ $0,00957638922371 >$ $< 0,84341588559271;$ $0,00234292939601 >$
825	$< 4,49776534577556;$ $0,01206334764371 >$ $< 0,89960547501813;$ $0,00286180929165 >$	$< 4,49776534577554;$ $0,01206334764376 >$ $< 0,89960547501813;$ $0,00286180929166 >$	$< 4,49776570888044;$ $0,01242688057833 >$ $< 0,89960730181532;$ $0,00293787709699 >$

1 - ***Home page*** **de S. P. Shary [143]** versões adaptadas e compiladas com Dev-C++.

Tabela 6.21 Sistemas de equações lineares intervalares resolvidas por algoritmos usando $\mathbb{KR}$ e $I(\mathbb{R})$. Intervalos da média de espera por processamento e da produtividade do sistema para o modelo ISPN do "sistema paralelo" com marcações iniciais diferentes no lugar p_1 (Figura 6.1). O **T** é o número de *marcações tangíveis = número das equações*.

	Média de espera por processamento[1] e Produtividade do sistema[2]	
T	**ALOUTER2[3]**	**MKRAW[4]**
20	$< 1,50554675660367;$ $0,00291300479727 >$ $< 0,30111022594396;$ $0,00223982684302 >$	$< 1,50555067214211;$ $0,00294938007694 >$ $< 0,30111056223598;$ $0,00219925983362 >$
50	$< 2,19168205369629;$ $0,00799139041819 >$ $< 0,43897922941466;$ $0,00522494531007 >$	$< 2,19169821603979;$ $0,00935524444551 >$ $< 0,43898514705328;$ $0,00544240878998 >$
105	$< 2,81825356701818;$ $0,01652099355804 >$ $< 0,56513674302849;$ $0,00954121489173 >$	$< 2,81829920363226;$ $0,02110840244247 >$ $< 0,56515862442343;$ $0,01046343819049 >$
196	$< 3,37235722462879;$ $0,03044495756004 >$ $< 0,67730396115115;$ $0,01502127349220 >$	$< 3,37245777796249;$ $0,03977913470763 >$ $< 0,67734694762011;$ $0,01698745624921 >$
336	$< 3,84194502662544;$ $0,05152126411635 >$ $< 0,77307596649715;$ $0,02148203411456 >$	$< 3,84215095235479;$ $0,06641853211838 >$ $< 0,77317248517291;$ $0,02485545767369 >$
540	$< 4,21845721498026;$ $0,07955472322269 >$ $< 0,85039521122917;$ $0,02944973760655 >$	$< 4,21884906822972;$ $0,10173395680436 >$ $< 0,85057844830081;$ $0,03436226526832 >$
825	$< 4,50047500628600;$ $0,11553164263253 >$ $< 0,90891310647103;$ $0,03919873354308 >$	$< 4,50117216864739;$ $0,14570970618961 >$ $< 0,90922513818704;$ $0,04539318017057 >$

1 - **Média de espera por processamento** $= [\mathrm{E}]\{\#P_1\}$.

2 - **Produtividade do sistema** $= \mathrm{prob}\{\#\mathrm{P}_9 > 0\} \cdot \lambda_{I/O}$.

3 - **Algoritmo usando** $\mathbb{KR}$.

Da *home page* de S. P. Shary [143] adaptado e compilado com Dev-C++.

4 - **Algoritmo usando** $I(\mathbb{R})$.

6.5 Considerações finais

Este capítulo foi dividido em 4 estudos de casos que representam a evolução na modelagem ISPN. A ISPN é usada como ferramenta de alto nível para estudar problemas relacionados a presença de incertezas e variabilidades nos modelos ICTMC subjacentes. O modelo ISPN considera como intervalos as taxas das transições exponenciais e os pesos das transições imediatas. Os intervalos podem ser degenerados sendo, neste caso, equivalentes aos pontos da análise pontual clássica.

O estudo de caso 1 serviu como subproduto final, para obtermos a ferramenta protótipo ISPN que foi usada no presente trabalho. Neste caso foi avaliado o tempo de processamento para as 3 tarefas críticas: a geração do espaço de estado/geração da matriz $[\mathbf{Q}]$; cálculo da matriz inversa $(1-[\mathbf{P}]^{\mathscr{V}\mathscr{V}})^{-1}$ e a solução do sistema linear intervalar. Com os resultados obtidos fizeram-se ajustes na ferramenta protótipo ISPN. Com o estudo de caso 1 constatou-se que a performance de análise ferramenta protótipo ISPN não foi influenciada pelo sistema operacional Linux/Windows XP e nem também pela quantidade de RAM presente, mas, foi influenciada pela frequência do *clock* (da máquina).

O estudo de caso 2 é uma aplicação na avaliação da dependabilidade. Dois multiplexadores foram utilizados para o estudo: um multiplexador ADM e um multiplexador SDH. Primeiro foram considerados somente intervalos degenerados. Os resultados das métricas (dependabilidade) têm garantia da inclusão dos valores pontuais reais através de limites, inferiores e superiores, verificados. Em seguida, se considerou que o tempo de reparo (MTTR) pertencentes aos intervalos $[7,4;\ 8,6]$ e $[7,8;\ 8,2]$ em horas e obteve-se os intervalos de disponibilidade.

No estudo de caso 3 os métodos intervalares foram aplicados na estimativa do conjunto envoltório da solução dos sistemas de equações lineares intervalares. Este estudo de caso foi conduzido através da avaliação dos resultados relacionados a um conjunto significativo de algoritmos para solução de sistemas de equações lineares intervalares aplicadas à solução do estado estacionário dos modelos ISPN aplicados na avaliação de desempenho. Estes algoritmos foram implantados usando o *framework* MATLAB *toolbox* INTLAB. Todos algoritmos usam aritmética intervalar clássica com arredondamentos direcionados para o exterior. Uma consequência do estudo de caso 3 é que a função MKRAW, especificamente projetada para ser usada na análise do estado estacionário dos modelos ISPN, passou a ser mais usada.

Finalmente, o estudo de caso 4 é dedicado à exploração do conjunto de algoritmos que usam a aritmética Kaucher para encontrar uma estimativa do conjunto solução unida de sistemas lineares intervalares associados à análise do estado estacionário dos modelos ISPN. O conjunto de códigos executáveis (SUBDIFF, RESPLIT, ALOUTER0 e ALOUTER2) [143] usam a aritmética de Kaucher. A função MKRAW (que usa a aritmética intervalar clássica) é usada para comparação de resultados. Os códigos executáveis SUBDIFF, RE_SPLIT, ALOUTER0 e ALOUTER2 têm o melhor tempo de solução para os sistemas < 200 *equações*. Para o sistema com 825 equações os tempos para se encontrar a solução são melhores para a função MKRAW e para o código executável RE_SPLIT. Deve ser destacado os intervalos muito menores produzidos por SUBDIFF, RE_SPLIT e por ALOUTER0 se comparado com aqueles produzidos

por ALOUTER2 e por MKRAW para este estudo de caso. Este fato deve ser visto com cautela, pois o aritmética Kaucher não foi implementada com arredondamento direcionados para o exterior.

A análise preditiva de confiabilidade e disponibilidade são bons exemplos de uso potencial da análise ISPN. Neste caso é observado uma grande variabilidade nos parametros, por exemplo, MTBF (mean time between failure) numa longa lista de padrões aceitos: MIL-217, Telcordia(Bellcore), NSWC, China 299b, RDF, etc.

CAPÍTULO 7

Conclusão e Trabalhos Futuros

Este trabalho apresentou a proposta do formalismo ISPN para análise quantitativa de sistemas. ISPN é definida como um espaço de modelos de rede de Petri estocásticas (modelos intervalares), na qual o tempo (taxas) e pesos são representados por intervalos. Desenvolveu-se a fundamentação para modelos markovianos intervalares subjacentes a ISPN. Foram exploradas alternativas de solução através da adaptação dos algoritmos de análise estacionária das cadeias de Markov à análise intervalar. Aplicou-se o modelo proposto, através dos métodos de análise do estado estacionário, em problemas para avaliação do desempenho e de dependabilidade. O formalismo ISPN é principalmente aplicado para modelar situações nas quais dados de entrada são conhecidos com certo nível de precisão. As taxas e os pesos foram representados através de intervalos. Este *framework* provê meios para formalização e estudo de problemas relacionados à presença de incertezas que podem incluir erros de dados ocorridos durante os processos de medida e erros de arredondamento gerados durante cálculos.

O modelo proposto e os métodos de análises associados permitem que se considere variações simultânea de parâmetros. As incertezas associadas aos parâmetros dos modelos são comuns nas fases iniciais de projetos de sistemas.

As principais contribuições incluem: proposição e descrição do formalismo ISPN; desenvolvimento e adaptação de métodos para análise de estado estacionário dos modelos markovianos intervalares subjacentes ao formalismo ISPN. Este resultado foi alcançado através da adaptação de algoritmos numéricos à aritmética intervalar. Uma terceira contribuição foi o desenvolvimento de uma ferramenta protótipo para analisar modelos ISPN. Esta ferramenta gera o espaço de estados associado ao modelo ISPN, elimina as marcações voláteis, calcula as probabilidade intervalares estacionárias e os obtém os intervalos associados às métricas de desempenho e dependabilidade.

O Capítulo 4, ISPN, apresentou o modelo, sua fundamentação e o processo de análise. Os estudos de caso tiveram o objetivo de apresentar de forma clara as fases do processo de análise ISPN.

A análise do estado estacionário dos modelos ISPN depende da solução de um sistema de equações lineares intervalares. A solução de um sistema de equações lineares é um problema *NP-hard*. No entanto, é possível encontrar cercos (vetor intervalar contendo a solução exata relacionada ao USS) em tempo polinomial. O Capítulo 5 apresentou o conjunto de métodos que foram explorados e implementados no protótipo computacional para análise do estado estacionário do formalismo ISPN. Os algoritmos apresentados usam a aritméticas intervalares Clássica e de Kaucher. Muitos algoritmos foram adaptados para possibilitar a integração com a ferramenta protótipo ISPN.

O Capítulo 6 apresentou 4 estudos de casos, cujo objetivo amplo foi explorar a capacidade

de representação do modelo ISPN. Além deste objetivo, os estudos de caso tiveram também propósitos específicos. Na Seção 6.1 o objetivo específico foi avaliar os gargalos da ferramenta protótipo ISPN, que utiliza o *framework* MATLAB *toolbox* INTLAB. A Seção 6.2 foi um estudo para análise da disponibilidade. Neste caso, consideraram-se alguns parâmetros intervalares e os respectivos valores intervalares de disponibilidade foram obtidos e apresentados. A Seção 6.3 avaliou vários métodos usados na solução dos sistemas de equações intervalares subjacentes aos modelos ISPN para determinação do vetor *p.m.f.* intervalar do estado estacionário. O estudo comparativo das funções do MATLAB *toolbox* INTLAB observam dois aspectos importantes: 1) o tempo da solução do sistema; 2) a qualidade dos resultados. Este aspecto é avaliado através das espessuras dos intervalos dos índices de desempenho (intervalos mais estreito s são melhores). Finalmente, a Seção 6.4 investigou a aplicação da aritmética Kaucher, através da ferramenta protótipo ISPN, para avaliar o potencial desta aritmética para a análise ISPN. Portanto, o Capítulo 6 explorou um conjunto de soluções para análise, capacidade de representação do modelo concebido e da ferramenta protótipo ISPN.

Vários aspectos conceituais e teóricos são apresentados nos apêndices. O Apêndice A é um esforço para qualificar os modelos intervalares em uma classificação mais ampla de modelos. O Apêndice B apresenta alguns conceitos teóricos dos processos estocásticos e cadeias de Markov. O Apêndice C apresenta a função de inclusão intervalar, uma ferramenta fundamental para a análise intervalar, que abre inúmeras possibilidades para desenvolvimento de trabalhos futuros, como por exemplo, otimização global. O Apêndice D considera elementos da teoria de medidas para fundamentar o paradigma de a probabilidade intervalar, para suportar o arcabouço axiomático da teoria das probabilidades. Finalmente, o Apêndice E apresenta a extensão intervalar ICTMC e a deriva as equações do estado estacionário e do estado transiente, subjacentes aos modelos ISPN.

Os trabalhos [33, 34, 35, 36, 37] foram publicados durante o desenvolvimento do formalismo ISPN e apresentam a definição do modelo, os métodos adaptados para solução de estado estacionário do formalismo proposto e demonstram aplicações do modelo ISPN em estudos de casos, cujo objetivo foi a análise de desempenho e de dependabilidade, assim como permitiram apresentar um estudo comparativo dos resultados da análise intervalar como os providos através de ferramentas que suportam a análise de GSPN.

Este trabalho é ponto de partida para muitos outros. Um dos desafios é o desenvolvimento da ferramenta ISPN. No âmbito da análise no estado estacionário é possível explorar a alternativa para eliminação das marcações voláteis através da adaptação do algoritmo de eliminação "on the fly" utilizados na análise GSPN [10]. A classe de métodos de solução de sistemas lineares intervalares que são propostos para encontrar a solução exata dos limites da solução USS (geralmente, com complexidade exponencial) não foram ainda avaliados no *framework ISPN*. Esta linha de trabalhos tem por objetivo a busca por soluções para modelos de grandes dimensões ao mesmo tempo em que se procura conjuntos solução com intervalos estreitos.

