

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Lorena Rodrigues Bruno

**Uma abordagem para a Modelagem e
Simulação de Processos de Negócio Baseada
nas Redes de Petri Estocásticas e Redes de
Petri Contínuas Temporizadas**

Uberlândia, Brasil

2018

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Lorena Rodrigues Bruno

**Uma abordagem para a Modelagem e Simulação de
Processos de Negócio Baseada nas Redes de Petri
Estocásticas e Redes de Petri Contínuas Temporizadas**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau
de Bacharel em Ciência da Computação.

Orientador: Stéphane Julia

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2018

Aos meus pais
Reginaldo José Rodrigues Bruno e
Maria José Vieira Bruno

Agradecimentos

Agradeço primeiramente a Deus por ter me dado saúde, força e por ter me guiado durante toda minha jornada, permitindo que eu chegasse até aqui.

Aos meus pais pelo amor, incentivo e apoio que sempre me deram. Por sempre acreditarem em mim e não medirem esforços para que mesmo com tantas dificuldades eu pudesse ter uma boa formação, tanto pessoal quanto acadêmica.

À minha irmã Renata pelo carinho, apoio e incentivo.

Aos meus amigos, tanto àqueles que conheci durante a vida acadêmica quanto aos antigos, pelo auxílio e compreensão durante toda essa jornada.

A todos os professores que contribuíram para a minha formação, desde a educação básica até a graduação, tanto pelos ensinamentos técnicos quanto pelos conselhos e conversas que levarei para toda minha vida.

Ao meu orientador, Profº. Drº. Stéphane, pela paciência, prontidão e confiança. Por me auxiliar e guiar durante a execução deste trabalho.

Resumo

Este trabalho descreve um estudo sobre modelagem e simulação de processos de negócios utilizando redes de Petri, também conhecidas como *workflow net*, e o simulador de redes de Petri contínuas *SimHPN*. A importância da modelagem de processos se dá principalmente pelos benefícios gerados para as empresas que dela usufruem, que tendem a obter mais agilidade e se tornarem mais competitivas no mercado, o que gera consequências diretamente ligadas a sua produtividade e lucratividade. A abordagem apresentada neste trabalho impacta também na distribuição de recursos humanos e computacionais dentro das organizações, na mudança do perfil administrativo e na redução do tempo despendido para a realização de uma determinada tarefa, tudo isso devido a um planejamento prévio das atividades a serem realizadas por meio do uso de redes de Petri.

As redes de Petri contínuas foram apresentadas por muito autores como uma alternativa para a representação de sistemas com quantidade de estados discretos muito elevada. Dessa forma, elas são empregadas como uma alternativa para análise de processos de negócio complexos demais para uma representação em redes de Petri discretas tradicionais.

O objetivo do trabalho é modelar, analisar e simular processos de negócio utilizando redes de Petri contínuas com base na ferramenta *SimHPN* do *MATLAB*. Os resultados das simulações com modelos contínuos serão comparados com os obtidos utilizando redes de Petri estocásticas tradicionais. Tal estudo comparativo será útil para analisar as vantagens e desvantagens do uso de redes de Petri contínuas na representação de fluxos de trabalho geralmente representados por modelos discretos.

Palavras-chave: Rede de Petri, Rede de Petri Contínua, Rede de Petri Estocástica, *Workflow Net*, *SimHPN*.

Lista de ilustrações

Figura 1 – Elementos de uma Rede de Petri	15
Figura 2 – Exemplo de Rede de Petri com Marcação	15
Figura 3 – Exemplo de Sensibilização e Disparos em uma Rede de Petri	16
Figura 4 – Rede de Petri Estocástica (DAVID; ALLA, 1992)	17
Figura 5 – Exemplo de Função de Densidade	18
Figura 6 – Exemplo de Função de Distribuição Acumulada	18
Figura 7 – Exemplo de Rede de Petri	19
Figura 8 – Exemplo de Redes de Petri Temporizadas Estocásticas (DAVID; ALLA, 2010)	20
Figura 9 – Rede de Petri Contínua (DAVID; ALLA, 2010)	21
Figura 10 – Transformação de uma Rede de Petri Discreta em Contínua (DAVID; ALLA, 2010)	21
Figura 11 – Disparo de Transição em uma Rede de Petri Contínua (DAVID; ALLA, 2010)	22
Figura 12 – De uma Rede de Petri Discreta Temporizada para uma Contínua Temporizada (DAVID; ALLA, 2010)	23
Figura 13 – Tipos de Acionamento	25
Figura 14 – Exemplo do Hospital - <i>Workflow net</i> AALST W. van der; STAHL (2011)	26
Figura 15 – Esboço da tela inicial do SimHPN	27
Figura 16 – Interface Gráfica SimHPN	27
Figura 17 – Menu <i>Model</i>	28
Figura 18 – Menu <i>Barra de Ferramentas</i>	28
Figura 19 – Exemplo de Gráficos Resultantes das Simulações	29
Figura 20 – Exemplo de Rede de Petri Estocástica	30
Figura 21 – Dados de Entrada no <i>SimHPN</i>	31
Figura 22 – Rede de Petri Contínua	32
Figura 23 – Gráfico das Marcações da Rede de Petri Contínua	33
Figura 24 – Gráfico das Evolução dos Fluxos de uma Rede de Petri Contínua utilizando <i>Infinite Server Semantics</i>	34
Figura 25 – Gráfico das Evolução dos Fluxos de uma Rede de Petri Contínua utilizando <i>Product Server Semantics</i>	34
Figura 26 – <i>Workflow net</i> para o “Processo de Tratamento de Reclamações” (AALST; HEE, 2004)	37
Figura 27 – Modelo 1: Processo de Tratamento de Reclamações sem Recursos	38
Figura 28 – Modelo 1: Matriz Pre	39
Figura 29 – Modelo 1: Matriz Post	39

Figura 30 – Modelo 1: Vetor M_0	40
Figura 31 – Modelo 1: Vetor λ	40
Figura 32 – Vetores $T.Type$	40
Figura 33 – <i>Workflow net</i> para o “Processo de Tratamento de Reclamações” com recursos	41
Figura 34 – Modelo 2: Processo de Tratamento de Reclamações com Recursos . . .	42
Figura 35 – Modelo 2: Matriz Pre	43
Figura 36 – Modelo 2: Matriz Post	43
Figura 37 – Modelo 2: Vetor M_0	43
Figura 38 – Modelo 1 - Estocástico	45
Figura 39 – Modelo 1 - Discreto Temporizado Estocástico - Informações adicionais	46
Figura 40 – Modelo 1 - Contínuo	46
Figura 41 – Modelo 1 - Contínuo - Informações adicionais	46
Figura 42 – Modelo 1 - Simulação Discreta Estocástica Única	47
Figura 43 – Comparação do Modelo 1 Estocástico e Contínuo	48
Figura 44 – Modelo 2 - Simulação Inicial - Parte 1	49
Figura 45 – Modelo 2 - Simulação Inicial - Parte 2	49
Figura 46 – Modelo 2 - Simulação Inicial - Parte 3	50
Figura 47 – Modelo 2 - Estocástico	51
Figura 48 – Modelo 2 - Estocástico - Informações adicionais	51
Figura 49 – Modelo 2 - Contínuo	52
Figura 50 – Modelo 2 - Contínuo - Informações adicionais	52

Lista de tabelas

Tabela 1 – Comparação dos Resultados das Simulações	53
---	----

Lista de abreviaturas e siglas

PN	Petri Net
D-Lugar	Lugar Discreto
C-Transição	Transição Contínua
D-Transição	Transição Discreta
MATLAB	Ambiente de simulação matemática, utilizado em diversas áreas profissionais

Sumário

1	INTRODUÇÃO	11
1.1	Objetivos	12
1.1.1	Objetivos Específicos	12
1.2	Justificativa	13
2	REFERENCIAL TEÓRICO	14
2.1	Definição de uma Rede de Petri	14
2.1.1	Elementos gráficos de uma Rede de Petri	14
2.1.2	Disparo de uma Transição	15
2.2	Rede de Petri Estocástica	16
2.2.1	Distribuição Exponencial	17
2.2.2	Definição de uma Rede de Petri Estocástica	17
2.2.3	Rede de Petri Temporizada Estocástica	19
2.3	Redes de Petri Contínuas	20
2.3.1	Definição	22
2.3.2	Rede de Petri Contínua Temporizada	22
2.4	Workflow nets	24
2.4.1	Definição	24
2.4.2	Acionamentos	24
2.4.3	Modelagem de um Processo	25
2.5	<i>SimHPN: MATLAB Toolbox para Análise e Simulação de Redes de Petri Contínuas</i>	26
2.5.1	Interface Gráfica	26
2.5.2	Dados de Entrada	28
2.5.3	Semânticas de Disparo	31
2.5.3.1	Infinite Server Semantics	31
2.5.3.2	Product Server Semantics	32
2.5.3.3	Exemplo das Semânticas de Disparo	32
2.6	Trabalhos Relacionados	34
3	MODELAGEM DE UM PROCESSO DE NEGÓCIO POR MEIO DE UMA <i>WORKFLOW NET</i>	36
3.1	Introdução	36
3.2	Implementação de uma <i>Workflow net</i> sem Recursos	38
3.3	Implementação de uma <i>Workflow net</i> com Recursos	41

4	SIMULAÇÃO: ANÁLISE QUANTITATIVA	44
4.1	Simulação: Modelo 1 - Discreto Temporizado Estocástico	44
4.2	Simulação: Modelo 1 - Contínuo	46
4.3	Simulação Para Eliminar Possíveis Gargalos do Modelo 2 Utilizando Recursos Contínuos e Transições Contínuos	48
4.4	Simulação: Modelo 2 com Recursos Discretos e Transições Discretas Temporizadas Estocásticas	50
4.5	Simulação: Modelo 2 com Recursos Contínuos e Transições Contínuas Temporizadas	51
5	CONSIDERAÇÕES FINAIS	54
	REFERÊNCIAS	56

1 Introdução

A tradução do termo em inglês *workflow* é fluxo de trabalho. Dessa forma, *Workflow Management Systems* são sistemas que gerenciam a execução dos processos de um caso, o que representa seu fluxo de trabalho. Um processo consiste em um conjunto de tarefas a serem realizadas de acordo com um conjunto de condições que determinam a ordem de execução dessas tarefas (AALST; HEE, 2001).

Com o avanço nas áreas de tecnologia da informação e telecomunicação, surge a necessidade de buscar novas maneiras de modelar processos de negócio, de forma que atendam às necessidades das empresas nessa área. Quanto mais rápido uma empresa se adapta às mudanças, mais chances ela tem de se destacar no mercado. Porém, para isso, é necessário reestruturar seus processos internos, sejam eles administrativos ou técnicos, melhorar a gerência desses processos e buscar integrá-los da melhor forma possível.

A grande importância dada à modelagem de processos se dá principalmente pelos benefícios gerados às empresas, sendo um caminho para conseguir agilidade e vantagem competitiva, e que estão diretamente ligados com sua produtividade e lucratividade. Dessa forma, surgem *softwares* especializados em gerenciar processos de negócios os *Workflow Management Systems*.

Para tanto, um recurso muito usado para modelagem e simulação de processos de *workflow* são as redes de Petri. Tal ferramenta de modelagem apresenta vantagens como permitir que processos sejam descritos graficamente. Além disso, graças à sua base formal, é possível fazer declarações sólidas com relação às propriedades dos processos que estão sendo modelados. Uma rede de Petri que representa um processo de *workflow* é chamada de *workflow net*. (AALST; HEE, 2001)

Muitas pesquisas já foram realizadas com relação às redes de Petri e apontam seu intenso uso na modelagem de sistemas e eventos discretos como controle de oficinas de fabricação, concepção de software em tempo real, sistemas de informação, logística, entre outras aplicações (MURATA, 1989; CARDOSO; VALETTE, 1997). Porém, passou-se a observar que para sistemas discretos com uma grande quantidade de fichas (elementos discretos dos modelos), o número de estados do sistema pode explodir, o que torna inviável o uso dessas redes na análise de alguns casos.

Na tentativa de solucionar este problema, o conceito de rede de Petri contínua foi introduzido em (DAVID, 1987); já o conceito de rede de Petri híbrida, introduzido no mesmo artigo, foi desenvolvido em (BAIL, 1991). A partir de então, a quantidade de fichas de um modelo passa a ser representada por números reais e não mais por inteiros positivos como acontecia nas redes discretas, eliminando então o problema de explosão

de estados discretos de sistemas complexos (DAVID; ALLA, 2010). No entanto, quando uma situação é modelada de forma a conter variáveis discretas e contínuas (pelo menos uma de cada tipo), existe uma forma do modelo chamada de rede de Petri híbrida.

1.1 Objetivos

O objetivo deste trabalho é, portanto, modelar, analisar e simular processos de *workflow* utilizando redes de Petri contínuas com base na ferramenta *MATLAB Petri net Toolbox*. O uso de redes contínuas pode ser justificado pelo fato de que os sistemas de gerenciamento de processos de negócios atuais, apesar de serem classificados como sistemas discretos, tendem a lidar com uma grande quantidade de dados, o que pode levar à explosão do número de estados discretos a serem produzidos durante o processo de análise.

Os resultados das simulações usando modelos contínuos deverão ser comparados com os obtidos utilizando redes de Petri tradicionais (variáveis de estados discretas e temporizações aleatórias que seguem a lei exponencial). Tal estudo comparativo servirá para analisar as vantagens e desvantagens de utilizar redes de Petri contínuas para representar fluxos de trabalho que são normalmente especificados por modelos discretos.

1.1.1 Objetivos Específicos

Entre os objetivos específicos que este estudo visa alcançar, estão:

1. Mostrar como produzir modelos baseados em redes de Petri contínuas para representar processos de *workflow*;
2. A apresentação da ferramenta *Matlab Petri net Toolbox* utilizada para simular modelos baseados em redes de Petri contínuas;
3. A seleção de um estudo de caso de processo de *workflow* a ser modelado;
4. A modelagem, análise e simulação do estudo de caso selecionado considerando uma abordagem tradicional baseada nas redes de Petri temporizadas estocásticas (puramente discreta);
5. A modelagem, análise e simulação do estudo de caso selecionado considerando uma abordagem contínua baseada nas redes de Petri contínuas;
6. A elaboração de um estudo comparativo entre os resultados das simulações do modelo discreto e do modelo contínuo.

1.2 Justificativa

Como a tecnologia vem ganhando cada vez mais espaço no mercado de trabalho, empresas que a utilizam para benefício próprio, acabam garantindo vantagem competitiva. Porém, para que isso aconteça, é necessário que elas tenham um plano de negócios sólido e bem definido. Dessa forma, o estudo sobre o uso de redes de Petri contínuas para modelar os fluxos de trabalho dessas empresas, principalmente relacionado aos sistemas de informação, se forem provados mais eficientes que o uso de redes discretas, podem ser um diferencial no mercado. Em particular para alcançar resultados satisfatórios nas etapas de dimensionamento e gerenciamento dos recursos envolvidos nas atividades dos processos administrativos e de negócios das empresas.

O objetivo essencial desse trabalho corresponde então a analisar as vantagens de se utilizar uma simulação contínua no âmbito de sistemas cujas variáveis de estado são dadas geralmente através de valores discretos. Em particular, os modelos de simulação devem ser aplicados à processos de negócio que existem dentro das empresas. A validação da abordagem será então realizada através de um estudo comparativo entre modelos tradicionais de simulação de modelos discretos estocásticos (rede de Petri estocástica) e modelos de simulação contínuos determinísticos (redes de Petri contínuas temporizadas).

2 Referencial Teórico

O modelo de redes de Petri foi proposto por *Carl Adam Petri* na tese (PETRI, 1962). A partir de então, diversos pesquisadores passaram a se aprofundar mais no assunto, como *Hassane Alla* e *René David* que introduziram as redes de Petri contínuas (ALLA; DAVID, 1998), trazendo grandes contribuições para a área.

Uma rede de Petri é uma ferramenta gráfica e matemática que pode ser utilizada em diferentes aplicações para representar sistemas constituídos por eventos discretos (sistemas nos quais as mudanças de estado ocorrem em instantes precisos). Um dos principais benefícios do uso de uma rede de Petri é que ela torna possível a modelagem e visualização de comportamentos de sistemas com características como paralelismo, concorrência, sincronização e compartilhamento de recursos (DAVID; ALLA, 2010).

A produção de uma rede é baseada em três elementos básicos: lugar, transição e ficha. O lugar, que pode ser interpretado, entre outras abordagens, como uma condição, um procedimento ou um estado de espera; a transição, que está associada a um evento que ocorre no sistema; e a ficha, que funciona como um indicador para representar que a condição associada ao lugar é verificada (CARDOSO; VALETTE, 1997).

2.1 Definição de uma Rede de Petri

Uma Rede de Petri (RP), com marcação inicial, é formalmente definida como uma quintupla $= (P, T, I, O, M_0)$, onde:

1. $P = \{p_1, p_2, \dots, p_n\}$ é um conjunto finito de lugares;
2. $T = \{t_1, t_2, \dots, t_n\}$ é um conjunto finito de transições;
3. $I : P \times T \rightarrow \mathbb{N}$ é uma relação de entrada que define arcos que ligam os lugares as transições;
4. $O : T \times P \rightarrow \mathbb{N}$ é uma relação de saída que define arcos que ligam as transições aos lugares;
5. $M_0 : P \rightarrow \mathbb{N}$ é a marcação inicial.

2.1.1 Elementos gráficos de uma Rede de Petri

Um lugar P_i é representado por um círculo, uma transição T_j é representada por uma barra ou retângulo, e uma marcação, ficha ou *token* é representada por um ponto

preto. Os arcos são direcionados e podem ligar tanto lugares a transições quanto o contrário. A Figura 1 apresenta os elementos gráficos de uma Rede de Petri.



