

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**REDUÇÃO AUTOMATIZADA DE REDES DE PETRI
BASEADA EM ALGORITMOS GENÉTICOS**

ANDRÉ BEVILAQUA

ORIENTADOR: PROF. DR. EDILSON REIS RODRIGUES KATO

São Carlos - SP
Maio/2014

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**REDUÇÃO AUTOMATIZADA DE REDES DE
PETRI BASEADA EM ALGORITMOS GENÉTICOS**

ANDRÉ BEVILAQUA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Inteligência Artificial.

Orientador: Prof. Dr. Edilson Reis Rodrigues Kato.

São Carlos - SP
Maio/2014

**Ficha catalográfica elaborada pelo DePT da
Biblioteca Comunitária da UFSCar**

B571ra

Bevilaqua, André.

Redução automatizada de redes de petri baseada em algoritmos genéticos / André Bevilaqua. -- São Carlos : UFSCar, 2014.

120 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2014.

1. Redes de computação. 2. Sistemas de manufatura. 3. Computação evolutiva. I. Título.

CDD: 004.6 (20^a)

Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

**“Redução Automatizada de Redes de Petri
Baseada em Algoritmos Genéticos”**

André Bevilaqua

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação

Membros da Banca:



Prof. Dr. Edilson Reis Rodrigues Kato
(Orientador - DC/UFSCar)



Prof. Dr. Paulo Rogério Politano
(DC/UFSCar)



Prof. Dr. Roberto Hideaki Tsunaki
(EESC/USP)



Prof. Dr. Ricardo Yassushi Inamasu
(EMBRAPA/CNPDIA)

São Carlos
Junho/2014

Dedico este trabalho à meu amigo Aurélio Borges Carmo, prematuramente retirado de nosso convívio após um acidente automobilístico. Descanse em paz grande amigo.

AGRADECIMENTOS

Primeiramente a Deus por ter permitido a conclusão dessa dissertação e também de todas as outras atividades realizadas durante os dois últimos anos.

Ao meu orientador Edilson Reis Rodrigues Kato pela paciência, dedicação e flexibilidade com as quais me instruiu e guiou por todas as fases de desenvolvimento da pesquisa e realização de todas as atividades requisitadas.

A minha mãe Neuza Maria Bevilaqua, meu pai Olisses Bevilaqua, meu irmão Diego Bevilaqua, minha tia e segunda mãe Nelcita e minha namorada e companheira Samila Carolina Costa por terem sempre acreditado na minha capacidade e me apoiado em todos os momentos.

Ao meu irmão Diego Bevilaqua um agradecimento especial por ter sido de fundamental importância com seus conhecimentos da linguagem de programação Matlab e que em breve estará também seguindo o caminho da pós-graduação.

Aos meus amigos que de diversas formas também participaram desse processo e que há vários anos fazem parte das conquistas e derrotas que fomentaram minha evolução acadêmica e pessoal.

Agradecimentos também para todos os professores do departamento de computação e demais pessoas que de alguma maneira contribuíram no decorrer desta pesquisa.

"O mundo está cheio de boas citações, só falta aplicá-las."

Blaise Pascal

RESUMO

Rede de Petri (RP) é uma ferramenta de modelagem com rigor matemático usada para especificar e formalizar sistemas e suas características. Modelos de RP podem ser avaliados quanto à seu comportamento dinâmico e suas propriedades estruturais. Diversos tipos de modelos podem ser especificados usando RP's, por exemplo, modelos de Sistemas de Manufatura. De maneira geral, modelos de sistemas complexos sofrem do problema da grande profusão de elementos. Esse problema é um limitante para diversas metodologias de verificação de propriedades e simulação de comportamento dinâmico dos modelos. Frequentemente metodologias baseadas em regras de redução são propostas para superar o problema da grande profusão de elementos. O presente trabalho apresenta uma ferramenta para automatizar a aplicação de regras de redução em modelos de RP. Um conjunto de regras foi selecionado para montar uma base de regras de redução e uma heurística baseada em algoritmos genéticos foi criada para aplicar reduções em modelos de RP Lugar-Transição. Os resultados obtidos mostraram que a ferramenta proposta, apresentada como uma *toolbox* do Matlab, pode ser utilizada para auxiliar no processo de análise, simulação e verificação de propriedades de sistemas modelados por meio de RP's.

Palavras-chave: Redes de Petri, Algoritmo Genético, Aplicações de Manufatura, Redução de Redes de Petri.

ABSTRACT

Petri Net (PN) is a graphical mathematical tool used to create models that provide a formal specification of systems, creating the possibility of running behavior analysis and property verification. PN models can be used to model several system types, for example, Manufacturing Systems. In general, using PN to model systems with too many characteristics cause the so-called state-space explosion problem, in other words, a great element profusion in the model. In order to mitigate the referred problem researchers developed reduction rules to synthesize models. Reduced models maintain original system properties and can be easily analyzed and simulated. A method to automate reduction rules application is proposed on the present research. A Genetic Algorithm heuristic is responsible for applying six reductions rules on Place-Transition Petri Nets given as input. The proposed algorithm, presented as a Matlab toolbox, may contribute when simulating and running property verification methodologies on Petri Net models.

Keywords: Petri Net, Genetic Algorithm, Manufacturing System, Petri Net Reduction.

LISTA DE FIGURAS

Figura 1. Modelo de Rede de Petri de um Sistema de Processamento Paralelo (SHEN et. al. 2013).	27
Figura 2. Representação Matricial da RP da Figura 2 (SHEN et. al. 2013).....	28
Figura 3. Possíveis conexões entre lugares e transições (SHEN et. al. 2013).....	29
Figura 4. Regras de Redução: FSP (a), FST (b), FPT (c), FPP (d), EST (e), ESP (f) (MURATA, 1989).....	38
Figura 5. Regra de Fusão Serial (SLOAN, 1996).....	44
Figura 6. Ilustração da necessidade da quarta pré-condição para redes temporizadas (SLOAN, 1996).....	45
Figura 7. Regras de Redução para Redes de Petri Temporizadas (XIA, 2010).....	53
Figura 8. Aplicação da Regra Número 3 em um Modelo de Gerência de Banco de Dados (XIA; LIU, 2010).	54
Figura 9. Características Desejáveis e Indesejáveis em Modelos de <i>WorkFlow</i> (QING-XIU et. al. 2010).	55
Figura 10. Redução de Estruturas Interativas (partindo de transições ou de lugares) (QING-XIU et. al. 2010).....	56
Figura 11. Transformação de Modelo Não-Livre para Modelo Livre (QING-XIU et. al. 2010).....	57
Figura 12. À esquerda um modelo de RP com três pontos de redução identificados e à direita a RP reduzida (SHEN et. al. 2013).....	62
Figura 13. Representação do padrão que define a regra Fusão de Lugares Seriais (FSP) gráfica e matricialmente (representação usada pelo algoritmo) (MURATA, 1989).....	72
Figura 14. Representação do padrão que define a regra Fusão de Transições Seriais (FST) gráfica e matricialmente (representação usada pelo algoritmo) (MURATA, 1989).....	72
Figura 15. Representação do padrão que define a regra Fusão de Lugares Paralelos (FPP) gráfica e matricialmente (representação usada pelo algoritmo) (MURATA, 1989).....	73
Figura 16. Representação do padrão que define a regra Fusão de Transições Paralelas (FPT) gráfica e matricialmente (representação usada pelo algoritmo) (MURATA, 1989).....	73

Figura 17. Representação do padrão que define a regra Eliminação de Lugares com Ciclo (ESP) gráfica e matricialmente (representação usada pelo algoritmo) (MURATA, 1989).....	74
Figura 18. Representação do padrão que define a regra Eliminação de Transições com Ciclo (EST) gráfica e matricialmente (representação usada pelo algoritmo) (MURATA, 1989).....	74
Figura 19. Modelo de Rede de Petri Usado para Demonstração de Reduções (DESROCHERS; AL-JAAR, 1995).....	75
Figura 20. Modelo de Rede de Petri em Representação Matricial Usado para Demonstração de Reduções (DESROCHERS; AL-JAAR, 1995).....	76
Figura 21. Identificados na Cor Vermelha os Padrões da Rede de Petri que Possibilitam Aplicação de Regras de Redução e à Direita a Representação da Rede já Reduzida.....	78
Figura 22. Identificados na Cor Vermelha os Padrões da Rede de Petri que Possibilitam Aplicação de Regras de Redução e na Parte Inferior a Matriz da Rede Reduzida.....	79
Figura 23. Identificados na Cor Vermelha os Padrões da Rede de Petri que Possibilitam Aplicação de Regras de Redução e à Direita a Representação da Rede já Reduzida.....	80
Figura 24. Identificados na Cor Vermelha os Padrões da Rede de Petri que Possibilitam Aplicação de Regras de Redução e na Parte Inferior a Matriz da Rede Reduzida.....	81
Figura 25. Identificados na Cor Vermelha os Padrões da Rede de Petri que Possibilitam Aplicação de Regras de Redução e à Direita a Representação da Rede Totalmente Reduzida.....	81
Figura 26. Identificados na Cor Vermelha os Padrões da Rede de Petri que Possibilitam Aplicação de Regras de Redução e na Parte Inferior a Matriz da Rede Totalmente Reduzida.....	82
Figura 27. Esboço do fluxo seguido pelo Algoritmo Genético em busca da redução para uma RP.	83
Figura 28. Exemplo de RP Visualizada na Ferramenta PIPE.....	90
Figura 29. Matriz de Incidência Gerada pela Ferramenta PIPE para a RP Mostrada na Figura 28.	91
Figura 30. Modelo de Controlador Usado na Montagem de Motor (DESROCHERS; AL-JAAR, 1995).	94
Figura 31. Matriz de Incidência do Controlador Usado na Montagem de Motor.....	95
Figura 32. Matriz de Incidência do Melhor Indivíduo e Informações Gerais do Primeiro Teste.	96

Figura 33. Modelo de Sistema Flexível de Manufatura (DESROCHERS; AL-JAAR, 1995).	98
Figura 34. Matriz de Incidência para Modelo de Sistema Flexível de Manufatura. ...	99
Figura 35. Matriz de Incidência do Melhor Indivíduo e Informações Gerais do Segundo Teste.	99
Figura 36. Modelo de Sistema Flexível de Manufatura (DESROCHERS; AL-JAAR, 1995).	101
Figura 37. Matriz de Incidência para Modelo de Sistema de Manufatura.	102
Figura 38. Matriz de Incidência do Melhor Indivíduo e Informações Gerais do Terceiro Teste.	102
Figura 39. Modelo de CAVE de Realidade Virtual (CRUZ-NEIRA et al., 1993)	105
Figura 40. Matriz de Incidência CAVE de Realidade Virtual.	106
Figura 41. Matriz de Incidência do Melhor Indivíduo e Informações Gerais do Quarto Teste.	107
Figura 42. Primeira Parte Modelo de FMS Usado no Quinto Teste	110
Figura 43. Segunda Parte do Modelo de FMS Usado no Quinto Teste	111
Figura 44. Matriz de Incidência do Modelo de FMS do Quinto Teste.....	112
Figura 45. Informações Gerais do Quarto Teste.	113
Figura 46. Matriz de Incidência do Melhor Indivíduo Obtido no Quinto Teste.	114
Figura 47. Carregando Scripts para Workspace do Matlab.....	127
Figura 48. Carregando Matriz de Incidência para Variável "teste_modelo_CAVE".	127
Figura 49. Execução de Teste na Ferramenta AGRRP.	129

LISTA DE TABELAS

Tabela 1. Interpretações Típicas de Lugares e Transições (MURATA, 1989).	297
Tabela 2. Comparativo de Pesquisas Baseadas Direta ou Indiretamente em Redução de Modelos de Redes de Petri.	63
Tabela 2. (continuação) Comparativo de Pesquisas Baseadas Direta ou Indiretamente em Redução de Modelos de Redes de Petri.	64
Tabela 2. (continuação) Comparativo de Pesquisas Baseadas Direta ou Indiretamente em Redução de Modelos de Redes de Petri.	65
Tabela 3. Comparação entre Metodologias Baseadas em Macro e Micro Reduções.	104

LISTA DE ABREVIATURAS E SIGLAS

RP – Redes de Petri

AG – Algoritmo Genético

AGRRP – Algoritmo Genético para Redução de Redes de Petri

SFM – Sistemas Flexíveis de Manufatura

FST – Fusão de Transições Seriais (do inglês, *Fusion of Series Transitions*)

FSP - Fusão de Lugares Seriais (do inglês, *Fusion of Series Places*)

FPP - Fusão de Lugares Paralelos (do inglês, *Fusion of Parallel Places*)

FPT - Fusão de Transições Paralelas (do inglês, *Fusion of Parallel Transitions*)

EST – Eliminação de Transições com Ciclo (do inglês, *Elimination of Self-Loop Transitions*)

ESP - Eliminação de Lugares com Ciclo (do inglês, *Elimination of Self-Loop Places*)

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO	16
1.1 Objetivos	18
1.2 Delimitação de Escopo.....	20
1.3 Metodologia.....	21
1.4 Organização do Trabalho	23
CAPÍTULO 2 - CONCEITOS FUNDAMENTAIS	24
2.1 Modelagem de Sistemas com Redes de Petri.....	25
2.2 Redes de Petri (RP)	25
2.3 Propriedades Comportamentais.....	30
2.4 Modelagem de Sistemas Usando Redes de Petri.....	34
2.5 Métodos de Análise de Redes de Petri	36
Existem três métodos principais para analisar RP's: árvore de alcançabilidade, abordagem de equações matriciais e técnicas de redução ou decomposição (MURATA, 1989).....	36
Outra alternativa para analisar grande sistemas modelados com RP's é reduzir os modelos mantendo suas propriedades comportamentais para que a análise seja feita com menos esforço (MURATA, 1989).....	37
2.6 Algoritmos Genéticos	39
CAPÍTULO 3 - REVISÃO BIBLIOGRÁFICA	42
3.1 Regras de Redução para Redes de Petri Temporizadas	43
3.2 Política de Prevenção de <i>Deadlocks</i> Otimizada Aplicada à Sistemas Flexíveis de Manufatura Baseada em Reduções de Redes de Petri	46
3.3 Novas e Eficientes Regras de Redução de Redes de Petri para Verificação de Programas Paralelos.....	48
3.4 Ferramenta de Software para Transformação de Redes de Petri Coloridas - TransCPN.....	50
3.5 Preservação de Propriedades em Reduções de Redes de Petri Temporizadas.	52
3.6 Método de Verificação Aprimorado Baseado em Redução de Redes de Petri para Modelos de <i>Workflow</i>	55

3.7 Decisão de Substituição de Serviços Web Baseada em Técnicas de Redução de Redes de Petri.....	58
3.8 Identificação Automática de Reduções para Simulação Discreta e Estocástica .	59
3.9 Uma Nova Abordagem para Redução de Sistemas em Redes de Petri Baseada em Teoria de Regiões	61
3.10 Considerações Finais.....	63
CAPÍTULO 4 - PROPOSTA DO TRABALHO	69
4.1 Visão Geral da Proposta	69
4.2 Padrões Para Representação das Regras de Redução.....	71
4.3 Exemplos de Utilização das Regras de Redução.....	74
4.4 Algoritmo Genético (AG) para redução de Modelos de Redes de Petri (RP)	82
4.4.1 Estrutura do Cromossomo.....	83
4.4.2 Base de Regras.....	84
4.4.3 População Inicial	85
4.4.4 Cruzamento.....	86
4.4.5 Mutação.....	87
4.4.6 Função de Avaliação (Fitness)	87
4.4.7 Critério de Parada	88
4.5 Linguagem de Programação e Ferramentas Auxiliares	88
4.5.1 MATLAB.....	88
4.5.2 PIPE	89
CAPÍTULO 5 - TESTES E RESULTADOS.....	92
5.1 Redução de Modelo de RP de um Controlador de Tempo Real Usado no Processo de Montagem de Motor	92
5.2 Redução de Modelo de RP de um Sistema Flexível de Manufatura	97
5.3 Redução de Modelo de RP Usado como Demonstração de Teoria de Síntese	100
5.4 Redução de Modelo de RP de uma CAVE de Realidade Virtual.....	103
5.5 Redução de um Modelo de RP de um Sistema Flexível de Manufatura	108
5.6 Contribuições e Limitação	115
5.6.1 Contribuições	115
5.6.2 Limitação	117
CAPÍTULO 6 - CONCLUSÃO.....	118

6.1 Trabalhos Futuros	120
REFERÊNCIAS.....	122
APÊNDICE A*	126

Capítulo 1

INTRODUÇÃO

Uma indústria pode ser definida como um grupo de empresas ou organizações envolvidas na produção e/ou distribuição de produtos e/ou serviços (GROOVER, 2007).

De maneira geral, esse tipo de indústria faz uso de diversos sistemas, dentre esses os sistemas de suporte à manufatura. Tais sistemas podem ter uma ou mais finalidades como, por exemplo, modelar equipamentos e processos, controlar e planejar a produção e satisfazer requisitos de qualidade dos produtos.

Para criar sistemas que auxiliem o processo de manufatura, é necessário modelar características relativas ao ambiente para que seja possível que o sistema atue em conformidade com o objetivo de produção da empresa. O problema de modelagem envolve eventos assíncronos e concorrentes que tipicamente ocorrem em sistemas dinâmicos onde ocorrem eventos discretos (DESROCHERS; AL-JAAR, 1995).

Os trabalhos propostos por Pan et. al. (2011); Hu et. al. (2012) abordam respectivamente sistemas flexíveis de manufatura (FMS, do inglês *Flexible Manufacturing Systems*) e sistemas automatizados de manufatura (AMS, do inglês *Automated Manufacturing Systems*). Esses são dois tipos de sistemas de manufatura existentes e tem como objetivo geral aumentar a flexibilidade e a resposta do ambiente de manufatura à demanda e variedade de produtos.

Ainda segundo Pan et. al. (2011); Hu et. al. (2012), a flexibilidade e automatização dos sistemas citados fazem com que seja necessário considerar diversas situações durante a modelagem dos mesmos, e citam que as Redes de Petri se encaixam perfeitamente para atender tais requisitos.

A leitura de bibliografias mais antigas e a comparação com outras mais recentes indica que as situações que devem ser consideradas na modelagem permanecem semelhantes. Alguns exemplos são: incerteza em relação à demanda de produtos, quantidade finita de máquinas e até mesmo possibilidade de falhas e necessidade de período para reparo de maquinário (DESROCHERS; AL-JAAR, 1995; PAN et. al., 2011; HU et al., 2012).

Portanto, entende-se que sistemas de manufatura envolvem uma grande gama de características que devem ser apuradas e consideradas, para posteriormente serem inseridas na modelagem dos ambientes de manufatura.

Existem diversas maneiras de se modelar sistemas de manufatura. Podem ser usadas cadeias de Markov, redes de Petri, autômatos finitos e até mesmo filas de prioridade simples.

O presente trabalho trata da utilização de redes de Petri (RP), propostas por C. A. Petri (1973), para modelagem de diversos tipos de sistema. Falando especificamente de Sistemas de Manufatura, pode-se considerar as RP na modelagem de sistemas de manufatura devido ao seu formalismo matemático que permite modelagem e análise de sistemas discretos e integrados. Esse formalismo também permite a criação de ferramentas de modelagem em RP com o poder de simulação e análises de desempenho.

A referida técnica mostra-se muito eficiente para modelagem, análise, simulação e controle de sistemas de manufatura (DESROCHERS; AL-JAAR, 1995; PAN et al., 2011; HU et al., 2012).

Alguns fatores principais justificam a utilização de RP para modelagem, são eles:

1. Capacidade de capturar relações de precedência e interações estruturais de eventos concorrentes, assíncronos e estocásticos.
2. Facilidade em modelar eficientemente conflitos e tamanhos de buffer.
3. Possibilidade de detecção de *deadlocks*.
4. Diversas extensões dessas redes como, por exemplo, redes de Petri coloridas, interpretadas, temporizadas, entre outras, permitem análises qualitativas e quantitativas de características do sistema modelado.

Porém, os modelos produzidos seguindo tal formalismo apresentam também uma severa limitação, a grande profusão de elementos quando se trata de modelar um sistema real de maneira mais detalhada (MURATA, 1989)

Os trabalhos propostos por Shen et. al. (2013) e indicam que a grande profusão de elementos ainda é um tema em destaque nas bibliografias atuais relacionadas às Redes de Petri.

De acordo com Shen et. al. (2013), diversas iniciativas para contornar a grande profusão de elementos em Redes de Petri vem sendo desenvolvidas desde a pesquisa de (MURATA, 1989). Muitas delas fazem uso de técnicas que quando aplicadas reduzem pequenos fragmentos da rede original e quando aplicadas em diversos pontos da rede, conseguem uma redução geral do grafo de alcançabilidade do modelo.

Para contornar a referida limitação, relativa à grande profusão de elementos nas RP's, diversas técnicas e regras de redução para tais modelos foram propostas. O objetivo é gerar redes que mantenham as propriedades da rede inicial, como vivacidade, ausência de *deadlocks*, entre outras, mas que também sejam menores do que a rede original. Isso faz com que a rede possa ser testada e verificada em relação às propriedades Murata (1989); Pan et. al. (2011); Hu et. al. (2012); Shen et. al. (2013).

As pesquisas desenvolvidas por: Mikolajczak & Singh (2009); Qing-xiu et. al. (2010); Xia (2010); Lu & Zhang (2010); Feng et. al. (2011); Pan et. al. (2011); Hu et. al. (2012); Wu et. al. (2012); Shen et. al. (2013) e outras exibidas nas seções seguintes, motivaram e justificam o desenvolvimento da presente pesquisa por mostrarem que ainda existem problemas relacionados a grande profusão de elementos em RP's, sendo assim, ainda é possível contribuir para a evolução desse tema de pesquisa. Explicações mais específicas de contribuições são exibidas nos próximos capítulos do trabalho.

1.1 Objetivos

Tendo em vista as dificuldades enfrentadas para mitigar o problema do número excessivo de elementos em modelos de RP, objetiva-se com a presente pesquisa criar um método de redução automatizado para Redes de Petri. O trabalho proposto por Mikolajczak & Singh (2009) segue essa linha de pesquisa, porém, o método proposto é semiautomático, fazendo com que o usuário escolha onde aplicar cada regra no

modelo, mas com a ajuda de um software que executa as reduções nos elementos requisitados pelo usuário.

O objetivo do trabalho é desenvolver um método de redução automática para modelo de Rede de Petri, permitindo que os mesmos possam ser validados e simulados com seu grafo de alcançabilidade reduzido. Ou seja, uma aplicação de software capaz de diminuir de maneira automatizada a quantidade de lugares e transições de um de RP, utilizando uma melhor sequência de técnicas de redução disponíveis e que mantenha as propriedades do modelo original, permitindo assim que a ferramenta possa auxiliar em pesquisas que busquem verificação de propriedades e simulação de modelos.

As regras escolhidas para a redução (mostradas nas seções 2 e 3) fazem com que propriedades de vivacidade, delimitação e segurança das redes sejam mantidas e por esse motivo, produzem modelos reduzidos que favorecem testes de propriedades comportamentais posteriores que sejam feitos usando árvores (ou grafos) de alcançabilidade, tendo em vista que para tais testes serem bem sucedidos é necessário que o modelo seja pequeno (para não sofrer com a explosão de estados) e seja delimitado (MURATA, 1989).

Para fins de clareza, é interessante dividir o objetivo em:

Objetivo Geral:

Criação de um sistema inteligente baseado em algoritmo genético para automatizar aplicação de regras de redução em modelos de Rede de Petri.

Objetivos Específicos:

Criação de base com regras de redução disponíveis para utilização do algoritmo genético.

Implementação do algoritmo genético para aplicação das regras.

Definição de Redes de Petri para validação.

Testes e validação da proposta.

Calculo de redução percentual de elementos.

Exibição de Lugares e Transições retirados dos modelos juntamente com as regras de redução aplicadas.

1.2 Delimitação de Escopo

O primeiro passo para o desenvolvimento do presente trabalho foi a pesquisa bibliográfica. Com base na pesquisa confirmou-se que as regras de redução são um tema ainda atual em diversos contextos de modelagem por meio das Redes de Petri e que estratégias para dar algum nível de automação ao processo de aplicação de regras de redução em modelos (normalmente com o objetivo de simplificar análise de propriedades) são exploradas, caso das pesquisas de Mikolajczak & Singh (2009) e Yang (2009).

Também foi possível delinear um caminho a ser seguido no desenvolvimento da proposta do presente trabalho, tendo como base os estudos já realizados em contextos semelhantes.

É importante frisar que existem várias maneiras de lidar com a grande profusão de elementos em modelos de Redes de Petri. Uma das maneiras é por meio da utilização de alguma das variações das Redes de Petri como, por exemplo, as Redes de Petri Coloridas. Frequentemente utiliza-se tais variações de formalismo de modelagem para contornar a grande profusão de elementos devido à sua capacidade de construção de modelos parametrizados e compactos. Redes de Petri Coloridas fazem parte da categoria de redes Alto-Nível (JENSEN, KRISTENSEN, 2009).

