



Construção de um Simulador de Redes de Petri Estocásticas

Trabalho de Conclusão de Curso

Engenharia de Computação

GABRIEL SANTOS OLIVEIRA

Orientador: Prof. Sérgio Murilo Maciel Fernandes



Gabriel Santos Oliveira

Construção de um Simulador de Redes de Petri Estocásticas

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Engenharia de Computação
Escola Politécnica de Pernambuco
Universidade de Pernambuco

Orientador: Prof. Sérgio Murilo Maciel Fernandes

Recife - PE, Brasil
dezembro de 2021

Gabriel Santos Oliveira Oliveira, Gabriel Santos
Construção de um Simulador de Redes de Petri Estocásticas/ Gabriel Santos
Oliveira. – Recife - PE, Brasil, dezembro de 2021
xv, 31 p.: il. ; 29 cm.

Orientador: Prof. Sérgio Murilo Maciel Fernandes

Trabalho de Conclusão de Curso – Engenharia de Computação
Escola Politécnica de Pernambuco
Universidade de Pernambuco, dezembro de 2021.

1. Redes de Petri Estocásticas. 2. Equações Diferenciais. 3. Cadeias de Markov. I.
Título. II. Fernandes, Sérgio Murilo Maciel. III. Universidade de Pernambuco.

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 28/12/2021, às 10h00min, reuniu-se para deliberar sobre a defesa da monografia de conclusão de curso do(a) discente **GABRIEL SANTOS OLIVEIRA**, orientado(a) pelo(a) professor(a) **SÉRGIO MURILO MACIEL FERNANDES**, sob título Construção e Análise de um Simulador de Redes de Petri Estocásticas, a banca composta pelos professores:

LUIS CARLOS DE SOUSA MENEZES (PRESIDENTE)
SÉRGIO MURILO MACIEL FERNANDES (ORIENTADOR)

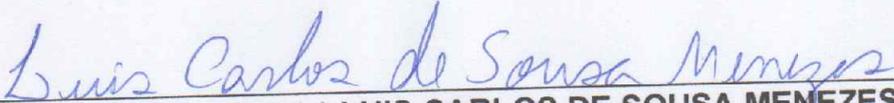
Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

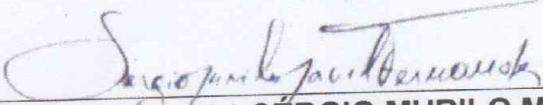
Aprovada Aprovada com Restrições* Reprovada

e foi-lhe atribuída nota: 9,0 (**NOVE**)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O(A) discente terá 7 dias para entrega da versão final da monografia a contar da data deste documento.


AVALIADOR 1: Prof (a) **LUIS CARLOS DE SOUSA MENEZES**


AVALIADOR 2: Prof (a) **SÉRGIO MURILO MACIEL FERNANDES**

AVALIADOR 3: Prof (a)

* Este documento deverá ser encadernado juntamente com a monografia em versão final.

Agradecimentos

Agradeço à minha família, em especial aos meus pais, José Paulo e Juliane Ferreira por todo o apoio e suporte durante a graduação e por acreditarem em mim durante todo o período em que estive na graduação.

Aos meus amigos que levarei para a vida toda, Adonias Luna, Edinaldo Júnior, Thiago Lourenço e Maria Gabrielly com os quais superei diversos desafios durante todos esses anos. Com eles eu pude viver alguns dos melhores anos da minha vida, com os quais tive a oportunidade de aproveitar ao máximo esse desafio que foi a minha formação acadêmica.

A todos os professores dedicados e comprometidos com os quais que tive a oportunidade de aprender, em especial aos do Ensino Fundamental e Médio, que muitas vezes não são lembrados, mas que para mim, foram também peças indispensáveis na minha decisão em me tornar um Engenheiro de Computação.

Ao meu professor orientador, Sérgio Murilo, pela paciência e dedicação durante o desenvolvimento deste trabalho, pelas discussões sobre arquiteturas de computadores, especialmente sobre arquiteturas especializadas e sistemas tolerantes a falhas, e por sempre me incentivar e apoiar, além de me introduzir a modelagem de sistemas com Redes de Petri. Ao professor Luis Menezes por despertar o meu interesse em linguagens de programação e pelas explicações sobre os diversos paradigmas de programação que existem.

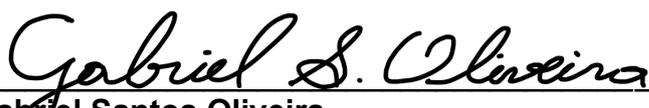
*“Truth is ever to be found
in simplicity, and not
in the multiplicity and
confusion of things.”*

Isaac Newton

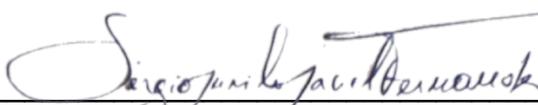
Autorização de publicação de PFC

Eu, **Gabriel Santos Oliveira** autor(a) do projeto de final de curso intitulado: **Construção e Análise de um Simulador de Redes de Petri Estocásticas**; autorizo a publicação de seu conteúdo na internet nos portais da Escola Politécnica de Pernambuco e Universidade de Pernambuco.

O conteúdo do projeto de final de curso é de responsabilidade do autor.



Gabriel Santos Oliveira



Orientador(a): **Sérgio Murilo Maciel Fernandes**

Coorientador(a):



Prof, de TCC: **Daniel Augusto Ribeiro Chaves**

Data: 28/12/2021

Resumo

Embora as redes de Petri sejam amplamente utilizadas para a descrição e análise de sistemas concorrentes, paralelos e distribuídos, existem poucas ferramentas de código aberto disponíveis atualmente. Apesar de algumas das mais proeminentes estarem iniciando a divulgação do seu código-fonte, o acesso a *frameworks* de alta qualidade permanece limitado aos centros de pesquisa onde foram desenvolvidas. O *software* desenvolvido por este projeto serve então como uma tentativa inicial para o desenvolvimento de uma *framework* para a descrição, análise e simulação de redes de Petri, inicialmente focado em redes de Petri estocásticas, em particular as redes reativas. O *software* desenvolvido comportou-se de maneira esperada ao simular corretamente dois modelos de redes de Petri estocásticas.

Palavras-chave: Redes de Petri Estocásticas. Equações Diferenciais. Cadeias de Markov. Simulação.

Abstract

Although Petri nets are widely used for the description and analysis of concurrent, parallel, and distributed systems, there are few open source tools currently available. Although some of the most promising ones are beginning to release their source code, access to high quality frameworks remains limited to the research centers where they were developed. The software developed by this project then serves as an initial attempt at developing a framework for the description, analysis and simulation of Petri nets, initially focused on stochastic Petri nets, in particular reactive nets. The developed software behaved as expected by correctly simulating two stochastic Petri net models.

Keywords: Stochastic Petri Nets. Differential Equations. Markov Chains. Simulation.