Figura 1 – Elementos de uma Rede de Petri

A Figura 2 representa uma Rede de Petri com 4 lugares, 4 transições e 8 arcos. O lugar P_3 é chamado lugar de entrada da transição T_4 pois existe um arco saindo de P_3 e chegando a T_4 . O lugar P_2 é chamado lugar de saída da transição T_1 pois existe um arco saindo de T_1 e chegando a P_2 . Os lugares P_1 e P_3 contêm fichas.

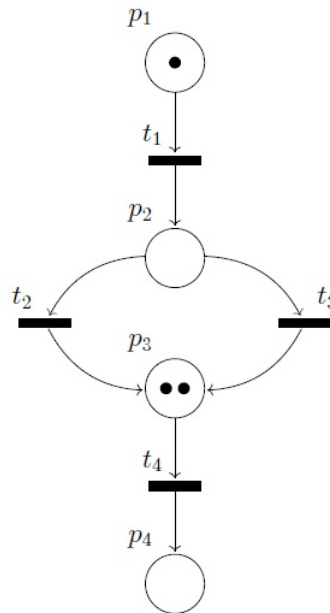


Figura 2 – Exemplo de Rede de Petri com Marcação

A marcação em uma Rede de Petri está relacionada com a quantidade de fichas que um lugar possui e será sempre zero ou um valor estritamente positivo, ou seja, maior que zero. O número de fichas pode variar de acordo com a ocorrência de certos eventos durante a execução da rede.

2.1.2 Disparo de uma Transição

A transição é um componente ativo na rede e é onde ocorre o chamado **disparo**. Uma transição pode ser disparada se cada um dos seus lugares de entrada contiver pelo menos uma ficha (*token*). A transição é, então, dita **sensibilizada**.

Disparar uma transição T_j significa consumir uma ficha de cada lugar de entrada de T_j e então produzir uma ficha em cada lugar de saída da transição. A Figura 3 mostra

o disparo da transição T_1 quando sensibilizada. Pode-se observar que na Figura 3a, o lugar P_2 não contém nenhuma ficha; assim a transição T_1 não pode ser disparada.

Na Figura 3b a transição T_1 está sensibilizada já que todos os lugares de entrada contêm pelo menos uma ficha; dessa forma, após o disparo de T_1 a Figura 3c é gerada, ou seja, é consumida uma ficha de cada lugar de entrada e gerada uma ficha em cada lugar de saída. Pode-se notar que o lugar P_5 contém duas fichas; isso ocorre porque antes do disparo o mesmo já continha uma ficha.

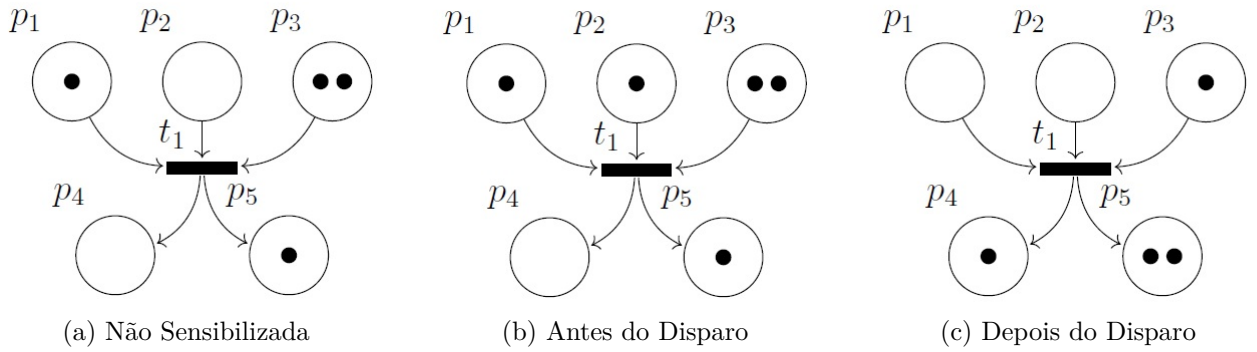


Figura 3 – Exemplo de Sensibilização e Disparos em uma Rede de Petri

2.2 Rede de Petri Estocástica

As redes de Petri estocásticas, introduzidas por (FLORIN; FRAIZE; NATKIN, 1991), são caracterizadas por terem um tempo aleatório associado a cada uma de suas transições. Em uma rede de Petri temporizada, um tempo fixo é associado a cada uma de suas transições ou lugares. No caso de sistemas de produção, por exemplo, o tempo de trabalho de uma máquina para tratar uma determinada peça é constante. Porém, existem fenômenos que não podem ser modelados adequadamente utilizando duração constante, que é o caso, por exemplo, do tempo que uma máquina consegue funcionar de forma adequada entre duas falhas. Esse tempo deve ser medido usando uma variável randômica e é neste caso que as redes de Petri estocásticas devem ser usadas, associando um tempo randômico a cada transição. Geralmente os tempos são atribuídos de acordo com uma distribuição exponencial (DAVID; ALLA, 1992).

Para melhor compreensão das redes de Petri temporizadas estocásticas, será apresentado, a seguir, um exemplo (Figura 4), onde, existe conflito entre as transições T_1 e T_2 , de forma que o disparo das duas depende da ficha em P_1 . A transição T_1 é disparada quando o evento *End of service* ocorre, ou seja, quando a execução de um serviço é finalizada. Já a transição T_2 é disparada com a ocorrência do evento *Servers brakes down*, ou seja, quando o servidor falha.

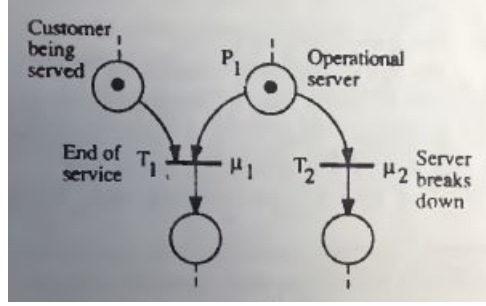


Figura 4 – Rede de Petri Estocástica (DAVID; ALLA, 1992)

A duração do serviço é uma variável randômica d_1 , cujo valor médio é $\frac{1}{\mu_1}$ (μ_i representa o parâmetro da distribuição exponencial e T_i representa a taxa de disparo de uma transição). O tempo durante o qual o servidor funciona de forma apropriada é uma variável randômica d_2 , cujo valor médio é $\frac{1}{\mu_2}$. Se $d_1 < d_2$, a transição T_1 é disparada e se $d_1 > d_2$, a transição T_2 que é disparada.

Sendo assim, podemos dizer que a Figura 4 representa uma rede de Petri estocástica, na qual a evolução é modelada por meio das variáveis aleatórias.

2.2.1 Distribuição Exponencial

Alguns conceitos de rede de Petri estocástica necessitam do uso da distribuição exponencial, sendo assim, é importante relembrar a definição desta. A distribuição exponencial é uma distribuição contínua de probabilidade, representada por um parametro λ e sua função de densidade está representada em 2.1. A função mostra que a probabilidade de x assumir um valor maior ou igual a 0 é de $\lambda e^{-\lambda x}$ e a probabilidade de assumir um valor negativo é nula. Sua função de distribuição acumulada $F(x)$ é dada em 2.2.

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{se } x \geq 0 \\ 0 & \text{se } x < 0 \end{cases} \quad (2.1)$$

$$F(x) = \int_0^x f(s)ds = \begin{cases} 1 - e^{-\lambda x} & \text{se } x \geq 0 \\ 0 & \text{se } x < 0 \end{cases} \quad (2.2)$$

Na Figura 5 é apresentado um exemplo utilizando a função de densidade aplicada a diferentes λ , já a Figura 6 apresenta um exemplo utilizando a função de distribuição acumulada para os mesmos valores de λ .

2.2.2 Definição de uma Rede de Petri Estocástica

Antes de apresentar a definição de rede de Petri estocástica é importante definir o conceito de transição q-sensibilizada. Uma transição T_j é q-sensibilizada quando seus

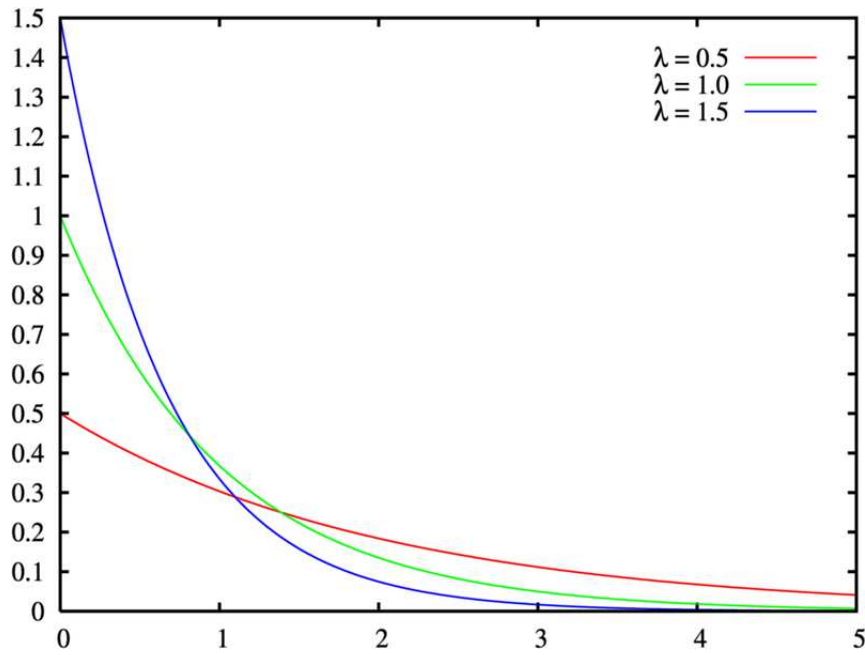


Figura 5 – Exemplo de Função de Densidade

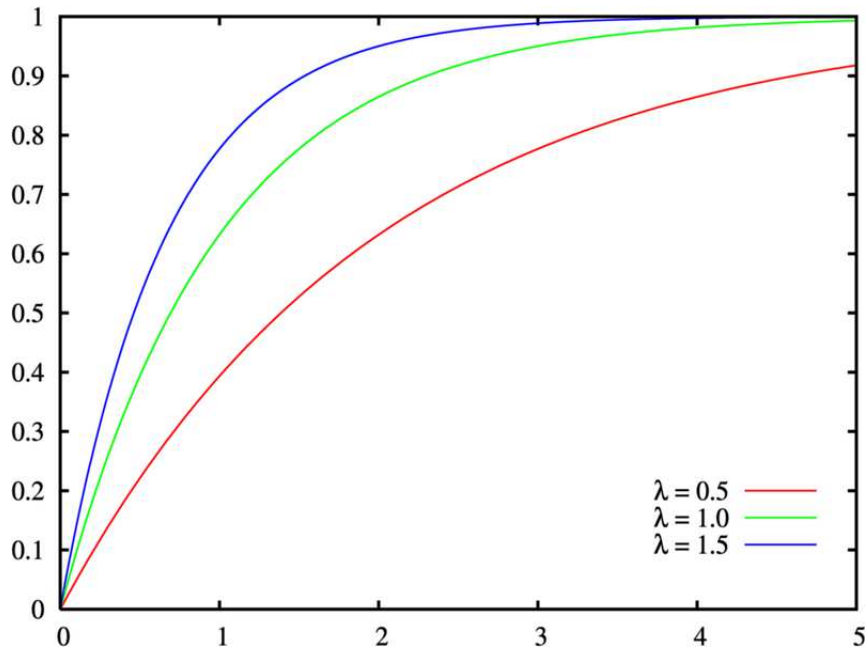


Figura 6 – Exemplo de Função de Distribuição Acumulada

lugares de entrada tem número de fichas maior ou igual a q , de forma que ela possa ser disparada q vezes consecutivas.

Na Figura 7, temos um exemplo de rede na qual apenas as transições T_1 e T_2 estão sensibilizadas e podem ser disparadas a uma taxa $2\mu_1$ e $2\mu_2$, respectivamente. Isso porque existem fichas suficientes em P_1 e P_3 para "disparar duas vezes" a transição T_1 , por exemplo, já que $P_1 > 2$ e $P_3 = 2$. Assim, podemos dizer que T_1 é uma transição 2-sensibilizada.

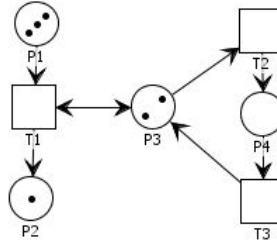


Figura 7 – Exemplo de Rede de Petri

Uma rede de Petri estocástica é um tipo de rede de Petri na qual um tempo randômico d_j é associado a cada transição T_j e cujo comportamento está de acordo com o algoritmo a seguir:

1. Inicialize a rede com a marcação inicial. O tempo inicial é $t = 0$.
2. Faça $X = \{T_1, T_2, T_3, \dots, T_j, \dots\}$ um conjunto de transições sensibilizadas.
3. Para cada transição T_j em X , realize um sorteio aleatório de d_j , respeitando a distribuição exponencial. Se a taxa de disparo de T_j for μ_j , o sorteio é realizado considerando uma taxa $q \cdot \mu_j$ para uma transição q -sensibilizada.
4. Faça $d_m = \min(d_1, d_2, \dots, d_j, \dots)$ e T_m a transição correspondente. Substitua t por $t + d_m$.
5. Dispare a transição T_m no tempo t . Vá para o passo 2.

2.2.3 Rede de Petri Temporizada Estocástica

As redes de Petri temporizadas são utilizadas, em geral, para avaliação de desempenho. A temporização pode estar associada tanto à duração de uma operação quanto ao tempo de espera para a ocorrência de um evento. Sendo assim, o tempo pode estar associado aos lugares ou às transições nas redes de Petri. Costuma-se relacionar a duração de operações ou estados aos lugares na rede, e o tempo de espera para a ocorrência de um evento ao disparo da transição (DAVID; ALLA, 2010).

A Figura 8 apresenta exemplos de redes de Petri temporizadas, com o tempo associado tanto aos lugares quanto às transições. Nela é representado um sistema composto por duas máquinas, M_A e M_B . Os estados P_1 , P_2 , P_3 e P_4 representam, respectivamente, “esperando o funcionamento de M_A ”, “ M_A em funcionamento”, “esperando o funcionamento de M_B ”, “ M_B em funcionamento”. O tempo de processamento da máquina M_A é d_A e o da máquina M_B é d_B .

Na Figura 8a, temos o tempo associado aos lugares, de forma que as fichas em P_1 podem representar clientes esperando pelo funcionamento da máquina M_A , por exemplo.

Porém a transição T_1 não pode ser disparada pois não há nenhuma ficha em um de seus lugares de entrada, no caso, P'_2 . Além disso, a ficha em P_2 pode significar que existe um cliente sendo processado pela máquina M_A , por exemplo.

Ainda na Figura 8a mas fazendo referência agora à máquina M_B , como não existem fichas em P_3 , significa que não há clientes esperando pelo funcionamento da máquina M_B e, conseqüentemente, não é possível disparar a transição T_3 . A ficha em P_4 representa um cliente sendo processado pela máquina M_B .

Os tempos d_A e d_B são associados aos lugares P_2 e P_4 , de forma que quando uma ficha chega em P_2 , ela deve permanecer neste lugar por um tempo d_A , antes de poder ser usada para o disparo da transição T_2 . O mesmo deve ocorrer com o lugar P_4 com relação ao tempo d_B e à transição T_4 .

Com relação à Figura 8b, os tempos estão agora associados com as transições, sendo que T_2 representa “fim do funcionamento da máquina M_A ” e está associado ao tempo d_A , e a transição T_4 representa “fim do funcionamento da máquina M_B ” e está associado ao tempo d_B .

Os tempos representam então os atrasos nas transições, assim, d_A é o atraso na transição T_2 e d_B o atraso na transição T_4 . Dessa forma, as transições que não possuem um tempo relacionado a elas são disparadas imediatamente após serem habilitadas.

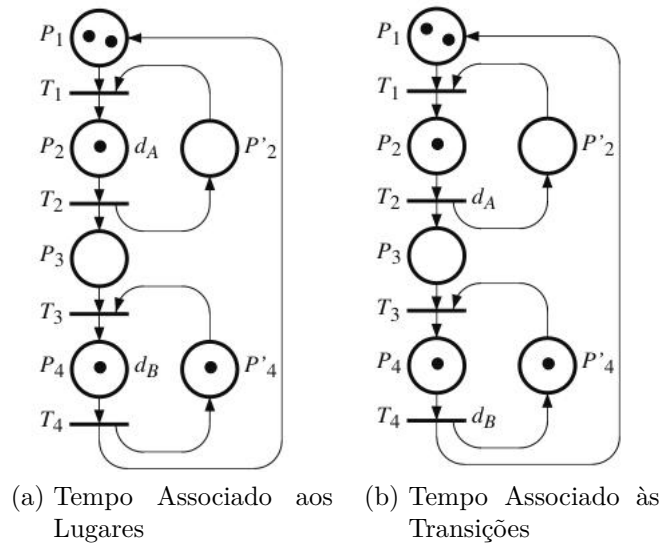


Figura 8 – Exemplo de Redes de Petri Temporizadas Estocásticas (DAVID; ALLA, 2010)

2.3 Redes de Petri Contínuas

As redes de Petri contínuas foram introduzidas em (DAVID, 1987) e representam um modelo no qual as fichas em cada lugar são representadas por números reais, e não inteiros como nas redes tradicionais. Conseqüentemente, em tal modelo, o disparo

das transições se torna um processo contínuo, semelhante à transferência de grãos de areia numa ampulheta. Em redes de Petri contínuas, lugares e transições são geralmente representados por linha dupla, como mostra o exemplo da Figura 9.