Ainda em se tratando de redes Alto-Nível pode-se citar as Redes de Petri Hierárquicas. Tais redes são consideradas importantes para reduzir a complexidade e simplificar a legibilidade de modelos que apresentem problemas relativos a grande profusão de elementos. O objetivo geral de tais redes é inserir sub-redes no modelo que podem se comportar como Lugares ou Transições, ou seja, o modelo passa a permitir a existência de dois novos tipos de elementos: os macro-lugares e as macro-transições. Tal estratégia é considerada importante para simplificar a análise e legibilidade de modelos com uma grande quantidade de elementos (BARROS, 1996).

Complementando a citação de possíveis estratégias usadas para contornar a grande profusão de elementos, cita-se a utilização de regras de redução. Diversas estratégias baseadas em reduções são abordadas de maneira mais detalhada no Capítulo 3 do presente documento.

Após esclarecer que não existe somente um tipo de estratégia para lidar com o problema previamente citado, é importante dizer que a presente pesquisa se limita a propor uma metodologia baseada em regras de redução.

O caminho trilhado se baseia em um questionamento simples: porque precisamos aplicar reduções em modelos de Redes de Petri?

Dentro do contexto de reduções, existem diversas iniciativas de aplicação, porém, a que se considera mais importante para o contexto do sistema proposto na presente pesquisa está relacionada com a verificação de propriedades e a simulação de modelos com uma grande profusão de elementos. Ou seja, considera-se necessário a criação da estratégia de automatização da aplicação de reduções para servir de ferramenta de auxílio nas análises de propriedades e simulação de modelos.

Um exemplo de pesquisa que considera importante a utilização de uma ferramenta para auxiliar na redução de modelos para posterior aplicação de metodologias de análise de propriedades é a proposta por Mikolajczak & Singh (2009). No referido caso, uma metodologia de análise e verificação de propriedades baseada em grafos de alcançabilidade é testada em modelos reduzidos e não-reduzidos, obtendo resultados que indicam que a aplicação da metodologia em modelos reduzidos é mais rápida e eficiente.

1.3 Metodologia

A seguir estão descritos os principais pontos norteadores e a sequência das principais etapas do desenvolvimento da proposta do trabalho, bem como as ferramentas de *software* e *hardware* utilizadas durante a pesquisa.

Com base nas observações realizadas durante a revisão bibliográfica considerou-se importante dar foco as Redes de Petri Generalizadas, tendo em vista que as outras categorias de RP's como, por exemplo, as redes coloridas ou temporizadas agregam características mais complexas na modelagem e também por levar em consideração que essas variações do formalismo das RP's são, isoladamente, estratégias para diminuir a grande profusão de elementos nos modelos.

A estratégia de criação do algoritmo genético teve como maior complexidade a criação de uma estrutura de indivíduo e a própria criação de um modelo de

cruzamento, tendo em vista que as formas de representação comuns em AG's não eram perfeitamente aplicáveis ao tipo de indivíduo proposto para a pesquisa. Os detalhes de criação do algoritmo genético são descritos de maneira aprofundada no Capítulo 4 do presente documento.

Após a definição da estrutura de indivíduo e do modelo de cruzamento, foi necessário definir qual a linguagem de programação seria utilizada na pesquisa. Devido à estrutura do indivíduo ser baseada em matrizes e vetores, escolheu-se a linguagem de programação Matlab, que possui diversas funções já programadas para operação e manipulação de matrizes e vetores.

A definição de qual *software* seria utilizado para criar os modelos de Rede de Petri que seriam utilizados durante os testes da ferramenta criada foi paralela à definição da linguagem de programação, pois buscava-se criar uma ferramenta que pudesse ser utilizada posteriormente em conjunto com o *software* de modelagem.

A escolha do *software* foi baseada na possibilidade de criação de modelos, simulação e análise de propriedades e definiu-se que a ferramenta *Platform Independent Petri Net Editor* (PIPE) era a mais apropriada, em primeiro lugar por atender às requisições listadas previamente e em segundo lugar por ser uma ferramenta desenvolvida na linguagem de programação Java. A linguagem de programação Java permite intrinsecamente a criação de *softwares* facilmente portáveis e também permite a importação de códigos em outras linguagens para funcionarem como funções auxiliares. As características citadas fazem com que a ferramenta desenvolvida em Matlab possa ser importada e utilizada em conjunto com as funções do *software* PIPE.

Os modelos escolhidos para os testes tem por objetivo demonstrar que a ferramenta é genérica, ou seja, é capaz de reduzir modelos independentemente do tipo de sistema, e que também permite ao usuário limitar os Lugares e Transições a serem removidos (dando a opção de impedir a retirada de certas transições ou lugares do modelo) e limitar também o percentual de redução (por meio do número de gerações). Tais fatores se mostram importantes pois cada redução de modelos pode ter um objetivo diferente, variando em relação ao nível de redução necessário, elementos que por ventura não deseja-se excluir do modelo e demais situações que ficam a critério do usuário e de seus objetivos ao aplicar uma metodologia de redução.

Para realizar os testes foi utilizado um *notebook* da marca Avell, modelo G1511 com processador Intel® Core™ i7-4700MQ CPU @ 2.4Ghz e memória de 8GB. A ferramenta foi desenvolvida e testada no Sistema Operacional Windows 8.1

1.4 Organização do Trabalho

O Capítulo 2 apresenta os conceitos básicos que se mostram necessários para compreender detalhes específicos relacionados à presente pesquisa.

O Capítulo 3 apresenta uma revisão teórica baseada em diversas bibliografias, exibindo como o objeto de estudo costuma ser abordado. São levantados também objetivos, motivações e resultados de trabalhos relacionados com a presente pesquisa para que seja possível estabelecer posteriormente um paralelo entre as bibliografias e a pesquisa conduzida no presente trabalho.

O capítulo 4 apresenta a proposta da presente pesquisa os passos seguidos para chegar até ao algoritmo final são expostos e exemplificados para que seja possível reproduzir o caminho seguido durante a pesquisa. Informações relativas as regras de redução usadas e alguns detalhes de implementação do Algoritmo Genético são também abordados.

No Capítulo 5 são apresentados os resultados obtidos com a abordagem proposta para o problema de grande profusão de elementos, sendo usados cinco modelos de RP para verificar o funcionamento da ferramenta desenvolvida.

O Capítulo 6 exhibe as conclusões dos autores após implementar e validar o método proposto e define alguns trabalhos futuros que podem ser desenvolvidos para estender a presente pesquisa obtendo mais contribuições para o objeto de estudo abordado.

Capítulo 2

CONCEITOS FUNDAMENTAIS

As indústrias fazem cada dia mais uso de sistemas de manufatura automatizados para melhorar seus processos de produção, e esses sistemas vem se tornando cada vez mais complexos devido às exigências tecnológicas e de competitividade de mercado.

Várias técnicas e formas de modelagem vêm sendo pesquisadas no intuito de se implementar tais sistemas, sendo que dentre essas técnicas as Redes de Petri tem papel destacado. Essas redes possuem uma base matemática e são reconhecidamente eficientes em lidar com problemas de caráter assíncrono, concorrente, sequencial e estocástico (C. A. PETRI, 1973; MURATA, 1989). Porém, um grande problema apontado na modelagem de problemas potencialmente reais, ou seja, sem abstrair o problema para um escopo reduzido, é a grande profusão de elementos presentes no modelo (DESROCHERS; AL-JAAR, 1995).

A implementação de controle a partir de modelos vem sendo crescentemente explorada. Esses modelos necessitam de avaliações relativas a propriedades e principalmente precisam ser simulados para que seja possível identificar características e possíveis falhas antes de se criar um controlador baseado no modelo.

Esse capítulo expõe os conceitos básicos necessários para o entendimento da proposta do presente trabalho, pautando-se em diversas pesquisas e exemplificando de maneira mais detalhada, os pontos motivadores da hipótese levantada, que culminou na especificação da proposta apresentada no capítulo 3 do trabalho.

2.1 Modelagem de Sistemas com Redes de Petri

Para ser bem sucedida no processo de produção, uma indústria deve planejar e controlar as atividades envolvidas, de forma que o produto final tenha passado por etapas estudadas e executadas a fim de tornar o processo o mais eficiente possível. Diversas medidas podem ser feitas para avaliar se um processo foi ou não eficiente, confrontando resultados com os objetivos da empresa, porém, o relevante na presente pesquisa é mostrar que o controle em um ambiente de manufatura qualquer, é extremamente importante e sempre levado em consideração pelas empresas. (GROOVER, 2007).

O controle em um ambiente de manufatura está relacionado à gestão das operações físicas da fábrica a fim de atingir os objetivos de produção (GROOVER, 2007).

Existem diversas maneiras de se modelar o controle em um ambiente de manufatura a fim de expressar previamente o comportamento e a disposição dos elementos dentro do ambiente de produção que se deseja estabelecer.

Desrochers (1995) cita a larga utilização das Redes de Petri para essa modelagem, existem ainda diversos outros trabalhos abordando a utilização desse tipo de modelagem nos dias de hoje, a apresentação destes trabalhos a título de contextualização com a proposta do trabalho será feita ainda durante o presente capítulo.

As RP's também são frequentemente usadas em outros contextos, por exemplo, para modelagem de sistemas paralelos, sistemas de Realidade Virtual, entre outros. O aprofundamento em abordagens usando RP para modelagem pode ser observado no Capítulo 3 do presente documento.

2.2 Redes de Petri (RP)

Apresentada por C. A. Petri (1973), as Rede de Petri são uma ferramenta matemática utilizada para modelagem e visualização aplicável em qualquer escopo de problema onde exista a noção de eventos.

O resultado da modelagem feita com tal formalismo é uma representação gráfica do sistema e de seu comportamento. Essa ferramenta é muito útil para a descrição e/ou estudo de sistemas que tenham caráter concorrente, distribuído, paralelo, assíncrono, não determinístico e estocástico (MURATA, 1989).

Uma Rede de Petri Lugar-Transição é composta de quatro partes básicas: um conjunto de lugares P (do inglês, *places*), um conjunto de transições T (do inglês, *transitions*), uma função de entrada I (do inglês, *input*) e uma função de saída O (do inglês, *output*). Podemos descrever uma RP, em termos matemáticos, como uma 5-tupla, $RP = (P, T, F, W, M_0)$. A letra P representa o conjunto de lugares, T representa o conjunto de transições, F representa o conjunto de arcos que ligam transições e lugares, W é a função peso (do inglês, *weight*) e M_0 representa as marcações iniciais da rede (MURATA, 1989; SHEN et al., 2013).

Podemos representar formalmente esses cinco componentes que definem uma Rede de Petri da seguinte forma:

- $P = \{p_1, p_2, \dots, p_m\}$
- $T = \{t_1, t_2, \dots, t_n\}$
- $F \subseteq (P \times T) \cup (T \times P)$
- $W : F \rightarrow 1, 2, 3, \dots$
- $M_0 : P \rightarrow 0, 1, 2, 3, \dots$

Quando temos uma Rede de Petri sem marcações iniciais especificadas podemos representá-la por $RP = (P, T, F, W)$. Quando temos uma RP e suas marcações iniciais denotamos como (PN, M_0) . Existem duas atividades principais que demonstram como é o comportamento de uma RP, ativar e disparar. Uma transição t é dita ativada se cada lugar que dá entrada nessa transição tem pelo menos $w(p, t)$ marcações, sendo que $w(p, t)$ representa o peso do arco que conecta p e t (MURATA, 1989; SHEN et al. 2013).

Por meio dos componentes citados, é possível representar um sistema de maneira estática e também podem ser simuladas atividades concorrentes e dinâmicas por meio do uso de marcações (MURATA, 1989).

A Figura 1 mostra um exemplo de um modelo de RP's constituído de lugares, transições, arcos dirigidos e marcações iniciais.

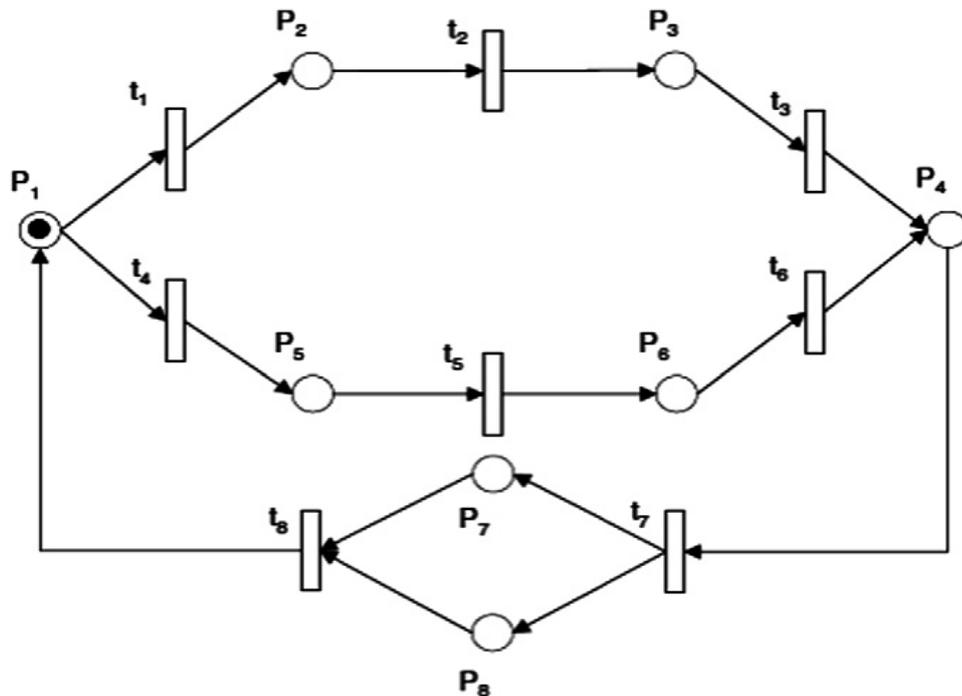


Figura 1. Modelo de Rede de Petri de um Sistema de Processamento Paralelo (SHEN et. al. 2013).

Devido a sua formalização matemática, as Redes de Petri permitem a utilização de equações e técnicas rígidas que possibilitam a caracterização do comportamento de sistemas. Isso pode ser feito de maneira simplificada ou aprofundada, ou seja, o modelo pode ser muito realista, muito abstraído ou ainda um meio termo (MURATA, 1989).

A utilização de abordagens de visualização diferentes para modelos de RP também é útil em determinadas situações. Utilizar o modelo gráfico para produzir um algoritmo, por exemplo, pode se tornar complexo por exigir um grande esforço computacional e, para atenuar esse efeito, existem outras formas de se representar a rede, como, por exemplo, em forma de matriz (MURATA, 1989).

Utilizar essa abordagem permite que os lugares sejam dispostos em forma de linhas, as transições como colunas e os números que preenchem a matriz representam os arcos que interligam lugares e transições. O estudo desenvolvido por Murata, (1989) trata da representação de modelos de RP por meio de matrizes como uma alternativa aos modelos que utilizam representações gráficas com círculos, setas e retângulos.

Com base nessa forma de representação pode se tornar mais simples implementar um algoritmo, fazendo com que as operações feitas no modelo sejam apenas operações executadas na matriz.

A matriz principal do modelo é chamada de matriz de incidência. A Figura 2 exibe a mesma RP da Figura 1, porém, utilizando uma representação matricial. Cada linha da matriz de incidência representa um lugar e cada coluna representa uma transição. Os pares ordenados (p, t) com o valor 1 representam que o lugar P é quem ativa a transição T. Os pares ordenados marcados o valor -1 representam que a transição T está enviando marcações ao lugar P.

$$A = \begin{matrix} & \begin{matrix} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 \end{matrix} \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \\ P_7 \\ P_8 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \end{matrix}$$

Figura 2. Representação Matricial da RP da Figura 2 (SHEN et. al. 2013).

Pode-se definir quatro tipos de conexões possíveis entre transições e lugares. A Figura 3 exibe um exemplo de cada tipo de conexão possível. Pode-se ter um lugar P1 como entrada (*input*) de uma transição T1 que direciona uma saída (*output*) para um lugar P2 (caso 1). Pode-se ter um lugar como entrada de uma transição que tem como saída mais de um lugar (caso 2). Pode-se ter mais de um lugar como entrada de uma transição com saída para apenas um P (caso 3). Por fim, podem existir muitos lugares como entrada de uma transição que tem como saída muitos lugares (caso 4).

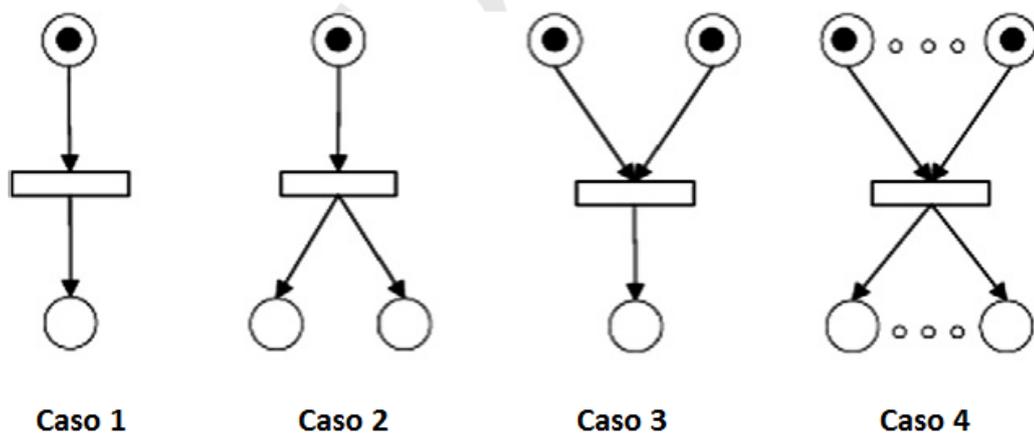


Figura 3. Possíveis conexões entre lugares e transições (SHEN et. al. 2013)

Por fim, existem também os casos em que lugares se ligam a mais de uma transição, dessa forma, mais de uma transição T pode ser ativada por apenas um lugar P, sendo que cada T ativada envia saídas somente para os lugares conectados a ela.

O trabalho desenvolvido por Murata (1989) exhibe algumas das interpretações típicas que lugares e transições podem receber. Essas interpretações são mostradas na Tabela 1.

Tabela 1. Interpretações Típicas de Lugares e Transições (MURATA, 1989).

Lugares de Entrada (<i>input places</i>)	Transições	Lugares de Saída (<i>output places</i>)
Pré-Condições	Evento	Pós-Condições
Entrada de Dados	Passo de Computação	Saída de Dados
Entrada de Sinais	Processador de Sinal	Saída de Sinais
Recursos Necessários	Tarefa/Atividade	Recursos Devolvidos
Condições	Cláusula Lógica	Conclusão
Armazenamento Temporário (<i>buffer</i>)	Processador	Armazenamento Temporário (<i>buffer</i>)

Caso seja necessária uma explanação mais ampla e com mais exemplos que ajudem a ilustrar utilizações e conceitos sobre o formalismo conhecido como Redes de Petri, além dos trabalhos já citados acima, pode-se indicar os seguintes: Peterson, (1981); Bernardinello & Fiorella, (1992). O estudo dos dois artigos ajuda a esclarecer

os conceitos envolvidos na definição do formalismo e também auxiliam o entendimento das diferentes classificações de RP's existentes.

Após a breve apresentação das propriedades e possíveis maneiras de visualização de RP's Lugar-Transição, abordar-se-á especificamente a modelagem do controle de Sistemas de Manufatura utilizando tal formalismo.

2.3 Propriedades Comportamentais

Murata (1989) cita que a maior importância dos modelos criados com RP's está na capacidade de analisar propriedades e verificar problemas existentes nos sistemas modelados. O autor considera que duas grandes classes de propriedades podem ser verificadas nos modelos: as propriedades dependentes das marcações iniciais (propriedades comportamentais) e as não-dependentes dessas marcações (propriedades estruturais).

No presente trabalho, o foco está nas propriedades comportamentais dos modelos e em como pode-se reduzir modelos sem perder tais propriedades, portanto, serão abordadas essas características para fomentar as bases da pesquisa.

Existem oito principais propriedades comportamentais que podem ser verificadas em modelos de Redes de Petri (MURATA, 1989). Essas principais propriedades serão abordadas superficialmente na presente pesquisa com o objetivo de contextualização. Qualquer definição mais profunda e com maior quantidade de detalhes pode ser verificada na pesquisa desenvolvida por Murata (1989).

As oito principais propriedades são descritas abaixo (MURATA, 1989):

1. Alcançabilidade (do inglês, *reachability*):

O conceito de alcançabilidade é fundamental para o estudo de propriedades dinâmicas de sistemas. Disparos efetuados por transições habilitadas fazem com que marcações fluam na rede, dessa forma, a distribuição de marcações muda com base nas regras de transição.

Um estado é dito alcançável se existe uma sequência de disparos que fazem com que marcações M_0 se tornem marcações M_n , ou seja, se marcações vindas de um estado inicial fluam até um estado n da rede.

O problema da alcançabilidade em Redes de Petri envolve um grafo que represente todos os estados alcançáveis da rede por meio de disparos das marcações iniciais da rede. Ou ainda, o problema pode envolver apenas subgrafos de alcançabilidade da rede, isso ocorre quando o interesse está apenas em saber se algum subconjunto de estados podem ser alcançáveis por meio de marcações partindo de um ponto inicial da rede.

Verificações relativas a alcançabilidade em RP's são extremamente complexas e envolvem espaço e tempo exponenciais, tornando o problema extremamente complexo para grandes modelos.

2. Delimitação (do inglês, *boundedness*):

Uma Rede de Petri é dita delimitada ou limitada quando o número de marcações em cada lugar P da rede não excede um número finito k , para qualquer lugar que seja alcançável por meio das marcações iniciais.

Lugares nas RP's são frequentemente usados para representar buffers ou algum armazenamento intermediário, como mostra a Tabela 1, portanto, a propriedade de delimitação é importante de ser avaliada para evitar que exista uma sobrecarga nos buffers mesmo depois de diversas sequências de disparo.

3. Vivacidade (do inglês, *liveness*):

O conceito de vivacidade está intimamente ligado com a ausência de deadlocks em sistemas. Uma rede é considerada viva se qualquer estado alcançado por meio de M_0 , seja possível realizar uma sequência de disparos que atinja qualquer transição do modelo.

Se uma RP é viva ela garante a ausência de deadlocks com qualquer sequência de disparos que tenha sido realizada durante a execução do sistema.

Essa propriedade é muito custosa e complexa de ser verificada para grandes sistemas, por isso existem quatro classificações para a vivacidade de RP's. Uma rede L0-viva tem transições que nunca podem ser ativadas, uma rede L1-viva tem transições que podem ser ativadas pelo menos uma

vez, uma rede L2-viva tem transições que podem ser ativadas pelo menos um número k finito de vezes, uma rede L3-viva tem transições que podem ser ativadas aparentemente infinitas vezes com uma sequência de disparos e uma rede é L4-viva se as transições forem L1-vivas para cada marcação M em $R(M_0)$.

4. Reversibilidade (do inglês, *reversibility*):

Uma RP é reversível se, para cada marcação M que estiver em algum ponto da rede $R(M_0)$, seja possível alcançar a situação de marcações iniciais M_0 com uma sequência de disparos partindo de M .

As propriedades mostradas até agora são independentes, ou seja, podem existir diversas combinações entre as mesmas em uma determinada RP.

5. Coberta (do inglês, *coverability*):

Considera-se que uma marcação M em uma rede (PN, M_0) é coberta se existem marcações M' em $R(M_0)$ de maneira que $M'(p) \geq M(p)$ para qualquer lugar P da rede. É um conceito relacionado com L1-viva, pois se temos M como o mínimo de marcações necessárias para ativar uma transição T , podemos dizer que T não é L1-viva somente se M não for coberta.

6. Persistência (do inglês, *persistence*):

Em uma rede persistente, se em algum momento duas transições estão ativas, o disparo de uma das duas não desativa a outra, ou seja, se em algum momento uma transição T for ativada ela não será desativada sem disparar.

Esse conceito é importante para avaliação de sistemas paralelos, onde existe disputa por recursos e mais de uma ação costuma estar vinculada a um mesmo recurso ou conjunto de recursos.

7. Distância sincrônica (do inglês, *synchronic distance*):

O conceito de distância sincrônica é fundamental para sistemas de condição/evento, pois representa o grau de dependência mútua de dois eventos.

A definição da distância sincrônica entre duas transições T1 e T2 em uma RP (PN, M0) é feita por meio de (1)

Equação 1. Equação para Cálculo de Distância Sincrônica

$$d_{12} = \max_{\sigma} |\sigma(t_1) - \sigma(t_2)|$$

A variável “ σ ” representa uma sequência de disparos começando em uma marcação M em R(M0). A expressão $\sigma(t_i)$ representa o número de vezes que a transição t_i , com $i = 1$ ou 2 , dispara durante a sequência de disparos σ .