Lista de ilustrações

| | |
|--|----|
| Figura 1 – Descrição gráfica de uma Rede de Petri, sua marcação inicial e a distribuição de <i>tokens</i> após o disparo das transições a , b , e , f e d | 15 |
| Figura 2 – Modelagem de uma rede <i>LAN</i> utilizando a ferramenta <i>CPN Tools</i> . . . | 17 |
| Figura 3 – Um <i>overview</i> da arquitetura da ferramenta <i>GreatSPN</i> | 20 |
| Figura 4 – A visualização do modelo SEIRD no editor gráfico da ferramenta <i>GreatSPN</i> | 24 |
| Figura 5 – A visualização do modelo Lotka-Volterra no editor gráfico da ferramenta <i>GreatSPN</i> | 25 |
| Figura 6 – O resultado da simulação do modelo SEIRD. | 27 |
| Figura 7 – O resultado da simulação do modelo Lotka-Volterra. | 27 |

Lista de tabelas

| | |
|--|----|
| Tabela 1 – Resultados para a simulação do modelo SEIRD | 26 |
|--|----|

Lista de abreviaturas e siglas

| | |
|---------|---|
| API | <i>Application Programming Interface</i> |
| PN | <i>Petri Net</i> ou <i>Place-Transition Net</i> |
| SPN | <i>Stochastic Petri Net</i> |
| GSPN | <i>Generalized Stochastic Petri Net</i> |
| ODE | <i>Ordinary Differential Equations</i> |
| SDE | <i>Stochastic Differential Equations</i> |
| GpenSIM | <i>General Purpose Petri Net Simulator</i> |

Sumário

| | | |
|----------|--|-----------|
| 1 | INTRODUÇÃO | 12 |
| 1.1 | Motivação e Caracterização do Problema | 12 |
| 1.2 | Objetivo Geral | 13 |
| 1.3 | Objetivos Específicos | 13 |
| 1.4 | Estrutura do Documento | 14 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 15 |
| 2.1 | Fundamentos e Conceito de Redes de Petri | 15 |
| 2.1.1 | Propriedades Comportamentais | 16 |
| 2.1.2 | Redes de Petri Coloridas | 16 |
| 2.2 | Redes de Petri Estocásticas | 17 |
| 2.2.1 | Comparação a Redes de Petri | 18 |
| 2.2.2 | Cadeias de Markov | 18 |
| 2.3 | De Cadeias de Markov à Equações Diferenciais | 19 |
| 2.4 | Simulando Redes de Petri e suas Extensões | 19 |
| 2.4.1 | Ferramentas Disponíveis | 20 |
| 2.5 | Comparação com Redes de Reações | 21 |
| 3 | PROJETO DE SOFTWARE | 22 |
| 3.1 | <i>Julia</i> | 22 |
| 3.2 | Organização do Simulador | 23 |
| 3.2.1 | Definição dos Modelos no Simulador | 23 |
| 3.3 | Simulação | 24 |
| 3.4 | Redes Seleccionadas | 24 |
| 4 | RESULTADOS | 26 |
| 4.1 | Modelo SEIRD | 26 |
| 4.2 | Modelo Lotka-Volterra | 26 |
| 5 | CONCLUSÕES E TRABALHOS FUTUROS | 28 |
| | REFERÊNCIAS | 29 |

1 Introdução

Este trabalho de conclusão de curso propõe-se ao estudo e desenvolvimento de um projeto de sistema computacional (de *software*), que seja capaz de simular Redes de Petri Estocásticas, em particular, rede de Petri Estocásticas *reativas*, comumente utilizadas na análise de processos bio-químicos e sistemas ecológicos.

A escassez de ferramentas de código aberto para a análise e estudo de Redes de Petri Estocásticas, e o alto custo computacional envolvido, foram alguns dos motivos para a escolha deste tema.

Este capítulo descreve a introdução da monografia, e está organizado em 3 seções. Na Seção 1.1 são descritas tanto a motivação para a execução deste trabalho, quanto a definição do problema.

Posteriormente, nas Seções 1.2 e 1.3 são apresentados os objetivos gerais e específicos, bem como a proposta de solução do projeto. Por fim, a Seção 1.4 detalha a organização da monografia.

1.1 Motivação e Caracterização do Problema

As Redes de Petri são uma ferramenta ideal para a descrição, análise e estudo de sistemas discretos, distribuídos e/ou concorrentes: elas permitem a descrição destes sistemas de forma simples, além de oferecer diversos métodos de análise para as mais diversas características do sistema em estudo. Um exemplo de aplicação prática é a detecção de *deadlocks* em um mecanismo de controle para processamento concorrente.

A simplicidade das Redes de Petri, entretanto, limita as suas aplicações. Descrever sistemas complexos utilizando a notação originalmente proposta torna-se inviável ou até mesmo impossível. Com isso, foram desenvolvidas diversas extensões para a notação original, visando facilitar a descrição destes sistemas: Redes Coloridas permitem uma descrição concisa de sistemas com elementos distintos para cada transição; Redes Hierárquicas facilitam a modelagem de subsistemas e as interações entre eles; Redes Priorizadas incluem a noção de prioridades e preferências no disparo de transições, eliminando problemas de indecisão.

Dentre as extensões disponíveis, Redes de Petri Estocásticas, redes que incluem o conceito de tempo, são as mais interessantes, pois, podem ser utilizada para realizar a análise de desempenho de um sistema, auxiliar na identificação de gargalos, identificar alternativas para maximizar o desempenho ou minimizar os recursos necessários para a execução de processos. Entretanto, devido à inclusão do conceito de tempo, estas redes

necessitam de métodos de análise alternativos, muitos dos quais são baseados em Cadeias de Markov e métodos estatísticos, possuindo um alto custo computacional e implementações complexas.

Diversas ferramentas estão disponíveis para se trabalhar com redes de Petri, no entanto, a quantidade e qualidade de ferramentas com *código aberto* disponíveis para se trabalhar com redes de Petri é pequeno: as ferramentas desenvolvidas pelos grupos de pesquisa, em sua maioria, tem acesso restrito ou limitado, tanto a própria ferramenta quanto ao seu código-fonte.

Isso dificulta o trabalho de pesquisadores e o estudo de modelos novos ou não-convencionais – sem a possibilidade de reutilizar algoritmos já desenvolvidos, o pesquisador é limitado a implementar e verificar pessoalmente tais algoritmos, ou abandonar o seu projeto de pesquisa. Mais recentemente, algumas das ferramentas mais conhecidas iniciaram a disponibilização parcial de algumas de suas bibliotecas, o que dá mais liberdade aos pesquisadores que não estão envolvidos com estes grandes centros de pesquisa, embora o problema ainda persista.

Portanto, este trabalho visa desenvolver um simulador de Redes de Petri Estocásticas baseado em equações diferenciais, capaz de simular estas redes de forma eficiente e que seja simples o suficiente para que um futuro usuário possa incluir funcionalidades desejadas.

1.2 Objetivo Geral

Este trabalho tem como objetivo principal o estudo e implementação de um *software* para a simulação de Redes de Petri Estocásticas que seja simples, eficiente e compacto, facilitando o seu entendimento e futuras extensões às suas funcionalidades.