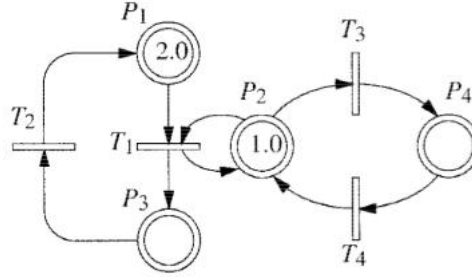


Figura 9 – Rede de Petri Contínua (DAVID; ALLA, 2010)

Em sistemas discretos de complexidade elevada, existe a possibilidade de ocorrer uma explosão combinatória da quantidade de estados alcançáveis (DAVID; ALLA, 2010). Dessa forma, surge a necessidade de buscar uma nova representação que solucione este problema, sendo essa uma das principais motivações para a definição das redes de Petri contínuas.

A passagem do modelo discreto para a representação contínua é dada da seguinte forma: consideremos uma rede de Petri R , definida pelo grafo Q e com marcação inicial m . A partir dela, aplica-se uma transformação que consiste em dividir cada ficha em k partes iguais, sem modificar nenhuma outra propriedade da rede. A nova rede de Petri vai ser denotada por R' e a nova marcação, por m' . Dessa forma, a rede da Figura 10a, por exemplo, dá origem à rede da Figura 10b, na qual as fichas em R' passam a ser representadas pela k -ésima parte das fichas de R . A rede de Petri R' em si, possui as mesmas características de uma rede de Petri discreta. Na Figura 10c, as frações da ficha, quando tendem a um número infinito, passam a ser representadas por um número real, o que caracteriza as redes de Petri contínuas.

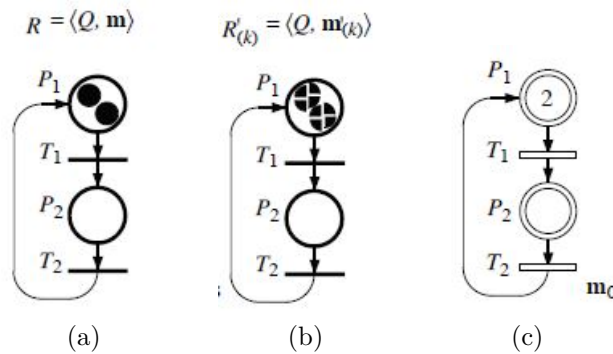


Figura 10 – Transformação de uma Rede de Petri Discreta em Contínua (DAVID; ALLA, 2010)

2.3.1 Definição

A estrutura formal de uma rede de Petri contínua é semelhante a de uma rede de Petri discreta, definida como uma quintupla $= (P, T, I, O, M_0)$, onde:

1. $P = \{p_1, p_2, \dots, p_n\}$ é um conjunto finito de lugares;
2. $T = \{t_1, t_2, \dots, t_n\}$ é um conjunto finito de transições;
3. $I : P \times T \rightarrow \mathbb{Q}^+$ é uma relação de entrada que define arcos que ligam os lugares as transições;
4. $O : T \times P \rightarrow \mathbb{Q}^+$ é uma relação de saída que define arcos que ligam as transições aos lugares;
5. $M_0 : P \rightarrow \mathbb{R}^+$ é a marcação inicial.

Contudo, a principal diferença entre redes de Petri discretas e contínuas está na regra de evolução das marcações, de forma que o disparo de transições numa rede contínua não é restrito a uma quantidade de fichas inteira. Consequentemente, as fichas não precisam mais ser representadas por números inteiros ([RECALDE; HADDAD; SILVA, 2007](#)). Na Figura 11, temos um exemplo de disparo de transição em uma rede de Petri contínua. Na rede representada no lado esquerdo da seta, temos a representação de seu estado antes do disparo; já a rede representada no lado direito da seta, mostra seu estado após o disparo da transição T_j , que consumiu 0.2 fichas.

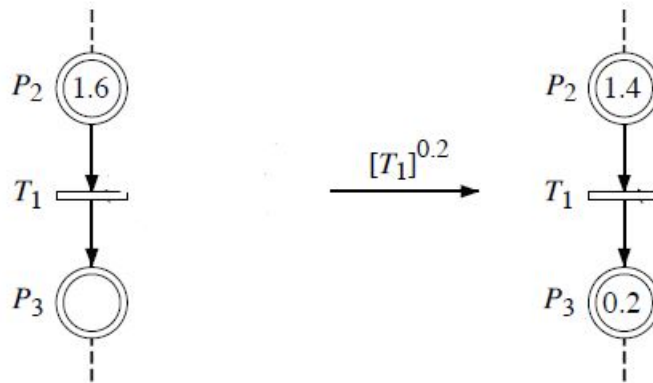


Figura 11 – Disparo de Transição em uma Rede de Petri Contínua ([DAVID; ALLA, 2010](#))

2.3.2 Rede de Petri Contínua Temporizada

Para entender o funcionamento de uma rede de Petri contínua temporizada, considere-se a rede de Petri discreta temporizada na Figura 12a. As fichas fluem de P_1 para P_2 , e P_3 especifica um comportamento de servidor único (apenas 1 ficha em P_3): toda vez que

o tempo d_1 é gasto, uma ficha é removida de P_1 e depositada em P_2 . Ou seja, a taxa de fluxo de P_1 para P_2 é $\frac{1}{d_1}$, contanto que P_1 não esteja vazio. Esse cenário é representado no gráfico da Figura 12d, que mostra que é gasto tempo d_1 para remover uma ficha de P_1 e transferí-la para P_2 , e tempo $2d_1$ para transferir as duas fichas de P_1 para P_2 .

A Figura 12b é obtida a partir da Figura 12a de forma que cada ficha é dividida em duas, e o tempo associado à transição também é dividido por dois. A taxa de fluxo da ficha passa então a ser $\frac{2}{d_1}$. Dado que o número de fichas na Figura 12b é o dobro do da Figura 12a, e considerando o aumento em duas vezes na taxa de fluxo, o tempo para esvaziar o lugar P_1 é o mesmo, como mostra o gráfico da Figura 12d.

Se cada ficha em P_1 é dividida em k fichas, e o tempo associado a T_1 é $\frac{d_1}{k}$, a taxa de fluxo passa a ser $\frac{k}{d_1}$, e o tempo para esvaziar P_1 continua o mesmo. Quando k tende ao infinito, as fichas passam a ser representadas por unidades de ficha e a taxa de fluxo $\frac{k}{d_1}$ é substituída por $\frac{1}{d_1}$, denotada por V_1 . Assim, obtemos a rede de Petri contínua temporizada da Figura 12c.

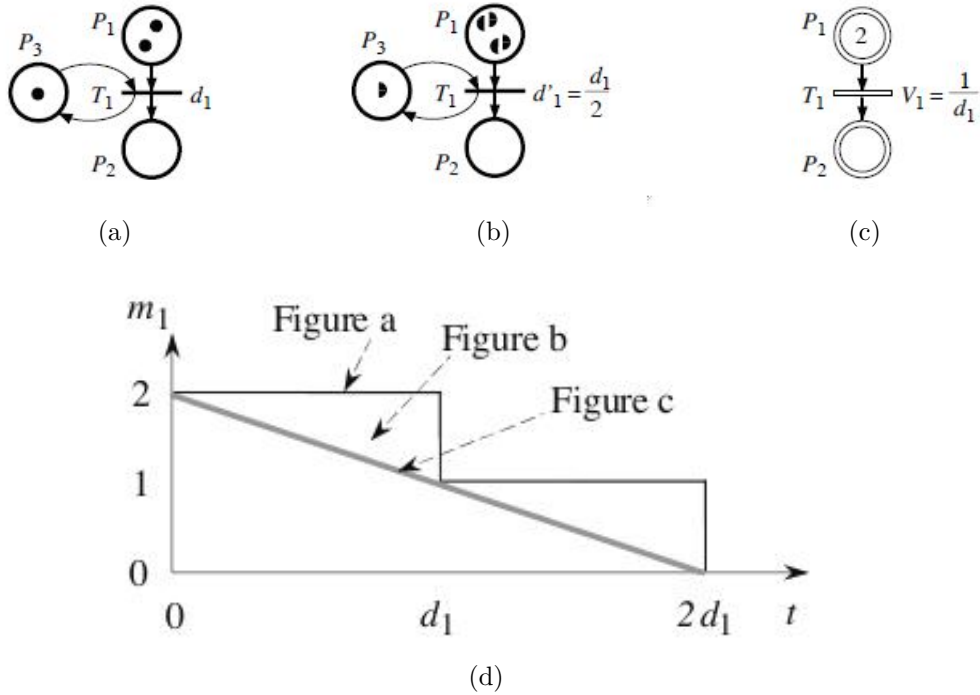


Figura 12 – De uma Rede de Petri Discreta Temporizada para uma Contínua Temporizada (DAVID; ALLA, 2010)

2.4 Workflow nets

2.4.1 Definição

Workflow é um termo que em português significa fluxo de trabalho. Contextualizando para a área tecnológica, *workflow* consiste na utilização de ferramentas computacionais para auxiliar nos passos necessários para a execução de um processo de negócio (REZENDE et al., 2013).

Um processo de *workflow* define quais tarefas de um processo de negócio precisam ser executadas e em qual ordem a execução deve ocorrer. Além disso, de acordo com Aalst (1998), processos de *workflow* são baseados em casos, isto é, cada parte do trabalho é executada para um caso específico. Modelar um processo de *workflow* em termos de uma Rede de Petri é bem direto: transições são componentes ativos e modelam as tarefas, lugares são componentes passivos e modelam as condições (pré e pós), e as fichas modelam os casos (AALST, 1998; AALST; HEE, 2004). Assim, uma Rede de Petri que modela um processo de *workflow* é denominada *workflow net* (AALST, 1998).

Quando as redes de Petri são utilizadas para modelar um *workflow*, obtém-se então as *workflow nets*. Entretanto, de acordo com Aalst e Hee (2004), uma rede de Petri *RN* é uma *workflow net* se, e somente se:

1. *RN* contém dois lugares especiais: *Início* e *Fim*, onde *Início* é o lugar de entrada, ou seja, não há arcos chegando nele, e *Fim* é o lugar de saída, ou seja não há arcos saindo dele.
2. Se *RN* contém uma transição T_j que liga o lugar *Fim* ao lugar *Início* então a Rede de Petri é fortemente conexa, ou seja, para quaisquer par de nós x e y existe um caminho de x até y Aalst (1998).

Os elementos de uma *workflow net* são os mesmos elementos da Rede de Petri apresentada na Figura 1. Porém, neste contexto, cada elemento possui um significado que corresponde aos elementos de um processo de negócio.

Os lugares estão relacionadas às condições encontradas no processo e às transições estão relacionadas as tarefas. Além disso, uma ficha no lugar *Início* representa um caso a ser tratado pelo processo e uma ficha no lugar *Fim* representa um caso já tratado.

2.4.2 Acionamentos

A noção de acionamento definida em Aalst e Hee (2004) representa uma condição externa que guia a execução de uma tarefa sensibilizada. Existem quatro tipos de acionamento:

- *usuário*: uma tarefa é acionada por um recurso humano;
- *mensagem*: um evento externo aciona uma tarefa sensibilizada;
- *tempo*: uma tarefa sensibilizada é acionada por um relógio, isto é, a tarefa é executada em um tempo pré definido;
- *automática*: uma tarefa é acionada no momento em que é habilitada e não requer interação humana.

A Figura 13 mostra como cada um dos quatro tipos de acionamento é associado às transições.

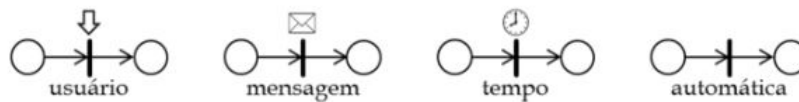


Figura 13 – Tipos de Acionamento

Nota-se que quando um acionamento do tipo “*usuário*” é considerado, a tarefa correspondente é acionada por um recurso humano, isto é, há uma alocação de recursos humanos associada.

2.4.3 Modelagem de um Processo

Para exemplificar a modelagem de um processo em *workflow net*, será considerado um exemplo simplificado apresentado em AALST W. van der; STAHL (2011). A Figura 14 mostra a *workflow net* que modela esse processo que representa a automação da estrutura administrativa de um hospital. Esse mesmo processo de automação da estrutura administrativa poderia também ser utilizado para um outro estabelecimento que não o hospital. Para este processo, foram definidas cinco tarefas a serem executadas.

1. *det_req*: determinar os requisitos funcionais que serão utilizados;
2. *comp_hwre*: fazer a compra dos equipamentos (*hardware*) a serem utilizados;
3. *comp_sftwre*: fazer a compra do programa (*software*) a ser utilizado;
4. *integr*: realizar a instalação e integração do *hardware* com o *software*;
5. *testes*: realizar os testes para verificar o sistema implantado.

A primeira fase do processo consiste em estabelecer os requisitos funcionais que serão utilizados durante a implantação do sistema. Em seguida as duas tarefas de compras podem ser realizadas simultaneamente. A próxima tarefa, a integração, só pode ser realizada quando as tarefas de compras forem concluídas. E por fim, os testes devem ser realizados para então finalizar o processo.

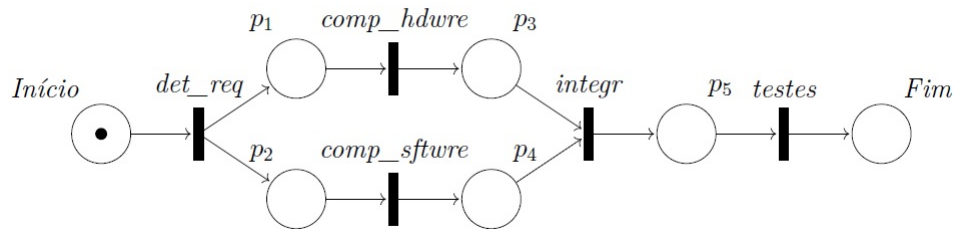


Figura 14 – Exemplo do Hospital - *Workflow net* AALST W. van der; STAHL (2011)

2.5 *SimHPN: MATLAB Toolbox* para Análise e Simulação de Redes de Petri Contínuas

O *SimHPN* é um software embutido no *MATLAB* que permite a análise e simulação de redes de Petri discretas, contínuas e híbridas. O diferencial oferecido por esse pacote do *MATLAB* é que permite a simulação das redes a partir de duas semânticas diferentes de disparo, que são *infinite server* e *product server semantics*. Ao final da simulação, o usuário tem a opção de exportar os resultados para a área de trabalho do *MATLAB* para que os dados sejam utilizados para análise posterior (JÚLVEZ; MAHULEA; VÁZQUEZ, 2012). A Figura 15 mostra um esboço da tela inicial do *SimHPN*.

2.5.1 Interface Gráfica

A ferramenta *SimHPN* (JÚLVEZ; MAHULEA; VÁZQUEZ, 2012) dispõe de uma interface gráfica amigável na qual podem ser desenvolvidas simulações e análise de dados. É possível ter acesso a essa interface a partir do *MATLAB* digitando o comando:

» *SimHPN*

em seu *prompt*.

Essa interface consiste de uma janela com uma Barra de Menu e três painéis de controle:

1. Área de desenho
2. Painel de opções

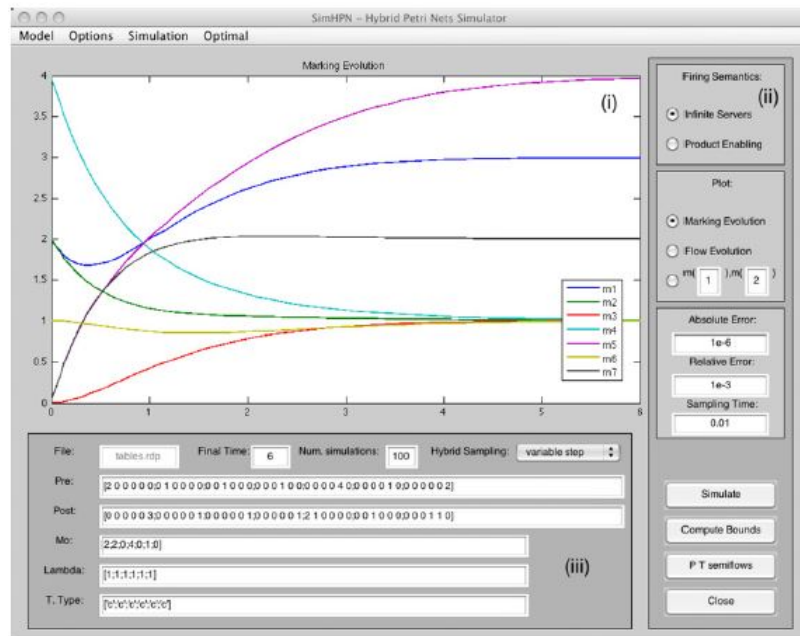


Figura 15 – Esboço da tela inicial do SimHPN

3. Painel de gerenciamento de modelo

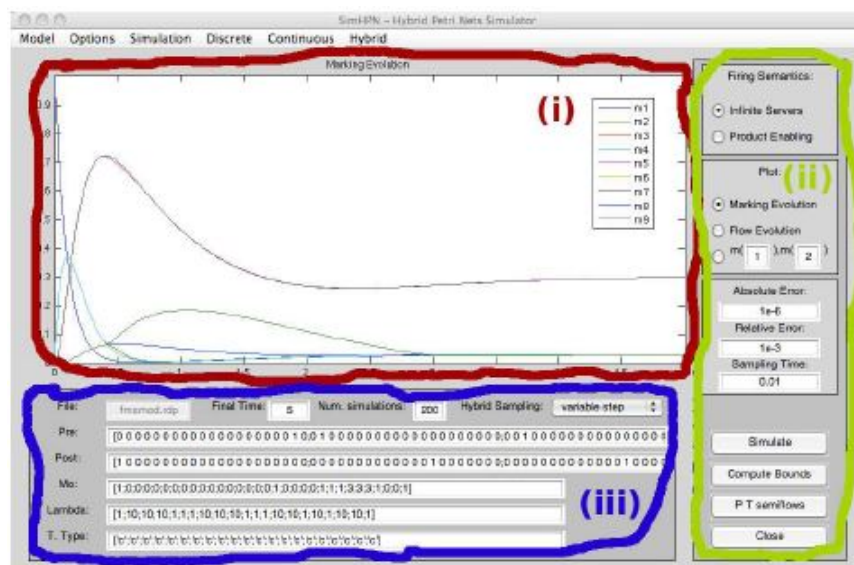


Figura 16 – Interface Gráfica SimHPN

A Figura 16 apresenta um *screenshot* da janela principal do *SimHPN*, no qual todos os componentes citados acima estão presentes. A Barra de Menu possui uma série de sub-menus nos quais o usuário pode acessar algumas das funcionalidades da ferramenta.