A distância sincrônica representada por (1) é uma importante métrica para redes condição/evento.

8. Equidade (do inglês, *fairness*):

Pode-se citar dois conceitos básicos de equidade em RP's: equidade delimitada (do inglês, *bounded-fairness*) ou equidade incondicional.

Em uma rede com equidade delimitada, para cada par de transições T, uma transição pode disparar somente um número limitado de vezes enquanto a outra não dispara.

Uma sequência de disparos σ tem equidade incondicional se ela é finita ou se qualquer transição existente na rede apareça infinitas vezes na sequência de disparo σ .

A última definição importante a ser citada é a de Rede de Petri com equidade incondicional. Para que isso ocorra, todas as sequências de disparo σ da RP devem apresentar equidade incondicional.

Na próxima subseção serão abordados conceitos iniciais sobre a modelagem de sistemas usando as Redes de Petri, juntamente com alguns exemplos de utilização.

2.4 Modelagem de Sistemas Usando Redes de Petri

A característica principal em um problema de modelagem está na presença de eventos concorrentes e assíncronos que devem ser considerados e formalizados corretamente. Um exemplo de área para a qual a modelagem de sistemas envolve essa característica são os sistemas de manufatura. Nesses sistemas, a utilização de redes de Petri se encaixa perfeitamente, pois ela permite capturar as relações de precedência e interações entre os eventos. Utilizar essa abordagem insere a possibilidade de se avaliar qualitativamente o modelo em busca da identificação de propriedades do sistema relativas a conflitos, deadlocks e limites de escopo (DESROCHERS; AL-JAAR, 1995)

A utilização de modelos de rede de Petri como base para produzir controladores de tempo real em ambientes de manufatura é frequentemente abordada em pesquisas. As marcações fluem através do modelo descrevendo uma sequência de eventos que caracteriza a execução de uma tarefa de controle do ambiente como um todo ou ainda, apenas de um dos módulos de produção de uma peça ou parte de um produto (DESROCHERS; AL-JAAR, 1995).

Outro exemplo de sistema que pode ser modelado por Redes de Petri são caves de realidade virtual. Um CAVE é um exemplo de ambiente de realidade virtual que consiste basicamente de quatro paredes, frontal, direita, esquerda e uma tela de projeção no chão. Por meio da utilização de diversos projetores apontado para as partes do ambiente, simula-se um ambiente de realidade virtual para o usuário, lembrando que alguns outros componentes como óculos e bastão de realidade virtual são usados para visualização e captura de movimentos no ambiente virtual (SHEN et. al. 2013).

Após a apresentação dos conceitos que exibem a possibilidade de se usar redes de Petri para a modelagem dos mais diversos sistemas, e mais especificamente os de manufatura, exhibe-se agora um breve levantamento bibliográfico, pautado em iniciativas atuais, referentes a alguns objetivos e áreas de aplicação, que podem ser buscados por meio da referida técnica de modelagem. Os trabalhos propostos por Yasuda & Bin (2010); Ballarini et. al. (2011); Hu et. al. (2012); Huang et. al. (2012); Lee et. al. (2012); Tarnauca et. al. (2012); Lv et. al. (2013); Shen et. al. (2013), são

todos exemplos de utilizações de modelagem em diferentes tipos de redes de Petri para modelagem de sistemas.

Para contextualizar melhor a utilização das RP's na modelagem de sistemas, serão apresentados agora os principais pontos de algumas das pesquisas citadas, mostrando objetivos e aplicações, sendo que o objetivo principal é exibir pesquisas atuais que utilizam o formalismo baseadas nas características desejáveis (dos problemas abordados) que podem ser representadas com a modelagem.

O trabalho proposto por Ballarini et. al. (2011) faz uso de RP's estocásticas para modelar um Sistema Flexível de Manufatura (FMS). Por ser um problema complexo e com muitas relações entre os elementos, as RP's são frequentemente usadas neste escopo de problema. A lógica *Hybrid Automata Stochastic Logic* (HASL) é utilizada para avaliar, juntamente com a RP, quesitos relativos à checagem e avaliação de performance dos modelos criados. O objetivo da abordagem é utilizar uma ferramenta que permita ao usuário a criação e avaliação de um modelo de maneira pouco complexa.

Huang et. al. (2012) é outro pesquisador que dedica seus esforços no tratamento de problemas relativos à modelagem de Sistemas Flexíveis de Manufatura (FMS), mais exatamente em uma política de prevenção de i. O formalismo das RP's é utilizado para construir o modelo, pois, de acordo com a pesquisa, este formalismo é considerado um dos mais eficientes para representar as relações entre elementos existentes nos FMS's. Aponta-se ainda a capacidade de verificar *deadlocks* por meio de grafos de alcançabilidade e avaliação estrutural como características desejáveis e providas pelos modelos de RP.

A pesquisa conduzida por Lv et. al. (2013) trata de um escopo de problema bastante atual, relacionado a Sistemas de Manufatura. Para serem capazes de executar processos em fábricas dispostas em diferentes localizações geográficas são utilizados Sistemas de Manufatura Distribuídos. Para lidar com a sincronização de processos e concorrência, escolheu-se um tipo de RP como formalismo de modelagem, as Redes de Petri Temporizadas Coloridas. Os resultados obtidos foram bastante satisfatórios e essa é mais uma das várias pesquisas que provam a eficiência da construção de modelos de RP para representação e interpretação de problemas relacionados com a Manufatura.

Mais uma classificação de problema relacionado à manufatura são os Sistemas Automatizados de Manufatura. Essa é a nomenclatura de problema abordada na

pesquisa conduzida por Hu et. al. (2012), o propósito desse tipo de sistema é supervisionar problemas de controle. Dentro dessa classe de problemas pode-se destacar a necessidade reforçar a vivacidade da rede e imparcialidade (*fairness*). A pesquisa utiliza um tipo específico de RP, dito como não explorado na literatura, juntamente com um método algébrico para reforçar vivacidade e imparcialidade, obtendo resultados promissores, mas que ainda precisam ser melhorados.

Os diversos componentes de uma CAVE de realidade virtual também podem ser modelados por Redes de Petri para que o comportamento do sistema seja estudado. A CAVE proposta inicialmente no trabalho de Cruz-Neira et. al. (1993), por exemplo, foi modelada como uma Rede de Petri por Shen et. al. (2013) com o objetivo de verificar uma nova abordagem de redução para RP's.

Após mostrar iniciativas atuais que utilizam RP's como uma ferramenta de modelagem, é interessante focar no que pode ser feito usando os modelos criados.

2.5 Métodos de Análise de Redes de Petri

Existem três métodos principais para analisar RP's: árvore de alcançabilidade, abordagem de equações matriciais e técnicas de redução ou decomposição (MURATA, 1989).

Árvore de alcançabilidade:

A partir da marcação inicial uma RP (N, M_0) , é possível que um certo número de novas marcações seja obtido, sendo esse número igual ao número de transições habilitadas. O processo de obtenção de novas marcações segue da mesma forma para cada nova marcação observada (MURATA, 1989).

O processo de obtenção de novas marcações pode ser modelado como uma árvore onde os nós são as marcações (sendo M_0 o nó raiz) e os arcos que ligam os nós representam o disparo de uma transição que leva a transformação de uma marcação em outra (MURATA, 1989).

Um problema importante das árvores de alcançabilidade é que para uma RP não-delimitada (*unbounded*), a árvore crescerá eternamente. Quando uma RP é delimitada, o método de árvore de alcançabilidade pode ser usado para verificar todas

as propriedades comportamentais mostradas na seção 2.3. Porém, frequentemente a árvore de alcançabilidade de RP's se torna muito grande devido ao problema de explosão de estados no modelo e isso deve ser contornado com outras estratégias para que a árvore seja usada na verificação de propriedades comportamentais (MURATA, 1989).

Matrizes e Equação de Estados:

A avaliação de comportamentos dinâmicos de sistemas em engenharia é frequentemente definida por meio de algumas equações diferenciais ou algébricas, porém, não é possível avaliar todo o comportamento dinâmico em RP's com equações (MURATA, 1989).

Algumas equações de matrizes podem ser usadas para descrever o comportamento de sistemas concorrentes baseados em RP's, mas as soluções obtidas com essas equações são limitadas pela natureza não determinística das RP's e porque as soluções devem ser definidas como valores inteiros não negativos (MURATA, 1989).

Regras de Redução para Análise:

Outra alternativa para analisar grande sistemas modelados com RP's é reduzir os modelos mantendo suas propriedades comportamentais para que a análise seja feita com menos esforço (MURATA, 1989).

Tendo uma rede (N, M_0) é possível criar, por meio das seis transformações que serão mostradas adiante, uma rede (N', M_0') com uma menor quantidade de elementos e que preserve as propriedades de vivacidade, delimitação e segurança (quando a rede k -delimitada tem $k=1$) (MURATA, 1989).

As transformações são: Fusion of Series Places (FSP), Fusion of Series Transitions (FST), Fusion of Parallel Places (FPP), Fusion of Parallel Transitions (FPT), Elimination of Self-loop Places (ESP) e Elimination of Self-loop Transitions (EST).

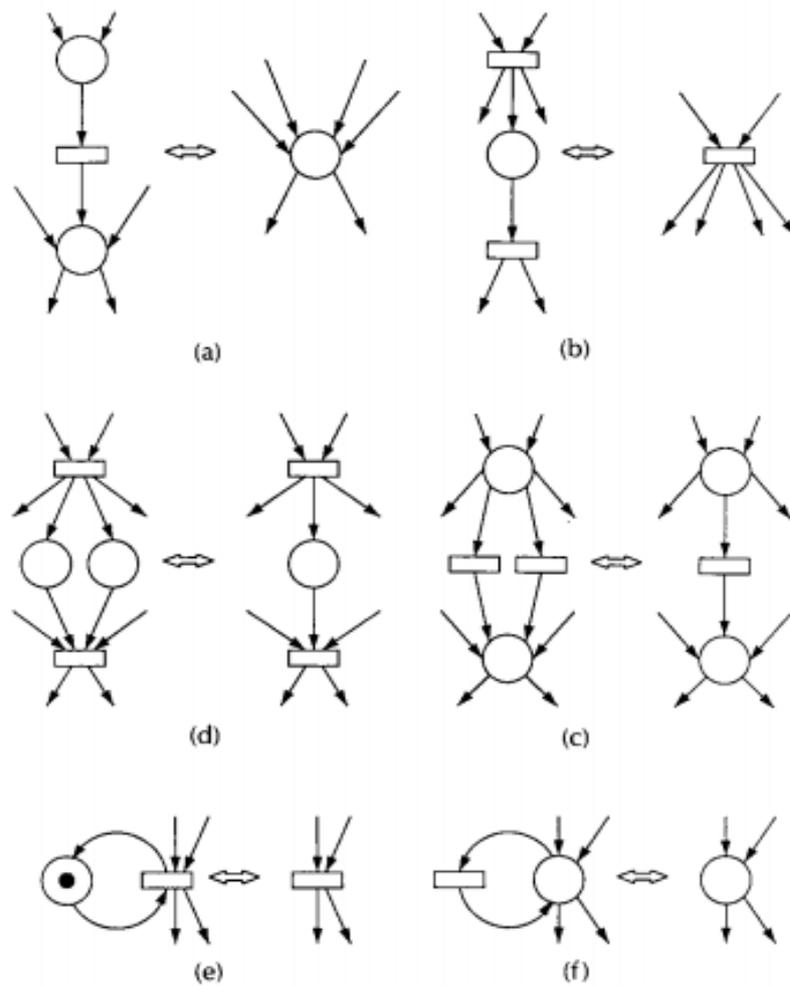


Figura 4. Regras de Redução: FSP (a), FST (b), FPT (c), FPP (d), EST (e), ESP (f) (MURATA, 1989).

O interesse do presente trabalho em utilizar tais regras de redução é simplificar o modelo para permitir, por exemplo, análises baseadas em árvores (ou grafos) de alcançabilidade (MURATA, 1989).

Para realizar essas análises os modelos devem seguir propriedades de delimitação e levando em consideração que as regras propostas permitem a manutenção dessa propriedade comportamental nos modelos reduzidos, a estratégia se torna viável (MURATA, 1989).

A propriedade de vivacidade das redes é também mantida quando aplicadas as referidas regras. Como dito anteriormente, a vivacidade está fortemente ligada com verificações do acontecimento de *deadlocks*, portanto, a criação de método de redução eficiente e automatizado usando as seis regras de redução citadas servirá para simplificar posteriores análises de propriedades comportamentais nas redes (MURATA, 1989).

A próxima subseção trata dos conceitos básicos relacionados aos Algoritmos Genéticos. Tais conceitos são importantes para entender a base do algoritmo de aplicação de reduções proposto na presente pesquisa.

2.6 Algoritmos Genéticos

A resolução de problemas por meio de busca é extensamente abordada na área de Inteligência Artificial. Essa abordagem se baseia na definição do problema e na definição de passos à serem executados por um agente em busca de uma solução para o problema (GOLDBERG, 1989; RUSSELL; NORVIG, 2009).

As buscas são divididas normalmente em buscas sem informação e com informação. As buscas sem informação trabalham gerando sistematicamente novos estados (de um problema modelado) e comparando-os com o objetivo. Porém, tais modelos se mostram bastante ineficientes na maioria dos problemas que envolvem um espaço de busca grande. Já as buscas com informação, que utilizam conhecimentos específicos do problema, podem encontrar soluções mais eficientes varrendo grandes espaços de busca (RUSSEL; NORVING, 2004; GOLDBERG, 1989).

Os Algoritmos Genéticos são inspirados pela biologia evolucionária e realizam buscas (com informação) que exploram o espaço de estados (do modelo de um problema) sem explorar de maneira sistemática cada caminho possível. As buscas são realizadas em todo o espaço de estado e todas as soluções geradas são avaliadas (RUSSEL; NORVING, 2004; GOLDBERG, 1989).

De maneira geral, um Algoritmo Genético codifica as variáveis de decisão de um problema como cadeias de caracteres de um alfabeto (ou alguma estrutura alternativa) de tamanho finito. Essas cadeias ou estruturas são conhecidas como cromossomos ou indivíduos e são candidatos a solucionar o problema. Cada parte da estrutura que representa um cromossomo é conhecido como gene e o valor de cada gene é um alelo. No problema do caixeiro viajante (amplamente conhecido na computação), por exemplo, teríamos um cromossomo sendo uma rota, as cidades seriam os genes e as distâncias seriam os alelos.

Outro componente essencial para o funcionamento de um Algoritmo Genético é a função de avaliação (do inglês, *fitness function*). A função de avaliação é a maneira

com a qual é possível determinar se uma solução é ou não boa para atingir os objetivos. As avaliações são usadas para guiar o algoritmo em busca de melhores soluções (RUSSEL; NORVING, 2004; GOLDBERG, 1989).

Todo o conceito de Algoritmos Genéticos se baseia na existência de uma população. Um número pré-definido de soluções candidatas à resolver o problema compõe a população do Algoritmo Genético. O tamanho da população do AG afeta diretamente na escalabilidade e o desempenho do algoritmo, sendo assim, é importante definir corretamente o tamanho da população para resolução de um problema, lembrando que uma população pequena pode causar convergência prematura e uma grande pode causar desperdício de recursos computacionais (RUSSEL; NORVING, 2004; GOLDBERG, 1989).

O trabalho proposto por Goldberg (1989) divide a execução de um Algoritmo Genético no seguintes passos:

Inicialização:

O primeiro passo de execução é a criação de uma população inicial (normalmente em um processo aleatório). Em algumas situações é possível usar informações específicas do problema para gerar a população inicial.

Avaliação:

Após a criação da população inicial, os indivíduos são avaliados e cada um recebe seu valor de *fitness* que será necessário para em outros passos da execução.

Seleção:

Nessa fase os indivíduos que serão usados na etapa de cruzamento são selecionados. Existem diversos procedimentos de seleção como, por exemplo, seleção por roleta, seleção estocástica, seleção ranqueada e seleção por torneio. Para aprofundar no funcionamento de cada tipo de seleção indica-se o trabalho de Goldberg, (1989).

Recombinação ou Cruzamento (do inglês, *crossover*):

Nessa etapa do processo duas ou mais (dependendo da estratégia de recombinação) partes dos pais selecionados na etapa anterior são recombinados com

o objetivo de criar filhos (soluções) mais aptos para resolver o problema. Os filhos são uma mistura de características dos pais selecionados e a estratégia de recombinação escolhida define como soluções diferentes podem ser criadas.

Mutação:

A etapa de recombinação envolve duas soluções candidatas, já a etapa de mutação ocorre aleatoriamente e modifica a estrutura de apenas uma solução candidata. Assim como a recombinação existem várias maneiras de se realizar a etapa de mutação em um Algoritmo Genético.

Substituição:

Na etapa de substituição, os filhos criados por meio de seleção, recombinação e mutação substituem pais presentes na população. Assim como na seleção, na etapa de substituição existem diferentes métodos que podem ser adotados.

Condição de Parada:

As etapas de 2 a 6 são repetidas até que uma condição de parada seja atingida pelo algoritmo. Dois exemplos de condição de parada são: número de gerações, quando se define um número fixo de gerações que serão executadas, e definição de uma função de avaliação, que quando atendida por uma solução candidata, provoca o término da execução do algoritmo.

Esclarecimentos mais aprofundados sobre Algoritmos Genéticos podem ser obtidos nos trabalhos de Holland (1975); Goldberg (1989); Russell & Norvig (2004).

Capítulo 3

REVISÃO BIBLIOGRÁFICA

O presente capítulo tem por objetivo evidenciar que a utilização de Redes de Petri para modelagem de sistemas é um tema atualmente abordado e explorado com os mais diversos objetivos e motivações. Ainda neste contexto pode-se perceber que nas mais diferentes áreas nas quais as Redes de Petri são utilizadas como formalismo de modelagem um problema é frequentemente citado, a quantidade de elementos necessários para modelar sistemas que tenham muitos componentes a serem descritos se torna um limitante para simulação e verificação de propriedades dos sistemas modelados.

Frequentemente, para simplificar a análise de sistemas grandes, são feitas reduções no modelo a fim de gerar um modelo reduzido que seja semanticamente equivalente e que mantenha as propriedades do sistema original (DESROCHERS; AL-JAAR, 1995; MURATA, 1989; JENG; DICESARE, 1993). Esta estratégia pode ser vista neste mesmo contexto, porém, com uma pequena mudança de perspectiva, na presente pesquisa. Reduzir um modelo de sistema de manufatura utilizando Redes de Petri, facilita a análise do mesmo, e essa síntese pode facilitar a implementação deste modelo, ou seja, transformar o modelo de controle em um hardware que possa realmente executar esse controle em um ambiente de manufatura, tornando essa uma tarefa menos complexa.

Existem diversas técnicas de síntese de redes de Petri, sendo que para a aplicação de cada uma delas, é necessário que a rede atenda a algumas propriedades comportamentais e estruturais. Como o problema de complexidade é frequentemente observado nas redes de Petri, é importante desenvolver métodos de transformação

que sejam capazes de sintetizar o modelo preservando suas características originais (MURATA, 1989).

Após o levantamento das considerações feitas por Murata, (1989), temos um passo inicial para avaliar a necessidade e importância de pesquisas realizadas a fim de sintetizar redes de Petri.

O próximo passo é contextualizar esse tipo de estratégias no escopo de ambientes de manufatura e também com outras aplicações, mostrando utilizações de redução em diversos modelos, e juntamente com essa contextualização mostrar que o tema continuou sendo abordado com considerável interesse.

Os trabalhos propostos por Lee-Kwang et. al. (1987); Murata, 1989; Dicesare, (1993) são importantes bases teóricas para pesquisas relativas a RP's. Desde a época em que tais trabalhos foram propostos as técnicas de síntese e redução de modelos de RP vem sendo extensamente abordadas, gerando propostas de grande contribuição para esse campo de pesquisa.

Os trabalhos abordados nas subseções a seguir apresentam diversas técnicas de redução de redes de Petri, bem como, diversos objetivos e justificativas para a aplicação de tais técnicas, mostrando que a síntese de redes de Petri é um caminho que, no mínimo, despertou e ainda desperta, certo interesse na comunidade científica.

3.1 Regras de Redução para Redes de Petri Temporizadas

Contextualização:

O trabalho proposto por Sloan (1996) foca na apresentação de regras de redução voltadas para modelos de RP temporizadas. O direcionamento proposto por Jeng (1993), que citou o caminho para futuras pesquisas como sendo criar regras e métodos voltados para variantes de RP's, é observado na presente pesquisa.

Objetivo:

O objetivo do trabalho foi adaptar reduções criadas para modelos genéricos de RP's, para torna-las aplicáveis a modelos baseados em redes temporizadas. Essa adaptação fez com que fosse necessário definir algumas novas pré-condições que

devem ser atendidas para que as regras criadas para as redes temporizadas tenham o mesmo efeito de suas correspondentes para redes genéricas.

Qualquer pesquisa que trate da redução de RP's, deve deixar claro quais as propriedades que o modelo reduzido manterá em relação ao modelo inicial. Os modelos produzidos usando as regras de redução para redes temporizadas mantêm a vivacidade da rede bem como características de tempo.

Contribuição:

As regras adaptadas são consideradas uma boa contribuição para quem busca realizar verificações do comportamento dinâmico de sistemas modelados (por exemplo, *deadlocks*, que estão intimamente ligados com a vivacidade da rede) usando Redes de Petri Temporizadas.

Regras de Redução:

As regras usadas para adaptação na pesquisa foram propostas por Berthelot (1985) e serviram, antes de sua adaptação, para realizar a verificação de *deadlocks* em programas da linguagem ADA.

No trabalho, a regra “fusão serial” é detalhada e a transformação é mostrada para provar que as propriedades sejam mantidas. As demais regras são citadas de maneira superficial.

A Figura 5 mostra a regra de fusão serial:

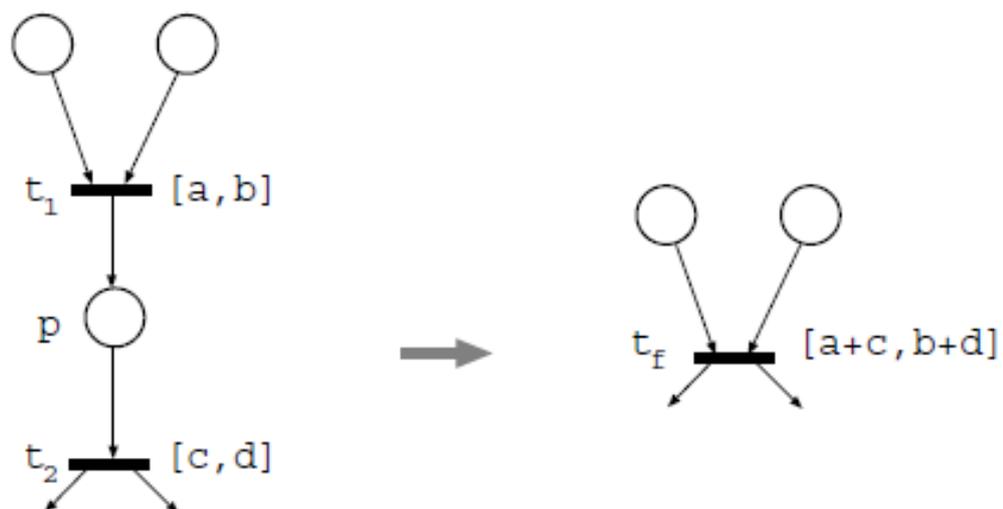


Figura 5. Regra de Fusão Serial (SLOAN, 1996).

Para provar que a regra pode ser adaptada para Redes de Petri Temporizadas são abordadas as pré-condições para que a redução possa ser realizada.

Existem três regras que devem ser satisfeitas para que a redução seja feita em modelos de RP não-temporizados:

1. O lugar "p" não deve ter marcações iniciais da rede.
2. O lugar "p" deve ser o único lugar que recebe saída (*output*) da transição "t1" e deve ser o único lugar a dar entrada (*input*) na transição "t2".
3. O lugar "p" não deve estar conectado a nenhuma outra transição.

Além das três regras acima, uma quarta regra é proposta e provada por Sloan (1996), para garantir que a regra de redução seja aplicável em redes temporizadas:

4. A transição "t1" não pode compartilhar suas entradas (*inputs*) com outras transições.

A Figura 6 é usada para ilustrar a necessidade da quarta regra nas redes temporizadas.