1.3 Objetivos Específicos

A implementação do projeto também envolveu outros fins, os quais foram atingidos:

- Estudo sobre Redes de Petri;
- Estudo sobre Redes de Petri Estocásticas;
- Proposta e desenvolvimento de um simulador em software;
- Aplicação do *software* desenvolvido para análise de algumas redes;
- Estudo dos desafios envolvidos no desenvolvimento de uma *framework* completa para especificação, análise e simulação de Redes de Petri e suas extensões;

- Identificação de possíveis melhorias sobre o *software* desenvolvido.

1.4 Estrutura do Documento

Este trabalho está dividido em 5 capítulos, incluindo este capítulo que conta com uma introdução a respeito do tema e objetivos do projeto. Em seguida o Capítulo 2 traz um estudo a respeito das Redes de Petri, suas características e como estendê-las para sistemas temporais, e demais conceitos necessários para a compreensão do sistema desenvolvido neste trabalho.

Já no Capítulo 3, é descrito as etapas do processo que resultou no sistema proposto. O Capítulo 4 apresenta os resultados obtidos e dados de testes efetuados com o sistema em funcionamento.

Finalmente, na Conclusão, encerro o trabalho com uma discussão a respeito do que foi desenvolvido, uma comparação com as ferramentas atualmente disponíveis e possíveis melhorias, além de novas ideias, para o projeto.

2 Fundamentação Teórica

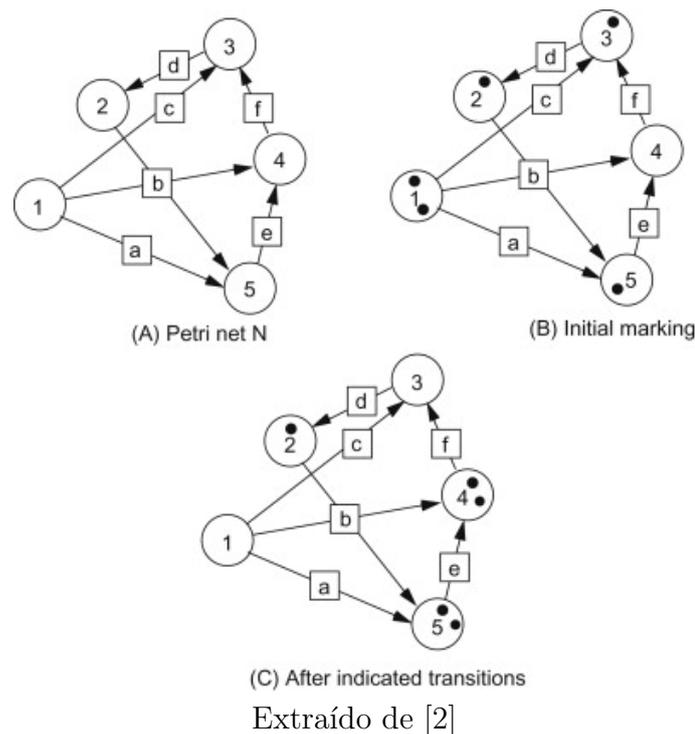
Este capítulo busca apresentar alguns conceitos fundamentais a respeito das Redes de Petri, Redes Estocásticas e os métodos de análise utilizados para cada uma delas.

2.1 Fundamentos e Conceito de Redes de Petri

As Redes de Petri foram originalmente desenvolvidas em 1962 por Carl Petri[1] para a análise de reações químicas de uma maneira simples. Logo após publicar a sua dissertação, Carl Petri notou que a notação que utilizou poderia ser estendida para descrever e analisar qualquer sistema discreto, concorrente e/ou paralelo.

Uma Rede de Petri, é composta por dois elementos: *Places* ou **Lugares**, elementos *passivos*, representados por um círculo – a representação mais comum – ou uma elipse, onde *tokens* são armazenados e; *Transitions* ou **Transições**, representados por um quadrado, um retângulo – ambos comumente utilizados – ou uma simples barra. As **transições** são elementos *ativos* que produzem, consomem ou transportam os *tokens* entre **lugares**. Uma **transição** "dispara" assim que possui o número de *tokens* necessários em suas entradas, consumindo os *tokens* nos locais de entrada e gerando *tokens* nos locais de saída.

Figura 1 – Descrição gráfica de uma Rede de Petri, sua marcação inicial e a distribuição de *tokens* após o disparo das transições **a**, **b**, **e**, **f** e **d**.



Além destes dois elementos, uma rede de Petri inclui mais dois componentes gráficos: os *tokens*, que representam algum recurso discreto no sistema modelado, e *Arcos*, setas que descrevem o fluxo de *tokens* entre os elementos da rede. Um *arco* só pode existir entre componentes distintos – entre um lugar e uma transição ou uma transição e um lugar. Estes componentes são utilizados apenas na notação gráfica.

Por fim, redes de Petri possuem o conceito de *pré-condições* e *pós-condições*, I^- e I^+ , respectivamente: as *pré-condições* representam a quantidade de *tokens* necessários para habilitar uma **transição**, os quais devem estar presentes em **lugares** específicos para cada transição; as *pós-condições* representam a localização e a quantidade de *tokens* após o disparo da **transição**. É comum a representação destes conceitos como uma única matriz de incidências $I = I^- - I^+$, simplificando a visualização do deslocamento de *tokens* após o disparo de determinada transição.