O primeiro sub-menu é o *Model*, a partir do qual temos a opção de importar as matrizes de dados necessárias para a simulação, a partir de Editores de rede de Petri como PMediteur e TimeNet, ou importar as matrizes diretamente de um arquivo *.mat* como mostrado na Figura 17. Os dados exigidos pelo sistema para que seja realizada a simulação

são: matrizes *Pre* e *Post*, marcação inicial M_0 , velocidade interna λ das transições e tipo das transições (discreta ou contínua). A última opção deste sub-menu permite salvar em um arquivo .mat as matrizes já definidas para o modelo.

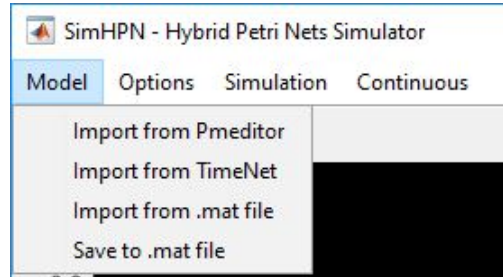


Figura 17 – Menu *Model*

O segundo sub-menu, *Options*, mostra uma barra de ferramentas que permite salvar o gráfico obtido pela simulação que é apresentado na Área de Desenho, aumentar ou diminuir o zoom, mostrar o gráfico em 3D, entre outras opções, como mostrado na Figura 18.

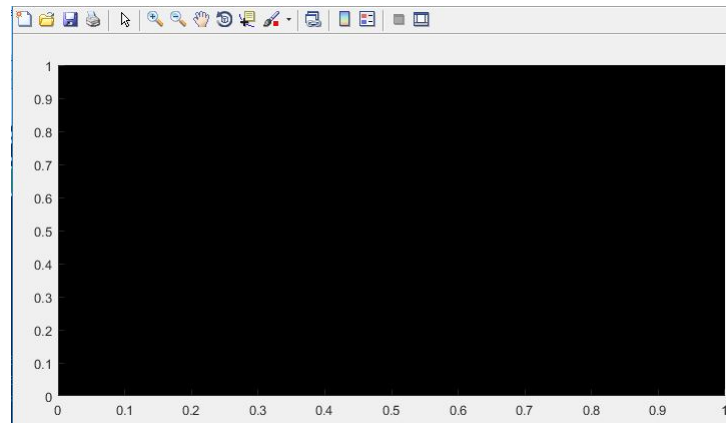


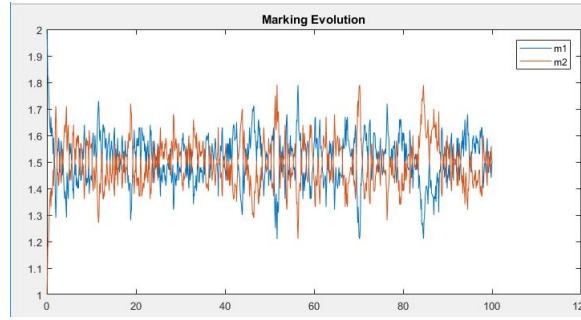
Figura 18 – Menu *Barra de Ferramentas*

O próximo sub-menu, *Simulation*, apresenta opções dos dados que serão plotados no gráfico, tendo como opções: *Markings to plot* ou *Flows to plot*. Ou seja, é possível plotar gráfico com a evolução das marcações (Figura 19a) ou com o fluxo das fichas (Figura 19b) durante a simulação. Os gráficos apresentados na Figura 19 são resultantes da simulação feita na rede de Petri definida na Figura 20.

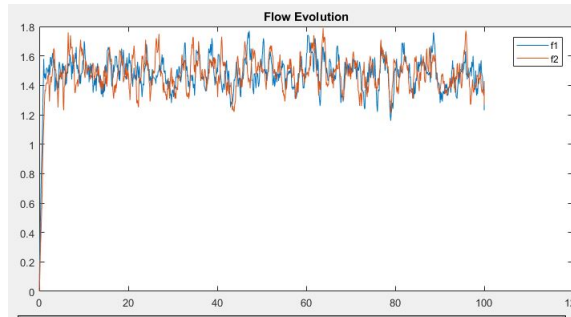
Além disso, em *Save results to workspace*, tem-se a opção de salvar os dados das marcações e fluxo das fichas em variáveis do *MATLAB*, isso após realizada a simulação (JÚLVEZ; MAHULEA, 2012).

2.5.2 Dados de Entrada

Para que a simulação seja realizada, a ferramenta exige que os dados sejam passados manualmente nos campos apresentados no Painel de Gerenciamento de Modelo



(a) Evolução das Fichas



(b) Evolução do Fluxo de Fichas

Figura 19 – Exemplo de Gráficos Resultantes das Simulações

(apresentado em iii, na Figura 16), importados de um arquivo *.mat* ou importados diretamente de editores de rede de Petri como PMEditur ou TimeNet.

Caso não seja utilizado nenhum tipo de editor, dados como o tempo final e o número de simulações devem obrigatoriamente estar presentes. Além disso, outros dados importantes devem ser considerados (lembrando que $|P|$ representa o conjunto dos lugares da rede e $|T|$ das transições da rede) :

- *Pre* - matriz $|P| \times |T|$ que contém os pesos dos arcos que ligam lugares a transições;
- *Post* - matriz $|P| \times |T|$ que contém os pesos dos arcos que ligam transições a lugares;
- M_0 - vetor coluna de dimensão $|P|$ contendo a marcação inicial da rede;
- *Lambda* - vetor coluna de dimensão $|T|$ contendo as taxas de disparo de cada transição;
- *T.Type* - vetor coluna de dimensão $|T|$ contendo o tipo da transição. Cada elemento pode receber um dos seguintes valores:
 - 'c' - para transição contínua;
 - 'd' - para transição estocástica discreta;
 - 'q' - para transição determinística discreta;

Para melhor compreensão do processo de simulação e como funcionam as matrizes e vetores com os dados de entrada, será apresentado a seguir um exemplo simples. A simulação tem como base a rede de Petri estocástica da Figura 20.

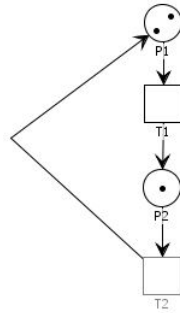


Figura 20 – Exemplo de Rede de Petri Estocástica

Com base na Figura 20, é possível determinar quais serão as dimensões das matrizes e vetores de entrada. As matrizes *Pre* e *Post* têm número de linhas igual ao número de lugares, e número de colunas igual ao de transições, portanto, para este exemplo, as matrizes tem dimensão 2×2 . O vetor coluna M_0 tem dimensão $|P|$, neste caso, 2. Os vetores coluna *Lambda* e *T.Type* têm dimensão $|T|$, neste caso, 2.

Primeiramente é preciso definir a matriz *Pre*, que corresponde aos pesos dos arcos que ligam lugares a transições. A rede de Petri possui dois lugares e duas transições, os lugares P_1 e P_2 possuem cada um, um arco para T_1 e T_2 , respectivamente, de peso igual a 1, o que resulta na matriz:

$$Pre = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

De forma semelhante, é preciso definir a matriz *Post*, que corresponde aos pesos dos arcos que ligam transições a lugares. A rede de Petri possui dois lugares e duas transições, as transições T_1 e T_2 possuem cada uma, um arco para P_2 e P_1 , respectivamente, de peso igual a 1, o que resulta na matriz:

$$Post = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

O vetor M_0 representa a marcação inicial da rede, ou seja, o número de fichas em cada lugar de entrada, que no caso são 2 em P_1 e 1 em P_2 , o que resulta no vetor:

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

O vetor Λ apresenta a taxa de disparo de cada transição, que neste caso, como não possui nenhuma especificação, está definida como disparo instantâneo, ou seja, igual a 1.

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

O vetor $T.Type$ define os tipos de cada transição (contínua, estocástica ou determinística). Neste caso, a rede de Petri apresentada é estocástica, o que resulta no vetor:

$$\begin{bmatrix} 'd' \\ 'd' \end{bmatrix}$$

Definidas todas as matrizes e vetores, os dados obtidos podem ser passados para o *SimHPN*, que ficaria como na Figura 21

The screenshot shows a software interface for SimHPN with the following fields and values:

Field	Value
File:	
Final Time:	100
Num. simulations:	100
Hybrid Sampling:	fixed step
Pre:	[1 0; 0 1]
Post:	[0 1; 1 0]
Mo:	[2; 1]
Lambda:	[1; 1]
T. Type:	['d'; 'd']

Figura 21 – Dados de Entrada no *SimHPN*

2.5.3 Semânticas de Disparo

Como já foi dito anteriormente, o SimHPN possui duas políticas principais de disparo de transições, e são elas *infinite server* e *product server semantics* que serão explicadas com mais detalhes nesta seção (JÚLVEZ; MAHULEA, 2010).

2.5.3.1 Infinite Server Semantics

Em um sistema de redes de Petri estocásticas, a semântica de *infinite servers* significa que o atraso entre a sensibilização e disparo de uma transição é o mínimo de n variáveis independentes distribuídas exponencialmente (sendo n o número de transições habilitadas), cada uma com seu respectivo λ_j . Podemos então interpretar, do ponto de vista contínuo, $f_j = \lambda \cdot enab(t_j)$ que corresponde à velocidade variável apresentada em (ALLA; DAVID, 1998).

No caso de *infinite servers* (velocidade variável), podemos apresentar o fluxo da transição t_j a partir da seguinte equação:

$$f_j = \lambda_j \cdot enab(t_j, m) = \lambda_j \cdot \min_{p_i \in \bullet t_j} \frac{m_i}{Pre(p_i, t_j)}$$

onde λ_j é um real positivo que representa a velocidade de disparo associada a t_j . O fluxo f_j relacionado à transição t_j terá valor igual à velocidade λ_j multiplicada pelo valor do menor número de fichas dentre os lugares de entrada de t_j .

2.5.3.2 Product Server Semantics

Redes de Petri Contínuas são geralmente utilizadas para modelar a evolução de um sistema de populações. Assim, quando modelado, o fluxo das transições torna-se o produto do grau de sensibilização de cada lugar de entrada, ou seja, o fluxo f_j relacionado à transição t_j terá valor igual à velocidade λ_j multiplicada pela multiplicação do número de fichas em todos os lugares de entrada de t_j (RECALDE; SILVA, 2000) :

$$f_j = \lambda_j \cdot \prod_{p_i \in \bullet t_j} \frac{m_i}{Pre(p_i, t_j)}$$

2.5.3.3 Exemplo das Semânticas de Disparo

Para melhor compreensão de como funciona cada semântica de disparo, será apresentado o exemplo de uma simulação de uma rede de Petri contínua simples, utilizando cada semântica de disparo e seus respectivos resultados. A rede de Petri contínua a ser simulada está representada na Figura 22. Além disso, as velocidades associadas a T_1 e T_2 são 0.5 e 0.25 respectivamente.

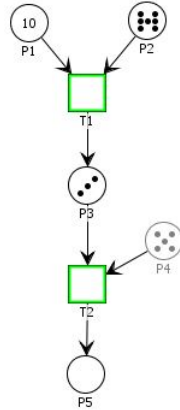


Figura 22 – Rede de Petri Contínua

O resultado da simulação desta rede na ferramenta *SimHPN* é o gráfico da evolução das marcações da rede apresentado na Figura 23.

A partir da simulação também é possível obter como resultado o gráfico da evolução dos fluxos de cada transição, tanto para *Infinite Server Semantics* quanto para *Product Server Semantics*. O gráfico apresentado em 23 é o mesmo para as duas semânticas já que está relacionado a informações relativas às marcações dos lugares, e não às transições. A seguir serão apresentados os resultados para as duas semânticas de disparo.

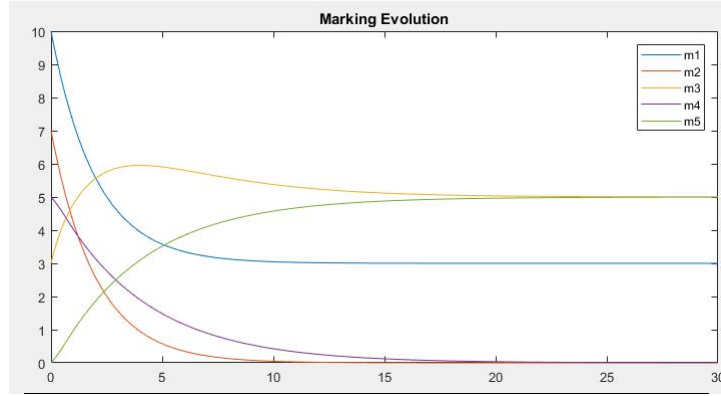


Figura 23 – Gráfico das Marcações da Rede de Petri Contínua

A primeira simulação foi realizada utilizando *Infinite Server Semantics*. Com relação à transição T_1 , no instante 0, por exemplo, temos 10 fichas em P_1 e 7 fichas em P_2 , que são os lugares de entrada de T_1 . Segundo a fórmula apresentada para o cálculo do fluxo em cada transição, o fluxo em T_1 deve ser igual ao λ_{T_1} , que é igual a 0.5, multiplicado pelo menor número de fichas presente em um de seus lugares de entrada, que neste caso é igual a 7 fichas no lugar P_2 , que resultaria em:

$$f_1 = 0.5 \cdot 7 = 3.5$$

Com relação a T_2 , temos 3 fichas em P_3 e 5 fichas em P_4 , que são os lugares de entrada de T_2 e segundo a fórmula, o fluxo deve ser igual ao λ_{T_2} que é igual a 0.25, multiplicado pelo menor número de fichas presente em um de seus lugares de entrada, que neste caso é igual a 3 fichas no lugar P_3 .

$$f_2 = 0.25 \cdot 3 = 0.75$$

Esses valores de fluxo para $t = 0$ podem ser observados no gráfico resultante da simulação, em 24.

A segunda simulação foi realizada utilizando *Product Server Semantics*. Com relação à transição T_1 no instante 0, por exemplo, temos 10 fichas em P_1 e 7 fichas em P_2 , que são os lugares de entrada de T_1 . Segundo a fórmula apresentada para o cálculo do fluxo em cada transição, o fluxo em T_1 deve ser igual ao λ_{T_1} , que é igual a 0.5, multiplicado pela multiplicação do número de fichas presente em cada um de seus lugares de entrada, que neste caso é igual a 70, que resultaria em:

$$f_1 = 0.5 \cdot 70 = 35$$

Com relação a T_2 , temos 3 fichas em P_3 e 5 fichas em P_4 , que são os lugares de entrada de T_2 e segundo a fórmula, o fluxo deve ser igual ao λ_{T_2} que é igual a 0.25, multiplicado pela multiplicação do número de fichas presente em cada um de seus lugares de entrada, que neste caso é igual a 15.

$$f_2 = 0.25 \cdot 15 = 3.75$$

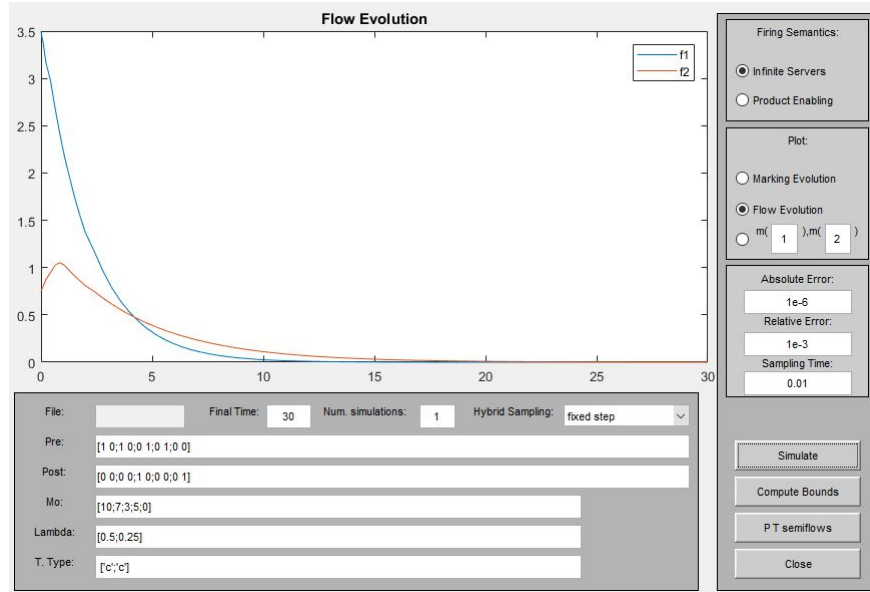


Figura 24 – Gráfico das Evolução dos Fluxos de uma Rede de Petri Contínua utilizando *Infinite Server Semantics*

Esses valores de fluxo para $t = 0$ podem ser observados no gráfico resultante da simulação, em 25.