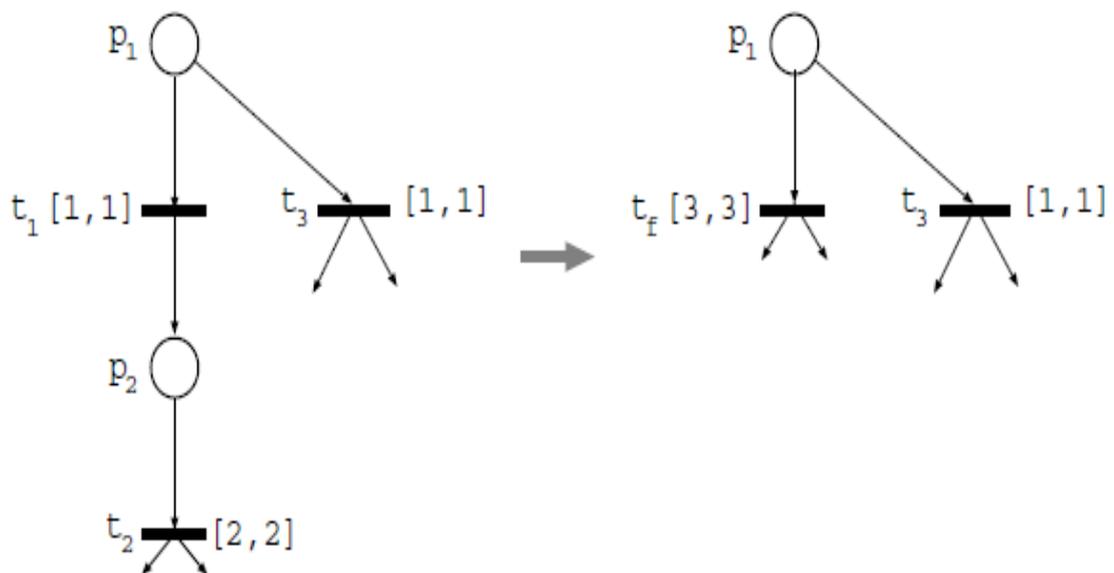


Figura 6. Ilustração da necessidade da quarta pré-condição para redes temporizadas (SLOAN, 1996).

A rede da esquerda na Figura 6 atende as três primeiras propriedades mas não atende a quarta propriedade. Na rede da esquerda é possível que quando uma marcação chegue até o lugar "p1" a transição "t1" ou a "t2" possam ser ativadas dependendo de uma função baseada em temporizador. Já nessa mesma situação na rede da direita (rede reduzida), a possibilidade de acontecer a ativação de "t1" ou "t2" seria perdida.

Validação:

O processo de validação na pesquisa é feito da forma mostrada acima, para cada regra de redução, o autor mostra quais as novas pré-condições que devem ser atendidas para que a regra passe a valer para redes temporizadas e prova por meio de equações matemáticas que essa utilização manterá propriedades comportamentais da rede original.

Conclusão:

O resultado final foi a obtenção das regras adaptadas para redes temporizadas. Essas regras foram consideradas bem sucedidas para facilitar a análise de sistemas em tempo real, seguindo o caminho dos resultados da pesquisa de Bethelot (1985), de onde foram retiradas as regras para a adaptação.

3.2 Política de Prevenção de *Deadlocks* Otimizada Aplicada à Sistemas Flexíveis de Manufatura Baseada em Reduções de Redes de Petri

Contextualização:

A pesquisa desenvolvida por Uzam (2004) trata do problema de existência de *deadlocks* em modelos de Sistemas Flexíveis de Manufatura (FMS, do inglês, *Flexible Manufacturing System*). De acordo com o autor o problema de *deadlocks* pode ser tratado com políticas para prevenir, detectar ou evitar. Dentro das três classes citadas, escolheu-se desenvolver a pesquisa focando em uma política para prevenção proposta na pesquisa de Uzam (2002) que se mostrava ineficiente para sistemas que sofriam com o problema da grande profusão de elementos.

Objetivo:

O objetivo da pesquisa é mostrar que a aplicação de reduções em modelos de FMS pode tornar viável a utilização de uma técnica de prevenção de *deadlocks*. A técnica de prevenção de *deadlocks* foi proposta em uma terceira pesquisa, se mostrava bastante eficiente, porém, sua aplicação se tornava ineficiente em modelos

que eram compostos por muitos elementos. Para atenuar essa limitação, foi proposta a aplicação de um método baseado em regras de redução, para que os modelos pudessem ficar menores e a técnica de prevenção pudesse ser aplicada eficientemente.

Contribuição:

A contribuição da pesquisa está em provar que políticas para prevenir *deadlocks* em modelos de RP's de FMS's podem se beneficiar da prévia aplicação de regras de redução nos modelos. Ou seja, a contribuição não está em propor novas técnicas ou em criar um mecanismo de prevenção de *deadlocks*, está em mostrar que a utilização de regras de redução nos modelos diminui a rede preservando propriedades como delimitação, vivacidade e reversibilidade, fazendo com que estratégias para prevenir *deadlocks* usando grafos de alcançabilidade possam ser melhor aplicadas.

Regras de Redução:

As seis regras de redução abordadas na pesquisa são apresentadas por Murata (1989), e são as mesmas escolhidas para o método de redução apresentado na presente pesquisa. A Figura 4 apresenta as referidas regras de redução.

Na pesquisa, o autor opta pela utilização das seis regras propostas por Murata (1989) devido as propriedades comportamentais que são mantidas na rede reduzida ao aplicar as regras. Com elas é possível garantir que a rede mantenha propriedades de vivacidade, delimitação e reversibilidade. Essas propriedades são importantes pois se elas fossem modificadas, principalmente a propriedade de delimitação, a utilização de uma política de prevenção de *deadlocks* baseada em grafos de alcançabilidade não poderia ser aplicada corretamente nas redes reduzidas.

Validação:

A validação é dividida em três passos:

Primeiramente o autor seleciona um modelo de RP de um FMS e aplica uma sequência de reduções (manualmente) no modelo, tendo como resultado uma RPR (Rede de Petri Reduzida).

O segundo passo é criar o grafo de alcançabilidade da rede reduzida.

Em terceiro lugar, utiliza-se a teoria de regiões para identificar áreas potencialmente causadoras de *deadlocks* para que se possa realizar as modificações necessárias e evitar a ocorrência *deadlocks*.

Como último passo de validação o autor compara a quantidade de estados alcançáveis em uma rede não reduzida e em uma rede reduzida para evidenciar a quantidade de estados a menos que devem ser considerados, tornando a estratégia de identificação de *deadlocks* baseada em teoria de regiões muito mais simples de ser aplicada em grafos de alcançabilidade reduzidos.

É importante frisar que essa estratégia só é possível pois as propriedades comportamentais de vivacidade, delimitação e reversibilidade da rede são mantidas com a redução.

Conclusão:

A conclusão da pesquisa é que as estratégias para prevenção de *deadlocks* em sistemas modelados por RP's podem ter sua utilização otimizada se aplicadas em redes reduzidas.

É importante observar que um método como o proposto na presente pesquisa (de redução automática de modelos de RP) traria benefícios para a pesquisa de Uzam (2004), pois os modelos poderiam ser reduzidos de maneira automática, fazendo com que o primeiro passo de validação da pesquisa fosse otimizado.

3.3 Novas e Eficientes Regras de Redução de Redes de Petri para Verificação de Programas Paralelos

Contextualização:

O método proposto na pesquisa conduzida por Letters (2006) é focado na avaliação de modelos de RP de programas paralelos. A avaliação de modelos é dividida em duas categorias: avaliação estrutural usando algoritmos ou enumeração dos estados alcançáveis no modelo do programa. As avaliações estruturais nem sempre garantem que os modelos gerados mantenham todas as propriedades dos modelos originais. Já as avaliações de enumeração de estados alcançáveis podem sofrer com o problema da explosão combinatória de estados (ou grande profusão de

elementos) e isso limita a eficiência desse tipo de análise. Para aproveitar das vantagens dos dois métodos, cita-se a estratégia de reduzir o modelo e posteriormente enumerar os estados alcançáveis.

Objetivo:

Busca-se com a pesquisa criar a generalização de duas regras de reduções propostas por Berthelot (1985). AS regras de pré-aglomeração e pós-aglomeração são usadas para fundir transições sequenciais. Porém, essas regras são consideradas ineficientes para avaliar programas paralelos, pois estes devem ter seu comportamento avaliado sem que exista um fim, ou seja, uma avaliação das propriedades considerando que o sistema irá executar para sempre. O objetivo do trabalho é prover uma generalização usando restrições algébricas para que as duas regras passem a preservar uma classe mais ampla de propriedades e possam ser usadas mais eficientemente na avaliação de programas paralelos.

Contribuição:

Permitir que uma maior gama de modelos de RP possam ser reduzidos por meio do enfraquecimento de condições estruturais locais das regras de redução abordadas. Ou seja, a contribuição está na generalização de regras para que as mesmas tenham menos limitações de estrutura local para serem aplicadas em modelos.

Regras de Redução:

As regras de redução usadas na pesquisa são: pós-aglomeração e pré-aglomeração. Ambas as regras são usadas para fundir transições sequenciais e foram retiradas do trabalho de Berthelot (1985) para serem generalizadas.

Validação:

Para validar a pesquisa o autor utilizou a ferramenta Quasar, de análise de programas concorrentes em ADA, e criou um método de aplicação das regras generalizadas que trabalha juntamente com a ferramenta de análise. Assim, era possível avaliar o modelo gráfico e identificar onde reduzir a rede e automaticamente testar o modelo reduzido.

O método de redução é semiautomático, pois é necessário que o usuário da ferramenta identifique onde reduzir a rede e depois teste os modelos reduzidos.

Conclusão:

Conclui-se que as regras generalizadas não são tão limitadas com relação a propriedades estruturais. A generalização permitiu que monitores, acessos a variáveis compartilhadas e controle de *deadlocks* (características que frequentemente são usadas em programação paralela) pudessem ser menos afetados pelas reduções e conseqüentemente, melhorou a avaliação dos modelos de programas paralelos em ADA.

3.4 Ferramenta de Software para Transformação de Redes de Petri Coloridas - TransCPN

Contextualização:

A evolução das pesquisas relacionadas às Redes de Petri deu origem a extensões das mesmas a fim de melhorar o poder dos modelos para representar situações com características bastante específicas. O trabalho desenvolvido por Mikolajczak (2009) cita a existência da extensão das RP's chamada Rede de Petri Colorida (RPC). De acordo com o autor, essa extensão é frequentemente usada para especificar e analisar protocolos de redes, desenvolvimento de softwares, sistemas de manufatura, entre outros.

A pesquisa é voltada para as Redes de Petri coloridas, mais exatamente, focada na redução desse tipo de redes para posterior avaliação usando a estratégia de grafo de alcançabilidade.

Objetivo:

A pesquisa foi desenvolvida com o objetivo de criar um software que permita aos usuários abrir um modelo de RP colorida e escolher regras de redução para a aplicação no modelo. Ou seja, o software simplifica o trabalho dos pesquisadores, que não precisam criar algoritmos específicos para reduzir seus problemas. Possibilita-se escolher as regras e requisitar a redução escolhendo o ponto da estrutura da rede na qual a regra será aplicada.

Contribuição:

A contribuição é a criação de uma *toolbox* para o software TransCPN. O algoritmo desenvolvido para a *toolbox* é um método semiautomático de redução que permite ao usuário dar uma Rede de Petri colorida como entrada e escolher onde aplicar as regras de redução. Com isso, pesquisadores que precisem criar grafos de alcançabilidade de Redes Coloridas para verificar propriedades, podem usar o método semiautomático proposto para reduzir o problema da grande profusão de elementos e posteriormente criar o grafo de alcançabilidade da rede reduzida.

Regras de Redução:

As regras de redução escolhidas para o desenvolvimento do software são as seis propostas por Murata (1989) e exibidas na Figura 4. As mesmas seis regras escolhidas para a presente pesquisa, que mantém as propriedades de vivacidade, delimitação e segurança e permitem avaliações relativas a *deadlocks* e também outros comportamentos dinâmicos do modelo.

Validação:

A validação da pesquisa é feita por meio de duas operações.

Primeiramente, aplica-se as regras de redução com o software criado na pesquisa em uma Rede de Petri Colorida, criando uma rede equivalente reduzida.

Depois disso são feitas duas análises baseadas em grafos de alcançabilidade (usando a ferramenta *CPNtools*), uma na rede original e outra na rede reduzida.

Com essa estratégia o autor prova que a utilização de regras de redução em modelos simplifica a criação de grafos (ou árvores) de alcançabilidade para testes de propriedades e comportamentos dinâmicos da rede. Ou seja, mostra-se que o grafo de alcançabilidade de uma rede reduzida é mais facilmente criado e permite avaliar as propriedades assim como quando o grafo é criado em uma rede não reduzida, sempre tendo em vista que essa estratégia depende de quais propriedades dos modelos são mantidas pelas regras de redução escolhidas.

Conclusão:

Após a série de testes realizadas com a ferramenta concluiu-se que a *toolbox* com o objetivo de tornar semiautomático o processo de redução de RPC's é uma boa

ferramenta para pesquisadores que desejam diminuir o problema da grande profusão de elementos nos modelos de RPC, tornando a estratégia de verificação de propriedades dos modelos por meio de grafo de alcançabilidade menos complexa de ser executada.

3.5 Preservação de Propriedades em Reduções de Redes de Petri Temporizadas

Contextualização:

O trabalho aborda a possibilidade de criação de novas regras de redução para as Redes de Petri Temporizadas. De acordo com o autor, a utilização de reduções em modelos gera modelos equivalentes, porém, cita-se que a equivalência não tem relação direta com a estrutura da rede, mas sim com as propriedades comportamentais que podem ser verificadas. Cita-se ainda que a validade de uma redução depende do tipo de equivalência de propriedades que é necessária no contexto do problema.

Objetivo:

O objetivo da pesquisa é propor novas regras de redução que podem ser usadas em modelos de Rede de Petri Temporizadas. Pode-se considerar que a pesquisa segue o caminho do trabalho proposto por Sloan (1996).

Contribuição:

Criação de regras de redução que podem ser usadas para simplificar a avaliação de propriedades de modelos de Redes de Petri Temporizadas focando na manutenção das propriedades comportamentais da rede original na rede reduzida.

Regras de Redução:

As regras de redução propostas não recebem nomes, define-se apenas com o nome “regra de redução” e os números de um a quatro. Como as regras não são amplamente conhecidas e não são baseadas diretamente em regras já existentes, é importante ilustra-las para melhor compreensão da proposta. A Figura 7 mostra as quatro regras de redução propostas.

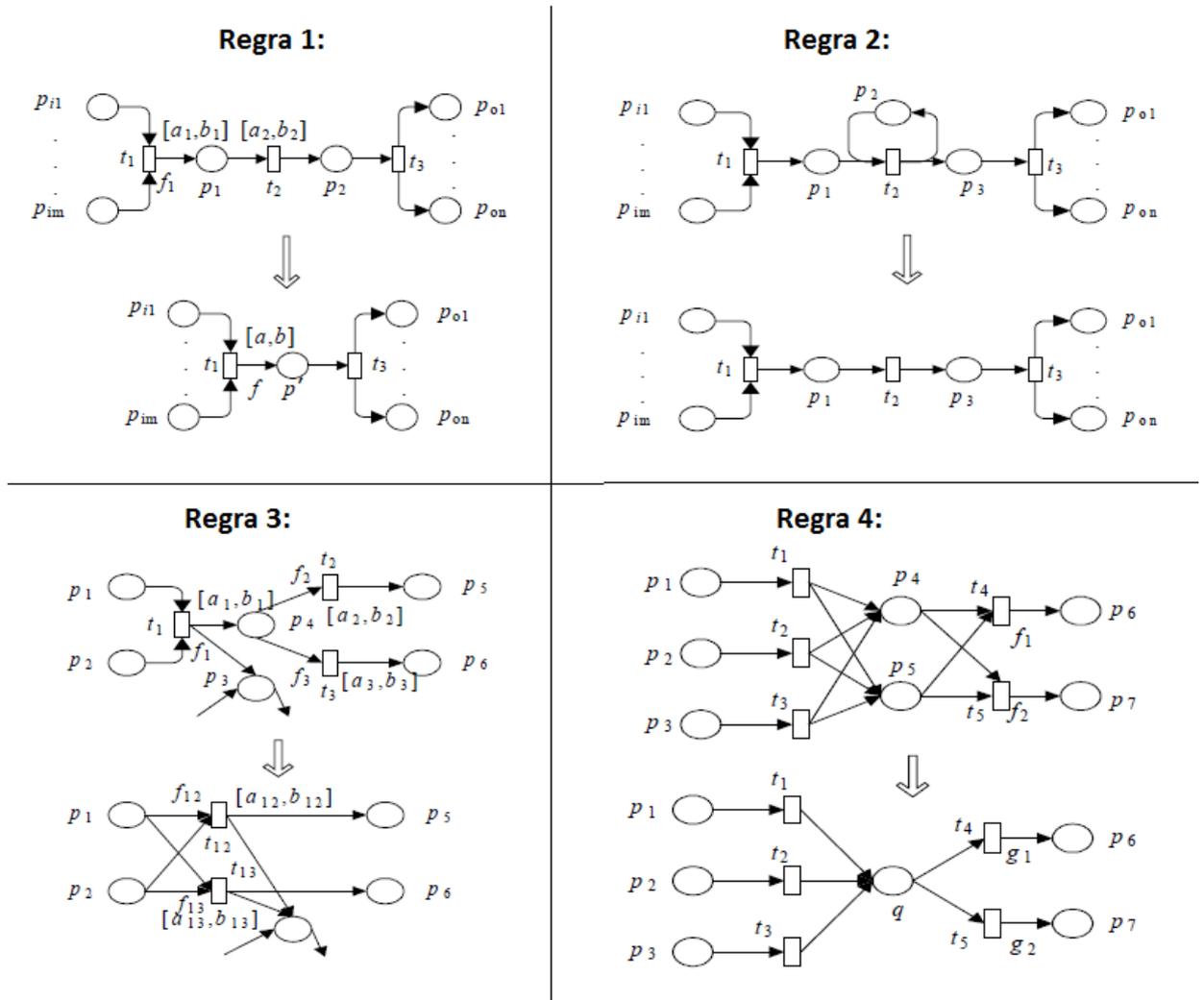


Figura 7. Regras de Redução para Redes de Petri Temporizadas (XIA, 2010).

Validação:

Para provar que as regras propostas são efetivas o autor utiliza-as em modelos de gerência de banco de dados com múltiplas cópias. Os modelos de gerência de banco de dados levam em consideração características de exclusão mútua para acesso a recursos compartilhados e acesso simultâneo a diferentes recursos do sistema.

A Figura 8 mostra o exemplo de aplicação da regra 3 em um modelo.

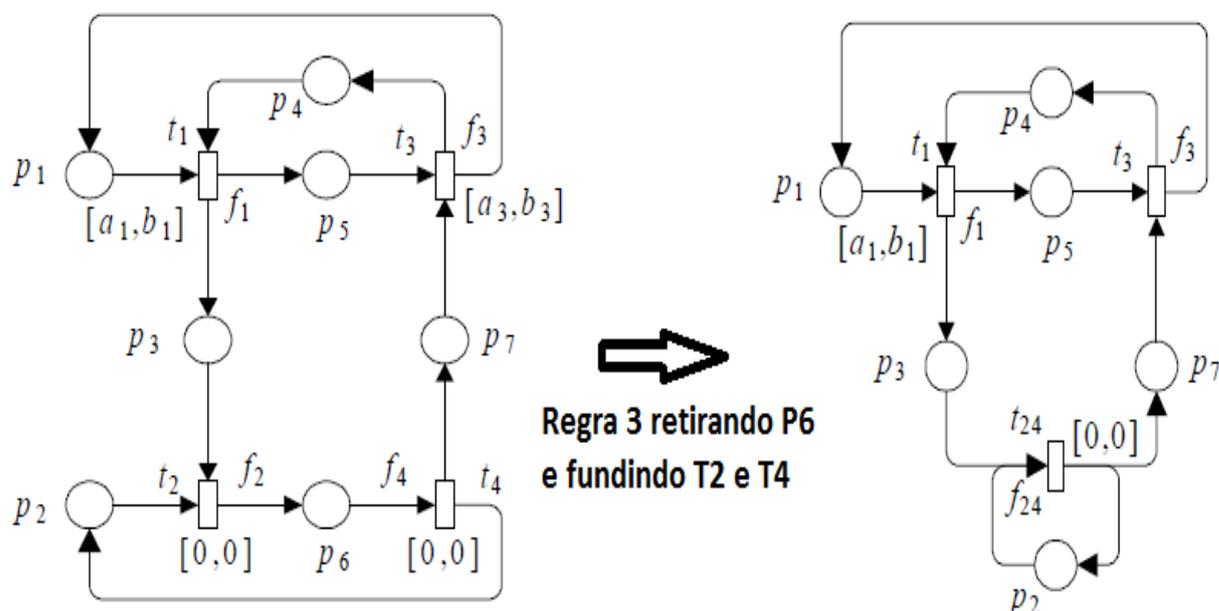


Figura 8. Aplicação da Regra Número 3 em um Modelo de Gerência de Banco de Dados (XIA; LIU, 2010).

Conclusão:

A conclusão do trabalho cita que as regras preservam as propriedades da rede como era necessário, porém, cita também que estudos mais aprofundados podem ser necessários para que se criem regras mais genéricas para as redes temporizadas. Ou seja, as regras criadas foram consideradas muito complexas e específicas para serem aplicadas com eficiência, dessa forma, sua utilização fica comprometida.

3.6 Método de Verificação Aprimorado Baseado em Redução de Redes de Petri para Modelos de *Workflow*

Contextualização:

Um modelo de workflow estrutura e semanticamente correto é um premissa básica para atingir objetivos de negócio. Porém, esse tipo de verificação em modelos não pode ser feita depois que o modelo já apresentou problemas, ele deve ser exaustivamente testado e verificado antes de entrar em uso real para evitar um custo de manutenção muito elevado. Existem dois tipos de verificação que podem ser realizadas em modelos: verificação estrutural e semântica. A verificação semântica envolve a simulação dos modelos e considera os dados dessa simulação, porém, o autor trabalha com reduções apenas para verificação estrutural.

A Figura 9 mostra, na parte esquerda, características estruturais desejáveis no modelo e, na parte direita, características estruturais que podem causar problemas de *deadlock* e de sincronização.

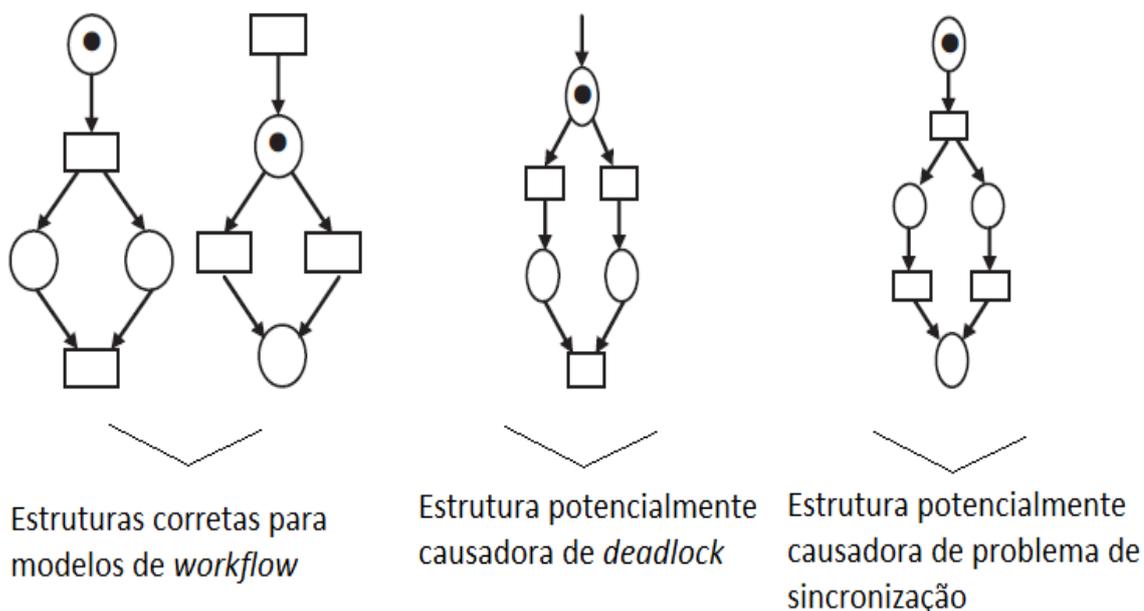


Figura 9. Características Desejáveis e Indesejáveis em Modelos de *Workflow* (QING-XIU et. al. 2010).

Objetivo:

O objetivo da pesquisa é mostrar que a aplicação de regras de redução em modelos de *workflow* pode ajudar a eliminar propriedades estruturais indesejáveis no

modelo, fazendo com que o modelo tenha maior integridade (por exemplo, estando livre de *deadlocks*) e possa ser levado para a execução com menor custo de manutenção.