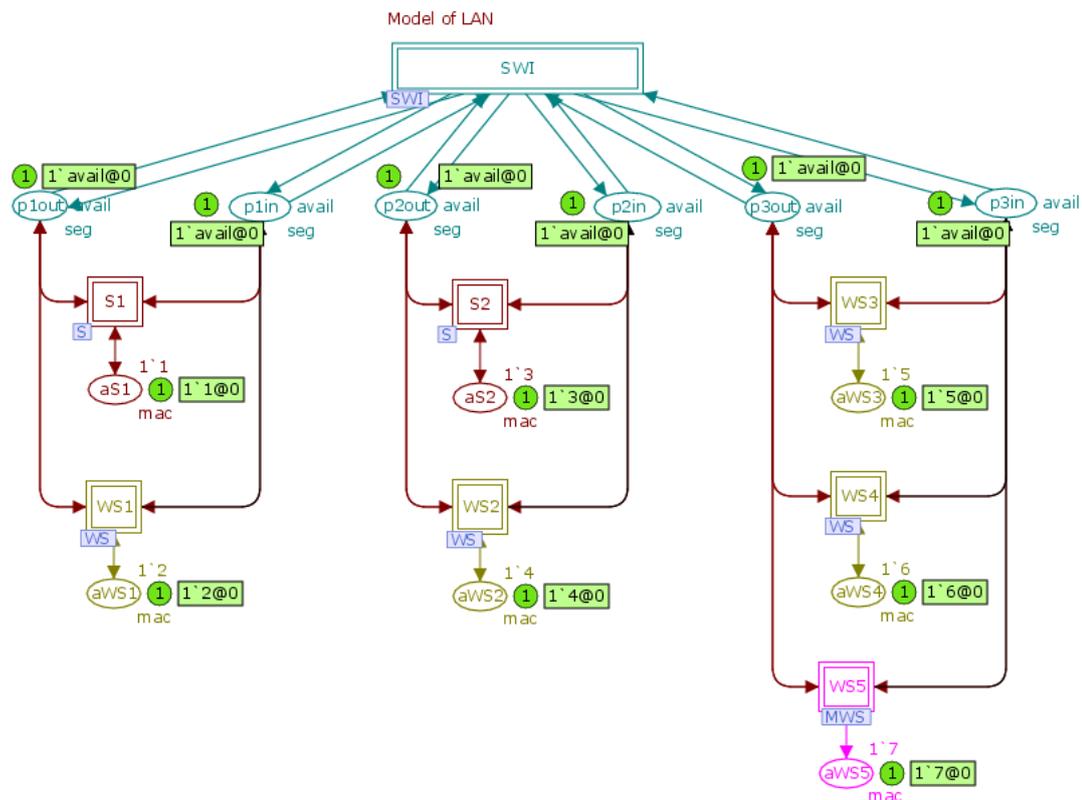
2.1.1 Propriedades Comportamentais

A principal utilidade das Redes de Petri é a análise de propriedades comportamentais e *funcionais* do sistema descrito. São elas:

- **Alcançabilidade:** A alcançabilidade consiste em determinar se um dado estado M_n pode ser alcançado a partir de um estado inicial M_0 e qual a sequência de disparos realizada para alcançar tal estado.
- **Limitabilidade:** o número máximo de *tokens* que um lugar qualquer pode conter para todas as marcações possíveis. A rede é dita *k-limitada* se o número de *tokens* não exceder um valor finito. Uma rede de Petri é dita *segura* ou *estritamente limitada* se é 1-limitada.
- **Conservatividade:** Uma rede é dita *conservativa* se o número de *tokens* permanece constante para todas as marcações possíveis.
- **Vivacidade:** Uma rede é *viva* se, para todas as marcações possíveis da rede, existir pelo menos uma transição habilitada para disparo. Se uma rede é *viva*, é impossível que ela sofra de *deadlocks*.
- **Reversibilidade:** Uma rede é *reversível* se, para cada uma das marcações possíveis, há uma sequência de transições que retorne a rede para a sua marcação original M_0 .

2.1.2 Redes de Petri Coloridas

Devido a sua notação simples, descrever certos sistemas através de redes de Petri se torna demasiadamente complexo, levando ao desenvolvimento de diversas extensões. Enquanto algumas extensões mantêm a "compatibilidade" com os métodos de análise

Figura 2 – Modelagem de uma rede LAN utilizando a ferramenta *CPN Tools*

Fonte: extraído de [3]

aplicáveis as redes de Petri, outras quebram essa compatibilidade adicionando conceitos mais complexos. Dentre as extensões que mantém compatibilidade, as redes de Petri *coloridas* são as mais utilizadas.

Uma rede de Petri colorida [4][5] estende o conceito de *tokens*, permitindo a existência de múltiplos *tokens* com "cores" diferentes, simplificando a descrição de sistemas com recursos distintos. Esta extensão é tão utilizada que é mais comum que as ferramentas disponíveis a implementem. Dentre as diversas ferramentas disponíveis para a análise de redes de Petri, a mais conhecida é a *CPN Tools*, desenvolvida pela Universidade de Eindhoven, disponível em [6].

2.2 Redes de Petri Estocásticas

Enquanto redes de Petri são bastante utilizadas para a verificação de características *funcionais* de sistemas, não é possível utilizá-las para a verificação de características *quantitativas* dos sistemas descritos, visto que não possuem um conceito de *tempo* para o disparo de transições. Em particular, duas características importantes não podem ser analisadas: a *performance* do sistema sob diferentes condições, e a *dependabilidade* do sistema. Devido ao desejo de utilizar a mesma descrição para verificação de características

quantitativas e qualitativas do sistema foram desenvolvidas as redes de Petri *estocásticas*[7].

Estas redes, incluem o conceito de tempo definindo que o disparo das transições ocorre após um retardo exponencialmente distribuído. Isso implica que as *SPNs* são isomórficas às Cadeias de Markov, permitindo a utilização de métodos já conhecidos para determinar o desempenho destas redes. Além das *SPNs*, também foram desenvolvidas as redes de Petri estocásticas generalizadas[8], que incluem transições não-estocásticas de disparo imediato. Conceitualmente, tanto *SPNs* quanto *GSPNs* incluem um conjunto não-nulo $\Lambda = (\lambda_1, \dots, \lambda_n)$ que representa o retardo de disparo para cada transição $T = (t_1, \dots, t_n)$.

2.2.1 Comparação a Redes de Petri

Ao adicionar tempo a notação das redes de Petri, o comportamento das redes muda consideravelmente: uma rede de Petri que normalmente sofreria de *dealock* pode "voltar a vida" em um momento posterior; redes reversíveis podem tornar-se irreversíveis devido à mudança de probabilidade de disparo de transição conforme o atual estado da rede.

2.2.2 Cadeias de Markov

Um processo estocástico qualquer é dito uma cadeia de Markov se, e somente se, ele respeitar a *propriedade de Markov*[9], descrita pela equação:

$$\begin{aligned} P[\chi(t)|\chi(t_n) = x_n, \chi(t_{n-1}) = x_{n-1}, \dots, \chi(t_0) = x_0] \\ = P[\chi(t) = x|\chi(t_n) = x_n], \quad t > t_n > t_{n-1} > \dots > t_0 \end{aligned} \quad (2.1)$$

Em termos mais simples, dado um processo estocástico, ele também pode ser tratado como uma cadeia de Markov se, e somente se, a *probabilidade do próximo estado depender apenas do estado atual do processo*. Outra característica desejável para processos estocásticos é a sua *estacionariedade* ou *homogeneidade*: processos nos quais as probabilidades são invariantes a mudanças no tempo, descrita pela equação

$$P[\chi(t+s) = x|\chi(t_n+s) = x_n] = P[\chi(t) = x|\chi(t_n) = x_n]. \quad (2.2)$$

Considerando as limitações impostas pela *propriedade de Markov* e o grande interesse em estudar apenas processos estacionários, apenas duas distribuições de probabilidade satisfazem ambas. São elas a distribuição *exponencial negativa*

$$f_\chi(y) = \lambda e^{-\lambda y}, \quad y \geq 0 \quad (2.3)$$

para processos estocásticos em tempo contínuo, e a *distribuição geométrica* de variáveis aleatórias

$$p_\eta(n) = P[\eta = n] = q^{n-1}(1-q), \quad n = 1, 2, 3, \dots; \quad 0 \leq q < 1. \quad (2.4)$$

para processos em tempo discreto. Uma prova de que estas distribuições satisfazem a propriedade (2.2) é descrita em detalhes em [10].

Enquanto a transformação de processos sem memória é um processo simples, modelar processos com memória se torna mais complexo devido à explosão de estados da cadeia de Markov resultante. Dado um sistema com memória, a cadeia de Markov resultante possuirá N^m estados, onde N é a quantidade de estados do processo estocástico e m é o número de estados necessários para decidir o próximo estado.