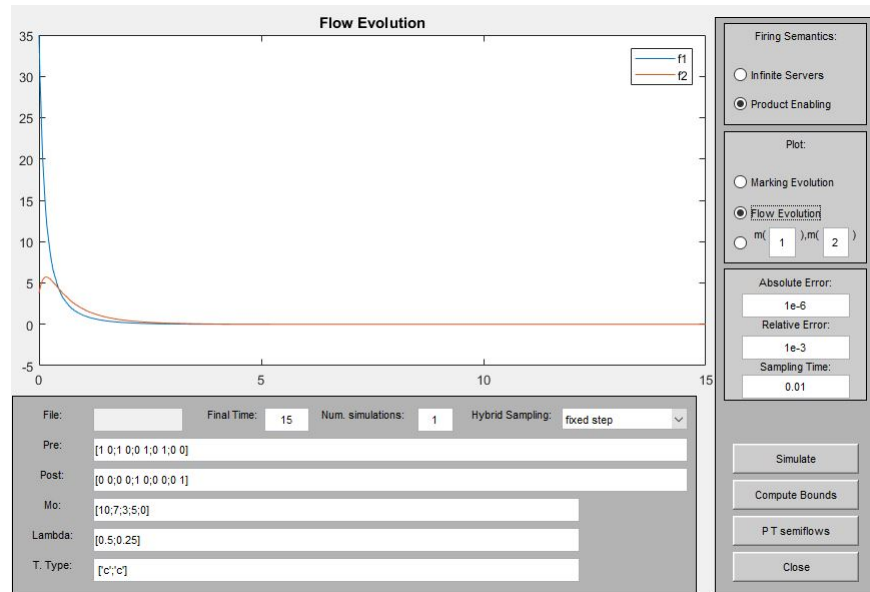


Figura 25 – Gráfico das Evolução dos Fluxos de uma Rede de Petri Contínua utilizando *Product Server Semantics*

2.6 Trabalhos Relacionados

Um dos principais trabalhos correlatos a essa pesquisa é apresentado em (PETRI, 1962). Nessa tese, Carl Adam Petri introduz um modelo matemático para descrever as relações existentes entre eventos e condições, o que corresponde hoje à chamada Rede de

Petri, que é a base para a pesquisa proposta para esse trabalho. Inicialmente, engenheiros apresentaram o modelo para modelagem de sistemas dinâmicos de eventos discretos. Porém, com o passar dos anos, baseado em várias pesquisas na área, observou-se alguns problemas relacionados às redes de Petri ordinárias, ou seja, redes discretas. Foi então que o conceito de rede Petri contínua foi introduzido em (DAVID, 1987) na qual suas fichas deixam de ser representadas por números inteiros, passando a ser representadas por números reais. Essa mudança visa solucionar o problema da explosão combinatorial de estados presente nas redes discretas.

O conceito de rede de Petri híbrida, também introduzido em (DAVID, 1987), só foi desenvolvido em (BAIL, 1991). As redes de Petri híbridas podem ser definidas como um modelo no qual variáveis de estados discretos são combinadas com variáveis de estados contínuos, misturando assim as características positivas de cada modelo. No livro (DAVID; ALLA, 2010), é feita uma abordagem completa sobre as redes de Petri, definindo desde as redes mais simples até as mais complexas.

As redes de Petri são conhecidas por serem uma ferramenta poderosa para a modelagem e análise de sistemas discretos. As redes de Petri contínuas e híbridas, extensões do modelo básico, permitem a modelagem e análise de sistemas contínuos e híbridos na mesma base conceitual (DAVID; ALLA, 2001).

Em (AALST; HEE, 2001), o autor define o que são processos de *workflow* e o que são as *workflow net* utilizadas no nosso modelo. Um processo de *workflow* compreende casos, recursos e *triggers* que se relacionam com um processo em particular. O ponto de ligação entre as redes de Petri e seu uso para modelagem de processos se dá na definição de *workflow net* que consiste em uma rede de Petri que representa um processo fluxo de trabalho.

As redes de Petri oferecem um rico poder de modelagem que tem sido explorado com sucesso em diferentes áreas como manufatura, logística, tráfego, informática e biologia (MURATA, 1989). Para desenvolver e analisar modelos de rede de Petri de forma eficiente são necessárias ferramentas de modelagem, análise e simulação adequadas. Entre as plataformas de *software* existentes para desenvolvimento de *software* científico, ambientes como o *MATLAB* disponibilizam ao usuário um grande conjunto de ferramentas com uma interface gráfica fácil de manipular (JÚLVEZ; MATCOVSCHI; PASTRAVANU, 2014). Dessa forma, ferramenta utilizada para este trabalho será, a princípio, um *MATLAB Petri Net Toolbox*.

3 Modelagem de um Processo de Negócio por meio de uma *Workflow net*

3.1 Introdução

A etapa inicial do trabalho consistiu na realização de um estudo da literatura com o auxílio de referências como (CARDOSO; VALETTE, 1997; AALST; HEE, 2001) que forneceram uma base para o entendimento dos conceitos básicos necessários para o desenvolvimento do trabalho como a definição das redes de Petri discretas, e os trabalhos relacionados ao problema de modelagem e simulação de fluxos de trabalhos (*workflow*) em sistemas de gerenciamento de processos de negócios. Em particular, o livro (DAVID; ALLA, 2010) introduz os principais conceitos considerados neste trabalho, que são as redes de Petri contínuas que correspondem aos modelos utilizados nesta proposta.

Depois do entendimento desses conceitos e do funcionamento de ferramentas como *Snoopy* (HERAJY; HEINER, 2013) e *MATLAB Petri Net Toolbox* (JÚLVEZ; MAHULEA; VÁZQUEZ, 2012), necessários para o desenvolvimento do trabalho, foi possível a realização da modelagem, análise e simulação, coerente e com resultados objetivos em função do estado atual do conhecimento na área. A base para a análise proposta consiste, primeiramente, na seleção um estudo de caso.

Após a definição do estudo de caso, este foi modelado utilizando uma rede de Petri temporizada estocástica com recursos discretos, assim como a simulação e análise do modelo produzido para o regime estacionário do processo. Para edição e simulação do modelo, foi utilizada a ferramenta *MATLAB Petri Net Toolbox* (simulador de redes de Petri discretas, contínuas e híbridas). Em seguida, transformou-se o modelo discreto em um modelo contínuo para simular e avaliar seu regime estacionário, utilizando a mesma ferramenta proposta para a simulação de redes de Petri temporizadas estocásticas.

Feito isso, foi realizada a comparação dos resultados das duas abordagens utilizando os seguintes parâmetros de análise quantitativa:

1. Comparação dos valores médios dos tempos de atividade;
2. Comparação das filas e gargalos existentes em ambos os modelos;
3. Avaliação das vantagens e limitações do modelo contínuo em relação ao discreto para modelagem e análise de processos de negócios, em particular em relação aos tempos de simulação necessários para se atingir o regime estacionário do processo.

Neste capítulo será apresentado o processo de modelagem de uma *workflow net* no *SimHPN* e no *Snoopy* (ferramenta utilizada apenas para modelagem gráfica da rede de Petri). O processo de negócio modelado foi uma adaptação do processo de tratamento de reclamações (“*Handle Complaint Process*”) apresentado em (AALST; HEE, 2004). A descrição do processo é apresentada a seguir:

An incoming complaint first is recorded. Then the client who has complained and the department affected by the complaint are contacted. The client is approached for more information. The department is informed of the complaint and may be asked for its initial reaction. These two tasks may be performed in parallel – that is, simultaneously or in any order. After this, the data are gathered and a decision is taken. Depending upon the decision, either a compensation payment is made or a letter is sent. Finally, the complaint is filed (AALST; HEE, 2004).

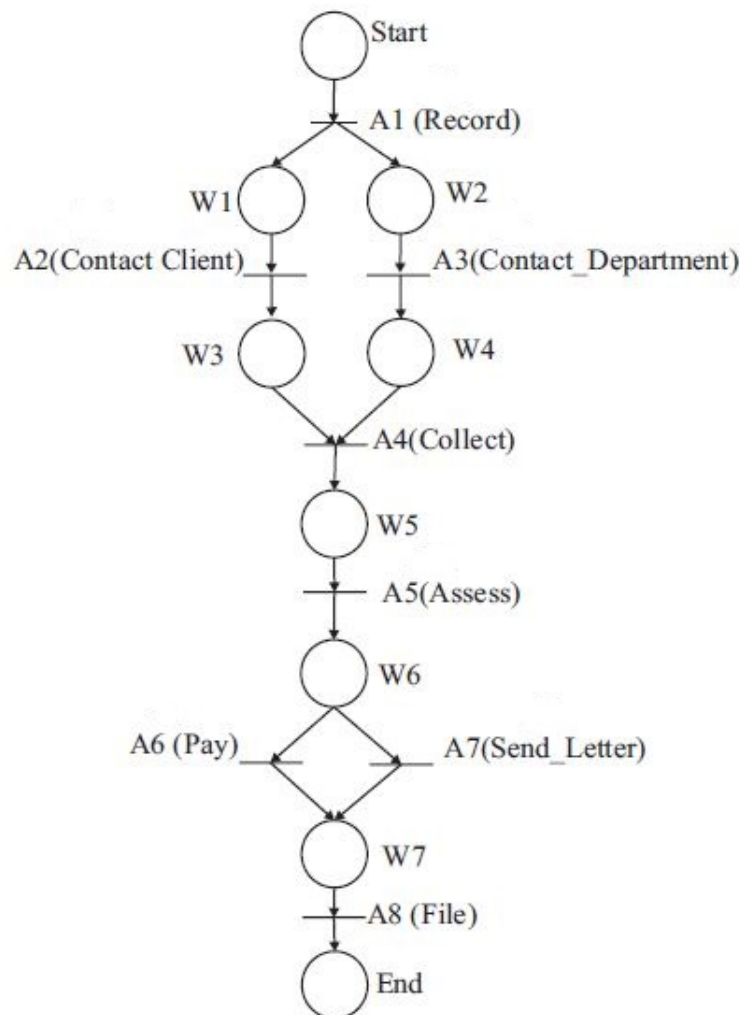


Figura 26 – Workflow net para o “Processo de Tratamento de Reclamações” (AALST; HEE, 2004)

A Figura 26 representa o processo de negócio descrito. Com base na figura e na descrição, alguns detalhes podem ser observados. O processo se inicia quando existir pelo

menos uma ficha no lugar *Start*, sendo que essa ficha representa o cliente. A presença de uma ficha em *Start* torna a transição A_1 sensibilizada e pronta para ser disparada, e seu disparo corresponderá ao registro da reclamação.

Em seguida, tanto o cliente quanto o departamento que sofreu a reclamação devem ser contactados, o que ocorre por meios das tarefas *Contact_Client* e *Contact_Departamento*, que podem ocorrer em paralelo, ou seja, as duas tarefas podem ser executadas simultaneamente ou em qualquer ordem. Assim que as duas atividades forem finalizadas, a transição A_4 estará pronta para ser disparada. Como as tarefas *Collect* e *Assess* só podem ser iniciadas quando a tarefa anterior a elas for concluída, são caracterizadas por um roteamento sequencial. As tarefas seguintes, *Pay* e *Send_Letter*, caracterizam um roteamento condicional no qual o processo pode “escolher” entre uma das duas para ser executada. Após essa escolha, a reclamação é arquivada e o processamento dessa reclamação chega ao fim.

3.2 Implementação de uma Workflow net sem Recursos

Inicialmente, será feita a modelagem do processo de forma semelhante ao apresentado na Figura 26, com algumas pequenas adaptações. Na seção seguinte, será feita uma nova modelagem do processo na qual serão associados recursos à algumas tarefas.

Para melhor visualização da rede de Petri a ser simulada no *SimHPN*, uma representação gráfica da rede foi criada no software *Snoopy*. A Figura 27 mostra um modelo de rede de Petri discreta no qual o lugar de entrada P_2 é alimentado de forma contínua por P_1 . Isso foi feito para simular de forma contínua a chegada de reclamações a serem tratadas. Cada transição T_i possui um intervalo de tempo associado. Esse intervalo é a base para o cálculo do tempo médio gasto em cada transição, que será utilizado no *SimHPN* para calcular a frequência de disparo λ_i de cada transição.

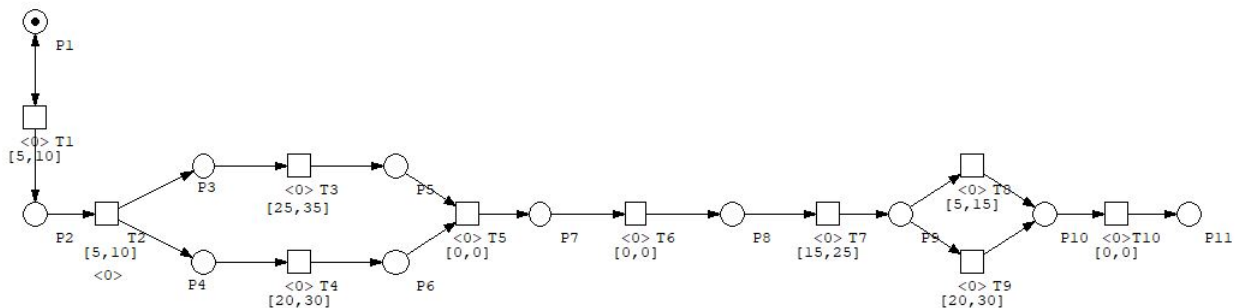


Figura 27 – Modelo 1: Processo de Tratamento de Reclamações sem Recursos

Para inserir essa rede de Petri no software *SimHPN*, é preciso gerar a notação matricial do grafo da Figura 27. A primeira matriz *Pre* possui $|P|$ linhas e $|T|$ colunas

e representa os arcos que vão de cada lugar para cada transição. Na rede em questão o peso de todos os arcos é igual a 1, portanto quando existir um arco ligando P_1 a T_1 , por exemplo, a posição na matriz correspondente (linha 1 coluna 1) será preenchida com o valor 1. Caso não exista nenhum arco entre esse lugar e essa transição, o valor na matriz será 0. A matriz da Figura 28 representa os arcos de entrada das transições da rede representada na Figura 27.

Pre	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
P1	1	0	0	0	0	0	0	0	0	0
P2	0	1	0	0	0	0	0	0	0	0
P3	0	0	1	0	0	0	0	0	0	0
P4	0	0	0	1	0	0	0	0	0	0
P5	0	0	0	0	1	0	0	0	0	0
P6	0	0	0	0	1	0	0	0	0	0
P7	0	0	0	0	0	1	0	0	0	0
P8	0	0	0	0	0	0	1	0	0	0
P9	0	0	0	0	0	0	0	1	1	0
P10	0	0	0	0	0	0	0	0	0	1
P11	0	0	0	0	0	0	0	0	0	0

Figura 28 – Modelo 1: Matriz Pre

A matriz *Post* possui $|P|$ linhas e $|T|$ colunas e representa os arcos que vão de cada transição para cada lugar. Como todos os arcos tem peso 1, quando existir um arco ligando T_3 a P_4 , por exemplo, a posição na matriz correspondente (linha 4 coluna 3) será preenchida com o valor 1. Caso não exista nenhum arco entre essa transição e esse lugar, o valor na matriz será 0. A matriz da Figura 29 representa os arcos de saída das transições da rede representada na Figura 27.

Post	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
P1	1	0	0	0	0	0	0	0	0	0
P2	1	0	0	0	0	0	0	0	0	0
P3	0	1	0	0	0	0	0	0	0	0
P4	0	1	0	0	0	0	0	0	0	0
P5	0	0	1	0	0	0	0	0	0	0
P6	0	0	0	1	0	0	0	0	0	0
P7	0	0	0	0	1	0	0	0	0	0
P8	0	0	0	0	0	1	0	0	0	0
P9	0	0	0	0	0	0	1	0	0	0
P10	0	0	0	0	0	0	0	1	1	0
P11	0	0	0	0	0	0	0	0	0	1

Figura 29 – Modelo 1: Matriz Post

M_0 é um vetor coluna de dimensão igual ao número de lugares na rede no qual é apresentada a marcação inicial da rede que corresponde à distribuição inicial das fichas de cada um dos lugares, representando em particular o estado inicial do processo. Na rede da Figura 27 existe apenas uma ficha em P_1 e o restante dos lugares não possui nenhuma. A Figura 30 representa M_0 .

M0	
P1	1
P2	0
P3	0
P4	0
P5	0
P6	0
P7	0
P8	0
P9	0
P10	0
P11	0

Figura 30 – Modelo 1: Vetor M_0

Ao vetor coluna λ estão associadas as taxas (frequências) de disparo de cada transição. Observa-se na Figura 27 que existe um intervalo de tempo associado à maioria das transições. Aquelas que não possuem nenhum valor associado, considera-se que os disparos são instantâneos, contanto que estejam sensibilizadas. Por exemplo, considerando que a transição T_4 tem um intervalo de tempo associado de $[20,30]$, para encontrar λ_4 é feita a média de 20 e 30 ($\frac{20+30}{2}$) e o valor resultante é utilizado como denominador de uma fração que tem numerador igual a 1, ou seja, $\lambda_4 = \frac{1}{25} = 0.04$. A Figura 31 representa *lambda*.

Lambda	
T1	0.133
T2	0.133
T3	0.033
T4	0.04
T5	1
T6	1
T7	0.05
T8	0.1
T9	0.04
T10	1

Figura 31 – Modelo 1: Vetor λ

O vetor $T.Type$ tem dimensão igual ao número de transições e é preenchido com o tipo de cada transição. Para o modelo em estudo, serão realizadas simulações tanto com transições discretas temporizadas com distribuições estocásticas exponenciais quanto com transições contínuas. A Figura 32 representa o vetor $T.Type$ para o caso de simulação discreta estocástica (Figura 32a) e para o caso de simulação contínua (Figura 32b).