Contribuição:

A principal contribuição da pesquisa é identificar duas características estruturais que implicam em problemas de sincronização e *deadlock*, e mostrar que com a aplicação sequencial de quatro regras de redução no modelo de *workflow* essas características estruturais são modificadas, evitando problemas de *deadlock* e sincronização no modelo.

Regras de Redução:

A pesquisa foi realizada com o uso de quatro regras de redução identificadas para modelos de *workflow*: redução de estruturas sequenciais, redução de estruturas interativas, redução de estruturas adjacentes e redução de estruturas fechadas. A Figura 10 ilustra a regra de redução de estruturas interativas e serve como exemplo para melhorar a compreensão da proposta.

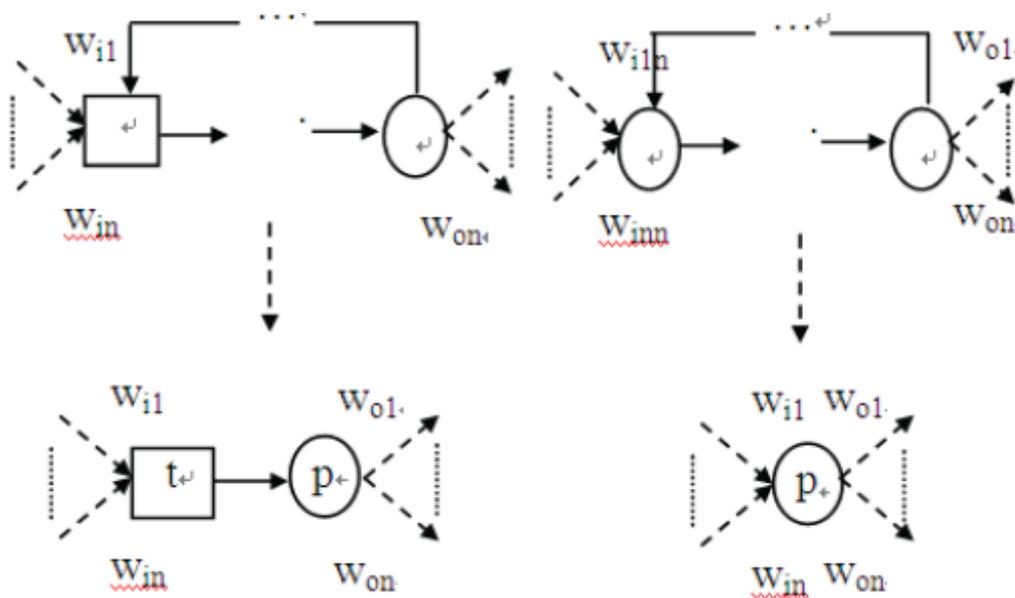


Figura 10. Redução de Estruturas Interativas (partindo de transições ou de lugares) (QING-XIU et. al. 2010).

Validação:

O primeiro passo para a validação é a transformação de um modelo não-livre de *workflow* em um modelo livre de *workflow*. Essa transformação consiste em

adicionar transições vazias para fazer com que o modelo tenha uma estrutura mais apropriada para a aplicação de regras de redução. A Figura 11 mostra a transformação de um modelo não-livre em livre com a adição de duas transições vazias, T' e T'' , e fusão dos lugares P_{21} e P_{23} .

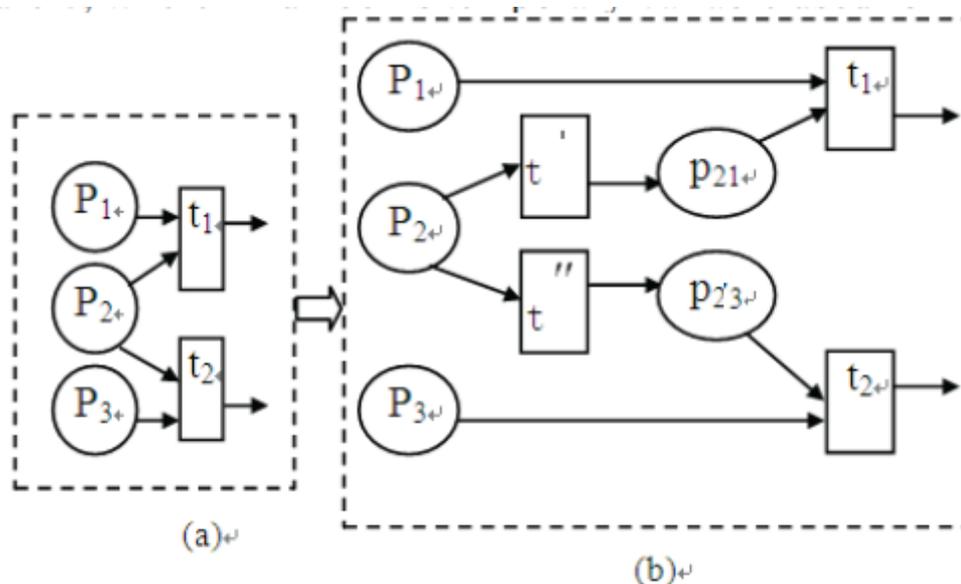


Figura 11. Transformação de Modelo Não-Livre para Modelo Livre (QING-XIU et. al. 2010).

Depois de criado um modelo livre de *workflow* são aplicadas regras de redução até que o modelo possa ser reduzido a um modelo com apenas um lugar. A proposta só é passível de ser feita por meio das modificações no modelo original e a aplicação das regras na sequência: redução sequencial, redução interativa, redução adjacente e redução de estruturas fechadas. Ou seja, o método é dependente da preparação do modelo (não aceita qualquer modelo como entrada) e aplica uma sequência pré-definida de reduções até que seja possível deixar o modelo com apenas um lugar.

Conclusão:

A pesquisa conclui que com a sequência de passos proposta é possível transformar o modelo não-livre em livre para que uma sequência de reduções seja aplicada a fim de verificar única e exclusivamente propriedades estruturais do modelo de RP, fazendo com que os modelos de *workflow* reduzidos estejam íntegros quanto a suas propriedades estruturais.

3.7 Decisão de Substituição de Serviços Web Baseada em Técnicas de Redução de Redes de Petri

Contextualização:

A pesquisa desenvolvida por Feng et. al. (2011) ajuda a dar a dimensão de como as reduções de redes de Petri, bem como a utilização das RP's para criação de modelos é atual e utilizada nos mais diversos contextos.

O autor trabalha com a Computação Orientada à Serviços (COS) para a criação de aplicações com serviços descentralizados providos por servidores web. Parte-se da premissa que não existe somente um servidor web capaz de prover um determinado serviço, portanto, em diversas situações pode ser necessário que um serviço seja substituído por outro de mesmo objetivo mas que é provido por outro servidor.

Uma das maneiras de modelar uma COS é por meio das Redes de Petri, assim, é possível realizar uma simulação para decidir quando e como substituir um dos serviços por outro equivalente sem que existam perdas para a aplicação como um todo. Para ser possível distinguir entre características comportamentais de interface e de processamento de cada serviço não é suficiente a utilização de uma Rede de Petri comum. A opção foi pela utilização de Redes de Petri Anotadas (LON, do inglês, *Labeled Open Net*).

Objetivo:

Propor uma metodologia para decidir a possibilidade de substituição de um serviço por outro que seja equivalente utilizando regras de redução no modelo de RP.

Contribuição:

Proposição de adaptações em regras de redução de RP's para Redes de Petri Anotadas que permitem simplificar modelos de COS e simplificar a decisão de substituição de serviços nos mesmos.

Regras de Redução:

As regras de redução utilizadas na proposta do método são adaptações das seis propostas por Murata (1989) e mostradas na Figura 4. As regras de redução foram adaptadas para trabalhar com Redes de Petri Anotadas.

Validação:

A validação da pesquisa é feita por meio de provas matemáticas das regras adaptadas. Nenhum cenário de testes utilizando algum modelo de COS é apresentado.

Conclusão:

Conclui-se que a pesquisa tem por objetivo propor uma nova metodologia para a decisão de substituição de serviços em modelos de COS, porém, não apresenta testes da metodologia. A validação fica restrita nas provas matemáticas das adaptações e em como estas podem vir a ser úteis para simplificar o trabalho de projetistas ao decidir quando e como substituir serviços.

3.8 Identificação Automática de Reduções para Simulação Discreta e Estocástica

Contextualização:

A pesquisa desenvolvida por Wu et. al. (2012) mostra outra perspectiva para utilização de modelos de PN e também de regras de redução.

Sistemas Químicos Celulares (SQC) costumam exibir comportamentos dinâmicos que são tanto estocásticos quanto discretos, devido a essas características, modelagem desses sistemas costumam ser feitas utilizando redes de Petri. O problema levantado na pesquisa é relacionado ao tamanho dos modelos SQC's, que fazem com que sendo necessário aplicar reduções nesses modelos para que seja possível simulá-los eficientemente.

Objetivo:

Criar um algoritmo que seja capaz de reduzir automaticamente (sem intervenções humanas) um modelo SQC, identificando situações em que reduções

específicas possam ser realizadas de maneira segura, eficiente e dinâmica (durante a simulação das reações químicas no sistema).

Contribuição:

Algoritmo baseado em Redes de Petri com Pesos (do inglês, *Weighted Petri net*) para identificar automaticamente possibilidades de aplicação de reduções em modelos SQC, tornando algoritmos de simulação mais facilmente aplicáveis e, ao contrário de outras pesquisas, sem necessidade de inserção de conhecimento específico por parte de um analista humano.

Regras de Redução:

As regras de redução para modelos SQC são bastante específicas para esse contexto de problema. De maneira geral, os dois tipos de redução que podem ser encontradas pelo algoritmo são reduções de reações rápidas ou reações lentas. As reações são modeladas por meio de transições, enquanto os elementos são modelados por lugares. O algoritmo busca reduções rápidas ou lentas levando em consideração quais elementos estão envolvidos com qual reação e faz a redução dos elementos e reação, tornando o sistema menor para continuar as simulações.

Validação:

A validação da pesquisa é feita aplicando o algoritmo em modelos de SQC e identificando as reduções propostas. O algoritmo identifica as possibilidades de redução com base em qual algoritmo de simulação será usado naquele modelo, tendo em vista que os algoritmos de simulação tem características distintas.

Conclusão:

O autor cita a existência de softwares capazes de capturar características desses sistemas por meio de simulações. O problema é que devido à necessidade de simular todas as reações levando em consideração tempo e situações dinâmicas, os softwares acabam sofrendo do problema de grande profusão de elementos.

Objetivando mitigar a referida dificuldade foi proposto no artigo um método que, valendo-se de regras de redução, cria submodelos reduzidos que podem ser simulados. As regras de redução foram adaptadas para identificar padrões estruturais dentro do modelo que permitem remover submodelos para serem simulados separadamente pelo algoritmo. Ou seja, executa-se um método em tempo real que

identifica dinamicamente oportunidades de reduções na estrutura do modelo, fazendo com que o mesmo possa ser testado separadamente por meio dos submodelos. Essas reduções são baseadas na existência de reações rápidas e lentas e no software de simulação que será usado no modelo reduzido.

O método se mostrou eficiente, fazendo com que as simulações pudessem ser executadas mesmo em modelos com uma grande quantidade de elementos.

3.9 Uma Nova Abordagem para Redução de Sistemas em Redes de Petri Baseada em Teoria de Regiões

Contextualização:

A pesquisa desenvolvida por Shen et. al. (2013) cita que um modelo de Rede de Petri passa a ser ineficiente quando o grafo de alcançabilidade da rede se torna grande ao ponto de dificultar o processo de identificação ou simulação das características do modelo. Cita-se que muitos pesquisadores desenvolveram métodos (algoritmos para reduzir modelos de RP) e metodologias (regras e técnicas dedicadas a tipos específicos de RP como, por exemplo, as temporizadas).

O método proposto na pesquisa tem como foco mitigar o problema da grande profusão de elementos, porém, buscou-se considerar reduções amplas. Ainda segundo o autor, a maioria das regras de redução propostas por pesquisadores ao longo dos anos teve como objetivo reduzir mínimos fragmentos das redes e isso parece ineficiente para modelos muito grandes.

Objetivo:

Criar um método para reduzir modelos de Rede de Petri usando Teoria de Regiões como foco na estrutura completa do modelo, ou seja, um método de redução que não se baseia na aplicação de muitas regras de redução que atuam em micro características estruturais do modelo.

Contribuição:

Criação de um método de redução que atua na estrutura completa de uma RP usando Teoria de Regiões.

Regras de Redução:

Não são usadas regras de redução típicas (como a maioria das apresentadas neste capítulo). São atribuídos pesos às transições com base na relação entre elas e os lugares à elas conectados. Repetindo o processo para todas as transições do modelo identifica-se a transição mais conexa do modelo bem como a menos conexa e por meio da última é possível identificar lugares redundantes. Esses lugares redundantes são eliminados por meio de fusão com as transições com maior peso, reduzindo a estrutura completa do modelo.

A Figura 12 mostra um exemplo de como a Teoria de Regiões é aplicada para criar modelos reduzidos de RP. São identificados três estruturas que podem ser removidas e cria-se uma nova rede com os lugares m_1 , m_2 e m_3 sendo os equivalentes da compressão realizada. O lugar R_1 da nova rede foi reduzido por uma regra de redução de transições seriais e assim foi criada a rede apresentada à direita na figura.

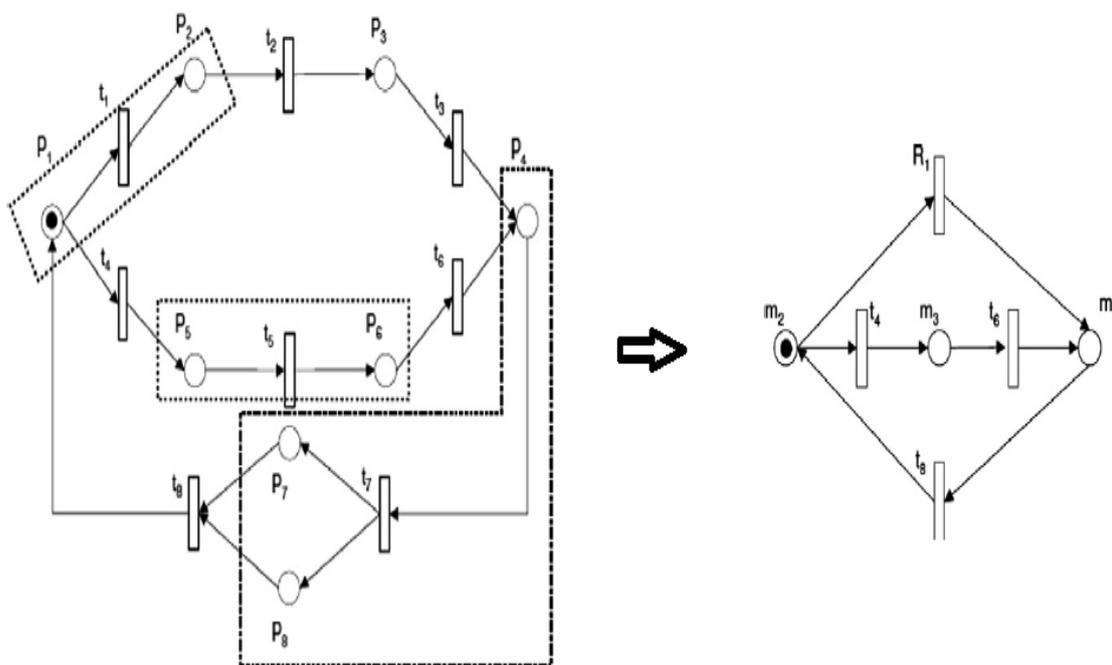


Figura 12. À esquerda um modelo de RP com três pontos de redução identificados e à direita a RP reduzida (SHEN et. al. 2013).

Validação:

Para validar a proposta o autor utiliza, além de modelos menores como o apresentado na Figura 12, um modelo de uma CAVE de realidade virtual modelada com RP. Para validar a estratégia são necessários quatro algoritmos. Primeiramente

um algoritmo para atribuir pesos às transições do modelo, depois um algoritmo para encontrar e marcar lugares que podem ser comprimidos, em terceiro lugar um algoritmo que comprime a matriz nos pontos marcados e por último um algoritmo que cria a matriz final, reduzida e comprimida.

Conclusão:

O trabalho apresenta uma metodologia que permite encontrar a transição com maior peso no modelo e os lugares redundantes. Com tais informações é possível comprimir uma RP com pesos antes de aplicar reduções na mesma, fazendo com que a rede final gerada seja ainda menor do que a gerada apenas com reduções.

A metodologia criada é apontada como um facilitador para análise de RP's, porém não é citado quais as propriedades comportamentais que são mantidas após a compressão do modelo.

O autor cita ainda que outros pesquisadores podem melhorar a metodologia e torna-la aplicável a outros tipos de Redes de Petri.

3.10 Considerações Finais

Após a apresentação de pesquisas relacionadas a modelagem de sistemas usando Redes de Petri e também pesquisas que focam na utilização de regras de redução em modelos com o objetivo de simplificar verificação de propriedades e simulação dos mesmos, pode-se citar algumas considerações sobre o tema.

Primeiramente, são observadas pesquisas que utilizam diferentes tipos de RP's para modelagem de sistemas. A escolha de qual tipo de RP utilizar deve levar em consideração diversas características do problema e do objetivo da modelagem, dessa forma, pode-se decidir qual tipo de Rede de Petri utilizar, podendo ou não optar por redes específicas (temporizadas, anotadas e coloridas, por exemplo).

Em segundo lugar, percebe-se que diversas pesquisas (começando em meados da década de 80 e surgindo até os dias de hoje) tratam da utilização de regras de redução de Redes de Petri com o objetivo geral de simplificar a análise e simulação de modelos preservando suas propriedades.

O foco de grande parte das pesquisas é propor novas regras de redução. Isso se deve ao fato de as Redes de Petri evoluírem com o tempo, são adicionadas novas características e restrições, criando assim vertentes das RP básicas apresentadas nas primeiras pesquisas encontradas na literatura. Juntamente com a evolução dos tipos de modelos que podem ser criados, também se mostra necessário o surgimento de novas regras de redução, pois a grande profusão de elementos nos modelos é um problema sempre abordado.

Os testes realizados em cada um dos métodos citados são baseados nos objetivos que a metodologia proposta pretende atingir. Pode-se buscar métodos que visam obter uma redução que seja eficiente do ponto de vista do tempo necessário para ser executado, como na pesquisa conduzida por Yang (2009).

Outra abordagem de validação utilizada no escopo de redução é relativa ao potencial de redução do método, ou seja, a quantidade de elementos que são retirados do modelo após a aplicação das reduções, essa abordagem é vista no trabalho proposto por Uzam (2004). Essa forma de validação é um dos focos da presente pesquisa.

No trabalho proposto por WU et. al. (2012) o potencial da metodologia apresentada é medido pela capacidade de desmembrar um modelo que representa ligações estruturais de compostos químicos, para que este possa ser simulado de maneira mais fácil.

Já Mikolajczak (2009) faz uso de uma abordagem baseada em análise de espaço de estados para validar a ferramenta proposta. O objetivo é provar que para grandes modelos de RP (nesse caso as Redes Coloridas) pode-se tornar complexo aplicar uma metodologia de avaliação de propriedades baseada em análise de espaço de estados, porém, utilizando a ferramenta de redução proposta pode-se reduzir a rede mantendo suas propriedades estruturais e simplificando a aplicação da referida metodologia.

Pode-se perceber que a utilização e criação de regras para redução de modelos de Redes de Petri é frequentemente abordada e, baseado nos resultados observados nas pesquisas, costuma obter bons resultados para os mais diversos objetivos. Porém, o contexto relativo aos trabalhos relacionados que se mostra importante para a presente pesquisa está relacionado principalmente com a criação de uma ferramenta que permita aplicar automaticamente regras de redução em modelos de RP.

Algumas das pesquisas mostradas tem por objetivo fazer uso de regras de redução para simplificar modelos de RP (desenvolvidos para descrever os mais diversos tipos de sistema) para posterior aplicação de metodologias de verificação de propriedades. Nessas pesquisas o objetivo normalmente é criar um método de aplicação dessas regras, o que normalmente envolve um estudo prévio da estrutura do modelo, para definir quais regras usar e em que sequência usa-las. Depois disso, cria-se um algoritmo que execute iterativamente o processo de aplicação usando a sequência estudada.

A presente pesquisa é também um método de redução para simplificar aplicação de verificações, porém, o objetivo é retirar do pesquisador a necessidade de estudar a sequência de regras a serem aplicadas e como aplicar. A partir da leitura do modelo em RP, um sistema composto por inúmeras regras e uma máquina de inferência baseada em Algoritmos Genéticos, realiza uma avaliação das regras e sua sequência de aplicação de forma a reduzir o referido modelo mantendo as propriedades estruturais (as provas matemáticas da manutenção de propriedades são apresentadas pelos autores que propuseram as regras).

A Tabela 2 sintetiza as principais informações sobre os trabalhos relacionados elencados no presente capítulo.

O Capítulo 3 apresenta as características gerais da proposta e alguns exemplos que tem por objetivo ajudar na compreensão geral da pesquisa desenvolvida.

Tabela 2. Comparativo de Pesquisas Baseadas Direta ou Indiretamente em Redução de Modelos de Redes de Petri.

Pesquisas/ Critérios	Redes de Petri para modelagem de:	Tipo de Rede de Petri utilizado.	Qual o objetivo da utilização de Regras de Redução na Proposta?	Regras de Redução Usadas	Contribuição
(SLOAN; BUY, 1996)	Programas paralelos em ADA.	Redes de Petri Temporizadas	Simplificar avaliação de propriedades de sistemas paralelos em linguagem ADA.	Fusão Serial, Pré-Fusão, Pós-Fusão, Fusão-Lateral, Lugares Redundantes, Lugares Paralelos Redundantes, Remoção de Lugares Finais Desnecessários, Remoção de Lugares Iniciais Vazios, Lugares Iniciais Marcados Sem Necessidade	Adaptação de nove regras de redução de redes de petri não-temporizadas para redes temporizadas
(UZAM, 2004)	Sistemas Flexíveis de Manufatura	Redes de Petri	Simplificar a aplicação de uma política de prevenção de <i>deadlocks</i> baseada em teoria de regiões.	<i>Fusion of Series Places</i> (FSP), <i>Fusion of Series Transitions</i> (FST), <i>Fusion of Parallel Places</i> (FPP), <i>Fusion of Parallel Transitions</i> (FPT), <i>Elimination of Self-loop Places</i> (ESP) e <i>Elimination of Self-loop Transitions</i> (EST)	Mostrar que a política de prevenção de <i>deadlocks</i> baseada em teoria de regiões pode ser melhor aplicada em modelos reduzidos de RP.
(LETTERS et al., 2006)	Programas Paralelos.	Redes de Petri	Tornar Regras de Redução mais genéricas para simplificar avaliação de propriedades de programas paralelos.	Pós-aglomeração e Pré-aglomeração	Generalização de regras para diminuir limitações de estrutura local, facilitando assim a verificação de propriedades em programas paralelos.

Tabela 2. (continuação) Comparativo de Pesquisas Baseadas Direta ou Indiretamente em Redução de Modelos de Redes de Petri.

(MIKOLAJC ZAK; SINGH, 2009)	Modelos para Problemas de Alocação de Recursos.	Redes de Petri Coloridas.	Criar ferramenta semiautomática para redução de RP's dentro do software TransCPN.	<i>Fusion of Series Places</i> (FSP), <i>Fusion of Series Transitions</i> (FST), <i>Fusion of Parallel Places</i> (FPP), <i>Fusion of Parallel Transitions</i> (FPT), <i>Elimination of Self-loop Places</i> (ESP) e <i>Elimination of Self-loop Transitions</i> (EST)	Toolbox para o software TransCPN que permite selecionar e aplicar regras de redução em modelos de RPC.
(XIA, 2010)	Modelos de gerência de banco de dados com múltiplas cópias.	Redes de Petri Temporizadas.	Propor quatro novas regras de redução para modelos de Redes de Petri Temporizadas.	As regras usadas são novas e os autores não dão nomes específicos para as mesmas.	Regras de redução para avaliação de propriedades comportamentais em Redes de Petri Temporizadas.
(QING-XIU et al., 2010)	Modelos de <i>Workflow</i> .	Redes de Petri	Propor um método de avaliação de propriedades estruturais de RP's para modelos de <i>workflow</i> .	Regras criadas pelos autores: Redução de estruturas sequenciais, redução de estruturas interativas, redução de estruturas adjacentes e redução de estruturas fechadas.	Método de avaliação de propriedades estruturais baseado em transformação de modelos e regras de redução.
(FENG et al., 2011)	Computação Orientada a Serviços (COS)	Redes de Petri Anotadas	Simplificar modelos de COS e identificar maneiras de substituição de serviços mais facilmente.	Adaptações das 6 regras propostas por Murata, (1989): <i>Fusion of Series Places</i> (FSP), <i>Fusion of Series Transitions</i> (FST), <i>Fusion of Parallel Places</i> (FPP), <i>Fusion of Parallel Transitions</i> (FPT), <i>Elimination of Self-loop Places</i> (ESP) e <i>Elimination of Self-loop Transitions</i> (EST).	Contribuição relacionada a geração de regras de redução adaptadas de Redes de Petri Anotadas.