Uma linha de trabalho importante é o estudo de simulações estocásticas intervalares. Não foram encontrados trabalhos significativos que apliquem simulação estocástica a modelos intervalares. A simulação desempenha papel destacado dentre as opções de avaliação das GSPN, pois é a alternativa para avaliação de modelos intratáveis através de métodos numéricos (explosão de estado). A análise no estado transiente não foi explorada neste trabalho. Uma

pesquisa interessante será o desenvolvimento de um método específico para análise intervalar no estado transiente. O desenvolvimento do método de Jensen [61, 10] na versão intervalar será um bom começo.

7.1 Observações para a conclusão e trabalhos futuros

A principais contribuições desta tese foram:

- O formalismo ISPN (Interval Stochastic Petri Net) usado na:

- análise quantitativa de sistemas variações simultânea de parâmetros,

- análise com resultados verificados;

- O desenvolvimento dos algoritmos intervalares dos modelos ISPN;
- A avaliação dos métodos de solução e aritméticas intervalares da análise do estado estacionário;
- O desenvolvimento de uma ferramenta-protótipo ISPN;
- Os resultados intervalares de avaliação de desempenho e dependabilidade.

Como possíveis trabalhos futuros têm-se:

- Desenvolvimento da ferramenta ISPN;
- Eliminação das marcações voláteis através do algoritmo de eliminação “on the fl”;
- Métodos de solução de sistemas lineares intervalares que não foram ainda avaliados no framework ISPN;
- Busca soluções para modelos de grandes dimensões e conjuntos solução com intervalos estreitos;
- Simulação estocástica para modelos intervalares;
- Análise no estado transiente (método de Jensen intervalar);
- Integração entre ISPN + (I)RDB (Reliability Block Diagram) + (I)ADB (Availability Block Diagram) + (I)FT (Fault Tree)

.

Referências Bibliográficas

[1] T. Augustin: On the suboptimality of the generalized bayes rule and robust bayesian procedures from the decision theoretic point of view: A cautionary note on updating imprecise priors. ISIPTA'03 (3rd International Symposium on Imprecise Probabilities and Their Applications) , pp. 31–45, (2003)

[2] G. Balbo: Introduction to Stochastic Petri Nets. In: Lectures on formal methods and performance analysis: revised lectures. First EEF Summer School on Trends in Computer Science. Berg en Dal, The Netherlands, July 3-7, 2000. Springer , pp. 84–155, (2001)

[3] A. Bertoni & M. Torelli: Probabilistic Petri nets and semi Markov processes. In Proceedings 2-nd European Workshop on Petri Nets, (1981)

[4] M. Berz & K. Makino: Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. Reliable Computing, **4**(4), pp. 361–369, (1998)

[5] U. N. Bhat: Elements of applied sthochastic processes. John Wiley and Sons, Inc. New York, Chichester, Brisbane, Toronto, Singapore(1984)

[6] C. Bliek: Computer Methods for Design Automation. PhD thesis, Massachusetts Institute of Technology, August, (1992)

[7] A. Bobio & M. Telek: Transient analysis of a preemptive resume M/D/1/2/2 through Petri nets. Technical report, Department of telecommunications - Technical University of Budapest, (April 1994)

[8] A. Bobio & M. Telek: Markov regenerative SPN with non-overlapping activity cycles. In International Computer Performance and Dependability Symposium - IPD95, IEEE Computer Society Press, pp. 124–133, (1995)

[9] Ed Brinksma, editor: Lectures on formal methods and performance analysis: revised lectures/ First EEF Summer School on Trends in Computer Science. Berg en Dal, The Netherlands, July 3 – 7, 2000, 9ed. , Springer (2001)

[10] G. Bolch; S. Greiner; K.S. Trivedi: Queueing networks and Markov chains: modeling and performance evaluation with computer science applications. A Wiley-Inerscience publication, John Wiley and Sons, Inc, Second Edition, (2006)

[11] J.P. Buzen: A Modeler's View of Workload Characterization. Workload Characterization of Computer Systems and Computer Networks, pp. 67-72, North-Holland, (1986)

[12] C.G. Cassandras & S. Lafortune: Introduction to Discrete Event Systems. Kluwer Academic Publishers, Boston, Dordrecht, London 828 p. (1999)

[13] F.E. Cellier: Continuous System Modelling. Springer-Verlag, (1991)

[14] G. Chabert & A. Goldsztejn Extension of the Hansen-Bliek Method to Right-Quantified Linear Systems Reliable Computing, **13**(4), pp. 325–349, (2007)

[15] G. Chabert: Techniques d'intervalles pour la résolution de systèmes d'équations. Thèse de doctorat, Université de Nice-Sophia Antipolis, (2007)

[16] H.M. Chen & M.H. van Emden: Adding interval constraints to the Moore Skelboe global optimization algorithm. In V. Kreinovich, editor. Extended Abstracts of APIC'95, International Workshop on Applications of Interval Computations, pages 54–57, Reliable Computing (Supplement), (1995)

[17] H. Choi, V.G. Kulkarni & K. Trivedi: Transient analysis of deterministic and sthochastic Petri nets. In Proceedings of the 14-th International Conference on Application and Theory of Petri Nets, Chicago, June, (1993)

[18] H. Choi, V.G. Kulkarni & K. Trivedi: Markov regenerative sthocastic Petri nets. Performance Evaluation, 20:337–357, (1994)

[19] W.M. Chow, E.A. McNair, & C.H. Sauer: Analysis of manufacturing systems by Research Queueing Package. IBM Journal of Research and Development, 29:330–341 (1985)

[20] G. Ciardo & C. Lindemann: Analysis of deterministic and stochastic Petri nets. In Proceedings International Workshop on Petri Nets and Performance Models - PNPM93, pages 160-169. IEEE Computer Society (1993)

[21] COPRIN project home page: **http://www-sop.inria.fr/coprin/** (08 September 2008)

[22] S. Corsaro & M. Marino: Interval Linear Systems: the State of the Art. Computational Statistics, 21:365–383,(2006)

[23] J.A. Couvillon, R. Freire, R. Johnson, W.D. Obal, M.A. Qureshi, M. Rai, W. Sanders & J.E. Tvedt: Performability modeling with UltraSAN. IEEE Software, 8:69–80, September, (1991)

[24] A. Cumani: Esp - A package for the evaluation of stochastic Petri nets with phase-type distributed transition times. In: Proceedings International Workshop Timed Petri Nets, IEEE Computer Society Press no. 674, Torino (Italy) p. 144–151, (1985)

[25] P. Dauphin, F. Hartleb, M. Kienow, V. Mertsiotakis, & A. Quick: PEPP: Performance Evaluation of Parallel Programs. User's Guide - version 3.3,"Technical report 17/93, IMMD VII, Univ. of Erlagen-Nürnberg, Germany, Sept. (1993)

[26] D. Denneberg: Non-Additive Measure and Integral, Kluwer Academic Publishers, Dordrecht, (1994)

[27] A.A. Desrochers & R.Y. Al-Jaar: Applications of Petri nets in manufacturing systems: modeling, control, and performance analysis. IEEE Press, Piscataway, NJ, USA (1994)

[28] D. Dubois et al: Fuzzy interval analysis, in: D. Dubois, H. Prade (Eds.), The Handbook of Fuzzy Sets: Volume 1 Fundamentals of Fuzzy Sets, Kluwer Academic Publishers, Dordrecht, 483-581 (2000)

[29] D. Dubois, H. Prade & R.R. Yager (Eds.): Readings in Fuzzy Sets for Intelligent Systems. Morgan Kaufmann Publishers, (1993)

[30] D. Dubois & H. Prade: Random sets and fuzzy interval analysis. Fuzzy Sets and Systems 42:87-101 (1991)

[31] D. Dubois & H. Prade: Possibility Theory. Plenum Press, New York, NY, (1988)

[32] J.B. Dugan, K.S. Trivedi, R.M. Geist & V.F. Nicola: Extended stochastic Petri nets: Applications and analysis. In Proc. PERFORMANCE '84, Paris, France, December, (1984)

[33] S.M.L. Galdino & P.R.M. MACIEL: Interval Generalized Stochastic Petri Net Models in Performance Evaluation In: IEEE International Conference on Systems, Man, and Cybernetics, 2006, Taiwan, pp. 2723-2728 (2006)

[34] S.M.L. Galdino, P.R.M. MACIEL & N.S. Rosa: Interval Generalized Stochastic Petri Net Models in Dependability Evaluation. In: IEEE International Conference on Systems, Man, and Cybernetics, 2007, Montreal, pp. 2502-2507 (2007)

[35] S.M.L. Galdino, P.R.M. MACIEL & N.S. Rosa: Interval Markovian Models in Dependability Evaluation. International Journal of Pure and Applied Mathematics, **41** (2), pp. 151-176 (2007)

[36] S.M.L. GALDINO & P.R.M. MACIEL: Análise Estado Estacionário ISPN. In: VII ERMAC, 2007, Recife. Anais Digitais do VII Encontro Regional de Matemática Aplicada e Computacional, (2007)

[37] S.M.L. Galdino & P.R.M. MACIEL: Outer Estimates of Interval System of Linear Equations: ISPN Models in Dependability Evaluation. In: 2008 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2008), 2008, Singapore. Proceedings of 2008 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2008), 2008.

[38] E. Gardeñes, M. Á. Sainz, L. Jorba, R. Calm, R. Estela, H. Mielgo & A. Trepat: Modal Intervals. Reliable Computing **7** (2), pp. 77-111, (2001)

[39] E. Gardeñes. & A. Trepat: Fundamentals of SIGLA, an Interval Computing System over the Completed Set of Intervals. Computing **24**, pp. 161-179, (1980)

[40] E. Gardeñes, A. Trepat & J.M. Janer: Approaches to Simulation and to the Linear Problem in the SIGLA System. Freiburger Intervall-Berichte, **81** (8), pp. 1-28, (1981)

[41] E. Gardeñes, A. Trepat & J. M. Janer: SIGLA-PL/1 Development and Applications. in: Nickel, K .L. E. (ed.), Interval Mathematics 1980, Academic Press, pp. 301–315, (1980)

[42] E. Gardeñes, A. Trepat & H. Mielgo: Present Perspective of the SIGLA Interval System, Freiburger Intervall-Berichte, **82** (9), pp. 1-65, (1982)

[43] J. Garloff: The Bernstein algorithm. Interval Computations, **2**, pp. 154–168, (1977)

[44] J. Garloff: Application of Bernstein expansion to the solution of control problems. Reliable Computing, **6**(3), pp. 333–320, (2000)

[45] A.J.C. van Gemund: Compile-Time Performance Prediction with PAMELA. Proc. 4th Int. Work shop on Compilers for Parallel Computers, Delft, the Netherlands, pp. 428-435, Dec., (1993)

[46] R. German: Performance Analysis of Communicating Systems - Modeling with Non-Markovian Stochastic Petri Nets. Wiley, (2000)

[47] C. Girault & R. Valk: Petri Nets for Systems Engineering: a guide to modeling, verification, and applications. Springer, (2003)

[48] A. Goldsztejn: A Right-Preconditioning Process for the Formal-Algebraic Approach to Inner and Outer Estimation of AE-solution Sets Reliable Computing,**11**(6), pp. 443–478, (2005)

[49] A. Goldsztejn: Définition et Applications des Extensions des Fonctions Réelles aux Intervalles Généralisés: Nouvelle Formulation de la Théorie des Intervalles Modaux et Nouveaux Résultats, Thèse de doctorat, Université de Nice-Sophia Antipolis, (2005)

[50] GreatSPN home page: **http://www.di.unito.it/∼greatspn/index.html** (06 May 2008)

[51] A. Del Grosso & A. Zucchini: Applications of Interval Arithimetic to the Analysis and Control of Structural Systems. Smart Materials and Structures, **2**, pp. 103-112, (1993)

[52] G.I. Hargreaves: Interval analysis in matlab. Numerical Analysis Report No. 416, University of Manchester, Reports available from: Department of Mathematics **http://www.ma.man.ac.uk/MCCM, http://www.ma.man.ac.uk/∼nareports**, Manchester M13 9PL, England, December (2002)

[53] Gareth I. Hargreaves home page:
http://www.maths.manchester.ac.uk/ hargreaves/downloadint.html (06 January 2008)

[54] F. Hartleb & V. Mertsiotakis: Bounds for the Mean Runtime of Parallel Programs. Proc. Sixth Int. Conf. On Modelling Techniques and Tools for Computer performance Evaluation, pp. 197-210, (1992)

[55] E. R. Hansen & G. W. Walster. *Global Optimization Using Internal Analysis*. Second Edition, Revised and Expanded, Marcel Dekker, Inc., New York, (2004)

[56] E. R. Hansen: Bounding the solution of interval linear equations. SIAM Journal on Numerical Analysis, **29**(5), pp. 1493-1503, October (1992)

[57] P.J. Huber & V. Strassen: Minimax tests and the Neyman-Pearson lemma for capacities. Annals of Statistics, **1**, pp. 251-263, (1973)

[58] E. Hyvönen: Constraint reasoning based on interval arithmethic: the tolerance propagation appproach. Artificial Intelligence, **58**, pp. 71-112, (1992)

[59] C. Jansson: Calculation of Exact Bounds for the Solution Sets of Linear Interval Systems, Linear Algebra and Its Applications **251**, pp. 321-340, (1997)

[60] L. Jaulin, M. Kieffer, O. Didrit, & E. Walter: Applied Interval Analysis: with Examples in Parameter and State Estimation, Robust Control and Robotics. Springer-Verlag, (2001)

[61] A. Jensen: Markoff chains as an aid in the study of Markoff process. Skandinavian Aktuarietidskr, **36**, pp. 87-91, (1953)

[62] E. Kaucher: Algebraische Erweiterungen der Intervallrechnung unter Erhaltung Ordnungs- und Verbandsstrukturen, Computing Supplement, **1** , pp. 65–79, (1977)

[63] E. Kaucher: Interval Analysis in the Extended Interval Space $\mathbb{IR}$, Computing Supplement **2**, pp. 33–49, (1980)

[64] E. Kaucher: ¨Ubermetrische und algebraische Eigenschaften einiger beimnumerischen Rechnen auftretender Räume, Ph.D. dissertation, Universität Karlsruhe, Karlsruhe, (1973)

[65] A. Kaufmann & M.M. Gupta: Fuzzy Mathematical Models in Engineering and Management Science. North Holland, Amsterdam, (1988)

[66] P.R. Kearfott & V. Kreinovich: Applications of interval computations. Kluwer, (1996)

[67] R. Kirchner: Hardware Support for Interval Arithmetic. Reliable Computing 12: 225–237 (2006)

[68] S. Kempken, W. Lüther & G. Hassliner: A tool for verified analysis of transient and steady states of queues. ACM International Conference Proceeding Series; Vol. 201. Proceeding from the 2006 workshop on Tools for solving structured Markov chains, Pisa, Italy Article No. 8, (2006)

[69] L.V. Kolev: Interval Methods for Circuit Analysis. World Scientific Publishing Co., River Edge, New Jersey (1993)

[70] L. V. Kolev: Outer Interval Solution of the Eigenvalue Problem under General Form Parametric Dependencies Reliable Computing, **12**, pp. 193-202, (2006)

[71] L. V. Kolev: Improvement of a Direct Method for Outer solution of Linear Parametric Systems. Reliable Computing, **12**, pp. 121-140 (2006)

[72] V. Kreinovich G. Heindl & A. Lakeyev. Solving linear interval is np-hard even if we exclude overflow and underflow. *Reliable Computing*, **4**, pp. 383–388, 1998.