T.Type	
T1	'd'
T2	'd'
T3	'd'
T4	'd'
T5	'd'
T6	'd'
T7	'd'
T8	'd'
T9	'd'
T10	'd'

(a)

T.Type	
T1	'c'
T2	'c'
T3	'c'
T4	'c'
T5	'c'
T6	'c'
T7	'c'
T8	'c'
T9	'c'
T10	'c'

(b)

Figura 32 – Vetores $T.Type$

3.3 Implementação de uma *Workflow net* com Recursos

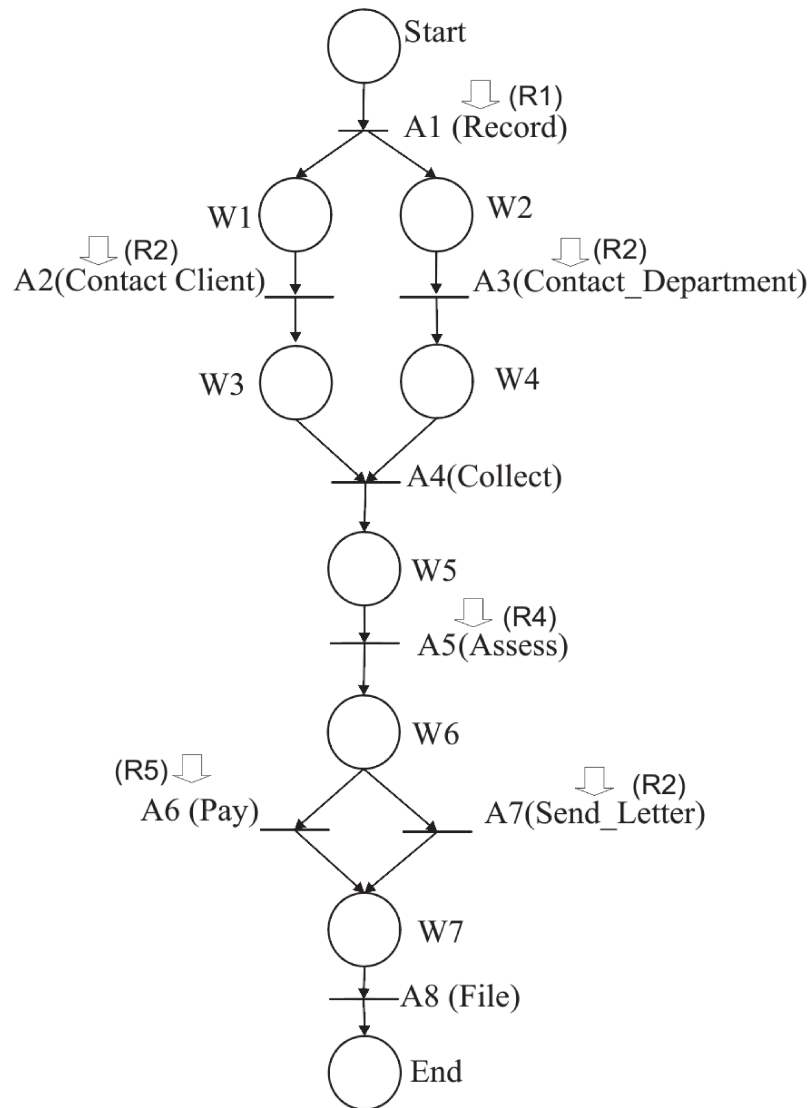


Figura 33 – *Workflow net* para o “Processo de Tratamento de Reclamações” com recursos

A Figura 33 representa o modelo de “Processo de Tratamento de Reclamações” apresentado na Figura 26, porém foram associados recursos (acionamento de tipo usuário) a algumas atividades do processo como *Record*, *Contact_Client*, *Pay*, *Contact_Department*, *Asses* e *Send_Letter*. Com base neste modelo, uma nova implementação de uma *workflow net* com alocação de recursos será gerada no *SimHPN* e, em seguida, suas propriedades e resultados serão comparadas com os mesmos parâmetros analisados para a *workflow net* sem recursos.

A adição de recursos ao modelo de rede de Petri contribui para uma maior precisão na representação do processo de reclamações. É possível em processos de negócios associar a maioria dos recursos representados num modelo de processo com funcionários que desempenham as tarefas para as quais eles foram designados. Algumas tarefas po-

dem demandar mais ou menos tempo e ainda maior ou menor quantidade de funcionários alocados.

Com a inclusão de recursos, é comum o aparecimento de gargalos que são acúmulos de fichas em determinados lugares da rede. Porém, este tipo de comportamento indica problemas de dimensionamento de recursos, de forma que os gargalos devem ser eliminados para que o processo fique balanceado.

O modelo de rede de Petri com recursos gerado utilizando o *Snoopy* é apresentado na Figura 34.

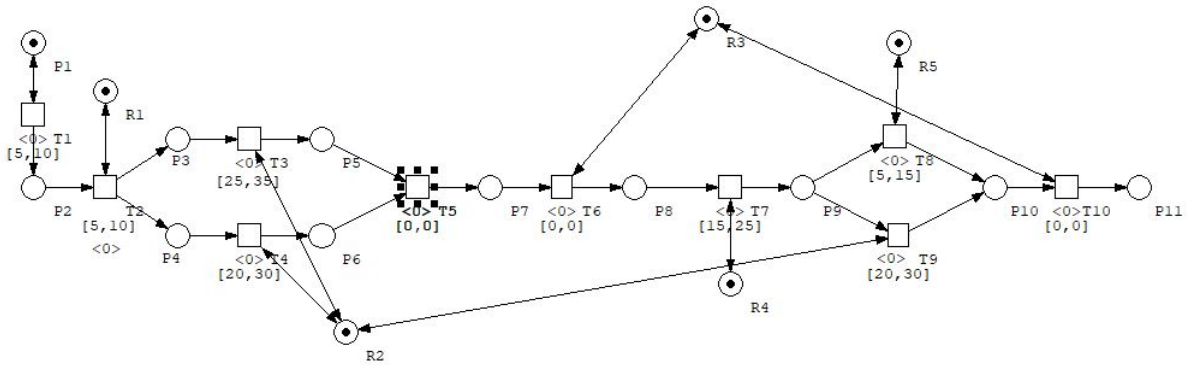


Figura 34 – Modelo 2: Processo de Tratamento de Reclamações com Recursos

No modelo anterior, apresentado na Figura 27, assim como no modelo agora proposto, a alimentação da rede, que são as reclamações recebidas, ocorre de forma contínua. Com a adição de um recurso R_1 ligado à transição T_2 , esta que antes estaria sensibilizada quando recebesse pelo menos uma ficha de P_2 , agora passa a depender também do recurso R_1 , de forma que só estará sensibilizada quando houver pelo menos uma ficha tanto em P_2 quanto em R_1 . A mesma dependência de sensibilização ocorre com as transições T_3 , T_4 e T_9 que agora dependem de R_2 , T_6 e T_{10} que dependem de R_3 , T_7 que depende de R_4 e T_8 que depende de R_5 .

Da mesma forma que no modelo anterior, para que a rede de Petri possa ser simulada utilizando o *SimHPN*, é preciso apresentar as matrizes que contenham os dados referentes aos arcos que ligam lugares às transições (matriz *Pre*) e os dados referentes aos arcos que ligam transições a lugares (matriz *Post*). Esses dados estão representados nas Figuras 35 e 36, respectivamente. Como já apresentado, ambas as matrizes possuem dimensão $|P| \times |T|$ e seus campos são preenchidos com os valores dos pesos dos arcos que ligam lugares e transições.

Além disso é preciso definir o vetor M_0 que representa a marcação inicial da rede e que, neste modelo, passa a ter mais 5 posições que são referentes às fichas iniciais de cada lugar que representa um recurso, e está representado na Figura 37. Os vetores λ e $T.Type$ são semelhantes aos vetores do modelo sem recursos (apresentados nas Figuras 31

Post	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
P1	1	0	0	0	0	0	0	0	0	0
P2	1	0	0	0	0	0	0	0	0	0
P3	0	1	0	0	0	0	0	0	0	0
P4	0	1	0	0	0	0	0	0	0	0
P5	0	0	1	0	0	0	0	0	0	0
P6	0	0	0	1	0	0	0	0	0	0
P7	0	0	0	0	1	0	0	0	0	0
P8	0	0	0	0	0	1	0	0	0	0
P9	0	0	0	0	0	0	1	0	0	0
P10	0	0	0	0	0	0	0	1	1	0
P11	0	0	0	0	0	0	0	0	0	1
P13(R1)	0	1	0	0	0	0	0	0	0	0
P14(R2)	0	0	1	1	0	0	0	0	1	0
P15(R3)	0	0	0	0	0	1	0	0	0	1
P16(R4)	0	0	0	0	0	0	1	0	0	0
P17(R5)	0	0	0	0	0	0	0	1	0	0

Figura 35 – Modelo 2: Matriz Pre

Pré	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
P1	1	0	0	0	0	0	0	0	0	0
P2	0	1	0	0	0	0	0	0	0	0
P3	0	0	1	0	0	0	0	0	0	0
P4	0	0	0	1	0	0	0	0	0	0
P5	0	0	0	0	1	0	0	0	0	0
P6	0	0	0	0	1	0	0	0	0	0
P7	0	0	0	0	0	1	0	0	0	0
P8	0	0	0	0	0	0	1	0	0	0
P9	0	0	0	0	0	0	0	1	1	0
P10	0	0	0	0	0	0	0	0	0	1
P11	0	0	0	0	0	0	0	0	0	0
P13(R1)	0	1	0	0	0	0	0	0	0	0
P14(R2)	0	0	1	1	0	0	0	0	1	0
P15(R3)	0	0	0	0	0	1	0	0	0	1
P16(R4)	0	0	0	0	0	0	1	0	0	0
P17(R5)	0	0	0	0	0	0	0	1	0	0

Figura 36 – Modelo 2: Matriz Post

e 32), isso porque ambos possuem dimensão igual ao número de transições, que não sofre nenhuma alteração com a adição de recursos à rede de Petri.

M0	T1
P1	1
P2	0
P3	0
P4	0
P5	0
P6	0
P7	0
P8	0
P9	0
P10	0
P11	0
P13(R1)	1
P14(R2)	4
P15(R3)	1
P16(R4)	2.7
P17(R5)	1

Figura 37 – Modelo 2: Vetor M_0

4 Simulação: Análise Quantitativa

O objetivo da análise quantitativa está relacionado com o tempo de execução e planejamento de recursos para o caso apresentado. O planejamento tem em vista a distribuição mais adequada no tempo dos recursos para que o processo de tratamento de reclamações ocorra de forma mais otimizada possível. Dessa forma, o objetivo da análise é prever a quantidade de recursos que deve ser associada a cada atividade e respeitar os prazos de entrega dos casos do processo. Em particular deverão ser detectados possíveis situações de gargalos e encontrar uma distribuição da quantidade de recursos que permitirá a eliminação dos gargalos na execução do processo.

Para realizar as simulações, como citado anteriormente, o ambiente utilizado será o *MATLAB Toolbox SimHPN*. A ferramenta possui os locais apropriados para preencher as matrizes e vetores de entrada, e gera como resultado um gráfico da evolução das marcações em relação ao tempo. A seguir serão apresentados resultados de simulações considerando os dois modelos apresentados anteriormente (modelo de processo de reclamação sem e com recursos).

Algumas informações com o tempo final da simulação e o número de simulações devem ser fornecidas. Utiliza-se para o modelo sem recursos, tempo final igual a 500 segundos, o número de simulações é igual a 200 no caso discreto estocástico, e 1 no caso contínuo. O modelo com adição de recursos também foi simulado tanto de forma discreta temporizada estocástica quanto contínua para comparação de resultados. O tempo final e o número de simulações foram iguais ao do modelo sem recursos, 500 segundos, 200 simulações para o caso discreto estocástico e 1 simulação para o caso contínuo. No caso contínuo, apenas uma simulação é suficiente para atingir o regime estacionário do processo, já que o modelo não é estocástico, mas determinístico.

4.1 Simulação: Modelo 1 - Discreto Temporizado Estocástico

A primeira simulação realizada foi referente à rede sem recursos, operando com recursos e transições discretas temporizadas estocásticas. Após preencher as matrizes e vetores de entrada, de acordo com o modelo proposto, a simulação foi iniciada, gerando como resultado o gráfico apresentado na Figura 38.

A partir do gráfico obtido, é possível observar que existe apenas uma marcação que cresce de forma exponencial, que corresponde à marcação final m_{11} . Isso ocorre porque o lugar P_{11} apenas recebe fichas e não as encaminha para um outro lugar e, como a alimentação da rede é constante, elas acabam se acumulando no lugar final. As outras

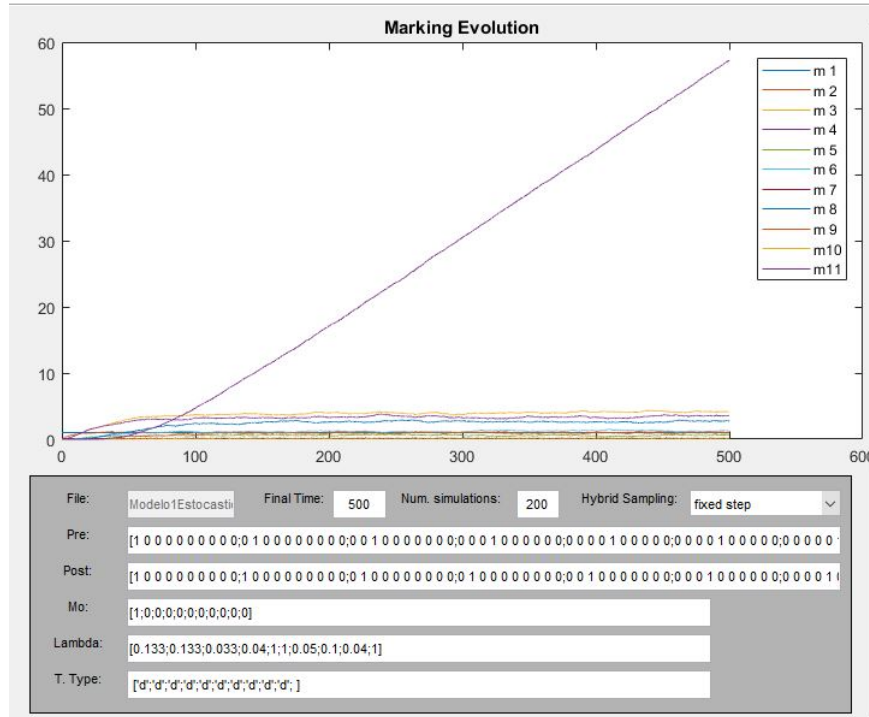


Figura 38 – Modelo 1 - Estocástico

marcações apresentam um crescimento muito pequeno, atingindo no máximo 4 fichas por lugar até que estabilizam seu crescimento e assim permanecem até o fim do tempo da simulação, que foi determinado como 500s. Dessa forma, é possível observar que a rede não apresenta nenhum gargalo.

Como o número de simulações definido foi igual 200, o gráfico apresenta a média dos valores obtidos nas 200 simulações. A partir da comparação desse gráfico com outros resultantes de simulações posteriores, será possível observar que quanto maior o número de simulações, mais o gráfico do modelo estocástico se aproxima do gráfico do modelo contínuo, por isso adotou-se uma quantidade tão expressiva de simulações.

Além das informações obtidas a partir do gráfico, no *prompt* de comando do *MATLAB* são apresentadas algumas informações adicionais como o tempo gasto para que a simulação fosse realizada, que neste caso foi de aproximadamente 365.3 segundos; a marcação final da rede, na qual se destaca a quantidade de fichas no lugar final (m_{11}) que é aproximadamente 57 fichas neste caso; e a taxa de disparo de cada transição ao final da simulação. Essas informações estão presentes na Figura 39.

A partir da informação do número de fichas no lugar final e do tempo de simulação, que é de 500 segundos, é possível concluir que o tempo para tratar cada caso é, em média, 8.7 segundos. Um detalhe importante sobre as marcações apresentadas nessa Figura é que cada marcação representa o valor médio das marcações apresentadas nas 200 simulações realizadas a partir deste modelo; portanto, esses números reais não significam que o número de fichas nas simulações não seja inteiro.

```
Elapsed time is 365.295980 seconds.  
Final marking: [1 1.02 4.155 3.575 0.68 1.26 0.14 2.72 1.05 0.13 57.245]  
Final flow: [0.11 0.135 0.15 0.1 0.135 0.125 0.175 0.075 0.02 0.13]
```

Figura 39 – Modelo 1 - Discreto Temporizado Estocástico - Informações adicionais

4.2 Simulação: Modelo 1 - Contínuo

A segunda simulação realizada é referente à rede sem recursos, com fichas e transições contínuas e dá origem ao gráfico da Figura 40. A Figura 41 apresenta informações como o tempo gasto para realizar a simulação, que foi de aproximadamente 1.26 segundos, e a quantidade de fichas no lugar final (m_{11}) que é de aproximadamente 57 fichas. Assim como para o caso discreto estocástico, como o tempo final da simulação é o mesmo e o número de fichas no lugar final também é o mesmo, a simulação contínua também gasta em média 8.7 segundos para tratar cada caso.