Tabela 2. (continuação) Comparativo de Pesquisas Baseadas Direta ou Indiretamente em Redução de Modelos de Redes de Petri.

(WU et al., 2012)	Sistemas Químicos Celulares (SQC)	Redes de Petri com Pesos	Simplificar modelos de SQC de maneira automática, ou seja, sem necessidade de análise de modelos por especialistas.	Regras de redução específicas para o contexto de Sistemas Químicos Celulares. Reações rápidas ou reações lentas, identificadas primordialmente por meio de transições (que representam reações).	Algoritmo para automatizar redução de modelos SQC.
(SHEN et al., 2013)	CAVE de realidade virtual	Redes de Petri	Simplificar modelos de Rede de Petri com Pesos utilizando método baseado em Teoria de Regiões.	A pesquisa não foca em micro regras de redução como a maioria das apresentadas no capítulo. O autor baseia seu método de redução em identificação de lugares redundantes e aplicação de macro reduções.	Nova abordagem para redução de RP baseada em Teoria de Regiões. Pode ser usada antes da aplicação de regras de redução, tornando o modelo final ainda menor.

Capítulo 4

PROPOSTA DO TRABALHO

4.1 Visão Geral da Proposta

O trabalho desenvolvido trata da criação de uma ferramenta de redução automatizada capaz de receber um modelo de RP com entrada e aplicar uma sequência de regras de redução no mesmo (usando o mecanismo de processamento baseado em algoritmos genéticos), com o objetivo de diminuir a quantidade de elementos no modelo e mitigar o problema da grande profusão de elementos, permitindo assim aplicar metodologias de verificação de propriedades baseadas em simulação de modelos ou grafo de alcançabilidade com menor complexidade.

O modelo do sistema de manufatura a ser usado para estruturar indivíduo e operadores genéticos do Algoritmo Genético, bem como, para efetuar a validação da proposta é representado por uma Rede de Petri do tipo Lugar-Transição. As redes são de um tipo clássico não tendo características diferenciadas como, por exemplo, serem coloridas.

As regras de redução escolhidas para compor o algoritmo foram observadas em (Murata, 1989 apud Murata & Koh, 1980; Silva, 1985).

. Para a criação de um protótipo inicial de sistema foram utilizadas seis regras de redução apresentadas no referido trabalho, essas técnicas são: Fusão de Lugares em Série (do inglês, *Fusion of Series Places* (FSP)), Fusão de Transições em Série (do inglês, *Fusion of Series Transitions* (FST)), Fusão de Lugares Paralelos (do inglês, *Fusion of Parallel Places* (FPP)), Fusão de Transições Paralelas (do inglês, *Fusion of Parallel Transitions* (FPT)), Eliminação de Lugares com Loop (do inglês, *Elimination of Self-loop Places* (ESP)) e Eliminação de Transições com Loop (do inglês, *Elimination of Self-loop Transitions* (EST)).

A Figura 4, exibida no capítulo 2 do presente documento, ilustra as referidas regras de redução que foram usadas para construir a base de regras mostrada no presente capítulo.

As regras de redução apresentadas foram citadas, utilizadas e/ou adaptadas em muitas pesquisas presentes no capítulo 2 do presente documento. Além disso, metodologias para evitar ou detectar *deadlocks* como grafo de alcançabilidade, por exemplo, podem ser impossíveis de aplicar quando as redes não atendem a propriedade de delimitação. Sabendo disso, reduzir um modelo usando as seis regras citadas pode ser um passo anterior à aplicação desse tipo de metodologia, tendo em vista que a redução usando essas regras mantém inalterada a propriedade de delimitação do modelo e também as propriedades de vivacidade e segurança (MURATA, 1989).

Metodologias para detectar e verificar a existência de *deadlocks* em modelos de RP usando teoria de regiões também podem ser limitadas pela grande profusão de elementos, dessa forma a estratégia de usar regras de redução antes da metodologia é sempre considerada viável. Na pesquisa conduzida por Uzam, (2004), por exemplo, a referida estratégia é aplicada e as regras escolhidas são as mesmas seis observadas no trabalho de Murata, (1989). Isso acontece porque o foco da pesquisa está na prevenção de *deadlocks* e as regras mantém propriedades fundamentais para esse escopo de avaliação.

Portanto, decidiu-se pelo uso das seis regras apresentadas por Murata, (1989) devido as propriedades que são mantidas nos modelos e também por observar-se que muitas metodologias relacionadas a prevenção, detecção ou recuperação de *deadlocks* em modelos de RP necessitam da manutenção dessas propriedades quando optam por reduzir os modelos antes de aplicar a metodologia.

Para o escopo da presente pesquisa não busca-se usar regras de redução para outros tipos de RP's, por exemplo, as variantes apresentadas por (BERNARDINELLO; FIORELLA, 1992). Para que o método seja utilizado, basta que o modelo se baseie nas redes conhecidas como "lugar-transição".

A tarefa de aplicar regras de redução nos modelos é realizada por um Algoritmo Genético. Uma matriz de incidência (representando o modelo de Rede de Petri) é usada como entrada pelo Algoritmo Genético que aplica uma sequência de regras de redução no modelo e exibe como resultado a matriz de incidência que representa o modelo

reduzido, tendo ainda um relatório com quais regras foram aplicadas em cada linha/coluna do modelo.

A escolha pela utilização de uma heurística para executar a redução vem de sua capacidade de varrer um espaço de busca amplo (nesse caso referente à estrutura do modelo representada como matriz de incidência) em pouco tempo. Outro fator que insere complexidade no modelo (além do número de elementos) é que ordem de aplicação das regras pode gerar novas características estruturais que permitam aplicar mais regras, ou seja, nesse aspecto o problema pode ser considerado combinatório, tendo em vista que a combinação de regras aplicadas influencia no tamanho final da rede reduzida.

A sequência das regras aplicadas no indivíduo de melhor resultado vai depender exclusivamente da regra de avaliação (Fitness) criada para o AG, ou seja, se a regra de fitness tiver sido criada para privilegiar redes que tenham poucos “lugares”, uma determinada sequência pode ser mais apropriada, se por acaso o objetivo for obter uma rede com uma baixa quantidade de transições, outra sequência pode se mostrar mais apropriada, e isso vai refletir diretamente no melhor indivíduo encontrado ao final da execução. Durante os testes o número que será usado na função de avaliação será relacionado a quantidade de elementos, ou seja, não serão privilegiadas redes com menos lugares ou menos transições, mas sim, redes com menos elementos no total.

Independentemente da regra de avaliação utilizada, após a execução do algoritmo, terá sido varrida uma grande área de busca, criada pela combinação de regras de redução, e uma determinada sequência de regras terá sido proposta (no melhor indivíduo criado). O caráter genérico da proposta vem da possibilidade de se mudar o fitness para que a execução do algoritmo siga um determinado objetivo.

4.2 Padrões Para Representação das Regras de Redução

Para a criação da ferramenta de redução automatizada proposta na presente pesquisa, além de definir quais as regras de redução seriam usadas para diminuir o número de elementos no modelo, também foi necessário adaptar as regras de redução mostradas como um padrão gráfico da rede, como mostrado na Figura 4, para um padrão que pudesse ser verificado na matriz de incidência.

A seguir são mostrados os padrões criados para se identificar as seis regras de redução em representações matriciais de modelos de RP. Detalhes passo-a-passo para aplicação de cada regra podem ser observados na seção 3.3 “Exemplos de Utilização das Regras de Redução”.

A Figura 13 exibe o padrão definido para que se identifique que a regra Fusão de Lugares Seriais (FSP) seja aplicada a um indivíduo.

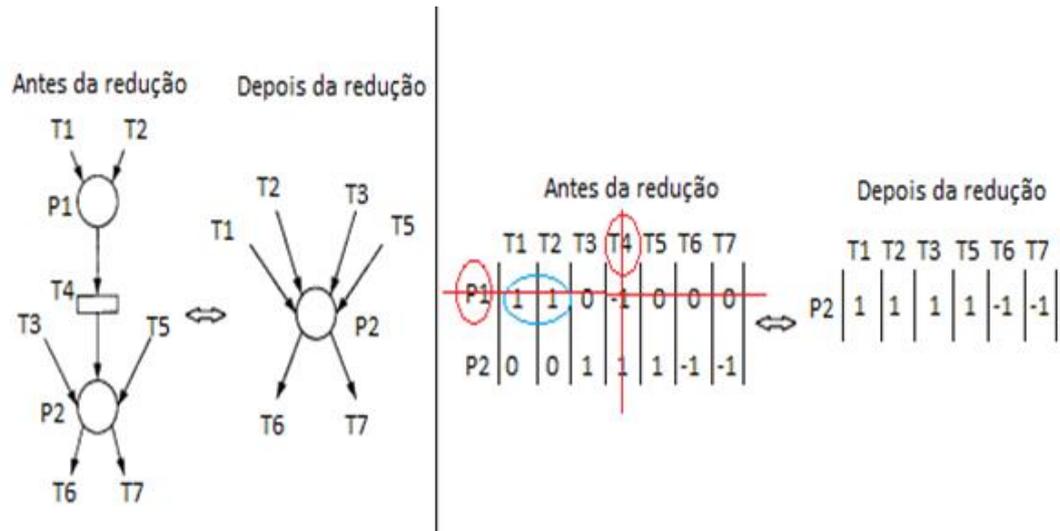


Figura 13. Representação do padrão que define a regra Fusão de Lugares Seriais (FSP) gráfica e matricialmente (representação usada pelo algoritmo) (MURATA, 1989).

A Figura 14 exibe o padrão definido para que se identifique a regra Fusão de Transições Seriais.

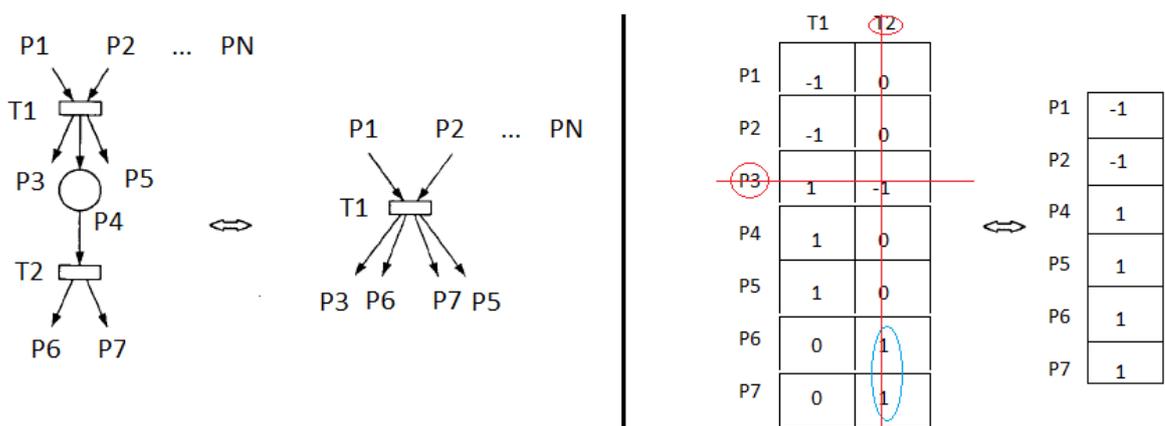


Figura 14. Representação do padrão que define a regra Fusão de Transições Seriais (FST) gráfica e matricialmente (representação usada pelo algoritmo) (MURATA, 1989).

A Figura 15 exibe o padrão definido para que se identifique a regra Fusão de Lugares Paralelos (FPP).

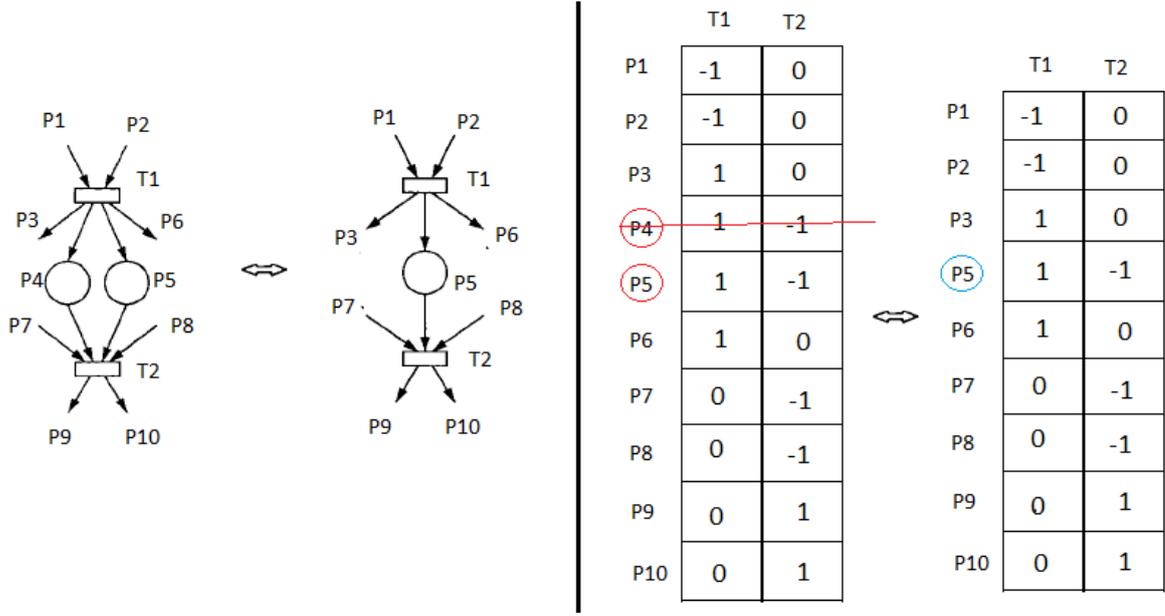


Figura 15. Representação do padrão que define a regra Fusão de Lugares Paralelos (FPP) gráfica e matricialmente (representação usada pelo algoritmo) (MURATA, 1989).

A Figura 16 exibe o padrão definido para que se identifique a regra Fusão de Transições Paralelas (FPT).

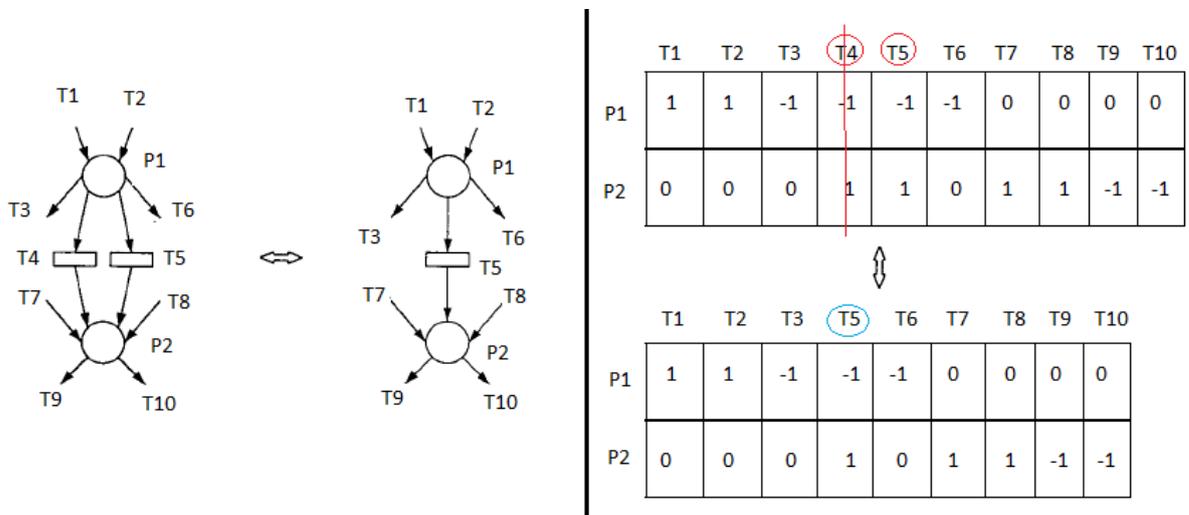


Figura 16. Representação do padrão que define a regra Fusão de Transições Paralelas (FPT) gráfica e matricialmente (representação usada pelo algoritmo) (MURATA, 1989).

A Figura 17 exibe o padrão definido para que se identifique a regra Eliminação de Lugares com Ciclo (ESP). Ciclos são representados no modelo matricial com o número dois associado a posição do par ordenado (lugar, transição) que tem o ciclo.

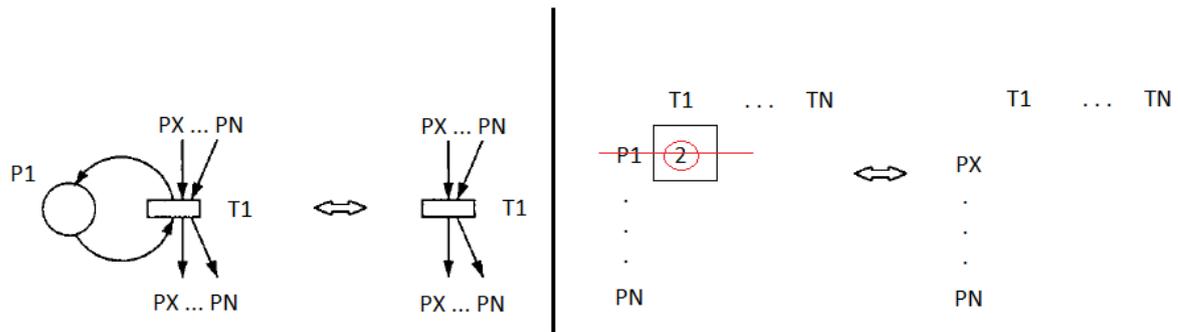


Figura 17. Representação do padrão que define a regra Eliminação de Lugares com Ciclo (ESP) gráfica e matricialmente (representação usada pelo algoritmo) (MURATA, 1989).

A Figura 18 exibe o padrão definido para que se identifique a regra Eliminação de Transições com Ciclo (EST). Ciclos são representados no modelo matricial com o número dois associado a posição do par ordenado (lugar, transição) que tem o ciclo.

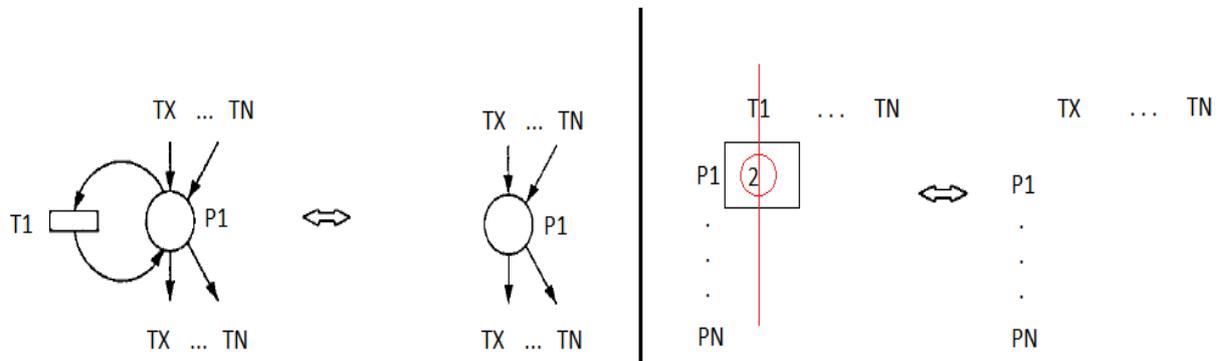


Figura 18. Representação do padrão que define a regra Eliminação de Transições com Ciclo (EST) gráfica e matricialmente (representação usada pelo algoritmo) (MURATA, 1989).

Para melhor compreensão das regras de redução, a próxima subseção é destinada a mostrar exemplos de aplicações das regras em um modelo de RP adaptado de Desrochers (1995).

4.3 Exemplos de Utilização das Regras de Redução

A demonstração das regras de redução escolhidas para pesquisa será realizada com base no modelo apresentado na Figura 19, sendo essa rede adaptada de Desrochers (1995).

É importante lembrar que as seis regras escolhidas para a pesquisa garantem a manutenção das propriedades de vivacidade, equidade e delimitação (MURATA, 1989).

Como observado na seção de trabalhos relacionados, essas regras são frequentemente usadas para reduzir modelos a fim de aplicar metodologias de verificação de *deadlock* e o algoritmo proposto na presente pesquisa pode facilitar o trabalho de pesquisadores que também tenham por objetivo testar metodologias para verificações de *deadlocks* em modelos de RP.

A Figura 20 mostra a rede da Figura 19 em formato de matriz, representação usada pelo algoritmo proposto na pesquisa, exibindo também o vetor de lugares marcados.

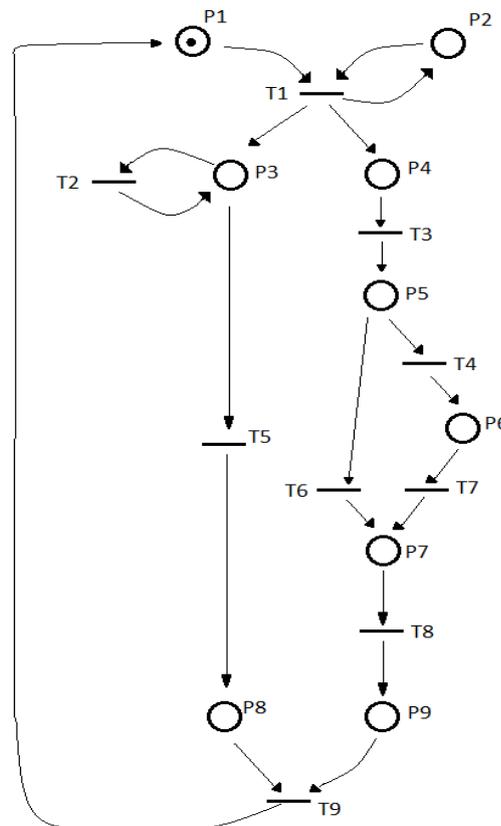


Figura 19. Modelo de Rede de Petri Usado para Demonstração de Reduções (DESROCHERS; AL-JAAR, 1995).

O processo de redução apresentado na presente seção tem por objetivo esclarecer como cada redução pode ser identificada e aplicada em modelos de RP Lugar—Transição. Busca-se ainda mostrar como esse processo se tornaria complexo se realizado manualmente por um pesquisador ou ainda, usando ferramentas para redução semiautomática como a proposta por Mikolajczak (2009).

Matriz de Incidência

	T1	T2	T3	T4	T5	T6	T7	T8	T9
P1	-1	0	0	0	0	0	0	0	1
P2	2	0	0	0	0	0	0	0	0
P3	1	2	0	0	-1	0	0	0	0
P4	1	0	-1	0	0	0	0	0	0
P5	0	0	1	-1	0	-1	0	0	0
P6	0	0	0	1	0	0	-1	0	0
P7	0	0	0	0	0	1	1	-1	0
P8	0	0	0	0	1	0	0	0	-1
P9	0	0	0	0	0	0	0	1	-1

Vetor de Lugares Marcados

P1

Figura 20. Modelo de Rede de Petri em Representação Matricial Usado para Demonstração de Reduções (DESROCHERS; AL-JAAR, 1995).

A primeira possibilidade de redução encontrada, ilustrada na Figura 21, é relacionada a regra Eliminação de Lugares com Ciclo (ESP). O Lugar P2 pode ser eliminado do modelo por meio dessa regra, pois tem apenas uma entrada e uma saída e ambas são relacionadas com a transição T1.

No caso da matriz mostrada na Figura 22 esse padrão pode ser identificado pois a linha que representa o lugar P2 tem apenas números dois, sendo esse valor usado para representar ciclos em Lugares ou Transições.

A segunda possibilidade de redução encontrada, ilustrada na Figura 21, é relacionada a regra Eliminação de Transições com Ciclo (EST). A Transição T2 pode ser eliminada do modelo por meio dessa regra, pois tem apenas uma entrada e uma saída e ambas são relacionadas com o Lugar P3.

No caso da matriz mostrada na Figura 22 esse padrão pode ser identificado pois a coluna que representa a Transição T2 tem apenas números dois, sendo esse valor usado para representar ciclos em Lugares ou Transições.

A terceira possibilidade de redução encontrada, ilustrada na Figura 21, é relacionada a regra Fusão de Transições Seriais (FST). A transição T7 tem somente uma entrada, vinda do Lugar P6. O lugar P6 por sua vez não tem marcações iniciais e tem somente uma entrada (vinda de T4) e uma saída (indo para T7). AS transições T4 e T7 são consideradas transições seriais e podem ser fundidas, sendo o Lugar P6 também fundido à transição T4 e promovendo a redução do modelo.