[73] G.J. Klir & B. Yuan: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice-Hall, PTR, Upper Saddle River, NJ, (1995)

[74] W. Kühn: Rigorously computed orbits of dynamical systems without the wrapping effect. Computing, **61**, 47-67, (1998)

[75] U.W. Kulish & W.L. Miranker: The Arithmetic of the Digital Computers: A New Approach. SIAM Review **28** (1), (1986)

[76] V.G. Kulkarni: Modeling and analysis of sthochastic systems Chapman and Hall. London, Glasgow, Weinheim, New York, Tokio, Melbourn, Madras (1995)

[77] Z. Kulpa: Diagrammatic analysis of interval linear equations. Realiable Computing, **9**, pp. 1-20, (2003)

[78] S. Kumar, W. Grassmann & R. Billington: A Stable Algorithm to Calculate Steady-State Probability and Frequency of a Markov System. IEEE Transactions on Reliability, R-**36**(l), April, (1987)

[79] S.E. Laveuve: Definition einer Kahan-Arithmetic und ihre Implementierung, in: Nickel, K. (ed.), Interval Mathematics, Lecture Notes in Computer Science 29, Springer Verlag, New York, pp. 236-245, (1975)

[80] E.D. Lazowska, J. Zahorjan, G.S. Graham & K.C. Sevcik: Quantitative System Performance - Computer System Analysis Using Queueing Network Models. Prentice-Hall, Inc., Englewood Cliffs, N.J., (1984)

[81] C. Lindemann: An improved numerical algorithm for calculating steady-state solutions of deterministic and stochastic Petri net models. Performance Evaluation, 18:75–95, (1993)

[82] C. Lindemann: DSPNexpress: a software package for the efficient solution of deterministic and stochastic Petri nets. Performance Evaluation, **22**, pp. 3-21, (1995)

[83] W.A. Lodwick & K.D. Jamison: Fuzzy Sets and Systems. **135** (1) , pp. 1-3 , April (2003)

[84] R. Lohner: Enclosing the solutions of ordinary initial and boundary value problems. In: E. Kaucher, U. Kulish and C. Ullrich (eds), Computer Arithmetic: Scientific Computation and Programming Languages, BG Teubner, Stuttgart, Germany, pp. 255-286, (1987)

[85] J. Lüthi & G. Haring: Fuzzy Queueing Network Models of Computing Systems. Proc. 13th UK Worshop on Performance Engineering of Computer and Telecommunication Systems, Ilkley, UK, July (1997)

[86] J. Lüthi, S. Majumdar, G. Kotsis, & G. Haring: Performance Bounds for Distributed Systems with Workload Variabilities & Uncertainties. Parallel Computing, **22**(13), pp. 1789-1806, Feb (1997)

[87] J. Lüthi: Interval Matrices for Bottleneck Analysis of Queueing Network Models with Histogram-Based Parameters. Proc. IEEE Int. Computer Performance & Dependability Symposium (IPDS'98, September 7-9, 1998, Durham, NC, USA), pp. 142-152, IEEE Computer Society Press, Los Alamitos, Sept. (1998)

[88] J. Lüthi & G. Haring: Fuzzy Queueing Network Models of Computing Systems. Proc. 13th UK Workshop on Performance Engineering of Computer and Telecommunication Systems, Ilkley, UK, July (1997)

[89] J. Lüthi & C.M. Lladó: Splitting techniques for interval parameters and their application to performance models Performance Evaluation , **51** (1), pp. 47-74, (2003)

[90] J. Lüthi & G. Haring: Mean Value Analysis for Queueing Network Models with Intervals as Input Parameters. Performance Evaluation, **32** (3), pp. 185–215, April (1998)

[91] M. Lyashko: The optimal solution of an interval system of linear algebraic equations. Reliable Computing, **11**, pp. 105-127, (2005)

[92] S. Majumdar & C. M. Woodside: Robust Bounds and Throughput Guarantees for Closed Multiclass Queueing Networks. Performance Evaluation, **32** (2), pp. 101–136, March (1997)

[93] S. Majumdar: Interval Arithmetic for Performance Analysis of Distributed Computing Systems. Proc. Canadian Conf. On Electrical and Computer Engineering, Quebec, Canada, Sep. (1991)

[94] S. Majumdar; J. Lüthi; G. Haring & R. Ramadoss: Characterization and Analysis of Software and Computer Systems with Uncertainties and Variabilities. In R. Dumke et al, editor, Performance Engineering. LNCS 2047, Springer-Verlag Berlin Heidelberg, pp. 202-221,(2001)

[95] S. Majumdar: Application of Relational Interval Arithmetic to Computer Systems. Proc. MASCOTS'95, pp. 345–351, IEEE Computer Society Press, Jan. (1995)

[96] S. Majumdar; J. E. Neilson; C. M. Woodside & D.C. Petriu: Robust Box Bonds: Network Performance Guarantees for Closed Multiclass Queueing Networks with Minimal Stochastic Assumptions. Proc. Infocom'92, pp. 2006-2016, May (1992)

[97] A. S. Malan, M. Milanese, M. Taragna & J. Garloff: B^3 algorithm for robust performance analysis in presence of mixed parametric and dynamic perturbations. Proceedings of the 31st IEEE Conference on Decision and Control, Tucson, AZ, pp. 128-133, (1992)

[98] M. Mares: Computation Over Fuzzy Quantities, CRC Press, Boca Raton, FL, (1994)

[99] M.A. Marsan, G. Balbo & G. Conte: A class of generalized sthochastic Petri nets for the performance evaluation of multiprocessor systems. ACM Transactions on Computer Systems, **2**, pp. 93–122 (1984)

[100] M.A. Marsan et all: Generalized stochastic Petri nets revisited: random switches and priorities. In Proceedings International Workshop on Petri Nets and Performance Models PNPN87, IEEE Computer Society, pp. 44-53, (1987)

[101] M.A. Marsan & G. Chiola: On Petri nets with deterministic and exponentially distributed firig times. In Lectures Notes in Computer Science,**266**, pp. 132–145, Springer Verlag (1987)

[102] M.A. Marsan: Stochastic Petri nets: an elementary introduction. In: G. rozenberg, ed, Advances in Petri Nets 1989. In Lectures Notes in Computer Science, Springer Verlag (1990)

[103] M.A. Marsan, A. Bobbio & S. Donatelli: Petri Nets in Performance Analysis: An Introduction. In: Lectures on Petri Nets I: Basic Models. Advances in Petri Nets. Wolfgang Reisig & Grzegorg Rozenberg(Eds). Springer, Berlin, pp. 211-256, (1998)

[104] M.A. Marsan et all: The effect of execution policies on semantics and analysis of stochastic Petri nets. IEEE Transactions on Software Engineering, SE **15** , pp. 832-846, (1989)

[105] M.A. Marsan, A. Bobbio, G. Conte & A. Cumani: Performance Analysis of Degradable Multiprocessor Systems using Generalized Stochastic Petri Nets. In: Dist. Proc. Newslett. **6**, pp. 47-54, (1984)

[106] M.A. Marsan et all: Generalized stochastic Petri nets: a definition at the net level and its implications. IEEE Transactions on Software Engineering, **19**, pp. 89-107, (1993)

[107] M.A. Marsan, G. Balbo, G. Conte, S. Donatelli & G. Franceschinis: Modelling with Generalized Stochastic Petri Nets. Wiley Series in Parallel Computing, John Wiley and Sons, (1995)

[108] P.M. Merlin, & D.J. Faber: Recoverability of Communication Protocols: Implications and Theoretical Study. IEEE Transaction on Communication, **24** (9), pp. 1036-1043, September (1976)

[109] J.F. Meyer, A. Movaghar, & W.H. Sanders: Stochastic activity networks: Structure, behavior, and application. In Proc. Inter. Workshop on Timed Petri Nets, Torino, Italy, pp. 106-115, July, (1985)

[110] M.K. Molloy: On the Integration of Delay and Troughput Measures in Distributed Processing Models. Tese PhD, UCLA, Los Angels, CA. , Ph.D. Thesis (1981)

[111] M.K. Molloy: Performnace analysis using stochastic Petri nets. IEEE Transactions on Computers, **31** (9), pp. 913-917, (1982)

[112] R.E. Moore: Interval Analysis. Prentice Hall, Englewood Clifs, NJ, USA (1966)

[113] R. E. Moore: Interval Arithmetic and Automatic Error Analysis in Digital Computing. PhD thesis, Stanford University, October (1962)

[114] R.E. Moore: Methods and applications of interval analysis. Studies in Applied Methemathics (SIAM), (1979)

[115] T. Murata: Petri Nets: Properties, Analysis and Applications, an invited survey paper. Proceedings of the IEEE **77**(4), pp. 541-580, (1989)

[116] S. Natkin: Les reseaux de Petri Sthochastiques et leur Application a l'Evaluation des Systems Informatiques. Tese PhD, CNAM, Paris, France, These de Docteur Ingegneur (1980)

[117] J.D. Noe & G.J. Nutt: Macro e-nets representation of parallel systems. IEEE Transaction on Computers, **31** (9), pp. 718-27, August, (1973)

[118] A. Neumaier: A simple derivation of the Hansen-Bliek-Rohn-Ning-Kearfott enclosure for linear interval equations. Reliable Computing, **5**(2), pp. 131-136, (1999)

[119] A. Neumaier: Interval methods for systems of equations. Cabbridje Univesity Press, Cambridge, (1990)

[120] M.F. Neuts: Matrix Geometric Solutions in Sthochastic Models. Johns Hopkins University Press, Baltimore, (1981)

[121] S. Ning & R. B. Kearfott: A comparison of some methods for solving linear interval equations. SIAM Journal on Numerical Analysis, **34**(4), pp. 1289-1305, August, (1997)

[122] H.T. Nguyen: A note on the extension principle, J. Math. Anal. Appl., **64**, pp. 359-380, (1978)

[123] P.W. Oliveira, T.A. Diverio & D.M. Cláudio: Fundamentos da Matemática Intervalar. SAGRA-Luzzatto, 1a. ed, Porto Alegre: Instituto de Informática da UFRGS, (1997)

[124] H.J. Ortolf: Eine Verallgemeinerung der Intervallarithmetik. Geselschaft fuer Mathematik und Datenverarbeitung, Bonn **11**, pp. 1-71 (1969)

[125] PEPA home page: **http://www.dcs.ed.ac.uk/pepa/** (06 May 2008)

[126] PEPSY-QNS home page: **http://www4.informatik.uni-erlangen.de/Projects/PEPSY/en/** (06 May 2008)

[127] J.L. Peterson: Petri net theory and the modeling of systems. Prentice Hall, Englewood Cliffs, (1981)

[128] C.A. Petri: Kommunikation mit automaten. Technical report, Doctoral Thesis, University of Bonn, 1962. (Available in English as: Communication with automata, Technical Report RADC-TR-65-377, Rome Air Development Center, Griffiss NY, (1966)

[129] L.B. Rall: Computacional Solution of Nonlinear Operator Equations. Wiley, New York, (1969)

[130] C. Ramchandani: Analysis of Asynchronous Concurrent Systems by Timed Petri Nets. PhD Thesis, MIT, Cambridge, MA, (1974)

[131] C. Ramchandani; Analysis of Asynchronous Concurrent Systems by Timed Petri Nets. Cambridge, Mass.: MIT, Project MAC, Technical Report 120, February, (1974)

[132] S. S. Rao & L. Berke: Analysis of Uncertain Structural Systems Using Interval Analysis. AiAA Journal, **35**(4), pp 727–735 (1997)