2.3 De Cadeias de Markov à Equações Diferenciais

Embora o estudo e análise das cadeias de Markov seja um campo bem estudado e fundamentado, os métodos de análise são computacionalmente custosos, limitando as redes que podem ser analisadas em tempo hábil. A aproximação de processos estocásticos pode ser realizada por um conjunto de *ODEs*, fato que Kurtz prova em [11] – quando o número de *tokens* no sistema é grande o suficiente, a aproximação é aceitável.

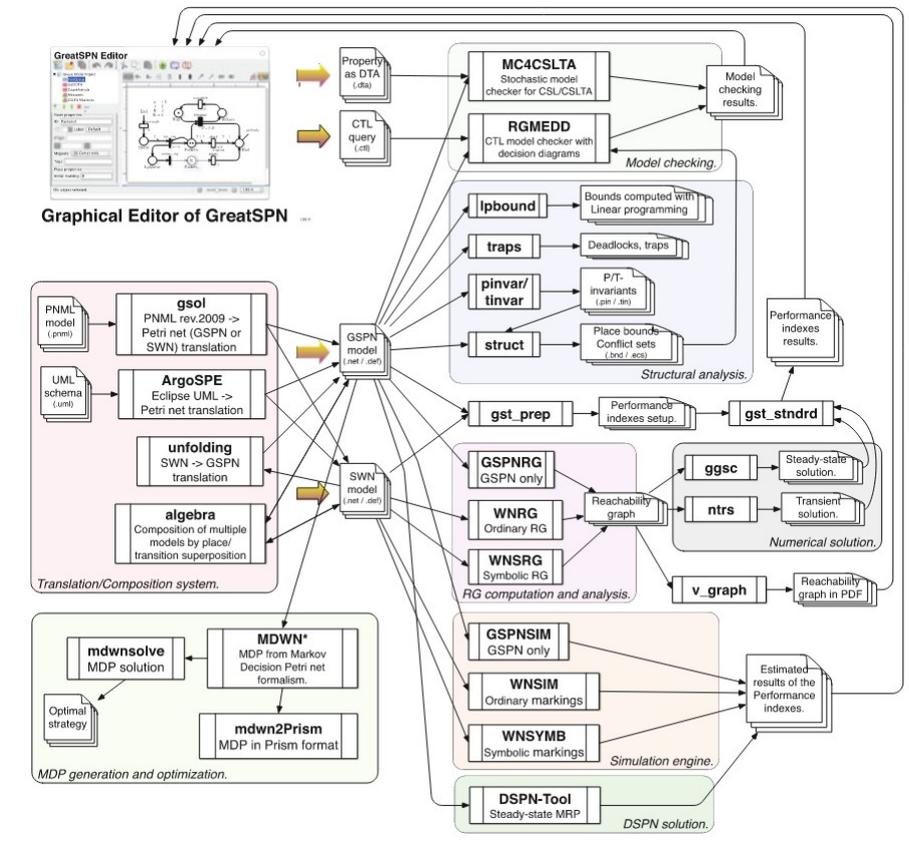
Não satisfeito com os resultados, Kurz propôs a *fluidização* do processo estocástico, ao qual se adiciona um ruído, gerando um conjunto de *SDEs* [12] que é mais estável que as *ODEs* originalmente propostas, gerando resultados superiores à solução original, embora não livres de erros de aproximação. Outras melhorias foram propostas e testadas, com base nos métodos propostos por Kurz em [13], solucionando alguns problemas envolvidos na simulação de redes com uma quantidade pequena de *tokens* ou com uma grande variação na quantidade de *tokens*, além de oferecer um desempenho melhor que a simulação da cadeia de Markov original.

Já em [14] foi realizado um estudo visando derivar um conjunto de *ODEs* e um conjunto de soluções iniciais para o sistema em estudo, a partir de uma rede de Petri discreta. Os autores também providenciam um pequeno tutorial para a aplicação dos métodos discutidos em outras redes além de uma discussão direcionada à aplicação de redes de Petri para o estudo de processos bioquímicos e as vantagens em comparação com outros métodos.

2.4 Simulando Redes de Petri e suas Extensões

O grande interesse em minimizar custos e prejuízos torna a modelagem, simulação e análise de sistemas um processo indispensável em diversas indústrias. Como as redes de Petri possuem uma notação simples, além de diversas extensões que permitem a análise de diversos aspectos e qualidades do sistema, elas ainda são uma ferramenta muito utilizada.

Desde a modelagem e verificação de bugs em componentes de hardware[15][16], a análise de sistemas de transporte público como, por exemplo, o metrô de Paris[17] e

Figura 3 – Um *overview* da arquitetura da ferramenta *GreatSPN*.

Fonte: extraído de [21]

a otimização das rotas para uma frota de veículos elétricos, incluindo minimização de paradas para reabastecimento[18], e a simulação da dependabilidade e do consumo de energia utilizada para a operação de diferentes organizações de um *data center*[19][20].

2.4.1 Ferramentas Disponíveis

Diversas ferramentas estão disponíveis para a modelagem, simulação e análise de redes de Petri, cada uma implementando um conjunto de análises e extensões conforme as necessidades dos centros de pesquisa onde são desenvolvidas. Alguns exemplos incluem:

- *Oris*[22], que suporta a simulação de redes de Petri discretas, redes de Petri estocásticas comuns e não-Markovianas. Estas funções podem ser utilizadas a partir de uma interface gráfica ou, de forma parametrizada, diretamente utilizando as bibliotecas desenvolvidas em Java através de uma API;
- O conjunto de ferramentas *GreatSPN*[21], oferecendo um conjunto de analisadores e simuladores para diversas extensões de redes de Petri, entre elas as Redes Estocásticas Simétricas[23], uma classe de redes de Petri estocásticas que incluem o conceito de

cores. Também oferece uma interface gráfica simples e bem integrada com as demais ferramentas;

- A biblioteca *SNAKES*[24], desenvolvido em Python, com um foco maior na modelagem e verificação das redes. Também oferece um simulador para diversos tipos de redes de Petri, além de um compilador especializado, acelerando a simulação de redes complexas;
- A *toolbox* de Redes de Petri disponível para Matlab[25], que possui uma integração simples com outras *toolboxes* para controle e análise de processos, além de oferecer uma interface gráfica relativamente simples;
- A biblioteca **GpenSIM**[26], uma alternativa mais modular e prática de utilizar em comparação a *toolbox* para redes de Petri, embora não possua uma interface gráfica;

2.5 Comparação com Redes de Reações

Um caso particular de Redes de Petri são as *Redes de Reações* utilizadas em pesquisas bio-químicas. Embora possuam nomes distintos, redes reativas são redes de Petri: ambas possuem os mesmo elementos, embora usem uma nomenclatura diferente – ao invés de lugar temos **espécies**, e ao invés de transições temos **reações**. De fato, redes de Petri foram originalmente desenvolvidas para a visualização e análise de reações químicas. Além desta diferença de nomenclatura, elas são amplamente utilizadas para a modelagem de sistemas bio-químicos, e mais recentemente, também são utilizadas para simular a estabilidade de sistemas e as concentrações dos componentes utilizados, como foi realizado em [14].