Na matriz apresentada na Figura 22 pode-se observar que T7 tem apenas uma entrada (um valor -1), vinda de P6, e que P6 por sua vez tem apenas uma entrada (um valor 1) e não possui outras saídas (valores -1) além de T7. Nesse caso, pode-se pegar todas as saídas de T7 (valores 1) e colocar na coluna da Transição T4, que da entrada no Lugar P6, que por sua vez é quem dá entrada em P7. Dessa forma, T7 e P6 são fundidos na Transição T4.

A quarta possibilidade de redução encontrada, ilustrada na Figura 21, é relacionada a regra Fusão de Lugares Seriais (FSP). O Lugar P5 e a Transição T3 podem ser eliminados do modelo, pois o lugar P5 não tem marcações iniciais, tem somente uma entrada, que vem da Transição T3, que por sua vez tem também somente uma entrada, vinda do Lugar P4. Os Lugares P4 e P5 são considerados seriais e podem ser fundidos, sendo que no processo de fusão a Transição T3 também é fundida ao Lugar P4.

Na matriz apresentada na Figura 22 pode-se observar que P5 tem apenas uma entrada (um valor 1), vinda de T3, e que T3 por sua vez tem apenas uma entrada (um valor -1) e não possui outras saídas (valores 1) além de P5. Nesse caso, pode-se pegar todas as saídas de P5 (valores -1) e colocar na linha do Lugar P3, que da entrada na transição T3, que por sua vez é quem dá entrada em P5. Dessa forma, P5 e T3 são fundidos no Lugar P4.

A Figura 21 mostra as possibilidades de redução citadas e também a estrutura da rede depois de aplicadas as reduções. A Figura 22 mostra a mesma situação da Figura 21, porém, com a representação matricial.

Pode-se perceber que a nova RP com estrutura modificada permite que novas regras de redução possam ser aplicadas ao modelo. Sabendo da possibilidade de redução que ainda existe, nas Figuras 23 e 24 mostra-se a rede e matriz geradas e exibe mais padrões de redução identificados na nova estrutura do modelo. Tais padrões serão detalhados a seguir.

As Transições T6 e T4 se encaixam na regra de redução Fusão de Transições Paralelas (FPT), como pode ser observado na Figura 23. Ambas as Transições citadas recebem entrada e dão saída para o mesmo Lugar, isso faz com que uma delas, nesse caso a transição com menor índice nas colunas da matriz (T4) seja a escolhida para eliminação. Esse padrão pode ser observado na representação matricial quando são identificadas duas colunas (representação de transições) que tem apenas um valor -1

(que representa uma entrada) e um valor 1, sendo que as entradas e saídas dessas duas colunas devem estar nas mesmas linhas (representação dos Lugares no modelo).

Seguindo o mesmo princípio da redução usada nos Lugares P4 e P5 e na Transição T3, mostrado na Figura 21, pode-se aplicar a regra FSP nos Lugares P3 e P8 e na Transição T5, e também nos Lugares P8 e P9 e na Transição T8. Essa redução é também ilustrada nas Figuras 23 e 24, mostrando respectivamente a representação gráfica e a representação matricial da RP.

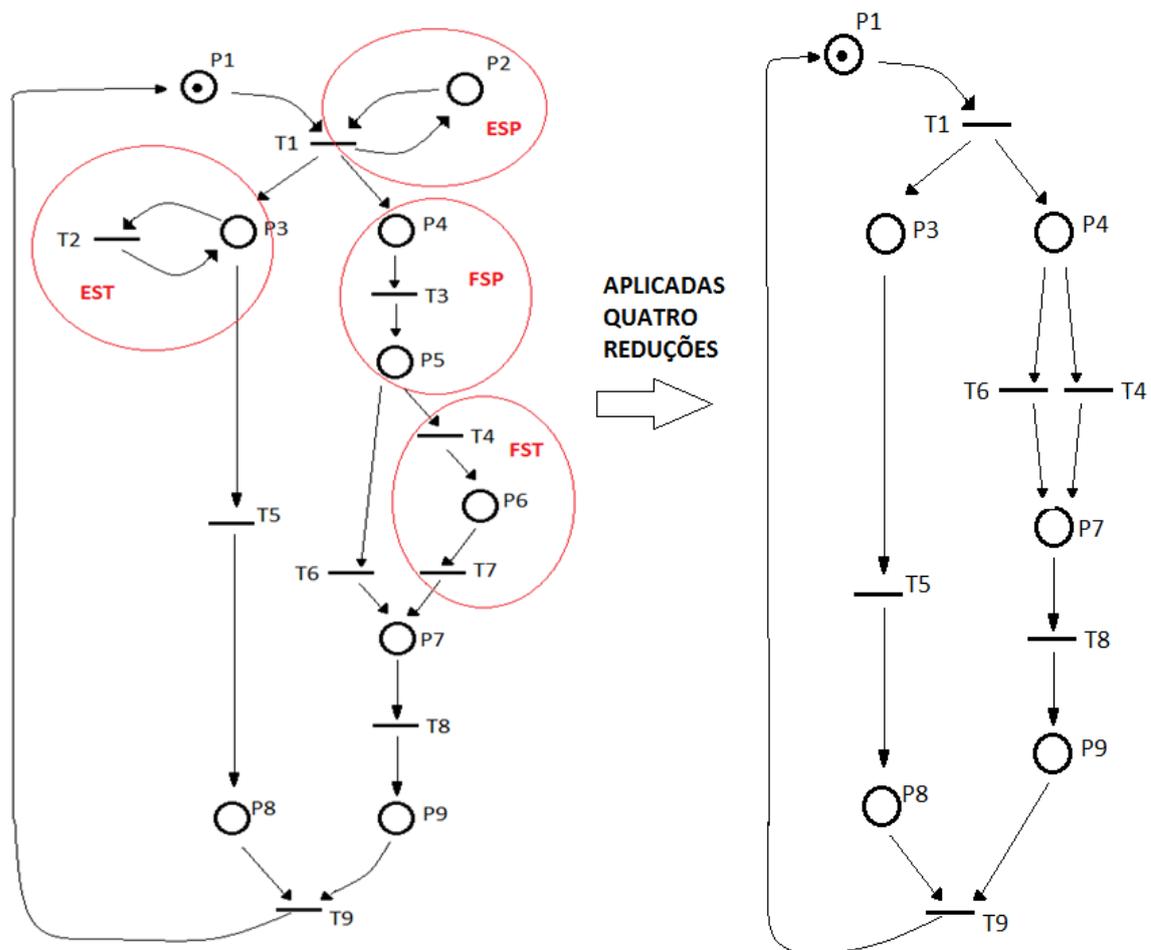


Figura 21. Identificados na Cor Vermelha os Padrões da Rede de Petri que Possibilitam Aplicação de Regras de Redução e à Direita a Representação da Rede já Reduzida.

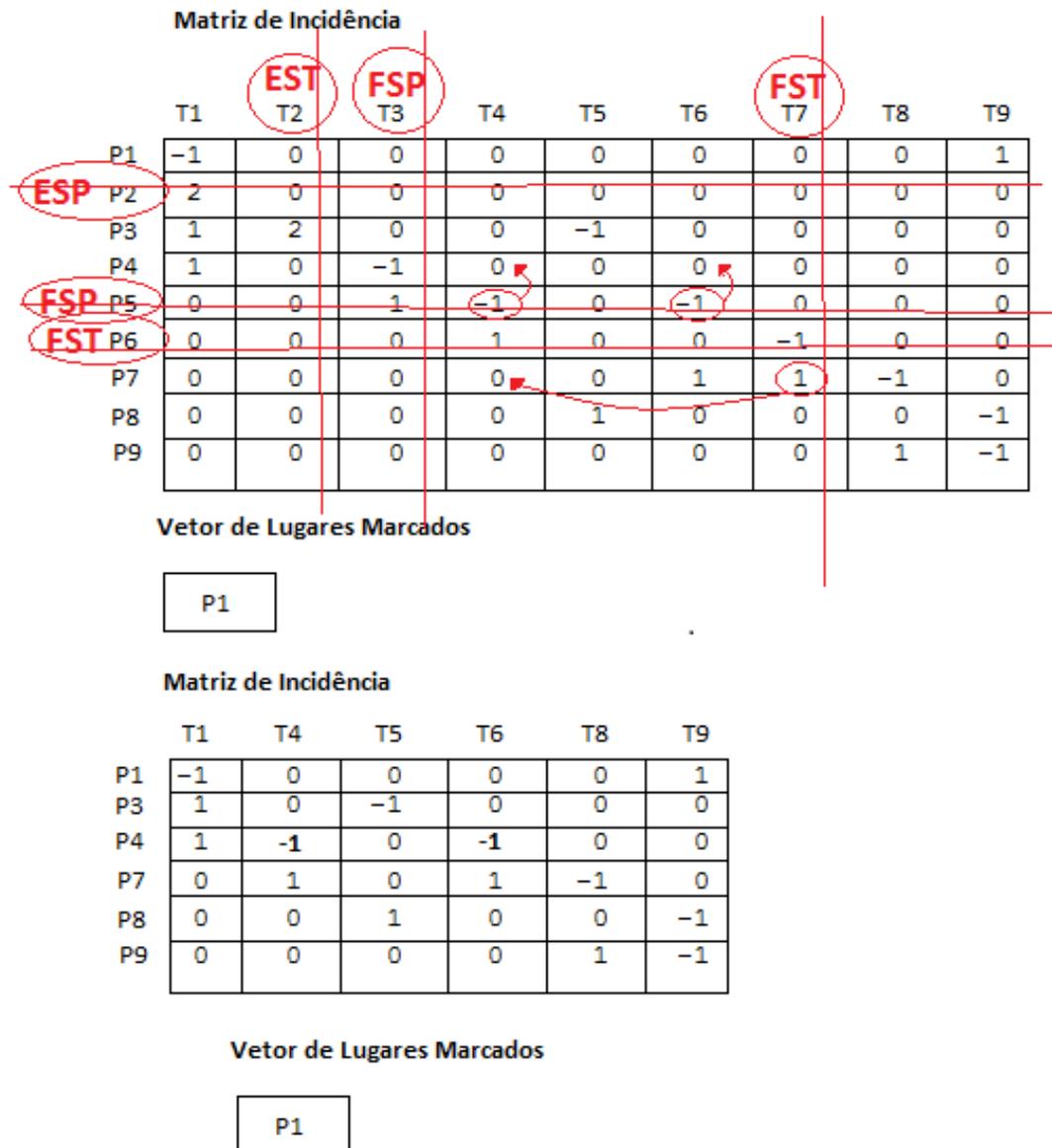


Figura 22. Identificados na Cor Vermelha os Padrões da Rede de Petri que Possibilitam Aplicação de Regras de Redução e na Parte Inferior a Matriz da Rede Reduzida.

Após a aplicação das sete regras citadas tem-se um modelo com quatro Lugares e três transições. Os próximos passos da redução podem ser verificados nas Figuras 25 e 26.

Seguindo o mesmo princípio de aplicação da regra Fusão de Lugares em Série (FSP), pode-se reduzir novamente o modelo usando os Lugares P4 e P7 e a transição T6. Essa redução causará a eliminação do Lugar P7 e da Transição T6 e cria a possibilidade estrutural necessária para aplicar a regra Fusão de Lugares Paralelos (FPP) nos Lugares P3 e P4. O lugar de menor índice é eliminado (P3) e apenas o Lugar de maior índice (P4) é mantido no modelo. A regra FPP pode ser identificada na matriz ao serem identificados dois Lugares com entradas (valores 1) vindas de uma mesma

Transição (nesse caso, T1) e saídas (valores -1) para uma mesma Transição (nesse caso, T9). Por fim, a regra Fusão de Transições Seriais (FST) pode ser aplicada nas Transições T1 e T9 e no Lugar P4. Após a aplicação dessa regra o modelo estará reduzido ao máximo, tendo apenas o Lugar P1 e a Transição T1 formando um ciclo.

Após a aplicação de todas as regras sabe-se que o modelo de RP, mostrado inicialmente na Figura 19, segue as propriedades de vivacidade, equidade e delimitação.

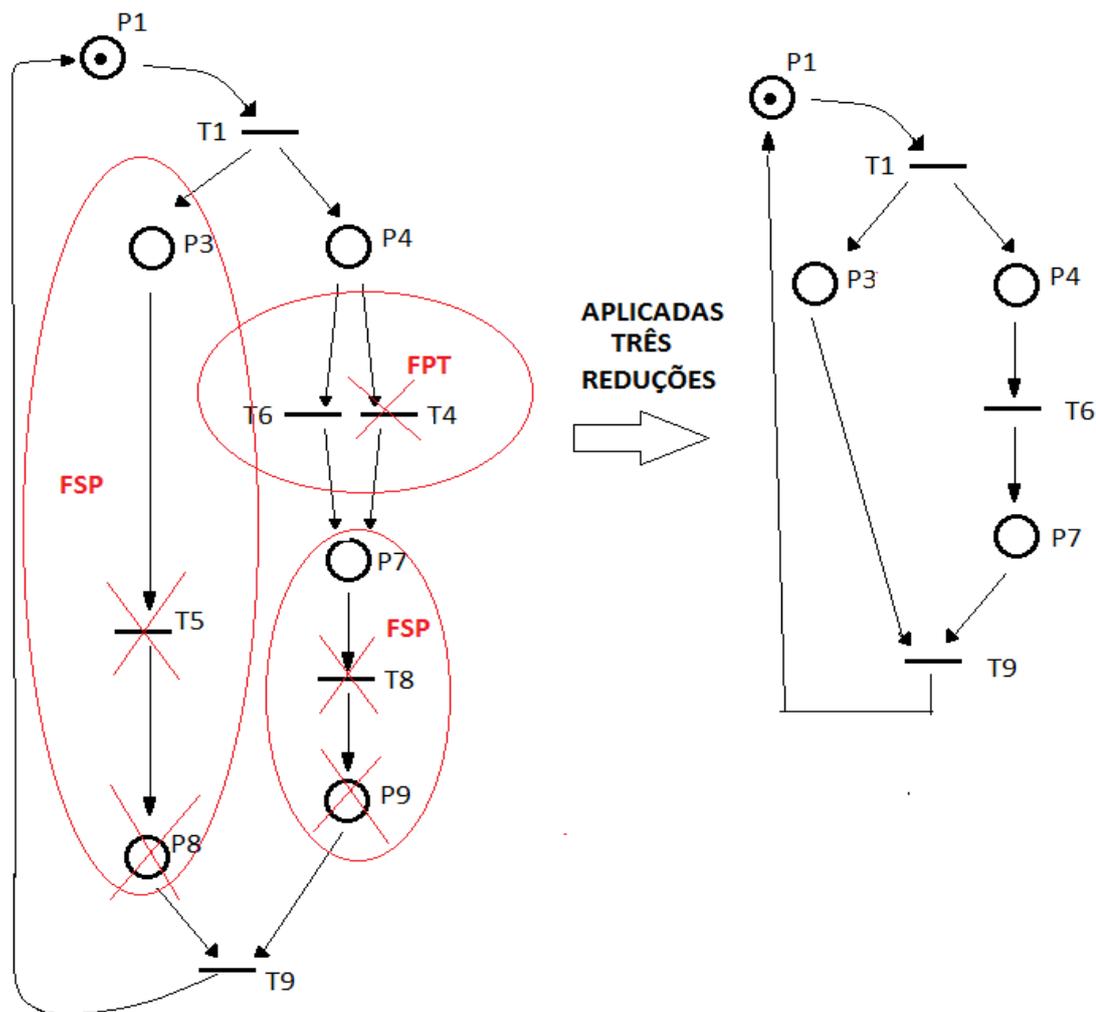


Figura 23. Identificados na Cor Vermelha os Padrões da Rede de Petri que Possibilitam Aplicação de Regras de Redução e à Direita a Representação da Rede já Reduzida.

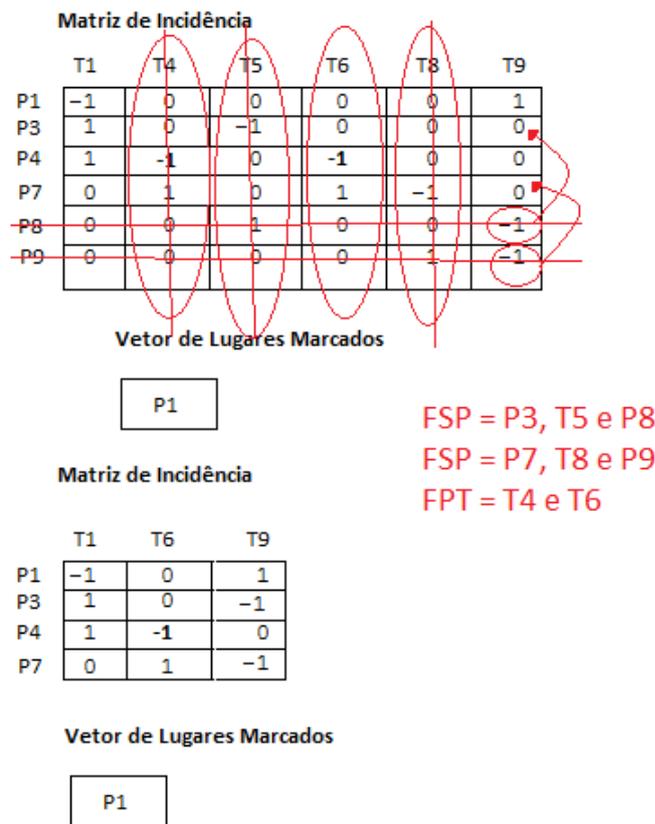


Figura 24. Identificados na Cor Vermelha os Padrões da Rede de Petri que Possibilitam Aplicação de Regras de Redução e na Parte Inferior a Matriz da Rede Reduzida.

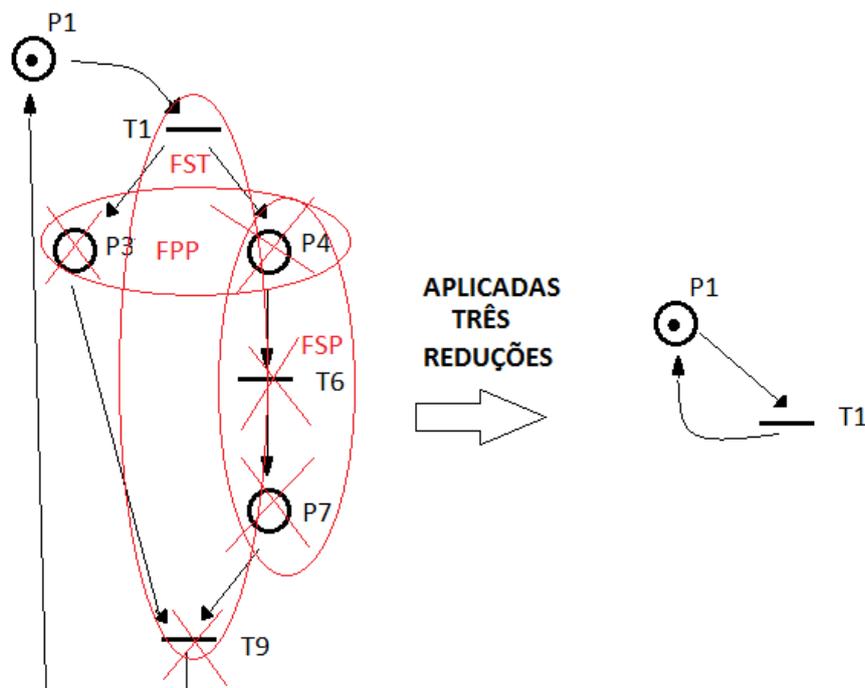


Figura 25. Identificados na Cor Vermelha os Padrões da Rede de Petri que Possibilitam Aplicação de Regras de Redução e à Direita a Representação da Rede Totalmente Reduzida.

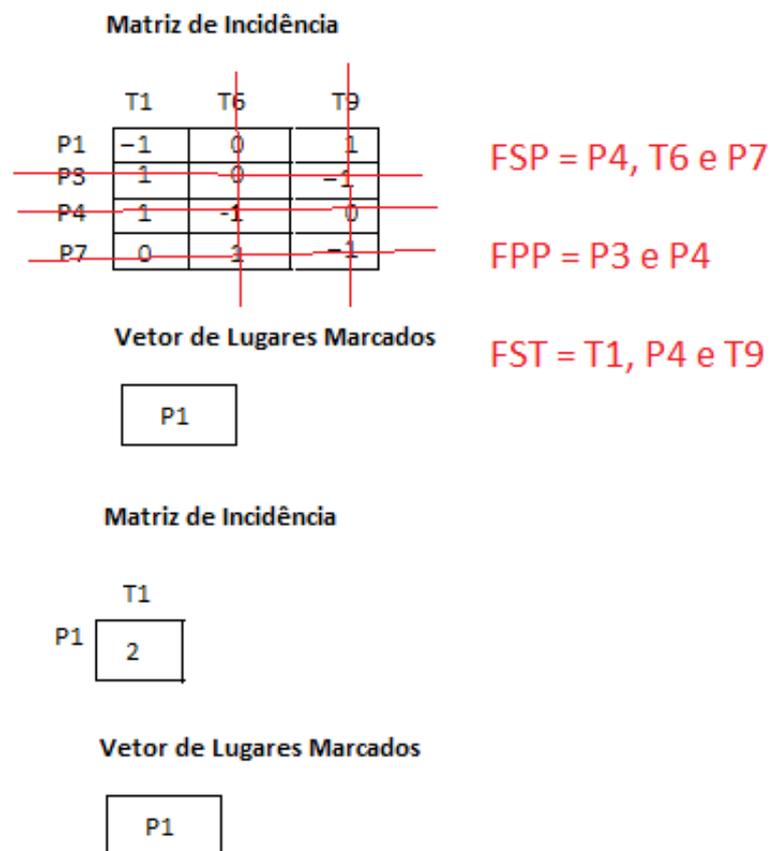


Figura 26. Identificados na Cor Vermelha os Padrões da Rede de Petri que Possibilitam Aplicação de Regras de Redução e na Parte Inferior a Matriz da Rede Totalmente Reduzida.

4.4 Algoritmo Genético (AG) para redução de Modelos de Redes de Petri (RP)

Informações gerais sobre a técnica de Algoritmos Genéticos bem como definições mais aprofundadas são apresentadas no capítulo 2 (Revisão Bibliográfica).

A Figura 27 mostra um esboço de funcionamento do Algoritmo Genético proposto para reduzir modelos em RP's.

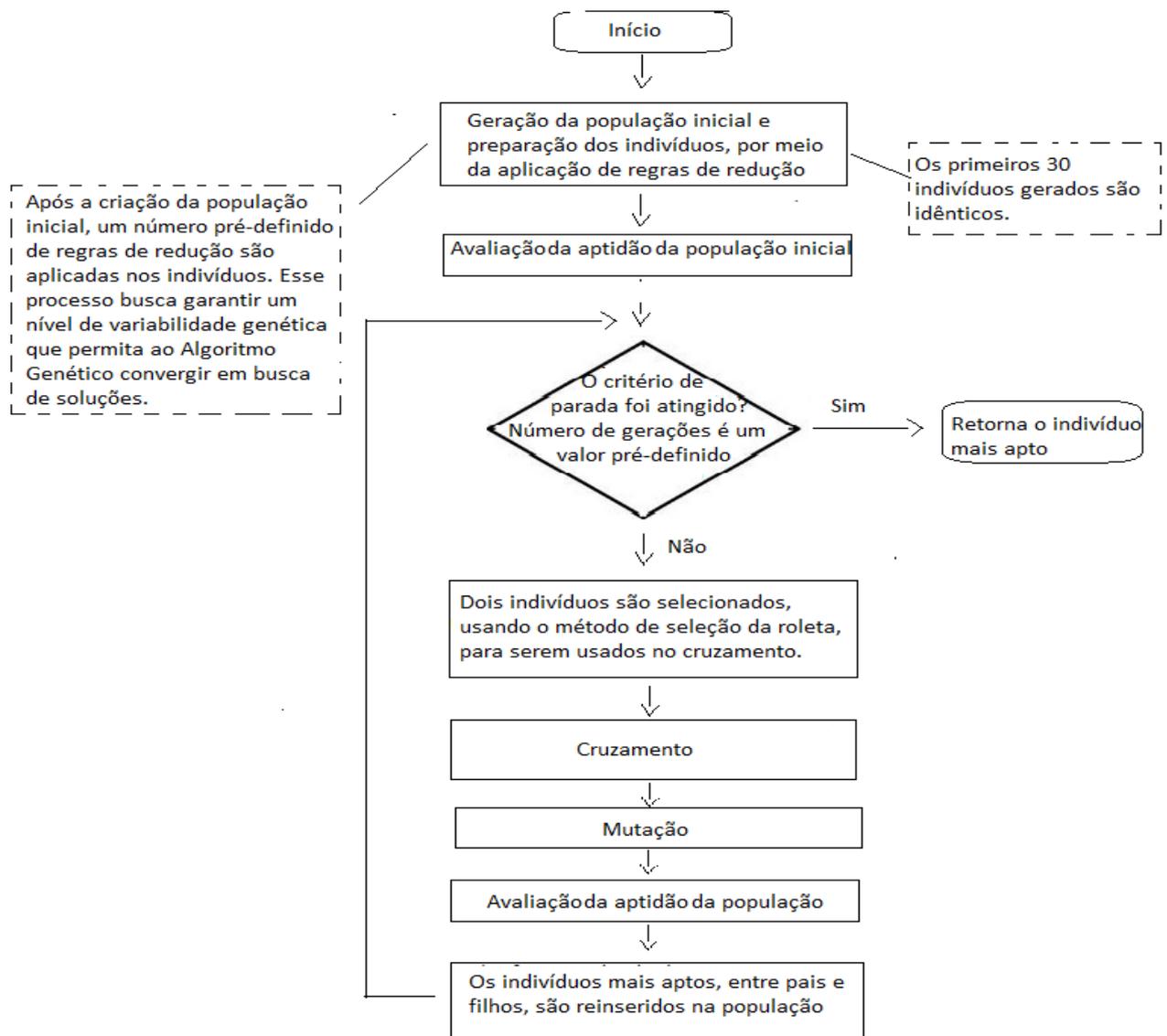


Figura 27. Esboço do fluxo seguido pelo Algoritmo Genético em busca da redução para uma RP.