[133] H. Ratscheck & J. Rokne: Computer methods for the range of functions. Ellis Horwood, (1984)

[134] A. Reibman, R. Smith & K. Trivedi: Markov and markov reward model transient analisys: an overview of numerical approaches. European Journal of Operational Research, **40**, pp. 257-267, (1989)

[135] W. Reisig: Petri nets - An introduction. Springer-Verlag, (1982)

[136] Jiri Rohn: A Handbook of Results on Interval Linear Problems. Czech Academy of Sciences Prague, Czech Republic, European Union. April 7, 2005. Posted at **http://www.cs.cas.cz/ rohn/handbook/** (16 January 2008)

[137] Jiri Rohn's home page:
http://www.uivt.cas.cz/∼rohn/matlab/index.html (25 September 2007)
http://www.cs.cas.cz/∼rohn/matlab/index.html (06 January 2008)

[138] J. Rhon: Systems of linear interval equations. Linear Algebra Appl., **126**, pp. 39-78, (1989)

[139] J. Rohn: Cheap and tight bounds: The recent result by E. Hansen can be made more efficient. Interval Computations, **4**, pp. 13-21, (1993)

[140] J. Rokne: Bounds for an interval polynomial. Computing, **18**, pp. 225–240, (1977)

[141] S.M. Rump: Fast and parallel interval arithmetic. BIT, **39**(3), pp. 534-554, (1999)

[142] Siegfried M. Rump's home page:
http://www.ti3.tu-harburg.de/rump/intlab/ (06 January 2008)

[143] Sergey P. Shary's home page: **http://old.ict.nsc.ru/rus/curvitae/shary/Codes/progr.html** (23 March 2008)

[144] S.P. Shary: Algebraic approach of the interval linear static identification, tolerance, and control problems, or one more application of Kaucher arithmetic. Realiable Computing, **2**(1), pp. 3-33, (1996)

[145] S.P. Shary: Algebraic approach in the "outer problem"for interval linear equations. Reliable Computing, **3**, pp. 103-135, (1997)

[146] S.P. Shary: Outer estimation of generalized solution sets to interval linear systems. Reliable Computing, **5**, pp. 323-335, (1999)

[147] W. Siler & J. J. Buckley: Fuzzy Expert Systems and Fuzzy Reasoning Wiley-Interscience, John Wiley and Sons, Inc (2005)

[148] I. Skalna: A Method for Outer Interval Solution of Systems of Linear Eqautions Depending Linearly on Interval Prameteres. Reliable computing, **12**, pp. 107-120, (2006)

[149] C.U. Smith: Performance Engineering of Software Systems. Addison-Wesley, Reading, MA, (1990)

[150] P. Spellucci & N. Krier: Untersuchungen der Grenzgenauigkeit von Algorithmen zur Auflösung linearer Gleichungssysteme mit Fehlererfassung, in: K. Nickel (ed.), Interval Mathmatics, (lectures Notes in Computer Science **29**), pp. 288-297, Springer, Berlin, (1975)

[151] W.J. Stewart: Introduction to Numerical Solution of Markov Chains. Princeton Univesity Press,New Jersey, (1994)

[152] W. Strother: Fixed points, fixed sets, and m-retracts. Duke Math. J. **22**, pp. 551-556, (1965)

[153] W. Strother: Continuous multi-valued functions. Bol. Soc. Mat. , São Paulo, **10**, pp. 87-120, (1958)

[154] T. Sunaga: Theory of an Interval Algebra and its Applications to Numerical Analysis. Gaukutsu Bunken, Fukeyu-kay, Tokyo (1958)

[155] F.J.W. Symons: Introduction to numerical Petri nets, a general graphical model of concurrent processing systems. Australian Telecommunications Research, **14** (1) ,pp. 28–33, January, (1980)

[156] A. Thomassian and P. F. Bay: Analytic Queueing Network Models for Parallel Processing of Task Systems IEEE Trans. On Computers, **35** (12), pp. 1045–1054, Dec (1986)

[157] TimeNET home page: **http://pdv.cs.tu-berlin.de/∼timenet/** (06 May 2008)

[158] Dr. Kishor S. Trivedi's home page: **http://www.ee.duke.edu/∼kst/**

[159] C.W. Ueberhuber: Numerical Computation: Methods, software and analysis. Springer Berlin Heidelberg, Vol. 1 (474 pages), Vol. 2 (495 pages) (1997)

[160] P. Walley: Statistical reasoning with imprecise probabilities. Chapman and Hall, London, New York, Tokyo, Melbourne, Madras, (1991)

[161] P. Walley & T.L. Fine: Towards a frequentist theory of upper and lower probability. Annals of Statistics, **10** (3), pp. 741-761, (1982)

[162] K. Weichselberger: Axiomatic foundations of the theory of intervalprobability. In: Proceedings of the Second Gauss Symposion, Section B, Mammitzsch and Schneeweiss, eds., Berlin: De Gruyter, pp. 47-64, (1995)

[163] K. Weichselberger: The theory of interval-probability as a unifying concept for uncertaintty. International Journal of Approximate Reasoning, **24**, pp. 149–170,(2000)

[164] K. Weichselberger: Elementare Grundbegriffe einer allgemeineren Wahrscheinlichkeitsrechnung, volume I. Intervallwahrscheinlichkeit als umfassendes Konzept. Physika, Heidelberg, (2001)

[165] W. Whitt: Blocking when service is required from several facilities simultaneously. AT&T Technical Journal, **64**, pp. 1807-1856 (1985)

[166] R. Yager: A characterization of the extension principle. Fuzzy Sets and Systems, **18**, pp. 205-217 (1986).

[167] A.G. Yakovlev: Computer Arithmetic ofMultiintervals, in: Problems of Cybernetics, Scientific Council on the Complex Problem "Cybernetics"of the Academy of Sciences of the USSR, **125** , pp. 66-81 (1986) (in Russian)

[168] R.C. Young: The algebra of many-valued quantities. Math. Ann. **104**, 260-290, (1931)

APÊNDICE A

Modelos

Este documento, como o título sugere, é sobre os modelos intervalares que são usados para estudar uma classe de sistema (sistemas de eventos discretos, que envolvem conceitos tais como "espaço de estado", "equação de estado" e "caminhos amostrais´´), que é muito comum ao nosso mundo cotidiano. Neste sentido é razoável fazer uma descrição simplificada do que significa "modelo", e apresentar alguns conceitos comuns na teoria dos modelos como eles vêm sendo desenvolvidos através dos anos. Isto define o objetivo deste apêndice, que é para ter o benefício de expor, superficialmente, material introdutório sobre classificação dos tipos de modelos.

Neste trabalho um modelo é uma abstração do mundo real. Seu objetivo é descrever o comportamento de um sistema. Entretanto, um modelo é geralmente obtido através de simplificações do sistema do mundo real, e não é único, *i.e.*, diferentes modelos para diferentes objetivos podem ser obtidos para um único sistema.

Um *modelo conceitual* é a descrição qualitativa de um sistema e sua representação (*e.g.*, descrição da geometria, condições iniciais, condições de contorno, parâmetros) relevante ao uso no modelo pretendido.

Um *modelo matemático* é uma representação matemática de um modelo conceitual para um modelo do mundo real, por expressões, com finalidade de ajudar no entendimento ou predizer o comportamento do sistema sobre determinadas condições.

A *verificação* é o ato de provar ou checar que um sistema formal tem uma propriedade formalmente declarada. Na interpretação estrita da palavra, o objetivo é encontrar exatamente uma evidência rigorosa para a reivindicação que o sistema é correto, no sentido de ter a propriedade. Se o sistema não é correto, então basta que a técnica de verificação falhe, terminando ou não, ou fornecendo um relatório de falha que deixa a questão de justeza sem resposta. Este tipo de técnica de verificação pode ser chamado unilateral - pode responder "sim, o sistema é correto", mas não pode responder "não, ele é incorreto". Em uma interpretação mais permissiva, a técnica de verificação pode também dar a resposta "incorreto" e, melhor ainda, dar alguma informação de depuração, tal como um exemplo de uma execução que viola a especificação. A técnica de verificação que, dando tempo bastante e memória, é eventualmente garantida terminar com a resposta "sim" ou "não" é um algoritmo de verificação. Um algoritmo de verificação unilateral irá eventualmente terminar e responder "sim" ou "não pode dizer".

A *validação* é o processo de obtenção assegurada que um sistema se comporta como planejado. O comportamento do sistema é comparado às expectativas do ser humano que está validando o sistema. Validação é, assim, sempre inerentemente informal.

Existem muitas classificações de tipos de modelos [159, 13, 12]. Vejamos alguns deles:

Não matemáticos

- Modelos linguísticos. Uma descrição linguística do comportamento de um sistema em diferentes circunstâncias.
- Modelos gráficos. Diagramas, grafos, mapas.
- Em escala. Escala reduzida de um sistema real.
- Maquetes.

Matemáticos

Descrevem a relação entre as variáveis do sistema por meio de equações que determinam seu comportamento (estado).

- Estáticos e dinâmicos.
- Linear e não linear.
- Parâmetros agrupados (*lumped*). Em um modelo parâmetro agrupado (modelo caixa-preta), as variações espaciais dos parâmetros são ignoradas e o sistema é descrito por parâmetros ajustáveis (ajuste de mínimos quadrados, por exemplo).
- Tempo contínuo e tempo discreto (incluídos os sistemas de eventos discretos).
- Determinísticos e não-determinísticos (probabilístico ou estocástico).

A.1 Modelagem e Incertezas

A modelagem pode se realizada através da obtenção do modelo, aplicando leis físicas que atuam nos sistemas. Em grande parte das aplicações práticas existe uma diferença entre o comportamento do sistema e o comportamento predito pelo modelo. Estas diferenças podem ser atribuídas às medidas (as medidas são somente aproximações dos valores reais) e ao modelo do sistema (construído com simplificações). As medidas podem até ser muito precisas, mas geralmente inexatas. A falta de exatidão do modelo é inerente, porque o modelo é uma abstração (aproximada) da realidade. Em muitos sistemas físicos é possível haver modelos exatos do sistema [159], porém, muito complexos e um modelo simplificado, embora mais restrito, é normalmente adotado. Existem muitos casos em que, devido a incertezas e imprecisões, fica muito difícil, e até impossível, a obtenção de modelos exatos. Algumas causas de inexatidão destes modelos são:

- O sistema real não existe (fase de projeto de uma fábrica ou de manufatura de um produto).

- O fenômeno físico é difícil de identificar ou prever.

- Os parâmetros do sistema podem mudar com o tempo devido a causas imprevisíveis, desconhecidas ou difíceis de serem modeladas.

As incertezas podem ser estruturadas (as equações do modelo são conhecidas, mas seus parâmetros não são) ou desestruturadas (as equações que modelam o sistema são desconhecidas ou muito difíceis de serem obtidas). As incertezas são geralmente representadas com modelos quantitativos, *i.e.*, modelos nos quais os parâmetros são números reais. Quando as incertezas não são representadas por números reais, outros tipos de modelos são necessários. Alguns tipos de modelos que podem representar as incertezas do sistema são qualitativos, nebulosos (fuzzy) [147, 73, 29, 31] e modelos intervalares.

A.2 Modelos Intervalares

O espaço de modelos representando incertezas é explicitamente expresso por um modelo intervalar, *i.e.*, um modelo no qual os parâmetros são representados por intervalos em vez de números reais. Por exemplo, um modelo deste tipo poderá ser representado pela seguinte equação diferencial:

$$[a]_2 \frac{d^2y}{dt^2} + [a]_1 \frac{dy}{dt} = f(x),$$

onde
$[a]_i$ são parâmetros intervalares do modelo;
x é a entrada do sistema;
y é a saída do sistema;
t é o tempo.

Portanto, um modelo intervalar é, na realidade, uma família, uma classe ou um espaço de modelos, e um modelo quantitativo pode ser visto como um modelo intervalar onde o intervalo tem largura zero (intervalo degenerado). Quando a largura do intervalo decresce, a precisão do modelo cresce. Um modelo intervalar também é referido como semiqualitativo, porque permite uma fácil integração do conhecimento qualitativo e quantitativo. Por exemplo, os intervalos dos parâmetros podem ser definidos por especialistas.

Um modelo intervalar representa adequadamente incertezas de modelos estruturados, i.e., a estrutura do modelo é conhecida e somente os parâmetros das equações sofrem imprecisões ou incertezas. É comum em sistemas reais, onde um valor específico para um parâmetro não pode ser estabelecido, mas pode ser limitado a uma gama de valores.

Este tipo de modelo pode ser usado numa variedade de casos:

- Para representar o comportamento de um modelo de um sistema submetido a diferentes situações.

- Para representar variabilidades temporais. Os parâmetros podem variar através do tempo segundo uma função desconhecida ou conhecida do tempo. Em qualquer caso, cada parâmetro pode ser limitado por um intervalo, incluindo todos valores possíveis.
- O modelo complexo pode ser representado por um modelo intervalar simplificado.