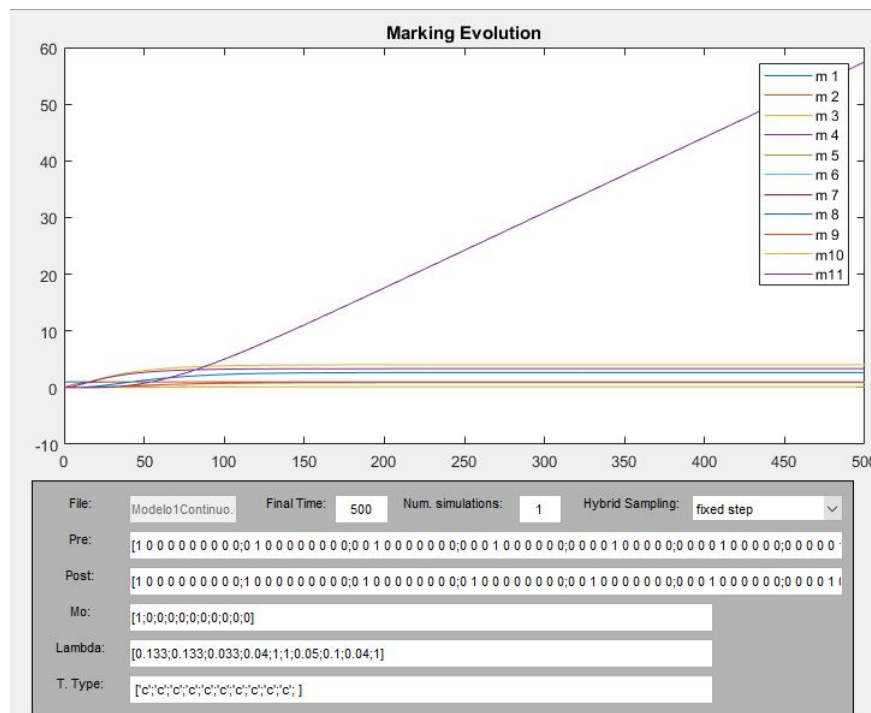


Figura 40 – Modelo 1 - Contínuo

```
Elapsed time is 1.261821 seconds.  
Final marking: [1 1 4.0303 3.325 0.133 0.838303 0.133027 2.65997 0.950001 0.1329 57.4608]  
Final flow: [0.133 0.133 0.133 0.133 0.133 0.132978 0.133001 0.0949998 0.0379999 0.133079]
```

Figura 41 – Modelo 1 - Contínuo - Informações adicionais

Assim como no modelo estocástico, existe apenas uma marcação que cresce de forma exponencial, a marcação m_{11} . Isso se deve ao fato de que a alimentação de fichas no lugar inicial é contínua e que, como o lugar final é o destino final das fichas, acaba ocorrendo um acúmulo de fichas.

Comparando os gráficos resultantes das simulações contínua e discreta do *Modelo 1* e o tempo gasto em cada simulação, conclui-se no caso da simulação de um modelo discreto temporizado estocástico é necessário um número significativo de replicações da simulação para que o gráfico se estabilize, mostrando o regime estacionário que será atingido pelo processo depois de um certo tempo de funcionamento. Isso se deve ao fato de que cada replicação gera um cenário de simulação distinto. A Figura 42 mostra o gráfico gerado por uma única simulação discreta estocástica.

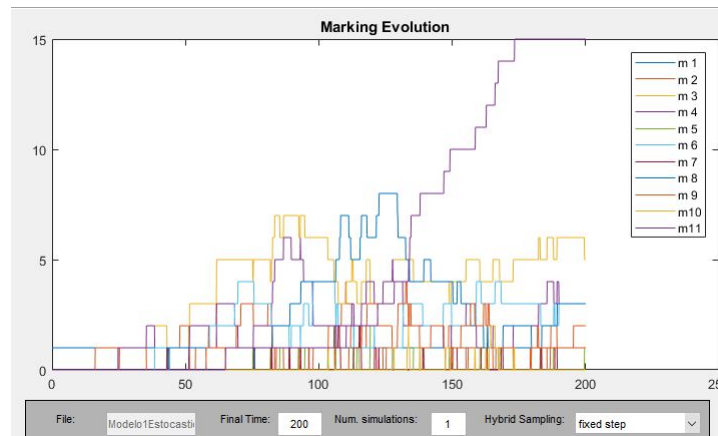


Figura 42 – Modelo 1 - Simulação Discreta Estocástica Única

Somente repetindo a simulação que valores médios das taxas de disparo das transições serão encontrados e que os gráficos de simulação vão se aproximar de funções contínuas no tempo. Dependendo da complexidade dos recursos simulados em relação ao número de atividades executadas, os tempos de simulação poderão crescer. Tanto o horizonte de simulação deverá aumentar quanto a quantidade de replicações para se atingir um tipo de gráfico estácionários (com poucas discontinuidades no formato das funções representadas).

Já no caso de um modelo de simulação contínuo que não depende de modelos de distribuições estocásticas e que considera diretamente valores contínuos, uma única replicação da simulação fornecerá o resultado correspondente ao regime estacionário médio do processo.

Esse raciocínio é melhor representado na Figura 43. O gráfico 43a, representa o modelo discreto estocástico obtido por 100 simulações, o gráfico 43b representa o modelo estocástico com 200 simulações, e o gráfico 43c representa o modelo contínuo com apenas 1 simulação.

Pela análise dos gráficos, é possível observar que o 43c é o que representa, em termos de regime estacionário (regime de equilíbrio que os processos atingem depois de um certo tempo de funcionamentos), a simulação com maior precisão.

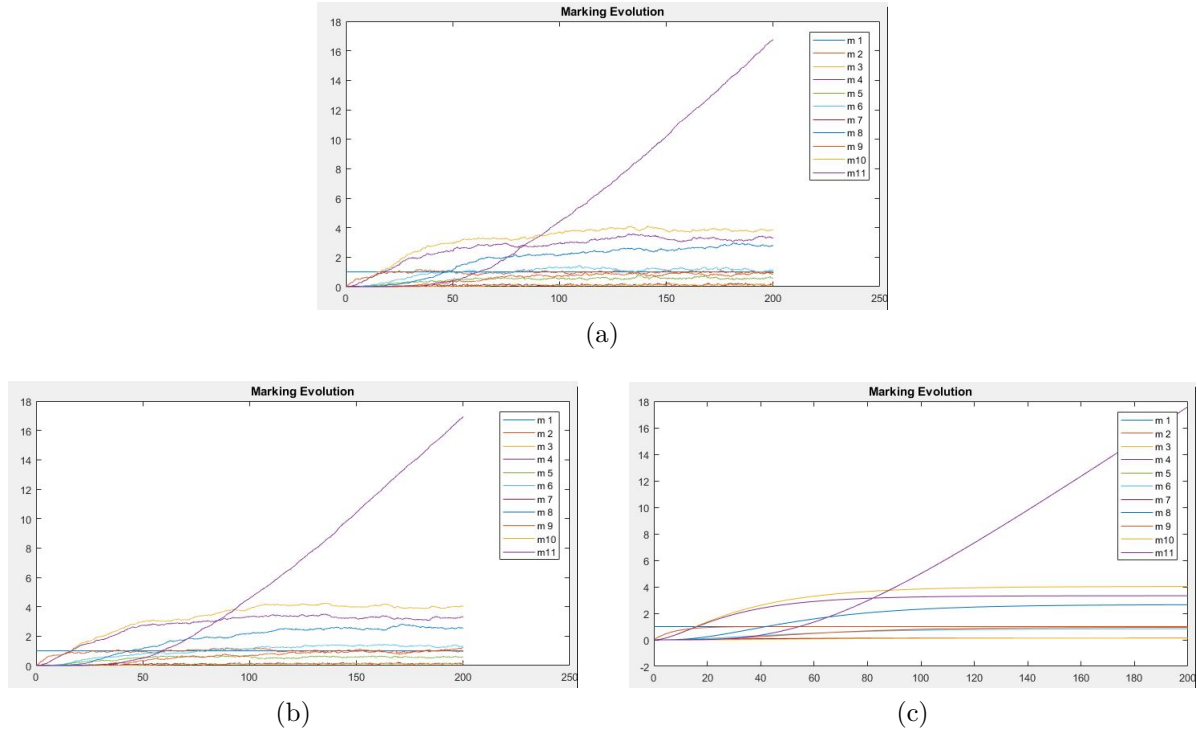


Figura 43 – Comparação do Modelo 1 Estocástico e Contínuo

4.3 Simulação Para Eliminar Possíveis Gargalos do Modelo 2 Utilizando Recursos Contínuos e Transições Contínuas

O *Modelo 2* é semelhante ao *Modelo 1*, porém com a adição de recursos às partes específicas da rede de Petri. Antes de começar a análise efetiva dos resultados dos modelos, foi feita uma simulação inicial com apenas 1 ficha em cada recurso (valor real igual a 1 associado a cada recurso contínuo) para verificar se existe a ocorrência de algum gargalo na rede e, caso haja, qual é a melhor forma de resolvê-lo. O resultado dessa simulação inicial está representado na Figura 44. A ideia é encontrar um número de fichas para cada recurso que permita um equilíbrio no fluxo da rede. Neste contexto, cada ficha no recurso representa um funcionário.

Na Figura 44 é possível observar que além do lugar final, representado por m_{11} existem duas outras marcações que também estão crescendo rapidamente, que são m_3 e m_4 que representam os lugares P_3 e P_4 na Figura 34. Para solucionar estes gargalos, foi preciso alterar a quantidade de fichas em R_2 e chegou-se a uma quantidade de fichas solução do problema, por tentativa e erro, que foi de 4 fichas. Porém, ao resolver um gargalo, surgiu outro.

O gráfico apresentado na Figura 45 mostra o gargalo que surgiu a partir da modificação do número de fichas em R_2 . O problema agora está no crescimento acelerado do número de fichas em P_8 , representado no gráfico por m_8 . Utilizando o mesmo método de tentativa e erro para tentar eliminar o gargalo, conclui-se que se R_4 tiver 3 fichas, o

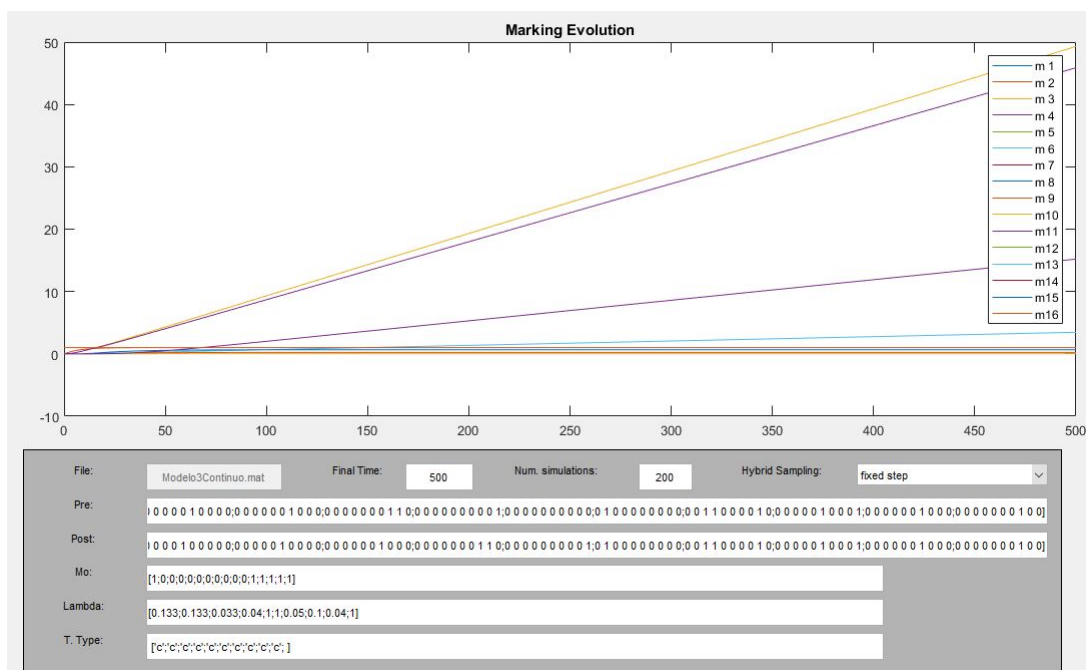


Figura 44 – Modelo 2 - Simulação Inicial - Parte 1

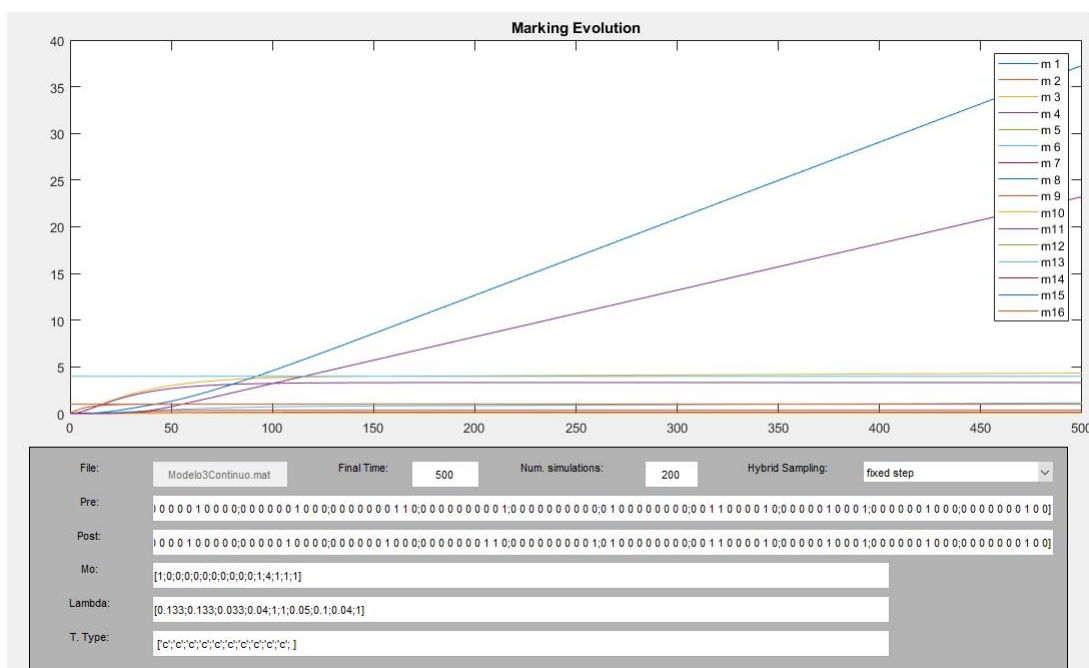


Figura 45 – Modelo 2 - Simulação Inicial - Parte 2

problema será resolvido e, felizmente, essa modificação não produz nenhum novo gargalo.

O número de fichas ideal para cada recursos poderia ter sido encontrado também a partir da análise do modelo utilizado como base para a simulação, apresentado na Figura 34. Cada recurso está ligado uma ou mais transições, sendo assim, é possível observar que para que haja um fluxo equilibrado de fichas na rede, os recursos ligados às transições com maior tempo associado devem possuir mais fichas. Assim, os recursos R_2 e R_4 devem

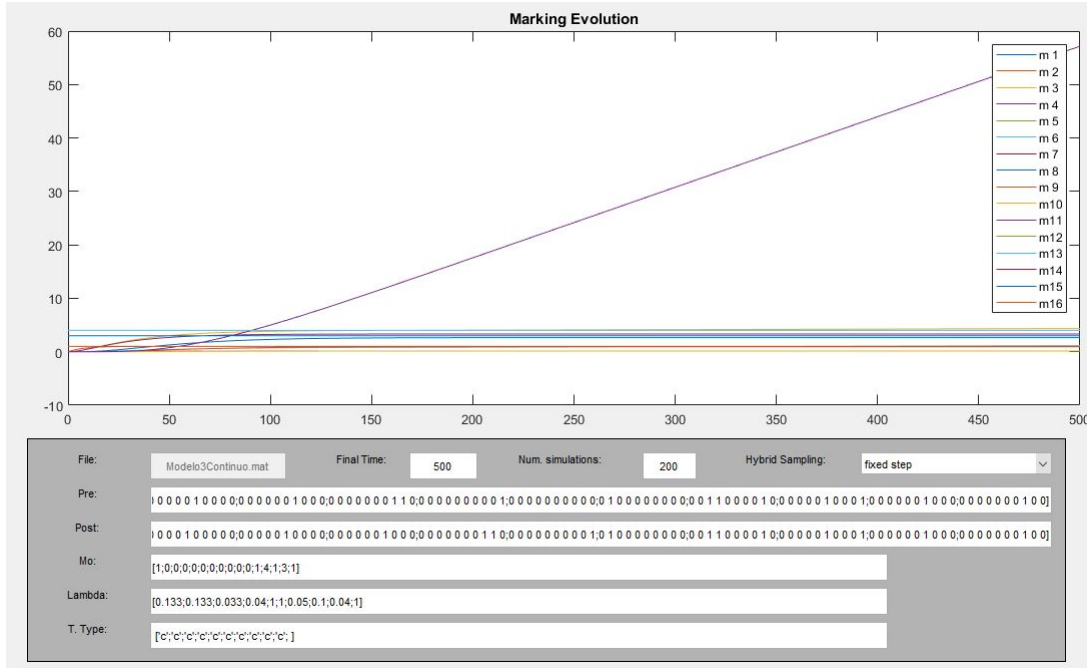


Figura 46 – Modelo 2 - Simulação Inicial - Parte 3

contar com mais de uma ficha, para garantir o equilíbrio no fluxo da rede.

A Figura 46 mostra o gráfico resultante da simulação do *Modelo 2* contínuo com a quantidade de recursos modificada. Esse número de fichas será utilizado para as simulações efetivas. O tipo contínuo foi utilizado para essa simulação inicial devido ao menor tempo de processamento.

4.4 Simulação: Modelo 2 com Recursos Discretos e Transições Discretas Temporizadas Estocásticas

A simulação agora é referente à rede com recursos discretos, operando de forma temporizada estocástica. Após preencher as matrizes e vetores de entrada de acordo com o modelo proposto, a simulação é iniciada e gera como resultado o gráfico apresentado na Figura 47.

O gráfico apresenta apenas uma linha que cresce de forma exponencial, que é a marcação m_{11} referente ao lugar final da rede, representado por P_{11} na Figura 34. É importante ressaltar também que os recursos representados por R_1 , R_2 , R_3 , R_4 e R_5 em 34, em 47 correspondem às marcações de m_{12} a m_{16} .