3 Projeto de Software

O simulador desenvolvido neste trabalho consistiu em uma pequena biblioteca, capaz de simular *SPNs* utilizando a modelagem proposta descrita em [27], onde uma rede de Petri discreta é convertida em uma série de equações diferenciais, similarmente ao que é proposto em [11] e [13]. As etapas do desenvolvimento do simulador são descritas em mais detalhes, além de explicações breves das ferramentas utilizadas.

3.1 *Julia*

Listing 3.1 – Definição da estrutura de Rede de Petri com lugares e transições em *Julia*.

```
struct Model{StateType , TransitionType }
    S:: StateType
    T:: TransitionType
end
```

O simulador foi desenvolvido na linguagem de programação *Julia*[28] devido a sua simplicidade, alto desempenho e a disponibilidade de bibliotecas para a análise de equações diferenciais de alta qualidade.

Julia foi desenvolvida para solucionar o problema das "duas linguagens", onde uma aplicação é desenvolvido em uma linguagem dinâmica de alto nível, como Python, Matlab ou R, devido a maior velocidade de desenvolvimento e simplicidade na implementação de algoritmos. Posteriormente, as funções com maior custo computacional são re-escritas em uma linguagem compilada, como C/C++ ou Fortran, com a linguagem de alto nível servindo apenas para preparar os dados que são repassados a estes *kernels* otimizados.

Isto dificulta o processo de desenvolvimento, visto que há uma grande diferença entre o conhecimento técnico necessário para o desenvolvimento em cada uma destas linguagens. É comum que os algoritmos sejam desenvolvidos por pesquisadores e especialistas de área, enquanto a otimização posterior é realizada por engenheiros.

Julia evita este problema, pois oferece uma sintaxe simples, similar a Python, mas com uma especificação mais rigorosa quanto a tipagem de variáveis e manipulação do ambiente de programação, minimizando as possíveis alterações em tempo de execução, resultando em um processo de otimização mais simples. Por fim, o processo de geração de código nativo é realizado pela framework LLVM, também utilizada na compilação de linguagens como C/C++, OpenCL, Cuda, Rust, entre outras. O código nativo é gerado em tempo de execução, assim que uma função é chamada, o que permite a especialização

do código gerado – não é preciso gerar funções que aceitam parâmetros genéricos, apenas para os parâmetros utilizados.

3.2 Organização do Simulador

Durante a construção do simulador, o foco era em ter uma representação compacta e simples para as redes de Petri, resultando em uma estrutura com apenas dois componentes: estados, denotados por S , e transições, denotados por T ; as taxas de retardo para o disparo das transições são definidas de maneira *ad-hoc* no momento da simulação, evitando a necessidade de re-criar a rede quando necessário alterar as taxas.

Foi utilizada a biblioteca *DiffEq.jl*[29], disponível também disponível nas linguagens Python e R, para solucionar as equações diferenciais geradas a partir das redes. Ela oferece diversos algoritmos para a resolução das equações diferenciais, um bom desempenho e aceleração da solução utilizando GPUs.

3.2.1 Definição dos Modelos no Simulador

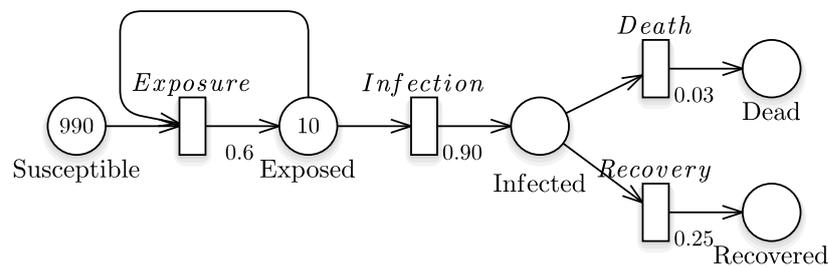
Listing 3.2 – Definição da rede SEIRD no simulador desenvolvido

```

1 S = [:S, :E, :I, :R, :D]
2
3 T = LVector(
4     exp=(LVector(S=1, I=1), LVector(I=1, E=1)),
5     inf=(LVector(E=1), LVector(I=1)),
6     rec=(LVector(I=1), LVector(R=1)),
7     die=(LVector(I=1), LVector(D=1)),
8 )
9
10 seird = Petri.Model(S, T)
11
12 u0 = LVector(S=990.0, E=10.0, I=0.0, R=0.0, D=0.0)
13 tspan = (0.0, 40.0)
14 delta = LVector(exp=0.9/sum(u0), inf=0.9, rec=0.25, die=0.03)
15
16 prob, cb = SDEProblem(seird, u0, tspan, delta)
17 sol = StochasticDiffEq.solve(prob, LambdaEM(), callback=cb)

```

Para definir uma Rede de Petri no simulador desenvolvido, são necessários apenas três elementos: os lugares, as transições e as taxas de disparo para cada transição. Como exemplo, temos a descrição do modelo SEIRD em 3.2: as transições são definidas na linha

Figura 4 – A visualização do modelo SEIRD no editor gráfico da ferramenta *GreatSPN*.

Fonte: o autor

1; as transições são definidas nas linhas 3 à 8; o modelo é então transformado em um conjunto de equações diferenciais na linha 10; nas linhas 12, 13 e 14 definimos a marcação inicial da rede M_0 , o espaço de tempo a ser simulado e as taxas de disparo das transições, respectivamente. Nas linhas 16 e 17 as equações são passadas para a biblioteca para serem simuladas.

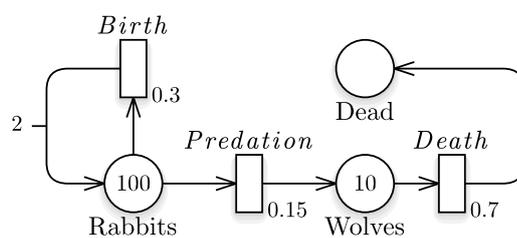
3.3 Simulação

No seu estado atual, o simulador desenvolvido consegue simular *SPNs* reativas, como aquelas utilizadas para o estudo de reações químicas e as interações entre espécies dentro de um ecossistema. Desta forma, é possível analisar o comportamento destes sistemas através do tempo e verificar, se possível para o sistema especificado, as condições de estabilização da rede.

3.4 Redes Selecionadas

Foram utilizadas duas redes de Petri Estocásticas para avaliar os resultados do simulador desenvolvido: o modelo epidemiológico SEIRD (*Susceptible-Exposed-Infected-Recovered-Dead*), e o modelo ecológico Lotka-Volterra. Ambos são modelos relativamente simples que modelam o comportamento da propagação de uma doença infecto-contagiosa em uma população, e a interação entre presas e predadores, respectivamente. As duas redes utilizadas já possuem descrições bem conhecidas por equações diferenciais, o que simplifica a verificação do correto funcionamento do simulador desenvolvido.