A.3 Observações

Existe, obviamente, um grande espectro de esforço devotado ao desenvolvimento da notação e semântica usada para classificar e descrever os diferentes tipos de modelos na modelagem de sistemas. Esperançosamente, ISPN faz parte do esforço para fornecer a infra-estrutura na solução de alguns problemas reais de engenharia e para construir sistemas práticos que desenvolvam um conjunto de funções desejadas de forma eficiente e econômica. Em termos gerais, os objetivos da ferramenta de modelagem de alto nível ISPN servem para: a) modelagem e análise - com finalidade de desenvolver um modelo para ver se ele reproduz o sistema físico (matematicamente, em laboratório ou através de simulação computacional). De posse do modelo dentro da precisão requerida, o objetivo passa a ser estudar o comportamento do sistema sobre diferentes condições; b) projeto e síntese - com técnicas de modelagem, assumidamente adequadas, pode-se explorar o tema de “como construir um sistema que tenha o comportamento desejado?”; c) controle - tenta-se selecionar os parâmetros variáveis do modelo para assegurar que o sistema comporte-se como desejado sobre uma variedade (possivelmente adversa) de condições. Neste caso precisa-se de um modelo que possa trabalhar com a validação e teste de vários mecanismos de controle que possam ser realizados; d) avaliação de desempenho - trata-se do tema "quão bem o sistema está realmente desempenhando?"; e) otimização - dado que um sistema possa ser projetado e controlado para alcançar um desempenho desejado, então surge naturalmente a questão: “o deve ser feito para alcançar a melhor performance possível?"

APÊNDICE B

Processos estocásticos

Este apêndice fornece definições básicas relativas à classe de processos produzidos pelos modelos GSPN [103]. Os processos estocásticos, que será o foco principal, são processos markovianos com espaço de estados finito em tempo contínuo (cadeia de Markov de tempo contínuo). O objetivo não é fornecer um tratamento formal, rigoroso e completo; em vez disso, desenvolve-se uma noção sobre a estrutura e a dinâmica dos processos estocásticos de interesse para fornecer as bases para modelos ISPN [33] - extensão intervalar dos modelos GSPN. Existe uma vasta literatura sobre processos estocásticos para quem tem interesse de aprofundar sobre o assunto. Este texto foi produzido consultando-se [76, 5, 107, 12].

Os modelos matemáticos podem ser categorizados como sendo probabilísticos e determinísticos. Nos casos em que modelos probabilísticos são usados, a representação é dada por uma coleção ou família de variáveis aleatórias. As coleções de variáveis aleatórias que são indexadas por um parâmetro tal como o tempo e o espaço são conhecidas como processo estocástico (ou "aleatório" ou processo de "chance"). Um processo estocástico é um modelo probabilístico de um sistema que envolve aleatoriedade. Se o sistema é observado em pontos discretos de tempo (épocas) $n = 0, 1, \cdots$ onde X_n é o estado do sistema no tempo n, então $\{X_n,\ n \geq 0\}$ é um processo estocástico que o descrevendo. No caso em que o sistema é observado em tempos discretos (que podem não ser igualmente espaçados no eixo do tempo) ele é chamado processo estocástico de tempo discreto. Alguns exemplos de X_n podem ser o INPC (Índice Nacional de Preço ao Consumidor - IBGE :: Instituto Brasileiro de Geografia e Estatística), observado a cada mês; o número de impressoras não vendidas de um lote no começo do n-ésimo dia de venda do produto; precipitação pluviométrica em Pernambuco neste século ou o número de assassinatos no Recife no n-ésimo dia de contagem, só para citar alguns exemplos.

Se o sistema é observado continuamente sobre o tempo com $X(t)$, sendo seu estado no tempo t, então ele é descrito por um processo estocástico de tempo contínuo $\{X(t);\ t \geq 0\}$. Por exemplo, $X(t)$ pode representar o número de microcomputadores sem funcionar em um laboratório num tempo t, a posição das nuvens de chuva sobre uma região no tempo t, ou o saldo bancário no tempo t.

Introduzindo um pouco de formalismo, um processo estocástico é uma coleção de variáveis aleatórias $\{X(\alpha),\ \alpha \in T\}$, indexado pelo parâmetro α com valores no conjunto T. As variáveis aleatórias têm valores de um conjunto S, chamado espaço de estado do processo estocástico. Em muitos casos o parâmetro α representa o tempo. Normalmente encontram-se dois casos: (i) $T = \{0, 1, 2, \cdots\}$, o qual se escreve $\{X_n,\ n \geq 0\}$, em vez de $\{X(\alpha),\ \alpha \in T\}$; e (ii) $T = [\,0,\ \infty)$, para o qual se escreve $\{X(t);\ t \geq 0\}$, em vez de $\{X(\alpha),\ \alpha \in T\}$. Também é comum encontrar $S \subset \{0, 1, 2, \cdots\}$ ou $S \subset (-\infty, \infty)$.

Considera-se $\{X(\alpha),\ \alpha \in T\}$ ser um processo estocástico com espaço de estado S e $X : T \rightarrow$

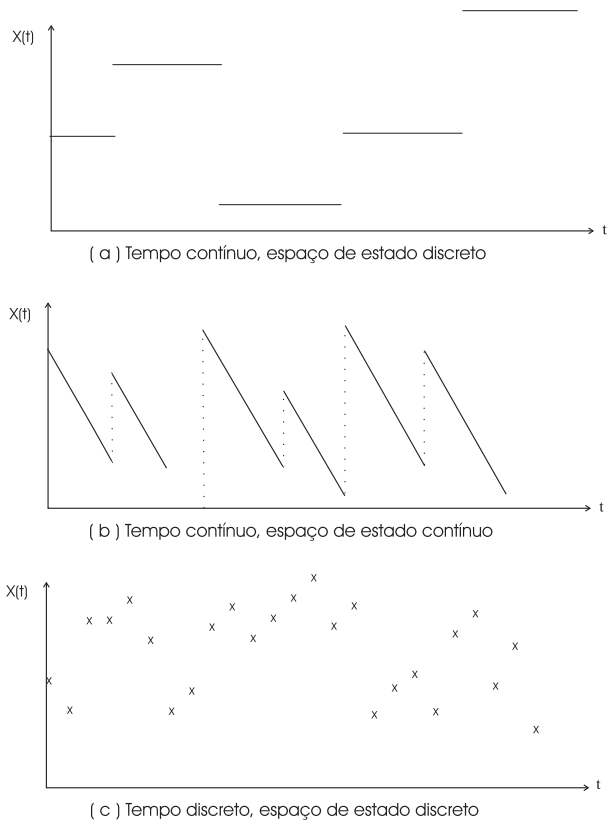


Figura B.1 Caminhos de amostragem típicos de processos estocásticos

S ser uma função. Pode-se pensar $\{X(\alpha),\ \alpha \in T\}$ como uma possível evolução (trajetória) de $\{X(\alpha),\ \alpha \in T\}$. As funções x são chamadas de caminhos amostrais do processo estocástico.

A Figura B.1 mostra alguns caminhos amostrais típicos de processos estocásticos. Como esse processos estocásticos seguem um caminho de amostragem aleatório, ele é algumas vezes chamado de função aleatória. Geralmente o conjunto de todos os caminhos amostrais, o chamado espaço de amostras do processo estocástico, é não contável (observe que isto é verdade até nos casos de processos estocásticos de tempo discreto $\{X_n,\ n \geq 0\}$ com espaço de estados finitos S. Um dos objetivos do estudo dos processos estocásticos é estudar o comportamento dos caminhos amostrais que o sistema segue, com o objetivo final de previsão e controle do futuro do sistema.

Na identificação da natureza do processo estocástico é importante classificá-lo com base na natureza de seus parâmetros: espaço de estado e parâmetro de espaço. As classes diferentes dos processos estocásticos são dados em forma de tabela (Tabela B.1).

Tabela B.1 Processo Estocástico

Espaço do parâmetro	Espaço do Estado	
	Discreto	Contínuo
Discreto	1	3
Contínuo	2	4

Eis exemplos correspondentes aos processos estocásticos da Tabela B.1:

1. Espaço de estados e espaço de parâmetros discreto: número de preferências de consumidores, observadas semanalmente.

2. Espaço de estados discreto e espaço de parâmetros contínuo: número de pessoas esperando numa fila em qualquer hora do dia.

3. Espaço de estados contínuo e espaço de parâmetros discreto: quantidade de gasolina em estoque observada em épocas de tempo discreto.

4. Espaço de estados contínuo e espaço de parâmetros contínuo: quantidade de água numa barragem observada num intervalo de tempo.

B.1 Cadeias de Markov

Uma cadeia de Markov é um processo estocástico que apresenta a propriedade markoviana em que os estados anteriores são irrelevantes para a predição dos estados seguintes, desde que o estado atual seja conhecido.

Uma maneira simples de visualizar um tipo específico de cadeia de Markov é através de uma máquina de estados finitos. Se algo está no estado y no tempo n, então a probabilidade de que se mova para o estado x no tempo $n+1$ não depende de n, e somente depende do estado atual y em que está. Assim em qualquer tempo n, uma cadeia de Markov finita pode ser caracterizada por uma matriz de probabilidades, cujo elemento $(x,\ y)$ é dado por $\Pr(X_{n+1} = x|X_n = y)$, e é independente do tempo n. Estes tipos de cadeia de Markov finitas e discretas podem também ser descritas por meio de um grafo dirigido, onde cada aresta é rotulada com as probabilidade de transição de um estado a outro, sendo esses estados representados como os nós conectados pelas arestas.

Propriedade Markoviana: Se o presente estado do sistema é conhecido, o futuro do sistema é independente do seu passado.

Em outras palavras, o sistema, tendo a propriedade acima, o passado afeta o futuro somente através do presente; ou, afirmado ainda de outra maneira, o estado atual do sistema contém todas as informações relevantes necessárias para produzir o futuro em sentido probabilístico. Embora existam cadeias de Markov para tempo discreto, na sequência só faremos a discussão de cadeias de Markov de tempo contínuo para permitir o estabelecimento da formulação que permita a análise estado do estacionário e do estado transiente da mesma.

B.1.1 Cadeias de Markov de tempo contínuo

A propriedade de Markov (ausência de memória) para uma cadeia de Markov de tempo contínuo é dada por:

$$\begin{aligned} P\left[X(t_{k+1}) = x_{k+1} \mid X(t_k) = x_k, X(t_{k-1}) = x_{k-1}, \ldots, X(t_0) = x_0\right] \\ = P\left[X(t_{k+1}) = x_{k+1} | X(t_k) = x_k\right] \end{aligned} \tag{B.1}$$

onde $t_0 \leq t_1 \leq \cdots \leq t_k \leq t_{k+1}$.

O valor de $X(t_{k+1})$ depende somente de x_k e não depende de qualquer história passada dos estados (sem memória de estado passado). O tempo gasto no presente estado também é irrelevante na determinação do próximo estado (sem memória de época).

O estudo de cadeias de Markov de tempo contínuo utiliza as taxas nas quais os eventos acontecem (transição de estados). Isto permite especificar um modelo para que se possa realizar a análise do mesmo.

B.2 Observações

O objetivo deste texto é introduzir os principais componentes da teoria e do formalismo ISPN e também dos exemplos de algumas de suas aplicações. É necessário para distinguir entre CTMC e ICTMC subjacentes a GSPN (clássica) e a ISPN (intervalar), respectivamente. Este apêndice foi incluído para relembrar conceitos básicos teóricos dos processos estocásticos com cadeias de Markov. Posteriormente, no Apêndice E, faremos a extensão intervalar dos conceitos para a ICTMC e derivar suas equações do estado estacionário e no estado transiente.

APÊNDICE C

Função de Inclusão Intervalar

Neste apêndice vamos introduzir a importante noção de função intervalar, como calculá-la e definir algumas funções intervalares básicas. Iniciaremos com a notação e a definição formal de função intervalar. Este é um tópico de importância fundamental na análise intervalar.

As funções são denotadas com a mesma convenção tipográfica que os elementos de seus espaços imagens, portanto $[f](\cdot)$ é uma função intervalar escalar, e $[\mathbf{f}](\cdot)$ é uma função intervalar vetorial.

Considere uma função $\mathbf{f}:\mathbb{R}^n \to \mathbb{R}^m$. A função intervalar $[\mathbf{f}]: I(\mathbb{R})^n \to I(\mathbb{R})^m$ é uma função de inclusão, se

$$\forall [\mathbf{x}] \in I(\mathbb{R})^n, \quad \mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{x}]).$$

A função intervalar $[\mathbf{f}]([\mathbf{x}]) = \mathbb{R}^m, \ \forall [\mathbf{x}] \in I(\mathbb{R})^n$, é um exemplo de uma função de inclusão (não muito útil) para todas funções $\mathbf{f}:\mathbb{R}^n \to \mathbb{R}^m$ [60]. Uma das finalidades da análise intervalar é fornecer, para uma grande classe de função f, funções de inclusão que podem ser calculadas de maneira razoavelmente rápida e tal que $[\mathbf{f}]([\mathbf{x}])$ não seja muito larga (superestimação intervalar). A função $\mathbf{f}$ pode ser, por exemplo, polinomial [97, 44], ou dada por um algoritmo [114]. Ela pode até mesmo ser definida como solução de um conjunto de equações diferencias [84, 4, 74].