Existe um crescimento pouco significativo das linha que representam m_2 e m_3 , mas isso não interfere nos resultados visto que entre 200 e 500 segundos, elas aumentam em apenas aproximadamente 5 fichas. Além disso, observa-se tanto pelo gráfico quanto pelas informações adicionais fornecidas pelo *MATLAB*, apresentadas na Figura 48, as

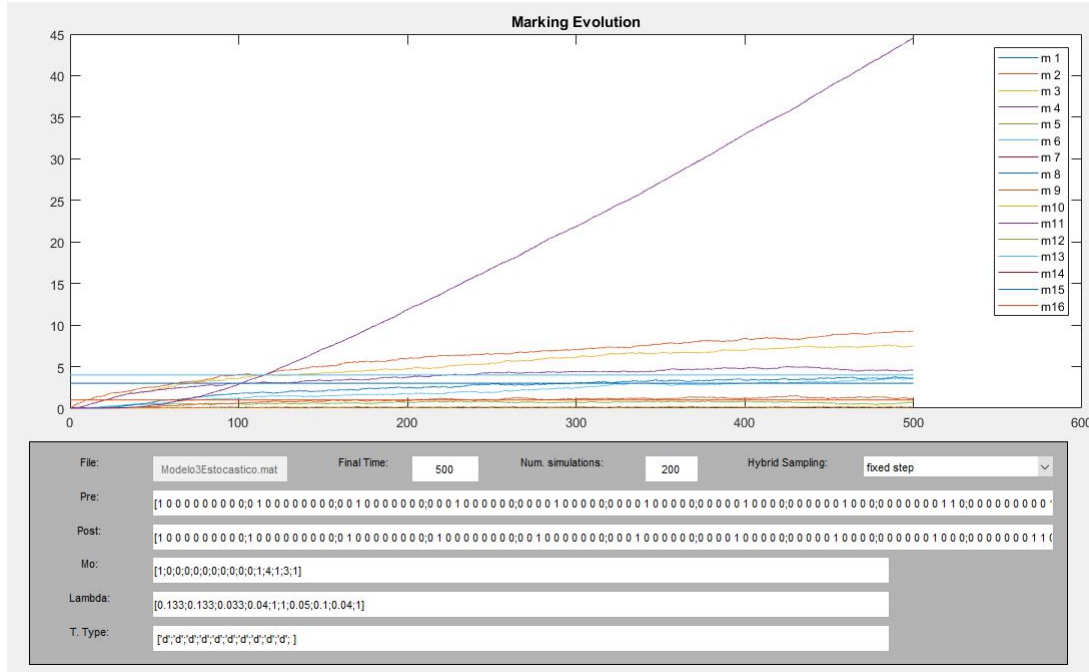


Figura 47 – Modelo 2 - Estocástico

```
Elapsed time is 613.703373 seconds.
Final marking: [1 9.275 7.49 4.635 0.705 3.56 0.125 3.615 1.12 0.12 44.555 1 4 1 3 1]
Final flow: [0.135 0.11 0.09 0.07 0.06 0.09 0.08 0.075 0.045 0.13]
```

Figura 48 – Modelo 2 - Estocástico - Informações adicionais

últimas marcações (de m_{12} até m_{16}), que são referentes aos recursos, possuem valor constante, e não sofrem nenhuma alteração durante toda a simulação. Além disso, a Figura 48 apresenta informações como o tempo gasto para realizar a simulação, que foi de aproximadamente 613.7 segundos, e a quantidade de fichas no lugar final (m_{11}) que é de aproximadamente 45 fichas. A partir da informação do número de fichas no lugar final e do tempo de simulação, que é de 500 segundos, é possível concluir que o tempo para tratar cada caso é em média 11.1 segundos.

Se comparados o tempo gasto para a realização da simulação estocástica do *Modelo 1* e *Modelo 2*, é possível concluir que o tempo de simulação gasto no *Modelo 2* é expressivamente maior. Isso se deve a alguns fatores como o aumento na complexidade da rede.

4.5 Simulação: Modelo 2 com Recursos Contínuos e Transições Contínuas Temporizadas

A simulação agora é referente à rede com recursos operando de forma contínua, com transições contínuas temporizadas. Após preencher as matrizes e vetores de entrada

de acordo com o modelo proposto, a simulação é iniciada e gera como resultado o gráfico apresentado na Figura 49. A Figura 50 apresenta informações como o tempo gasto para realizar a simulação, que foi de aproximadamente 1.16 segundos, e a quantidade de fichas no lugar final (m_{11}) que é de aproximadamente 57 fichas. Além disso, é possível concluir que assim como na simulação do modelo contínuo sem recursos, cada caso leva em média 8.7 segundos para ser tratado.

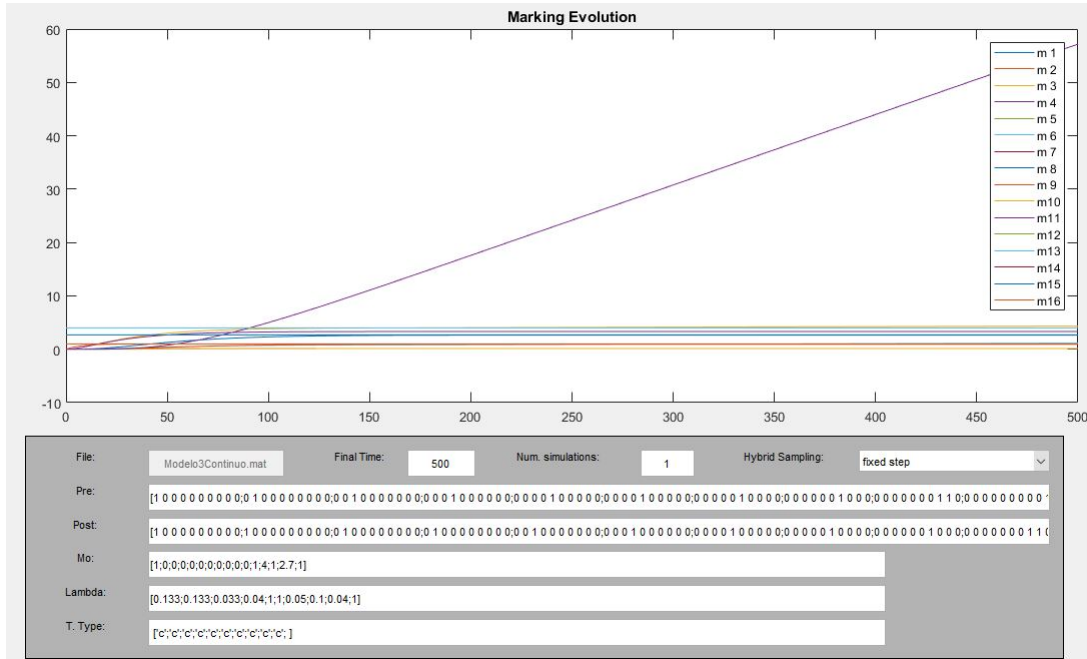


Figura 49 – Modelo 2 - Contínuo

```
Elapsed time is 1.159446 seconds.
Final marking: [1 1 4.34317 3.325 0.132 1.15017 0.131985 2.64002 0.942856 0.132039 57.1779 1 4 1 2.7 1]
Final flow: [0.133 0.133 0.132 0.133 0.132 0.132014 0.131999 0.0942858 0.0377143 0.131964]
```

Figura 50 – Modelo 2 - Contínuo - Informações adicionais

Observa-se, assim como na simulação estocástica, que os recursos (representados por m_{12} até m_{16}) possuem valores constantes do início ao fim e que a única marcação que possui crescimento exponencial é m_{11} , que se refere ao lugar final da rede. Nessa simulação, diferente do caso estocástico, foi utilizado para o recurso R_4 uma quantidade de fichas representada por um número real (2.7). Na prática, isso representaria, por exemplo, que um mesmo funcionário poderia dividir seu tempo entre a realização de duas tarefas, utilizando 70% no recurso R_2 e 30% em outro recurso.

Tal distribuição das atividades de um funcionário humano não poderia ser representada com recursos discretos. E no caso de trabalhos administrativos, é bem comum dividir o tempo de atividade de um funcionário entre diversas atividades, de modo simultâneo, o que torna este modelo mais próximo da realidade em termos de recursos usados em sistemas administrativos.

A partir da comparação entre as simulações estocástica e contínua do *Modelo 2* é possível observar que o tempo de simulação gasto no caso estocástico é significativamente maior que no caso contínuo, mesmo ambos apresentando o mesmo tempo final e a mesma complexidade da rede. O que muda é o número de simulações realizadas. Comparando com relação à simulação contínua do *Modelo 1*, há uma queda mínima no tempo de simulação do *Modelo 2*.

De forma geral, é possível observar que a complexidade da rede de Petri apresentada no segundo modelo é maior, porém com a vantagem de representar de forma mais precisa o processo de negócio. Como o tempo de simulação para ambos os modelos é igual, é possível concluir que, no caso das redes de Petri discretas estocásticas, a adição de recursos gera um aumento no tempo gasto com a simulação porém o número de casos tratados no mesmo período de tempo diminui, ou seja, a adição de recursos faz com que o processamento de cada caso seja mais lento. No caso das redes de Petri contínuas, a adição de recursos à rede provoca uma diminuição pouco significativa no tempo gasto com a simulação, porém o número de casos tratados permanece praticamente o mesmo, ou seja, a adição de recursos não contribui para que o processamento de um caso seja mais rápido, porém também não o torna mais lento. Esses resultados estão representados na Tabela 1.

Tabela 1 – Comparação dos Resultados das Simulações

	Modelo 1 Estocástico	Modelo 1 Contínuo	Modelo 2 Estocástico	Modelo 2 Contínuo
Tempo final da simulação (s)	500	500	500	500
Tempo gasto (s)	365.29	1.26	613.70	1.15
Número de casos tratados	57.24	57.46	44.55	57.17
Media de casos/ s	8.73	8.70	11.22	8.74

5 Considerações Finais

Este trabalho formaliza um processo de modelagem de processos utilizando uma ferramenta chamada *SimHPN* para análise e simulação dos modelos propostos, na qual valida-se o tempo de execução dos modelos e a quantidade de casos tratados em cada simulação. O *software SimHPN* possui inúmeras ferramentas que possibilitam a modelagem e simulação desses processos utilizando redes de Petri discretas estocásticas e contínuas. É possível simular um mesmo processo de *workflow* utilizando diferentes abordagens, que no caso deste trabalho foram as redes de Petri discretas estocásticas e contínuas, ambas combinadas com modelos com e sem recursos. O *SimHPN* é uma ferramenta projetada para simulação de redes de Petri híbridas, que não foram o alvo deste trabalho, porém foi o *software* que melhor supriu as condições técnicas necessárias para o desenvolvimento do projeto proposto. Além disso, poderá ser considerado numa abordagem futura o uso de redes de Petri híbridas.

Entre as dificuldades enfrentadas é possível destacar a busca por uma ferramenta que suprisse as necessidades impostas pelo projeto proposto, já que as redes de Petri mais comumente utilizadas são as discretas, foi preciso encontrar uma ferramenta que fosse além e apresentasse também a possibilidade de simulação de redes de Petri contínuas. Outro desafio foi entender como modelar um processo de negócios no *SimHPN* e como extrair os dados necessários para a análise proposta.

Foram apresentados dois modelos diferentes de um mesmo processo de *workflow*, modelados por redes de Petri, com o intuito de analisar principalmente o tempo gasto em cada simulação e a quantidade de casos tratados em determinado período de tempo. Esses modelos foram analisados tanto para as redes de Petri contínuas quanto para as redes de Petri discretas estocásticas temporizadas. Em particular, o modelo sem recursos mostrou o tempo médio mínimo de tratamento de um caso e o modelo com recurso forneceu uma proposta de dimensionamento aceitável (sem situações de gargalo) que forneceu uma estimativa realista do tempo médio de tratamento de um caso no processo estudado.

Os resultados mostraram que para o processo de tratamento de reclamações (AALST; HEE, 2004), é mais vantajosa a utilização de redes de Petri contínuas, se comparadas com as discretas estocásticas. Isso se deve ao tempo gasto para realizar cada simulação e também ao número de casos processados em determinado intervalo de tempo. Além disso, no caso discreto estocástico são necessárias inúmeras simulações para que se atinja o regime estacionário do processo, enquanto que no caso contínuo é necessária apenas uma simulação.

A simulação de processos de negócio pode ser um processo trabalhoso, porém

benéfico para as empresas a longo prazo, pois facilita o planejamento das atividades a serem desempenhadas, a alocação de recursos para cada tarefa e o cálculo do tempo médio gasto para se concluir determinada atividade. As redes de Petri contínuas se mostraram a forma mais rápida e precisa de simular processos, se comparadas com as redes discretas.

Com isso, uma abordagem futura seria a modelagem de processos de negócios reais (com grande quantidade de atividades e recursos a serem considerados) para análise de seu desempenho em redes de Petri contínuas. Uma outra proposta seria o estudo de modelos baseados em redes de Petri híbridas (que misturam transições e lugares discretos e contínuos) para verificar quais as vantagens de seu uso em modelagem de processos de *workflow*. Tal modelo poderia mostrar situações de falhas administrativas (um grupo de funcionários que interrompem de modo abrupto as suas atividades numa situação de greve por exemplo).

Referências

- AALST, W.; HEE, K. Workflow management: Models, methods, and systems. The MIT Press Cambridge, Massachusetts. London, England, 2004. Citado 4 vezes nas páginas 5, 24, 37 e 54.
- AALST, W. M. P. van der. The application of petri nets to workflow management. *In: The Journal of Circuits, Systems and Computers*, p. 21–66, 1998. Citado na página 24.
- AALST, W. van der; HEE, K. van. *Workflow management: Models, methods, and systems*. Eindhoven: BETA Working Paper Series, 2001. Citado 3 vezes nas páginas 11, 35 e 36.
- AALST W. VAN DER; STAHL, C. Modeling business processes. *MIT Press*, p. 20–25, 2011. Citado 3 vezes nas páginas 5, 25 e 26.
- ALLA, H.; DAVID, R. Continuous and hybrid petri nets. *Journal of Circuits, Systems, and Computers*, World Scientific, v. 8, n. 01, p. 159–188, 1998. Citado 2 vezes nas páginas 14 e 31.
- BAIL, J. Hybrid petri nets. In: *Proc. European Control Conference*. [S.l.: s.n.], 1991. v. 1472. Citado 2 vezes nas páginas 11 e 35.
- CARDOSO, J.; VALETTE, R. *Redes de Petri*. Florianópolis: Editora da UFSC, 1997. Citado 3 vezes nas páginas 11, 14 e 36.
- DAVID, R. Continuous petri nets. In: *8th European Workshop on Application and Theory of Petri nets*. [S.l.: s.n.], 1987. v. 275. Citado 3 vezes nas páginas 11, 20 e 35.
- DAVID, R.; ALLA, H. *Petri Nets and Grafcet: Tools for Modelling Discrete Event Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1992. ISBN 013327537X. Citado 3 vezes nas páginas 5, 16 e 17.
- DAVID, R.; ALLA, H. On hybrid petri nets. *Discrete Event Dynamic Systems*, Springer, v. 11, n. 1-2, p. 9–40, 2001. Citado na página 35.
- DAVID, R.; ALLA, H. *Discrete, Continuous, and Hybrid Petri Nets*. 2. ed. França: Springer, 2010. Citado 10 vezes nas páginas 5, 12, 14, 19, 20, 21, 22, 23, 35 e 36.
- FLORIN, G.; FRAIZE, C.; NATKIN, S. Stochastic petri nets: Properties, applications and tools. *Microelectronics Reliability*, Elsevier, v. 31, n. 4, p. 669–697, 1991. Citado na página 16.
- HERAJY, M.; HEINER, M. *Snoopy computational steering framework-user manual version 1.0*. [S.l.]: BTU Cottbus-Senftenberg, 2013. Citado na página 36.
- JÚLVEZ, J.; MAHULEA, C. Simhpn: a matlab toolbox for continuous petri nets. *IFAC Proceedings Volumes*, Elsevier, v. 43, n. 12, p. 21–26, 2010. Citado na página 31.
- JÚLVEZ, J.; MAHULEA, C. Simhpn: A matlab toolbox for hybrid petri nets. *User Manual*, Citeseer, 2012. Citado na página 28.

- JÚLVEZ, J.; MAHULEA, C.; VÁZQUEZ, C.-R. Simhpn: A matlab toolbox for simulation, analysis and design with hybrid petri nets. *Nonlinear Analysis: Hybrid Systems*, v. 6, n. 2, p. 806 – 817, 2012. ISSN 1751-570X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1751570X11000586>. Citado 2 vezes nas páginas 26 e 36.
- JÚLVEZ, J.; MATCOVSCHI, M. H.; PASTRAVANU, O. Matlab tools for the analysis of petri net models. In: IEEE. *Emerging Technology and Factory Automation (ETFA)*, 2014 IEEE. [S.l.], 2014. p. 1–12. Citado na página 35.
- MURATA, T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, IEEE, v. 77, n. 4, p. 541–580, 1989. Citado 2 vezes nas páginas 11 e 35.
- PETRI, C. *Kommunikation mit Automaten. Schriften des IIM Nr. 3, Institut für Instrumentelle Mathematik, Bonn, West Germany. See, also, Communication with Automata (in English). Gri ss Air Force Base.* [S.l.], 1962. Citado 2 vezes nas páginas 14 e 34.
- RECALDE, L.; HADDAD, S.; SILVA, M. Continuous petri nets: Expressive power and decidability issues. *Automated Technology for Verification and Analysis*, Springer, p. 362–377, 2007. Citado na página 22.
- RECALDE, L.; SILVA, M. Pn fluidification revisited: Semantics and steady state. *J. Zaytoon S. Engell, Automation of Mixed Processes: Hybrid Dynamics Systems*, p. 279–286, 2000. Citado na página 32.
- REZENDE, L. P. d. et al. Workflow net possibilística para problemas de não conformidade em processos de negócios. Universidade Federal de Uberlândia, 2013. Citado na página 24.