Figura 5 – A visualização do modelo Lotka-Volterra no editor gráfico da ferramenta *GreatSPN*.



Fonte: o autor

4 Resultados

Este capítulo descreve os parâmetros utilizados e os resultados obtidos para os dois modelos analisados

4.1 Modelo SEIRD

O modelo SEIRD analisado possui quatro transições: exposição de uma pessoa suscetível a uma pessoa já exposta, com uma taxa de disparo decrescente, inicialmente com valor 0.9 e decaindo linearmente conforme a fórmula $\frac{0.9}{\text{Indivíduos Suscetíveis}}$; a infecção de um indivíduo exposta, com uma taxa de 0.9; a recuperação de um indivíduo infectado, com uma taxa de 0.25, e; a morte de um indivíduo infectado, com uma taxa de 0.03.

Os resultados para a simulação deste modelo estão na tabela 1. Por eles, é possível notar a presença de erros de aproximação, visto que a simulação inicia com um total de 1000 indivíduos – 990 suscetíveis e 10 expostos – e termina com aproximadamente 1026 indivíduos.

4.2 Modelo Lotka-Volterra

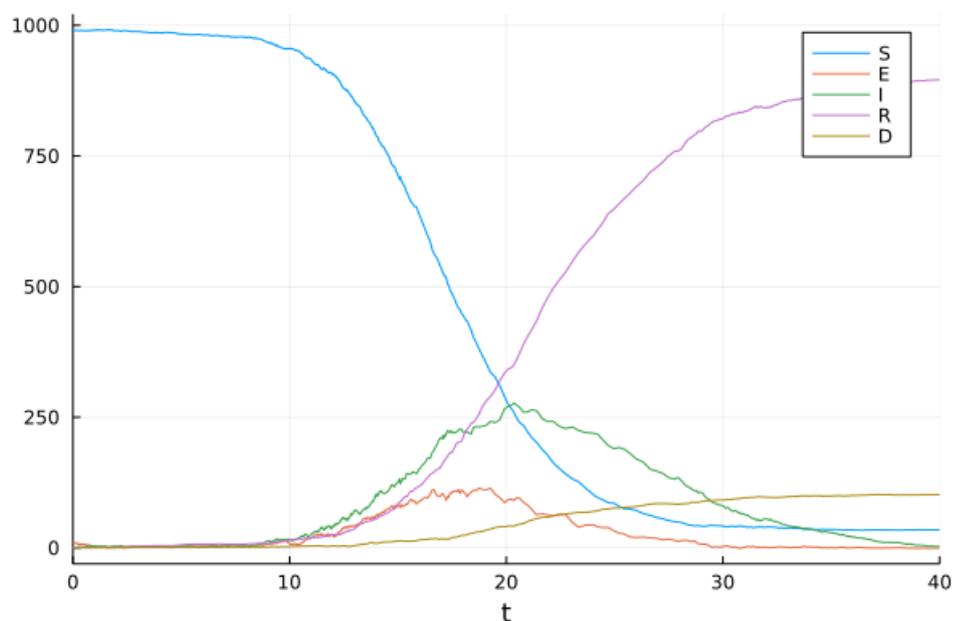
O modelo Lotka-Volterra é possui apenas três transições: nascimento, onde um coelho gera outro adicional, com uma taxa de 0.3; predação, onde um coelho é consumido e um lobo é gerado, com uma taxa de 0.15, e; morte, onde um lobo morre, com uma taxa de 0.7. Para este modelo em particular, não é simulado o lugar de morte, visto que o número de *tokens* neste local cresce infinitamente, sem acrescentar informações de interesse na simulação.

Com estes valores, a simulação nos dá os valores máximos e mínimos para cada população: temos um máximo de 107 coelhos e 65 lobos, e um mínimo de 15 coelhos e 3 lobos. Para este modelo em particular, o sistema tem um comportamento sazonal,

| Indivíduos | Quantidade |
|-------------|--------------------|
| Suscetíveis | 25.13847510876284 |
| Expostos | 1.1720510236991608 |
| Infectados | 29.691409725139142 |
| Recuperados | 859.8242955468187 |
| Mortos | 110.27504646816888 |

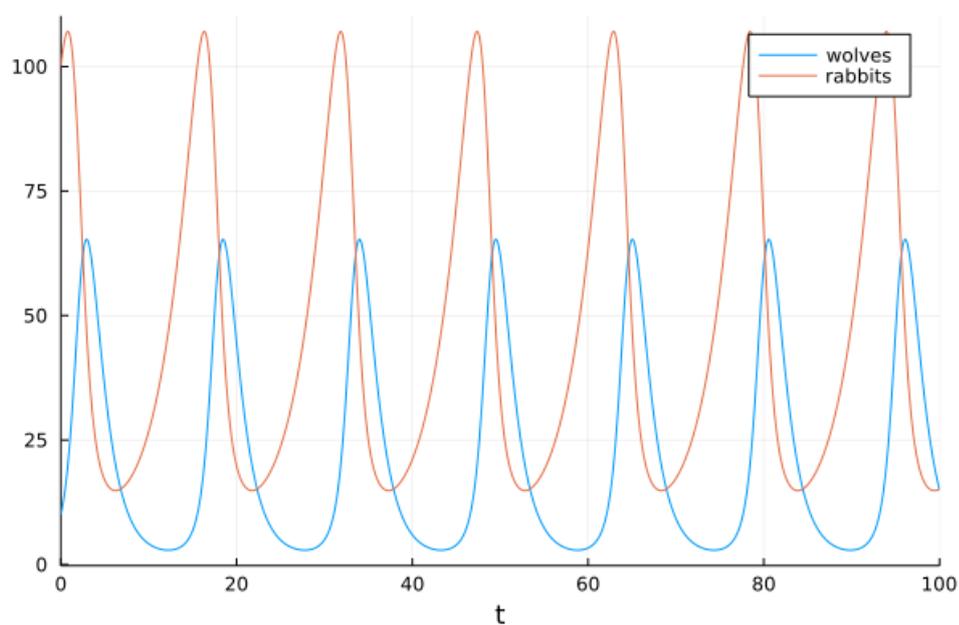
Tabela 1 – Resultados para a simulação do modelo SEIRD

Figura 6 – O resultado da simulação do modelo SEIRD.



Fonte: o autor

Figura 7 – O resultado da simulação do modelo Lotka-Volterra.



Fonte: o autor

uma relação inversamente proporcional entre a quantidade de indivíduos de cada uma das espécies.

5 Conclusões e Trabalhos Futuros

Neste trabalho foi descrito o processo de desenvolvimento de um simulador simples para redes de Petri Estocásticas, capaz de simular redes reativas através de equações diferenciais. Durante a realização dos testes foi possível perceber que o simulador gera os resultados esperados: ele é capaz de gerar o conjunto correto de equações diferenciais a partir de uma descrição por redes de Petri Estocásticas, além de solucionar estas equações.