Se $\mathbf{f}(\cdot)$ é uma função de $\mathbb{R}^n \to \mathbb{R}^m$ diferenciável uma vez, então a sua matriz jacobiana em $\mathbf{x}$ é

$$\mathbf{J_f}(\mathbf{x}) \triangleq \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_m}{\partial x_n}(\mathbf{x}) \end{pmatrix}.$$

Se $f(\cdot)$ é uma função de $\mathbb{R}^n \to \mathbb{R}$ diferenciável uma vez, então a seu gradiente em $\mathbf{x}$ é

$$\mathbf{g}_f(\mathbf{x}) \triangleq \begin{pmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\mathbf{x}) \end{pmatrix}.$$

Se $f(\cdot)$ é uma função diferenciável duas vezes, então a sua matriz hessiana em $\mathbf{x}$ é a matriz jacobiana (simétrica) associada com seu gradiente, *i.e.*,

$$\mathbf{H}_f(\mathbf{x}) \triangleq \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(\mathbf{x}) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(\mathbf{x}) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(\mathbf{x}) \end{pmatrix}.$$

Lembremos que $f : D \subseteq \mathbb{R} \rightarrow \mathbb{R}$ denota uma função real $y = f(x)$ de uma variável definida no domínio D, com valores em $\mathbb{R}$.

Definição C.1 (Função Intervalar) $[f] : D \subseteq I(\mathbb{R})^n \rightarrow I(\mathbb{R})$ denota uma função intervalar real $[y] = f([\mathbf{x}])$, onde $[\mathbf{x}] = ([x]_1,\ [x]_2, \cdots [x]_n)$ é um vetor intervalar n-dimensional e $[y]$ um intervalo. Dizemos que $[y]$ é uma função de $[\mathbf{x}]$, $[y] = [f]([\mathbf{x}])$, se para cada x no domínio $D \subseteq I(\mathbb{R})^n$ existe um intervalo $[y]$ correspondente. Então [f] é uma função intervalar de várias variáveis intervalares.

A avaliação intervalar de uma função $[f]$ é dependente da escolha da expressão para $[f]$. A extensão intervalar ideal de $[f]$ será uma função que fornece a imagem exata de $[f]$ no domínio $[\mathbf{x}]$.

Definição C.2 A extensão unida $[f]_u([\mathbf{x}]) : I(\mathbb{R})^n \rightarrow f([\mathbf{x}])$ de uma função contínua $f : \mathbb{R}^n \rightarrow \mathbb{R}$ é

$$[f]_u([\mathbf{x}]) \triangleq [min\,(f(\mathbf{x}))\,,\ máx\,(f(\mathbf{x}))],\ \mathbf{x} \in [\mathbf{x}], \tag{C.1}$$

denota o intervalo de todos valores da função f quando $\mathbf{x} \in [\mathbf{x}]$. Se na expressão ocorrem constantes $[a]_k$, $k = 0 \cdots m$, elas são consideradas independentes umas das outras. Esta definição é independente da expressão. No mínimo, em princípio, pode-se conseguir o menor intervalo contendo $[f]_u([\mathbf{x}])$. Este é o chamado intervalo envoltório de $[f]_u([\mathbf{x}])$. Contudo este problema fica bastante distante de ser trivial no caso geral.

Como, no caso geral, esta extensão é muito difícil de calcular, então, o objetivo da aritmética intervalar é encontrar extensões intervalares computáveis que incluam a imagem exata da função $[F]_u$. Esta alternativa é uma maneira muito mais tratável e é consequência direta do Teorema 2.1.1.

Definição C.3 Uma função intervalar é chamada de monótona inclusiva se $[\mathbf{x}]_i \subset [\mathbf{y}]_i$ $(i = 1 \cdots n) \Rightarrow [f]([\mathbf{x}]) \subset [f]([\mathbf{y}])$.
Portanto, se $[\mathbf{x}]_1 \cup [\mathbf{x}]_2 = [\mathbf{x}]$ é uma inclusão monotônica, verifica-se que

$$[f]_u([\mathbf{x}]) \subseteq [f]([\mathbf{x}]_1) \cup [f]([\mathbf{x}]_2) \subseteq [f]([\mathbf{x}]). \tag{C.2}$$

Este procedimento é muito utilizado para computar a imagem de uma função $[f]$ em um domínio especificado $[\mathbf{x}]$, que consiste em encontrar uma função monotônica inclusiva e calcular uma aproximação superestimada da imagem. As melhores aproximações são obtidas dividindo-se o domínio. Quando f é uma função racional, a extensão natural $[f]_r([\mathbf{x}])$ é obtida pela substituição de cada variável real pelo intervalo correspondente e as operações correspondentes feitas pela aritmética intervalar [114]. Demonstra-se que:

$$[f]_u([\mathbf{x}]) \subseteq [f]_r([\mathbf{x}]). \tag{C.3}$$

Portanto, as aproximações superestimadas de $[f]_u([\mathbf{x}])$ podem ser obtidas, simplesmente, computando-se a extensão natural através da aritmética intervalar.

Exemplo: A imagem exata da função:

$$y = x^4 - 10x^3 + 3x^2 - 50x + 24$$

no espaço de parâmetros $[x] = [0, 4]$ é $[y] = [-1, 24]$.
A aproximação superestimada para imagem que é obtida usando a extensão natural é

$$[y] = [x]^4 - 10[x]^3 + 3[x]^2 - 50[x] + 24 = [-816, 840],$$

usando-se $[x]^k = [min(x^k),\ máx(x^k)],\ \ x \in [x]$.

A maioria das funções usadas na ciência da computação e matemática aplicada é expressa como combinação de funções aritméticas básicas. Inclusões intervalares racionais de funções irracionais podem também ser encontradas, incluindo as funções trigonométricas básicas, exponenciais, logaritmos etc.

Uma linha de pesquisa importante na aritmética intervalar é para encontrar funções de inclusão racional de funções irracionais que permitam calcular imagens com superestimações tão estreitas quanto possível.

C.1 Computação de funções intervalares racionais

A principal limitação da aritmética intervalar é que ela não tem algumas propriedades importantes da aritmética dos números reais, como, por exemplo, a propriedade distributiva.

Exemplo: A função real

$$f(x) = \frac{1}{x^2 + \frac{1}{2}}$$

é definida para todo $I(\mathbb{R})$. A expressão

$$\tilde{f}(x) = \frac{1}{x * x + \frac{1}{2}}$$

não é definida no intervalo $[-1, 1]$, mas claramente está contida no domínio de f:

$$\tilde{f}([-1, 1]) = \frac{1}{[-1, 1] * [-1, 1] + \frac{1}{2}} = \frac{1}{[-1, 1] + \frac{1}{2}} = \frac{1}{[\frac{-1}{2}, \frac{3}{2}]}$$

Portanto, a imagem de uma função nem sempre é computável. Contudo a extensão natural tem propriedades interessantes:

- $[f]_r([\mathbf{x}]) \supseteq [f]_u([\mathbf{x}])$;
- $[f]_u([\mathbf{x}]) = [f]_r([\mathbf{x}])$ é obtida se não existe variáveis multi-incidentes em $[f]_r([\mathbf{x}])$.

Definição C.4 Uma variável é multi-incidente numa função se aparece mais de uma vez na expressão da função.

Exemplo: A imagem da função

$$f(x) = \frac{x}{x+1}$$

no espaço paramétrico $[x] = [2, 3]$ é $[\frac{2}{3}, \frac{3}{4}]$. Se a imagem da função é calculada usando sua extensão natural

$$[f]_r([2, 3]) = \frac{[2, 3]}{[2, 3] + 1} = [\frac{1}{2}, 1],$$

a qual é uma superestimação

$$[\frac{2}{3}, \frac{3}{4}] \subset [\frac{1}{2}, 1],$$

por outro lado,

$$\tilde{f}(x) = \frac{1}{1 + \frac{1}{x}}$$

é uma expressão algébrica equivalente:

$$\tilde{f}([2, 3]) = \frac{1}{1 + \frac{1}{[2,3]}} = \frac{1}{1 + [\frac{1}{3}, \frac{1}{2}]} = \frac{1}{[\frac{4}{3}, \frac{3}{2}]} = [\frac{2}{3}, \frac{3}{4}],$$

que é exata.

A pesquisa em aritmética intervalar tem produzido muitas técnicas para cálculo da imagem de funções. Seu interesse principal é a garantia sobre os resultados, os quais sempre serão aproximações superestimadas de imagens exatas. A precisão pode ser fixada ao valor desejado. As técnicas existem e podem ser algébricas e numéricas.

Algoritmos numéricos. Extensões inclusões monotônicas de funções intervalares produzem aproximações superestimadas de suas imagens. De acordo com a Equação (C.2), aproximações melhores podem ser obtidas dividindo o domínio da função. O espaço de parâmetros função, um hipercubo, é dividido em subespaços cada vez menores, e o resultado aproxima-se cada vez mais do exato. Em alguns casos a divisão de um subespaço em subespaços não produz melhor resultado e portanto é desnecessário. Contudo é possível fazer divisão por partes (*branch cutting*) nestes casos, o qual é baseado no estudo das propriedades da função e suas derivadas. Muitos algoritmos aplicam ambas as técnicas simultaneamente. Detalhes e exemplos destas divisões por partes e limitações (*branch and bound*) podem ser encontradas em [55].

Técnicas algébricas. Como uma consequência da propriedade de subdistributividade da aritmética intervalar, os resultados do cálculo da imagem de uma função por meio de extensão intervalar são altamente dependentes da expressão sintática da função.

Exemplo: A função $f(x) = x^2 - x$ pode ser escrita como $\tilde{f}(x) = (x - \frac{1}{2})^2 - \frac{1}{4}$. A extensão natural no espaço paramétrico $[x] = [0, 2]$ dá uma aproximação superestimada da sua imagem:

$$f([x]) = [x]^2 - [x] = [0, 2]^2 - [0, 2] = [-2, 4],$$

enquanto a extensão natural da segunda função fornece a imagem exata:

$$\tilde{f}([x]) = ([x] - \frac{1}{2})^2 - \frac{1}{4} = ([0, 2] - \frac{1}{2})^2 - \frac{1}{4} = [-\frac{1}{4}, 2].$$

A razão é que em $f(x)$, a variável x é multi-incidente e em $\tilde{f}(x)$ ela ocorre uma única vez.

Portanto, uma maneira de obter resultados estreitos, mas sempre garantidos, é expressar a função com o menor número de multi-incidências. Muitas técnicas pesquisam a melhor maneira de expressar uma função para obter resultados mais estreitos [133]. Algumas delas são:

- cálculo em cascata [58];
- esquema de Horner estendido [119];
- formas centradas [112]: valor médio, polinômio de Baumann, polinômio de Taylor;
- polinômio de Bernstein : uma variável [140], várias variáveis [43];
- inclusão adequada de funções racionais como $sen(x)$, $cos(x)$, $exp(x)$, ou $\sqrt{x}$ [123].

C.2 Observações

O tópico abordado no presente apêndice não foi explorado no presente documento de forma explicita. Entretanto ele dá suporte complementar ao formalismo teórico da análise intervalar. É importante complemento para compreensão dos Apêndices D e E, que por questões de natureza genérica, as funções de inclusão intervalar podem ser considerada como a extensão unida da Definição C.2. Na prática, por questões abordadas acima, teremos que usar métodos e aritméticas da análise intervalar. No cálculo dos índices, da análise do estado estacionário do modelo ISPN deste documento, foi utilizada a extensão natural, por ser adequada à natureza simples das expressões matemáticas das métricas calculadas. Em futuras aplicações será necessário utilizar técnicas para cálculo de função de inclusão intervalar. O presente estudo ficou restrito à avaliação com uso da extensão natural (através da aritmética intervalar). Vale a pena mencionar que a aritmética Kaucher, assim como outras aritméticas intervalares, é utilizada para cálculo de funções de inclusão intervalar [69].

APÊNDICE D

Terminologia da probabilidade intervalar

Consideremos o caso de uma álgebra finitamente gerada $\mathscr{A}$, baseado num espaço amostral Ω. Então, sem perda de generalidade, Ω é finito, e $\mathscr{A}$ e o conjunto potência de $\Omega = \{\omega_1, \omega_2, \cdots \omega_n\}$. A terminologia para cada medida de probabilidade no sentido usual, *i.e.*, cada conjunto de função $p(.)$, satisfazendo os axiomas de Kolmogorov é chamado uma *probabilidade clássica*[1]. O conjunto de todas as probabilidades clássicas no espaço de medidas $(\Omega, \mathscr{A})$ será denotado por $\mathscr{K}(\Omega, \mathscr{A})$. Os axiomas para probabilidade com valores intervalares $p(.) = [l(.), u(.)]$ podem ser obtidos observando-se a relação entre o conjunto não-aditivo de funções $l(.)$ e $u(.)$ e o conjunto clássico de probabilidades estando de acordo com ele [1]. Num espaço amostral finito, vários conceitos de probabilidade intervalar coincidem. Eles estão lidando com o conjunto de funções

$$\begin{array}{rcl} p(.) : \mathscr{A} & \longrightarrow & \mathscr{X}_0 := \{[l, u] | 0 \leq l \leq u \leq 1\} \\ A & \longmapsto & p(A) = [l(A), u(A)] \end{array}$$

com

$$\mathscr{M} := \{p(.) \in \mathscr{K}(\Omega, \mathscr{A}) \mid l(A) \leq p(A) \leq u(A),\ \forall A \in \mathscr{A}\} \neq 0, \tag{D.1}$$

e

$$\begin{array}{l} \underset{p(.) \in \mathscr{M}}{inf} \quad p(A) = l(A) \\ \underset{p(.) \in \mathscr{M}}{sup} \quad p(A) = u(A) \end{array} . \tag{D.2}$$

Tal p(.), e o conjunto correspondente de funções l(.) e u(.), são chamados probabilidades inferior e superior [57], envelopes [161, 26], probabilidade coerente [160] e probabilidade-F [162, 163, 164]. A terminologia de Weichselberger chama estrutura $\mathscr{M}$. Observe que, por (D.2), existe uma correspondência uma a uma entre p(.) e a estrutura $\mathscr{M}$.

Conjuntos mais gerais de probabilidades clássicas são obtidos pela teoria de previsões coerentes [160]. Pelo teorema do envelope inferior ([160], p.134) e o fato de que a esperança clássica e probabilidade clássica correspondem unicamente uma com a outra, a definição de coerência pode ser reescrita em uma forma similar a (D.2). Desde que Walley [160] não deu um nome para o conjunto clássico de probabilidades resultantes, ele poderá ser chamado de estrutura também. Muitos conceitos da teoria da probabilidade clássica podem ser generalizadas adequadamente.

D.1 Observações

A teoria de medidas é uma ferramenta matemática que é utilizada para dar suporte à teoria das probabilidades. A abordagem considerada acima considera elementos da teoria de medidas, estendidos para o paradigma da probabilidade intervalar [35]. No caso geral, os cálculos das probabilidades intervalares são intratáveis. Uma alternativa passa a ser utilizar métodos da análise intervalar para que os problemas relacionados possam ser resolvidos em tempo polinomial.

APÊNDICE E

ICTMC

Um espaço de modelo que representa incertezas ou imprecisão pode ser expresso explicitamente por um modelo intervalar, *i.e.*, um modelo no qual há parâmetros representados através de intervalos em vez de números reais. Então, um modelo intervalar é, de fato, uma família, uma classe ou um espaço de modelos, e um modelo clássico pode ser visto como um modelo intervalar particular no qual larguras de intervalo são zero. A propriedade markoviana (*memoryless*) para uma ICTMC (*Interval Continuous-Time Markov Chain* - Cadeia de Markov Intervalar de Tempo Contínuo) é determinado por:

$$\begin{aligned}[\mathbf{P}]\,\{&[\mathbf{X}](\tilde{t}_{k+1}) = x_{k+1} | [\mathbf{X}](\tilde{t}_k) = x_k, \\ &[\mathbf{X}](\tilde{t}_{k-1}) = x_{k-1}, \ldots, [\mathbf{X}](\tilde{t}_0) = x_0\} \\ &= \mathbf{P}\,\{[\mathbf{X}](\tilde{t}_{k+1}) = x_{k+1} | [\mathbf{X}](\tilde{t}_k) = x_k\}\,,\end{aligned} \tag{E.1}$$

onde $\tilde{t}_0 \leq \tilde{t}_1 \leq \cdots \leq \tilde{t}_k \leq \tilde{t}_{k+1}$, $\forall \tilde{t}_i \in [t_i]$, $[t_i] = [\check{t}_i - \Delta t_i, \check{t}_i + \Delta t_i] \equiv \check{t}_i \pm \Delta t_i$, $\check{t}_i = mid[_i]$, $\Delta t_i = rad[t_i]$, $i = 0, 1, 2, \cdots, k+1$.

O valor $[\mathbf{X}]([t_{k+1}])$ depende somente de x_k e não depende de qualquer história passada dos estados (sem memória de estado passado). O tempo gasto no presente estado também é irrelevante na determinação do próximo estado (sem memória de época) [12].

O estudo de cadeias de Markov intervalar de tempo contínuo utiliza as taxas intervalares nas quais os eventos acontecem (transição de estados). Isto permite especificar um modelo para que se possa realizar a análise intervalar do mesmo.

Considere χ denotar o conjunto de estados de tempo contínuo do processo estocástico markoviano, a probabilidade de transição entre estados $i, j \in \chi$ é especificado por uma matriz de intervalar $[\mathbf{P}](\chi)$, cuja entrada, $[p]_{ij}([t])$, $\forall \tilde{t} \in [t]$, é a probabilidade intervalar de uma transição de i para j dentro de um intervalo de tempo de duração $\tilde{t}$ para todo possível $\tilde{t} \in [t]$. Consequentemente, todo valor dentro do intervalo $[p]_{ij}([t])$ são possíveis transições de probabilidade. Uma possível solução é obtida pela introdução de informação relativa à taxa intervalar, na qual várias transições de estados (eventos) podem acontecer em qualquer hora.

As probabilidades de transição dependentes do tempo estão definidas como:

$$\begin{aligned}&[p]_{ij}([s],[t]) \equiv P\,[[X]([t]) = j \,|\, [X]([s]) = i]\,, \\ &\forall \tilde{s} \in [s], \forall \tilde{t} \in [t],\ \tilde{s} \leq \tilde{t}\end{aligned} \tag{E.2}$$

$$[s] = [\check{s} - \Delta s, \check{s} + \Delta s] \equiv \check{s} \pm \Delta s,$$

$$[t] = [\check{t} - \Delta t, \check{t} + \Delta t] \equiv \check{t} \pm \Delta t.$$

As funções intervalares de transição $[p]_{ij}([s],[t])$ são funções dos instantes de tempo $\tilde{s} \in [s]$ e $\tilde{t} \in [t]$ que satisfazem uma equação intervalar de Chapman-Kolmogorov de tempo contínuo

$$\begin{array}{c} [p]_{ij}([s],[t]) = \sum_{all\, r}[p]_{ir}([s],[u])[p]_{rj}([u],[t]), \\ \forall \tilde{s} \in [s],\ \forall \tilde{u} \in [u],\ \forall \tilde{t} \in [t],\ \tilde{s} \leq \tilde{u} \leq \tilde{t}. \end{array} \tag{E.3}$$

Para reescrever (E.3) na forma de matriz intervalar, é definido:

$$[\mathbf{H}]([s],[t]) \equiv [p]_{ij}([s],[t]),\ \ i,j = 0,\, 1,\, 2, \cdots \tag{E.4}$$

e observar que $[\mathbf{H}]([s],[s]) = \mathbf{I},\ \forall \tilde{s} \in [s]$ (a matriz identidade). Então, de (E.3):

$$\begin{array}{c} [\mathbf{H}]([s],[t]) = [\mathbf{H}]([s],[u]) \cdot [\mathbf{H}]([u],[t]), \\ \forall \tilde{s} \in [s],\ \forall \tilde{u} \in [u],\ \forall \tilde{t} \in [t]\ \ \tilde{s} \leq \tilde{u} \leq \tilde{t}. \end{array} \tag{E.5}$$

A equação intervalar de Chapman-Kolmogorov é a relação geral estendida para a propriedade de ausência de memória (*memoryless*) de Markov .

A cadeia de Markov intervalar de tempo-contínuo homogênea está definida quando toda a função de transição $[p]_{ij}([s],\,[t])$, descrita por (E.2), é independente dos instantes de tempo absolutos $[s]$, $[t]$ e só depende da diferença $[\tau] = [s] - [t]$. Para sistemas de tempo-contínuo homogêneos, a equação diferencial ascendente de Kolmogorov (E.6) é obtida

$$\frac{d[\mathbf{P}]([\tau])}{dt} = \mathbf{P}([\tau])[\mathbf{Q}]. \tag{E.6}$$

Com as condições iniciais seguintes (assumindo que uma transição de qualquer estado i para $j \neq i$ não pode acontecer em tempo zero):

$$[p]_{ij}(0) = \left\{ \begin{array}{l} 1\ if\ \ j = i \\ 0\ if\ \ j \neq i \end{array} \right. . \tag{E.7}$$

A solução de (E.6) é da forma

$$[\mathbf{P}]([\tau]) = e^{[\mathbf{Q}][\tau]}. \tag{E.8}$$

Definindo as probabilidades de estados intervalares

$$[\pi]_j(t) \equiv [\mathbf{P}][\mathbf{X}([t]) = j] \tag{E.9}$$

produz o vetor *intervalar de probabilidade de estado*

$$[\pi](t) = [[\pi]_0(t),\, [\pi]_1(t),\, [\pi]_2(t),\, \cdots].$$

Este é um vetor intervalar novo, cuja dimensão é determinada pela dimensão do espaço de estado da cadeia (não necessariamente finito).

Para muitas aplicações práticas, o comportamento estado-estacionário do sistema é de interesse. Em outras palavras, o sistema é observado após long-run, isto é,

$$[\pi]_j = \lim_{t \to \infty}[\pi]_j(t). \tag{E.10}$$

Se $[\pi_j]$ existe para algum estado j, ele é chamado um estado-estacionário, equilíbrio, ou vetor intervalo de probabilidade do estado-estacionário:

$$[\pi] = [[\pi]_0, [\pi]_1, [\pi]_2, \cdots].$$

Se o limite (E.10) existe, segue que

$$\lim_{t\to\infty} \frac{d[\pi](t)}{dt} \to 0$$

não depende de t. Então, a equação diferencial intervalar estado-estacionário se reduz à equação algébrica

$$[\pi][\mathbf{Q}] = \mathbf{0}. \tag{E.11}$$

O limite (E.10) é independente do vetor de probabilidade intervalar estado inicial. Além disso, o vetor intervalar $[\pi][\mathbf{Q}] = \mathbf{0}$ é determinado, resolvendo um sistema intervalar de equações lineares, onde o lado esquerdo $[\mathbf{Q}] \in I(\mathbb{R})^{n\times n}$ é a matriz de coeficiente, e o lado direito da equação é $\mathbf{0} = (0\,,0\,,0\,,\ldots,0)^t \in \mathbb{R}^n$. Consequentemente, $[\pi][\mathbf{Q}] = \mathbf{0}$ está definido como uma família de equações lineares, tal que:

$$\tilde{\pi}\cdot\tilde{\mathbf{Q}} = 0 \;\; (\tilde{Q} \in [\mathbf{Q}]),$$

com a condição

$$\tilde{\pi}\cdot\mathbf{1} = 1$$

ou, reescrevendo,

$$[\pi][\mathbf{Q}] = \mathbf{0}, \tag{E.12}$$

com a condição

$$\sum_{all\, j} [\pi]_j = 1\ .$$

Assim, nós consideramos um sistema linear de equações no qual os coeficientes são intervalos. Nós estamos interessados em inclusões para o conjunto solução de (E.12), dado por

$$\Sigma([\mathbf{A}]\,,b) = \tilde{\pi} \in \mathbb{R}^n \mid \tilde{\pi}\cdot\tilde{\mathbf{A}} = \mathbf{b}$$

$$\textit{para todo } \tilde{\mathbf{A}} \in [\mathbf{A}] \textit{ e } \mathbf{b} = (0\,,0\,,0\,,\ldots,1)^t\ ,$$

onde, como feito usualmente, $[\mathbf{A}]$ é obtida com a eliminação de uma das equações (E.12) e mantendo a condição de normalização $\tilde{\pi}\cdot\mathbf{1} = 1$.

Os métodos diretos e iterativos (ver Capítulo 5) são considerados para encontrar o vetor *p.m.f* intervalar contendo o conjunto solução unida $\sum([\mathbf{A}],b)$. A qualidade do cerco (superestimação) de $\sum([\mathbf{A}],b)$ depende do sistema subjacente.

E.1 Observação

Neste apêndice foi feita a extensão intervalar dos conceitos CTMC para a ICTMC e, assim, derivar suas equações do estado estacionário e do estado transiente.

APÊNDICE F

Listagem de arquivos-M

Listagem do arquivo **verinthull.m**:

```
function [x,As]=verinthull(A,b)
%       VERINTHULL          Verified interval hull of the solution set of
%                            a system of interval linear equations.
%
%       For a square interval matrix A and a matching interval vector b,
%       [x,As]=VERINTHULL(A,b)
%       either computes a verified interval hull x of the solution set of the
%       system A*X=b, or finds a tight (”almost thin”) interval matrix As
%       which is a part of A and is verified to contain a real singular
%       matrix, or yields no verified result.
%
%       There are three possible outputs:
%
%       ~isnan(x) & isnan(As) : x is a verified interval hull of the solution set of A*X=b,
%        isnan(x) &~isnan(As) : As is a tight verified singular interval matrix contained in A,
%        isnan(x) & isnan(As) : no verified result.
%
%       In the third case, you may run [x,flag,As]=INTERVALHULL(A,b) for UNVERIFIED output.
%       Copyright 2007 Jiri Rohn
%
%       Based on the file INTERVALHULL, with unverified output suppressed and
%       verified output adapted to INTLAB standards.
%
%       WARRANTY
%
%       Because the program is licensed free of charge, there is
%       no warranty for the program, to the extent permitted by applicable
%       law. Except when otherwise stated in writing the copyright holder
%       and/or other parties provide the program ”as is” without warranty
%       of any kind, either expressed or implied, including, but not
%       limited to, the implied warranties of merchantability and fitness
%       for a particular purpose. The entire risk as to the quality and
%       performance of the program is with you. Should the program prove
%       defective, you assume the cost of all necessary servicing, repair
%       or correction.
%
gr=getround;
setround(0);
b=b(:); [m,n]=size(A);
x=repmat(infsup(NaN,NaN),n,1); As=repmat(infsup(NaN,NaN),m,n);
```

```
if ~(m==n&n==length(b))
    % error('dimensions do not agree'),
    setround(gr); return
end
if ~isintval(A), A=infsup(A,A); end        % allows for real input
if ~isintval(b), b=infsup(b,b); end
[xx,flag,Asing]=intervalhull(A,b);         % main part; returns verified or unverified output
switch flag
    case 'verified interval hull computed '
        x=xx;                              % verified interval hull
    case 'interval matrix verified singular '   % verified singular
        if ~any(any(isempty(Asing)))&all(all(in(Asing,A)))      % excluding case of verified singularity
            As=Asing;                                           % without singular matrix in A found
        end
    otherwise
        % error('no verified result; you may run [x,flag,As]=intervalhull(A,b) for unverified output'),
end
setround(gr);
```