A adição de novos métodos de análise, incluindo a análise de soluções estacionárias e transientes para as redes descritas é um ponto de melhoria, permitindo a geração de dados relacionados a desempenho e *throughput* dos estados presentes na rede seria de grande valia. Além disto, a possibilidade de gerar cadeias de Markov e simular as redes por elas seria interessante para a simulação de redes menores, evitando assim os erros de aproximação introduzidos pelas equações diferenciais.

A inclusão de suporte a outras extensões para redes de Petri, em particular redes de Petri generalizadas e redes hierárquicas, além de melhor suporte a redes estocásticas, trata-se de um projeto futuro, visto que a estrutura atualmente utilizada para a descrição não é adequada para a análise de redes não-reativas.

Por fim, a implementação do projeto em uma linguagem de programação orientada a objetos permitiria uma melhor organização do código e simplificaria a interface para criação e simulação das redes, ao custo de um processo de desenvolvimento mais lento. Para isto, seria necessário um estudo do custo-benefício para cada uma das linguagens disponíveis, buscando minimizar o tempo de desenvolvimento de componentes fundamentais, como bibliotecas para trabalhar com equações diferenciais e cadeias de Markov.

Referências

- [1] PETRI, C. A. *Kommunikation mit Automaten*. Tese (Doutorado) — Universität Hamburg, 1962. Citado na página 15.
- [2] WWW.SCIENCEDIRECT.COM. *Petri Nets - an overview | ScienceDirect Topics*. Disponível em: <<https://www.sciencedirect.com/topics/physics-and-astronomy/petri-nets>>. Citado na página 15.
- [3] ZAITSEV, D. Switched lan simulation by colored petri nets. *Mathematics and Computers in Simulation*, v. 65, p. 245–249, 04 2004. Citado na página 17.
- [4] JENSEN, K. Coloured petri nets: Basic concepts, analysis methods and practical use. vol. 2, analysis methods. In: . [S.l.: s.n.], 1992. Citado na página 17.
- [5] JENSEN, K.; KRISTENSEN, L. M. *Coloured Petri Nets : Modelling and Validation of Concurrent Systems*. [S.l.]: Springer-Verlag Berlin Heidelberg, 2009. Citado na página 17.
- [6] CPN Tools – A tool for editing, simulating, and analyzing Colored Petri nets. Disponível em: <<http://cpntools.org/>>. Citado na página 17.
- [7] MOLLOY, M. Performance analysis using stochastic petri nets. *IEEE Transactions on Computers*, C-31, p. 913–917, 09 1982. Citado na página 18.
- [8] CHIOLA, G. et al. Generalized stochastic petri nets: a definition at the net level and its implications. *IEEE Transactions on Software Engineering*, v. 19, p. 89–107, 1993. Citado na página 18.
- [9] MARKOV, A. A.; NAGORNYI, N. M. *The Theory of Algorithms*. [S.l.]: Kluwer Academic, 1988. Citado na página 18.
- [10] BAUSE, F.; KRITZINGER, P. *Stochastic Petri Nets: An Introduction to the Theory*. [S.l.]: Braunschweig Wiesbaden Vieweg, 2002. 49-54 p. Citado na página 19.
- [11] KURTZ, T. G. Solutions of ordinary differential equations as limits of pure jump markov processes. *Journal of Applied Probability*, v. 7, p. 49–58, 04 1970. Citado 2 vezes nas páginas 19 e 22.
- [12] KURTZ, T. G. Strong approximation theorems for density dependent markov chains. *Stochastic Processes and their Applications*, v. 6, p. 223–240, 02 1978. Citado na página 19.

- [13] BECCUTI, M. et al. Analysis of Petri Net Models through Stochastic Differential Equations. abr. 2014. Citado 2 vezes nas páginas 19 e 22.
- [14] GILBERT, D.; HEINER, M. From petri nets to differential equations – an integrative approach for biochemical network analysis. In: DONATELLI, S.; THIAGARAJAN, P. S. (Ed.). *Petri Nets and Other Models of Concurrency - ICATPN 2006*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 181–200. ISBN 978-3-540-34700-2. Citado 2 vezes nas páginas 19 e 21.
- [15] CORTADELLA, J.; YAKOVLEV, A.; ROZENBERG, G. *Concurrency and hardware design : advances in Petri nets*. [S.l.]: Springer, 2002. Citado na página 19.
- [16] LIAO, H. et al. Concurrency bugs in multithreaded software: modeling and analysis using petri nets. *Discrete Event Dynamic Systems*, v. 23, p. 157–195, 2013. Citado na página 19.
- [17] STAINES, A. A colored petri net model for the france paris metro system. In: . [S.l.: s.n.], 2012. p. 15–19. Citado na página 19.
- [18] OUTALHA, A. et al. Electric vehicle routing problem using timed petri nets and optimization algorithm. *Journal of Advanced Research in Dynamical and Control Systems*, v. 12, p. 1109–1115, 05 2020. Citado na página 20.
- [19] MARWAH, M. et al. Quantifying the sustainability impact of data center availability. *SIGMETRICS Performance Evaluation Review*, v. 37, p. 64–68, 03 2010. Citado na página 20.
- [20] CALLOU, G. et al. A petri net-based approach to the quantification of data center dependability. In: _____. [S.l.: s.n.], 2012. p. 313–336. ISBN 978-953-51-0700-2. Citado na página 20.
- [21] AMPARORE, E. G. et al. 30 years of GreatSPN. In: *Principles of Performance and Reliability Modeling and Evaluation*. [S.l.]: Springer, 2016. p. 227–254. Citado na página 20.
- [22] BUCCI, G. et al. Oris: a tool for modeling, verification and evaluation of real-time systems. *International Journal on Software Tools for Technology Transfer*, v. 12, p. 391–403, 2010. Citado na página 20.
- [23] CHIOLA, G. et al. Stochastic well-formed colored nets and symmetric modeling applications. *IEEE Trans. Computers*, v. 42, p. 1343–1360, 1993. Citado na página 20.
- [24] POMMEREAU, F. Snakes: A flexible high-level petri nets library (tool paper). In: *Petri Nets*. [S.l.: s.n.], 2015. Citado na página 21.

-
- [25] PASTRAVANU, O. C.; MATCOVSCHI, M.-H.; MAHULEA, C. Petri Net Toolbox — Teaching Discrete Event Systems Under MATLAB. In: . [S.l.: s.n.], 2004. Citado na página 21.
- [26] DAVIDRAJUH, R.; SERVICE, S. O. *Modeling Discrete-Event Systems with GPenSIM : An Introduction*. [S.l.]: Springer International Publishing, 2018. Citado na página 21.
- [27] BAEZ, J. C.; POLLARD, B. S. A Compositional Framework for Reaction Networks. *Advances in Mathematical Physics Vol. 2018 (2018)*, 7676309, abr. 2017. Citado na página 22.
- [28] BEZANSON, J. et al. Julia: A fast dynamic language for technical computing. Citado na página 22.
- [29] RACKAUCKAS, C.; NIE, Q. Differentialequations.jl—a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, Ubiquity Press, v. 5, n. 1, 2017. Citado na página 23.