Listagem do arquivo **verenclinthull.m**:

```
function x=verenclinthull(A,b)
%     VERENCLINTHULL        Verified enclosure of the interval hull of the solution set
%                            of a system of interval linear equations.
%
%     This is an INTLAB file. It requires to have INTLAB installed under
%     MATLAB to function properly.
%
%     For a square interval matrix A and a matching interval vector b,
%     x=VERENCLINTHULL(A,b)
%     either computes a verified enclosure x of the interval hull of the
%     solution set of A*X=b, or yields no verified result.
%
%
%     There are two possible outputs:
%
%     ~isnan(x)  : x is a verified enclosure of the interval hull of A*X=b,
%      isnan(x) : no verified result.
%
%     See also INTERVALHULL, VERIFYLSS.
%
%     Copyright 2006-2007 Jiri Rohn
%
%     Based on the file INTERVALHULL, with unverified output suppressed and
%     verified output adapted to INTLAB standards.
%
%     WARRANTY
%
%     Because the program is licensed free of charge, there is
%     no warranty for the program, to the extent permitted by applicable
%     law. Except when otherwise stated in writing the copyright holder
%     and/or other parties provide the program "as is" without warranty
```

```
%       of any kind, either expressed or implied, including, but not
%       limited to, the implied warranties of merchantability and fitness
%       for a particular purpose. The entire risk as to the quality and
%       performance of the program is with you. Should the program prove
%       defective, you assume the cost of all necessary servicing, repair
%       or correction.
%
```

Listagem do arquivo **verintervalhull.m**:

```
function [x,As,C]=verintervalhull(A,b,t)
%       VERINTERVALHULL         Verified interval hull of the solution set of
%                               a system of interval linear equations.
%
%       This is an INTLAB file. It requires to have INTLAB installed under
%       MATLAB to function properly.
%
%       x=verintervalhull(A,b) computes a verified interval hull x of the
%       solution set of a system of interval linear equations A*x=b, A square,
%       or yields a vector of NaN's.
%
%       [x,As]=verintervalhull(A,b) gives either a verified interval hull x,
%       or a very tight ("almost thin") interval matrix As which is a part of
%       A and is verified to contain a singular matrix. If no verified result
%       is achieved, then both x and As consist of NaN's.
%
%       [x,As,C]=verintervalhull(A,b) produces additional information in the
%       structure C:
%           C.xun       is an unverified interval hull, issued in case a verified
%                       one could not be computed,
%           C.Asun      is a thin unverified singular matrix, issued in case a verified
%           one could not be computed,
%           C.flag      describes verbally the output; it has one of the forms
%                       'verified interval hull computed '
%                       'interval matrix verified singular '
%                       'interval hull computed, result not verified '
%                       'interval matrix singular, result not verified',
%           C.orth      is the number of orthants inspected by the algorithm,
%           C.iter      is the total number of iterations of the core algorithm,
%           C.D         is a (possibly huge) matrix with rows of the form [z xl' xu'], where
%                       z is a plus/minus-one vector, and
%                       infsup(xl,xu) is a verified (possibly not optimal)
%                          enclosure of the intersection of the solution set
%                          with the z-orthant;
%                       if some plus/minus vector z is not present, then it is
```

```
%                    verified that the intersection of the solution set with
%                    the z-orthant is empty,
%        C.Xlower    is a matrix such that for each i the lower bound x.inf(i) was
%                    attained as the lower bound of the solution of the system
%                      (mid(A)-diag(y)*rad(A)*diag(z))*x=mid(b)+diag(y)*rad(b),
%                    where [y z] is the ith row of C.Xupper.
%        C.Xupper    is a matrix such that for each i the upper bound x.sup(i) was
%                    attained as the upper bound of the solution of the system
%                      (mid(A)-diag(y)*rad(A)*diag(z))*x=mid(b)+diag(y)*rad(b),
%                    where [y z] is the ith row of C.Xlower,
%        C.D, C.Xlower, and C.Xupper are empty if x is empty.
%
%     [x,As,C]=verintervalhull(A,b,1) [i.e., with additional input argument "1"]
%                  is the same as before, but in course of the computation
%                  it also produces screen output of the form
%                    ”Expected remaining time: ... sec.”
%                  This is a useful feature, considering the NP-hardness of
%                  the problem; however, it slows down the actual computation.
%
%     See also VERIFYLSS.

%     Copyright 2007-2008 Jiri Rohn
%
%      Based on the algorithm ”hull” described in
%      J. Rohn, A Handbook of Results on Interval Linear Problems,
%      posted at http://www.cs.cas.cz/~rohn
%
%     WARRANTY
%
%     Because the program is licensed free of charge, there is
%     no warranty for the program, to the extent permitted by applicable
%     law. Except when otherwise stated in writing the copyright holder
%     and/or other parties provide the program ”as is” without warranty
%     of any kind, either expressed or implied, including, but not
%     limited to, the implied warranties of merchantability and fitness
%     for a particular purpose. The entire risk as to the quality and
%     performance of the program is with you. Should the program prove
%     defective, you assume the cost of all necessary servicing, repair
%     or correction.
%
%     History
%     2007-11-30 first posted
%     2008-01-07 t=0 for unassigned t added
```

```
%
if (nargin==3)&&isequal(t,1)                    % t==1: display remaining time
   t=1;
else
   t=0;
end
[x,As,C]=intervalhull(A,b,t);                   % computation done by INTERVALHULL
[m,n]=size(A);
if isempty(x)                                   % output adapted to INTLAB standards
   x=repmat(infsup(NaN,NaN),n,1);
end
if isempty(As)
   As=repmat(infsup(NaN,NaN),m,n);
end
if isempty(C.xun)
   C.xun=repmat(NaN,n,1);
end
if isempty(C.Asun)
   C.Asun=repmat(NaN,m,n);
end
```

F.1 Observação

O motivo da presença da listagem foi da volatilidade das home-pages.

APÊNDICE G

Repositórios

As comunidades científicas dedicadas a Redes de Petri e Análise Intervalar contam com repositórios completos atualizados nas páginas *web*:

`http://www.informatik.uni-hamburg.de/TGI/PetriNets/`
e
`http://www.cs.utep.edu/interval-comp/`.

A finalidade dos repositórios é fornecer uma grande variedade de serviços online para suas comunidades internacionais. Estes serviços constituem, entre muitas outras coisas, visão geral (artigos, livros, tutorias, perguntas geralmente feitas (*FAQ - Frequently Asked Questions*), *links* relacionados, informações sobre Conferências Internacionais sobre Teoria e Aplicações, bibliografias, banco de dados de ferramentas, grupos de notícias e endereços.

APÊNDICE H

Ferramentas de Software na Análise Intervalar

Nas aplicações práticas da Análise Intervalar é quase imprescindível o uso de ferramentas de software (*software tools*). Várias bibliotecas que dão suporte a cálculos intervalares têm sido desenvolvidos, assim como *toolboxes* integrados em softwares matemáticos, tais como MATLAB, SCILAB, Maple e Mathematica. Muitos incluem rotinas para encontrar o conjunto solução de sistemas lineares intervalares. Todos fornecem meios que permitem o desenvolvimento de algoritmos numéricos, incluindo a solução de sistemas intervalares. A Sun Microsystem possui Compiladores C++ e Fortran95 com suporte nativo a tipo de dados intervalares com características que oferecem alta qualidade, confiabilidade e um ambiente para desenvolvimento de aplicações da aritmética intervalar. Os ambientes de programação de alto-nível permite a implementação de algoritmos numéricos usando uma notação próxima da notação matemática para o cálculo intervalar (como, por exemplo, no MATLAB *toolbox* INTLAB) através de várias rotinas eficientes e precisas.

A seguir, serão relacionados alguns softwares usados na análise intervalar, com ênfase em sistemas lineares intervalares, por estarem relacionados à solução do estado estacionário dos modelos ISPN proposto nesta tese.

- **INTLAB** é *toolbox* da aritmética intervalar para MATLAB. Ele fornece uma função para o cálculo de um cerco da solução do envoltório para um sistema linear intervalar com base em um método híbrido, que combina o método de Krawczyk e o método de Hansen-Bliek-Ning-Keafott-Neumaier. Segundo descrito na *home-page* do INTLAB [142]:

  > "O INTLAB é o *toolbox* do MATLAB para algoritmos de auto-validação. Ele abrange:
  >
  > 1. aritmética intervalar para dados reais e complexos incluindo vetores e matrizes,
  > 2. aritmética intervalar para as matrizes esparsas reais e complexas ,
  > 3. diferenciação automática,
  > 4. gradientes, hessianas, inclinações automáticas "*automatic slopes*", polinômios de uma variável e várias variáveis, funções padrão intervalares reais e complexas e aritmética intervalar de múltipla precisão.
  >
  > INTLAB tem vários milhares de usuários em mais de 40 países. O INTLAB é usado em muitas áreas, da verificação do caos a biologia da população, do projeto de controladores a provas assistidas por computadores, de EDPs a Re-

des de Petri. Na *home-page* do INTALB[1] há um arquivo, com uma lista de referências de trabalhos que usaram a *toolbox* INTLAB."

- **Int4Sci** É uma interface da análise intervalar para Scilab, baseada na bibliteca da aritmética intervalar C/C++ BIAS/PROFIL (discutidas abaixo). Essa interface cria uma biblioteca de alto nível para o Scilab através do carregamento e construção de uma biblioteca de baixo nível. Ela está disponível em `http://www-sop.inria.fr/coprin/logiciels/Int4Sci/`.

- **BIAS** (*Basic Interval Arithmetic Subroutines*) é uma biblioteca C++. Ela contém rotinas para calcular operações básicas entre vetor e/ou matriz intervalar. BIAS é disponibilizada junto com o pacote PROFIL, discutido abaixo.

- **PROFIL** (*Programmer's Runtime Optimized Fast Interval Library*) é uma biblioteca C++, baseada em BIAS, fornece dois solucionadores de sistemas lineares intervalares, baseados sobre métodos devido a Rump [141]. Ele está disponível em `http://www.ti3.tu-harburg.de/Software/PROFILEnglisch.html`.

- **ALIAS** (*Algorithms Library of Interval Analysis for equation Systems*) é uma biblioteca C++. Ela fornece rotinas que implantam métodos clássicos para cerco do conjunto solução dos sistemas lineares intervalares (métodos de Krawczyk, IGS, IGE, com ou sem precondicionamento) assim como métodos adicionais baseados na metodologia de programação com restrições e programação linear. ALIAS é parcialmente interfaceada com Maple. Ela está disponível em `http://www-sop.inria.fr/coprin/logiciels/ALIAS/`.

- **INTLIB** é uma biblioteca Fortran77, que consiste de sub-rotinas e funções para as operações intervalares elementares, algumas funções Fortran padrões e certas operações lógicas e utilitárias. Ela é disponível em `http://interval.louisiana.edu/pub/interval\_math/intlib/`.

- **b4m** é um toolbox para Matlab baseado na BIAS biblioteca ANSI C da aritmética intervalar, disponível em `http://www.ti3.tu-harburg.de/zemke/b4m/`.

- **IntpakX** é um pacote Maple. Ele entende a capacidade do Maple para computações intervalares. Ele está disponível em `http://www.math.uni-wuppertal.de/$sim$xsc/software/intpakX/d\_l\_c.html`.

- **XSC** (eXtensions for Scientific Computation) é um ambiente de programação de alto-nível com suporte da linguagem disponível para FORTRAN, PASCAL, e C (ACRITH-XSC, Fortran-XSC, PASCAL-XSC, e C- XSC). As rotinas para solução de problemas com verificação automática do resultado foram desenvolvidas para muitos problemas padrão da análise numérica, assim como para os sistemas de equações lineares ou não-lineares, para equações diferenciais e equações integrais, etc., como também para um

[1] `http://www.ti3.tu-harburg.de/rump/intlab/INTLABref.pdf(em10/01/2008)`

grande número de aplicações na engenharia e nas ciências naturais. Ele está disponível em `http://www.math.uni-wuppertal.de/~xsc/`.

Na análise de sistemas de filas muitas metodologias foram desenvolvidas com propriedades diferentes em relação à complexidade e à estabilidade numérica da computação. Nos estudos de S. Kempken, W. Lüther & G. Hassliner [68], os métodos de fatoração provaram ser eficientes para modelagem do tráfego em sistemas da telecomunicação e também para sistemas de filas clássicos. Desde que a taxa do tráfego em cada *link* da transmissão é limitada por uma capacidade ascendente constante, o modelo de tempo discreto com distribuições de suporte finito é então apropriado, incluindo o serviço determinístico como o caso mais simples. Por outro lado, as várias estruturas de autocorrelação do tráfego da Internet são consideradas usando modelagem semi-markoviana. Foi desenvolvida a ferramenta InterVerdiKom [68], que começa com a análise do estado estacionário dos sistemas semi-markovianos de tempo discreto, baseados na fatoração Wiener-Hopf e na fatoração polinomial. A ferramenta foi aplicada para avaliar o desempenho de sistemas de chaveamento com entrada das fontes *on-off* sobrepostas e para o tráfego de vídeo. A ferramenta emprega a aritmética intervalar para a verificação da exatidão dos resultados dos métodos da solução do estado estacionário e tem sido estendida recentemente para incluir o comportamento transiente do sistema.

Este volume foi tipografado em LaTeX na classe UFPEThesis (www.cin.ufpe.br/~paguso/ufpethesis).