A seguir serão descritos as principais características e detalhes de implementação da estratégia algorítmica desenvolvida na presente pesquisa.

4.4.1 Estrutura do Cromossomo

Cada cromossomo (ou indivíduo) que compõe a população do AG proposto consiste de duas matrizes e quatro vetores.

O principal elemento que constitui os indivíduos é a matriz que representa a RP, essa ideia segue o conceito exibido no capítulo 2 do presente trabalho, ou seja, é uma

representação alternativa de uma RP que faz uso de uma matriz de incidência para exibir lugares, transições e arcos de um determinado modelo. Os cruzamentos, mutações e cálculos de Fitness são todos executados com base nessa matriz.

Outro componente do indivíduo, representado também por uma matriz, é responsável por armazenar os lugares e transições removidos de cada indivíduo, ou seja, é uma matriz com N linhas (representando as N reduções que ocorreram) e duas colunas, responsáveis por exibir quais lugares e transições foram removidos a cada redução. Esse componente do indivíduo é utilizado para diversas verificações feitas durante a execução do Algoritmo Genético, porém, assim como os vetores que serão apresentados, esse elemento não é usado para operações genéticas, apenas a matriz que representa a RP passa por operações genéticas.

Alguns outros componentes são responsáveis por armazenar informações para avaliar outras restrições durante a execução do algoritmo como, por exemplo, um vetor, no qual anota-se o índice de cada redução aplicada no indivíduo. Com base nas informações desse vetor pode-se inferir que determinadas sequências de redução aplicadas em um indivíduo possam ser mais ou menos efetivas para reduzir o tamanho de uma rede.

Um vetor que armazena onde estão posicionadas as marcações iniciais também faz parte do indivíduo, para que seja evitada a exclusão de Lugares que tenham marcações iniciais.

Para fins de compreensão, os dois componentes fundamentais existentes na representação do indivíduo no AG são a matriz de incidência e o vetor de lugares marcados e é essa a representação usada no presente documento. A Figura 20 mostra um exemplo de como o indivíduo da Figura 19 é representado internamente no algoritmo. Ambas as referidas figuras são mostradas na seção 3.3, onde explica-se passo a passo como ocorre a aplicação de regras de redução em modelos de RP.

4.4.2 Base de Regras

O algoritmo contará com uma base de regras de redução que serão utilizadas para preparar a população inicial para cruzamento, ou seja, criar variabilidade genética dentro da população que evoluirá em busca da solução, e também serão utilizadas na mutação dos indivíduos durante a execução do algoritmo. Inicialmente serão utilizadas somente as seis regras citadas no começo do presente capítulo, porém, outras regras

que sejam de propósito geral ou específico também podem vir a ser consideradas para implementações futuras.

Para cada regra, existe um conjunto de propriedades que devem ser atendidas pela rede para que a redução possa ser aplicada e é nesse princípio que se baseiam as reduções na rede.

Em resumo, a base de regras abastece o algoritmo genético com padrões que são usados para criar variabilidade genética na população inicial do algoritmo e também para permitir a aplicação da mutação em sequências genéticas de indivíduos durante a execução do algoritmo.

4.4.3 População Inicial

A população inicial do AG é composta por 30 indivíduos idênticos, que representam, com a matriz de incidência, a das marcações iniciais e o vetor da rede de Petri para a qual busca-se a redução. A quantidade de indivíduos escolhida é considerada suficiente para que, ao preparar a população inicial para os cruzamentos, passo descrito a seguir, crie-se uma variabilidade genética que permita ao algoritmo convergir de forma satisfatória.

Para definir a quantidade de indivíduos necessários para a obtenção de soluções satisfatórias foram testadas algumas quantidades e chegou-se ao número 30 como um número que atende tanto entradas (RP's) com poucos ou muitos elementos e por isso optou-se por usar essa quantidade de indivíduos durante os testes.

Após a criação da população inicial, são aplicadas cinco regras de redução aleatoriamente em cada indivíduo, dessa forma cria-se uma variabilidade genética que permite cruzar os indivíduos fazendo a população convergir em direção a uma solução que apresente o menor fitness possível. O número de regras aplicadas na preparação dos indivíduos pode mudar, aumentando ou diminuindo, conforme o usuário desejar. Novamente, a definição do número de regras usadas inicialmente para criar variabilidade genética foi feita por meio de testes em redes com variadas quantidades de elementos.

4.4.4 Cruzamento

Na etapa de cruzamento, dois indivíduos são selecionados dentro da população, utilizando o método da roleta. Nesse método, atribui-se a cada indivíduo da população uma fatia da roleta que é percentualmente equivalente à avaliação (*fitness*) desse indivíduo.

Nessa etapa verifica-se a necessidade da existência do vetor referido na subseção 3.4.1, o vetor de regras aplicadas. Esse vetor tem a função de tornar possível saber quais Lugares, Transições e Arcos foram eliminados de um indivíduo por meio de cada redução. Assim, se seleciona-se os indivíduos 1 e 2 para cruzamento e o indivíduo 1 recebeu X reduções e o indivíduo 2 recebeu Y, pode-se escolher por meio do vetor de regras aplicadas reduções aplicadas ao indivíduo 1 e aplicar no indivíduo 2, assim como o inverso, promovendo cruzamento de informações entre os diferentes indivíduos do algoritmo.

Dois filhos são gerados, sendo que o Filho 1 recebe a estrutura do indivíduo 1, e uma parte da estrutura do indivíduo 2 (obtida com a aplicação de uma redução do indivíduo 2 na estrutura do indivíduo 1) e o Filho 2 recebe a estrutura do indivíduo 2 e uma parte da estrutura do indivíduo 1. Dessa forma, modificações na estrutura dos pais são misturadas nas gerações de filhos, fazendo com que os mesmos evoluam em busca de redes cada vez mais reduzidas. É importante esclarecer que o modelo de cruzamento usado faz com que regras aplicadas em um indivíduo sejam escolhidas para serem aplicadas exatamente no mesmo ponto na estrutura do outro indivíduo escolhido para cruzamento, ou seja, se a regra FSP foi aplicada no indivíduo 1 e causou a eliminação do Lugar P6 e da Transição T5, bem como dos arcos que ligavam esses elementos, essa mesma regra será escolhida e será aplicada nos mesmos Lugares, Transições e Arcos do indivíduo 2.

Os indivíduos são reinsertos na população utilizando a técnica que recebe o nome de Elitismo. Os indivíduos mais aptos dentre os presentes na população e os novos indivíduos gerados

Após o cruzamento, podem ainda ocorrer mutações nos filhos gerados.

4.4.5 Mutação

Após realizar o cruzamento dos indivíduos, existe uma possibilidade de 5% de ocorrer uma mutação genética em um dos novos indivíduos. A probabilidade de mutação de 5% foi utilizada durante os testes do algoritmo e apresentou um acréscimo importante no tempo de convergência do AG.

Caso ocorra a mutação, escolhe-se de maneira estocástica uma das regras de redução da base de regras e aplica-se esta redução em um ponto no indivíduo. Essa mutação segue as características comuns dos Algoritmos Genéticos e proporciona a variação genética dentro da população, fazendo com que a convergência seja melhorada.

4.4.6 Função de Avaliação (Fitness)

A regra de avaliação do AG consiste em uma contagem de Transições, Lugares e Arcos presentes na matriz de incidência que é parte da representação dos indivíduos.

Cada Lugar é uma linha da referida matriz, cada Transição é uma coluna da mesma e cada arco é um número diferente de zero na matriz de incidência. Para efeitos de avaliação é importante ter em mente se um modelo seria mais útil tendo menos transições, menos lugares ou menos arcos, porém, nos testes realizados buscou-se identificar indivíduos que tivessem a menor quantidade de elementos, sem dar prioridade à um dos três tipos de elementos que compõe os indivíduos.

Sabendo que lugares, transições ou arcos podem ser mais úteis para as reduções, implementou-se no sistema a possibilidade de o usuário mudar a função de avaliação como lhe convier. A função de avaliação padrão do sistema consiste na contagem do número de Transições, Lugares e Arcos do modelo, ou seja, a soma de desses três componentes é o que totaliza a aptidão do indivíduo no sistema.

A Equação 2 representa a função de avaliação previamente descrita. “NT” é a variável que representa na equação o número de transições existentes no modelo, “NL” representa o número de lugares e “NA” representa o número de arcos.

$$Fitness = NT + NL + NA$$

Equação 2. Exemplo de função de avaliação (Fitness).

4.4.7 Critério de Parada

O sistema implementado tem duas opções que podem ser usadas como critério de parada.

O usuário pode optar pela definição de um número de gerações durante as quais o algoritmo genético será executado. O padrão usado no testes foi de 750 gerações e apresentou bons resultados, porém, redes muito grandes podem exigir um aumento no número de gerações pela necessidade de aplicação de mais regras de redução.

Se não optar por definir um número de gerações o usuário pode também definir a porcentagem de redução no número de elementos que o modelo deve atender. Nesse caso, o sistema pode ser economizar gerações, caso encontre rapidamente uma solução com o *fitness* (avaliação) definido pelo usuário, ou ainda pode nunca ser capaz de encontrar uma solução boa o suficiente. Para evitar que o AG fique eternamente sendo executado foi implementado um mecanismo que define o número máximo de 1000 gerações mesmo quando o usuário opte por um critério de parada baseado em percentual do valor de avaliação (*fitness*).

4.5 Linguagem de Programação e Ferramentas Auxiliares

4.5.1 MATLAB

O sistema desenvolvido na presente pesquisa foi implementado usando a linguagem de programação MATLAB, versão 2012b.

O MATLAB é uma linguagem de programação que contempla uma extensa biblioteca de rotinas usadas frequentemente para criar aplicações matemáticas, de engenharia e para propósitos científicos de maneira geral (DRISCOLL, 2009).

Define-se MATLAB como uma linguagem de alto desempenho que integra em seu ambiente programação, visualização e computação. A linguagem permite a criação de aplicações, principalmente para resolução de problemas que são especificados em função de matrizes e vetores, em uma fração do tempo total que seria necessário em uma linguagem não interativa como o C, por exemplo. Em indústrias a linguagem é a

escolha para quando necessita-se de produtividade em pesquisa, desenvolvimento e análise de variados sistemas (DRISCOLL, 2009).

Outra importante característica da linguagem MATLAB são as *toolboxes*, que são aplicações que permitem solucionar problemas em áreas específicas. Com as *toolboxes* é possível aprender e aplicar tecnologias especializadas, fazendo com que o usuário possa aumentar sua produtividade de desenvolvimento e dando auxílio à aprendizagem. Existem diversas áreas para as quais *toolboxes* já foram desenvolvidas como, por exemplo, lógica difusa, simulação, redes neurais, sistemas de controle e, falando especificamente no contexto do presente trabalho, Redes de Petri (DRISCOLL, 2009).

Portanto, a escolha da referida linguagem de programação foi motivada pela existência de uma grande gama de funções matemáticas pré-programadas que auxiliam a execução de operações envolvendo matrizes. Tendo em vista que o sistema proposto trabalha com representações matriciais (e algumas representações auxiliares usando vetores) das Redes de Petri, considerou-se que a existência de funções que facilitam o processo de implementação era uma característica importante na escolha da linguagem tanto em relação à simplificação da programação quanto a velocidade de execução do algoritmo.

A possibilidade de criação de uma *toolbox* com as funções desenvolvidas para criação do algoritmo genético para redução de redes de petri foi também um fator decisivo na escolha da linguagem, tendo em vista que essa *toolbox* pode ajudar pesquisadores que necessitem de ferramenta ou conhecimento especializado em relação à redução de Redes de Petri de forma acessível e estruturada.

4.5.2 PIPE

A ferramenta *Platform Independent Petri Net Editor* (PIPE) também foi utilizada como apoio durante o desenvolvimento do projeto. A referida ferramenta é um projeto para edição e análise de Redes de Petri construída na linguagem de programação Java (CHUNG et. al. 2007).

A ferramenta é de código livre (do inglês, *open-source*) e contabiliza mais de sete mil *downloads* ao redor do mundo. O objetivo principal é facilitar o trabalho de projetistas e pesquisadores que trabalham diariamente com Redes de Petri, auxiliando na construção, análise e simulação das mesmas (CHUNG et. al. 2007).

A referida ferramenta auxiliou na criação das Redes de Petri e na geração da matriz de incidência correspondente. A possibilidade de simplificar a obtenção da matriz de incidência mostrou-se importante para a preparação das Redes usadas durante os testes e explicações.

Construir a matriz de incidência de uma Rede de Petri sem o auxílio de uma ferramenta especializada é uma tarefa complexa. Além do tempo necessário para calcular a matriz de incidência, deve-se levar em consideração que a não utilização de uma ferramenta de apoio pode causar erros na matriz de incidência final e prejudicar o desempenho dos testes e simulações de um modelo.

Como dito anteriormente, a ferramenta permite simulações dinâmicas das redes geradas e pode se mostrar importante para testar as redes reduzidas para verificação de propriedades.

Na Figura 28 pode ser observada a interface da ferramenta PIPE, pode-se visualizar como uma RP pode ser criada no software.

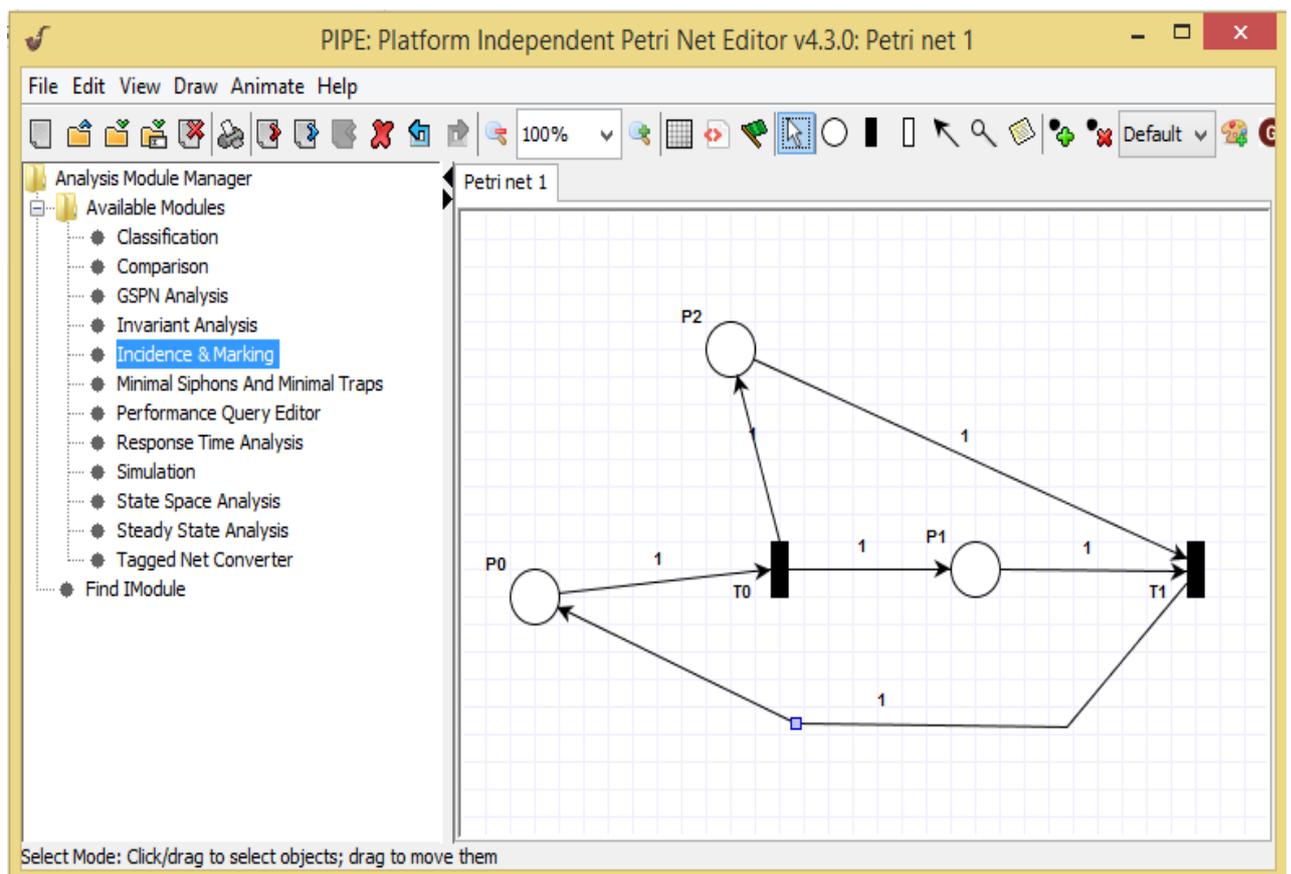


Figura 28. Exemplo de RP Visualizada na Ferramenta PIPE.

A ferramenta também permite a simulação e a análise usando espaço de estados (metodologia citada no capítulo 2 do presente documento) dos modelos, é possível visualizar as abas “*Simulation*” e “*State Space Analysis*” também na Figura 28.

A Figura 29 apresenta a funcionalidade fundamental para o propósito da presente pesquisa: geração da matriz de incidência do modelo, a aba “*Incidence & Marking*” é a que permite a obtenção da referida matriz.

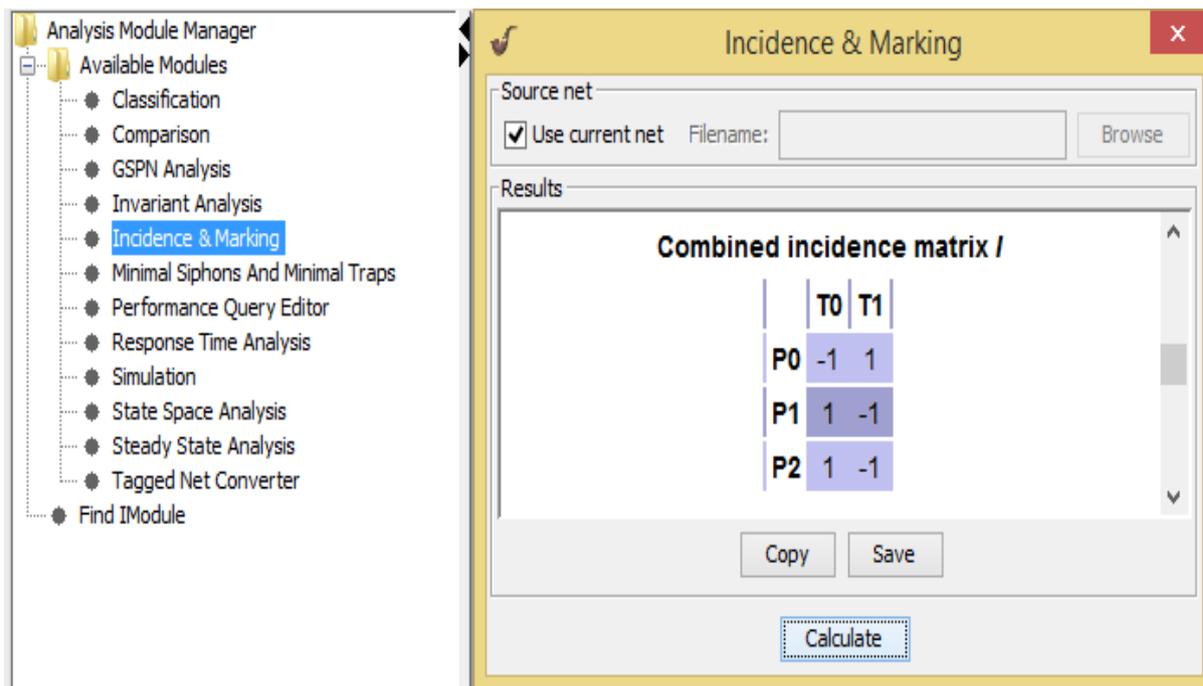


Figura 29. Matriz de Incidência Gerada pela Ferramenta PIPE para a RP Mostrada na Figura 28.

O capítulo a seguir tem por objetivo apresentar os testes realizados com a ferramenta Algoritmo Genético para Redução de Redes de Petri (AGRRP) e considerações relativas as contribuições e limitações observadas na ferramenta.

Capítulo 5

TESTES E RESULTADOS

A presente seção tem por objetivo validar a proposta da presente pesquisa por meio da aplicação do algoritmo em modelos de RP, começando com modelos com poucos elementos e finalizando com modelos com maior profusão de elementos.

Um modelo reduzido permite a aplicação de metodologias de verificação de propriedades, análises e simulação com menor complexidade por existirem menos elementos no modelo.

5.1 Redução de Modelo de RP de um Controlador de Tempo Real Usado no Processo de Montagem de Motor

O primeiro modelo usado na validação é mostrado na Figura 30. O referido modelo foi retirado de Desrochers & Al-Jaar, (1995) e representa um controlador de tempo real usado no processo de montagem dos pistões no bloco de motores.

A linha de montagem é constituída de dois robôs, duas partes usadas na montagem, motores, ferramentas, virabrequim, bastão de pistão e um sistema de visão. Com base nesses componentes é possível automatizar a linha de montagem e construir um modelo de Rede de Petri que represente essa automatização. Vale ressaltar que inicialmente é criada uma rede viva, reversível e segura para o sistema de manufatura. Após a construção do primeiro modelo e aplicando sequencialmente alguns teoremas o modelo foi melhorado até se tornar o modelo apresentado na Figura 30, modelo esse que tem uma representação mais refinada e, de acordo com o autor, reduz qualquer complexidade de interpretação das etapas do processo descrito.

O sistema abordado conta também com recursos compartilhados e devido a essa característica foi necessário, durante a fase de refinamento, que fossem inseridas algumas restrições para evitar que os recursos compartilhados fossem acessados de

maneira não desejada (ameaçando a integridade do modelo). Para isso, foram inseridas características estruturais da rede que representam o acesso a recursos compartilhados e garantem exclusão mútua sequencial e paralela.

De maneira geral o comportamento descrito no sistema envolve quatro etapas principais:

1. Um dos robôs move o virabrequim e alinha seu posicionamento.
2. O sistema de visão verifica o posicionamento.
3. O robô agora pega o bastão de pistão que interliga o virabrequim com o pistão e posiciona na estação de trabalho.
4. Por último, com o auxílio de ferramentas específicas o segundo robô acopla o pistão e as ligações com o virabrequim e insere no bloco de motor.

Os passos descritos são repetidos para montagem dos pistões e inserção nos motores que passam pela linha de montagem.

Portanto, essa descrição alto-nível sintetiza o processo que é demonstrado no sistema modelado por meio das redes de Petri que é utilizado para realizar o primeiro teste da ferramenta Algoritmo Genético Para Redução de Redes de Petri (AGRRP), proposta na presente pesquisa.

A Figura 31 apresenta a matriz de incidência gerada pela ferramenta PIPE para a referida rede e usada como entrada para o teste do algoritmo.

O modelo RP usado para o primeiro teste é constituído de dezesseis Lugares, oito Transições e trinta e dois Arcos, totalizando cinquenta e seis elementos. Os Lugares P0, P3, P4, P5, P10, P12, P14 e P15 estão no vetor de lugares com marcações iniciais.

A entrada de dados para a execução do algoritmo leva em consideração a matriz de incidência, o vetor de lugares marcados e os operadores genéticos definidos pelo usuário, nesse caso, os testes realizados usaram os seguintes parâmetros genéticos: população com 30 indivíduos, taxa de cruzamento de 30%, taxa de mutação de 5% e 1000 gerações.

A função de avaliação do indivíduo inicial é 56, ou seja, a soma de todos os elementos existentes na rede inicial.

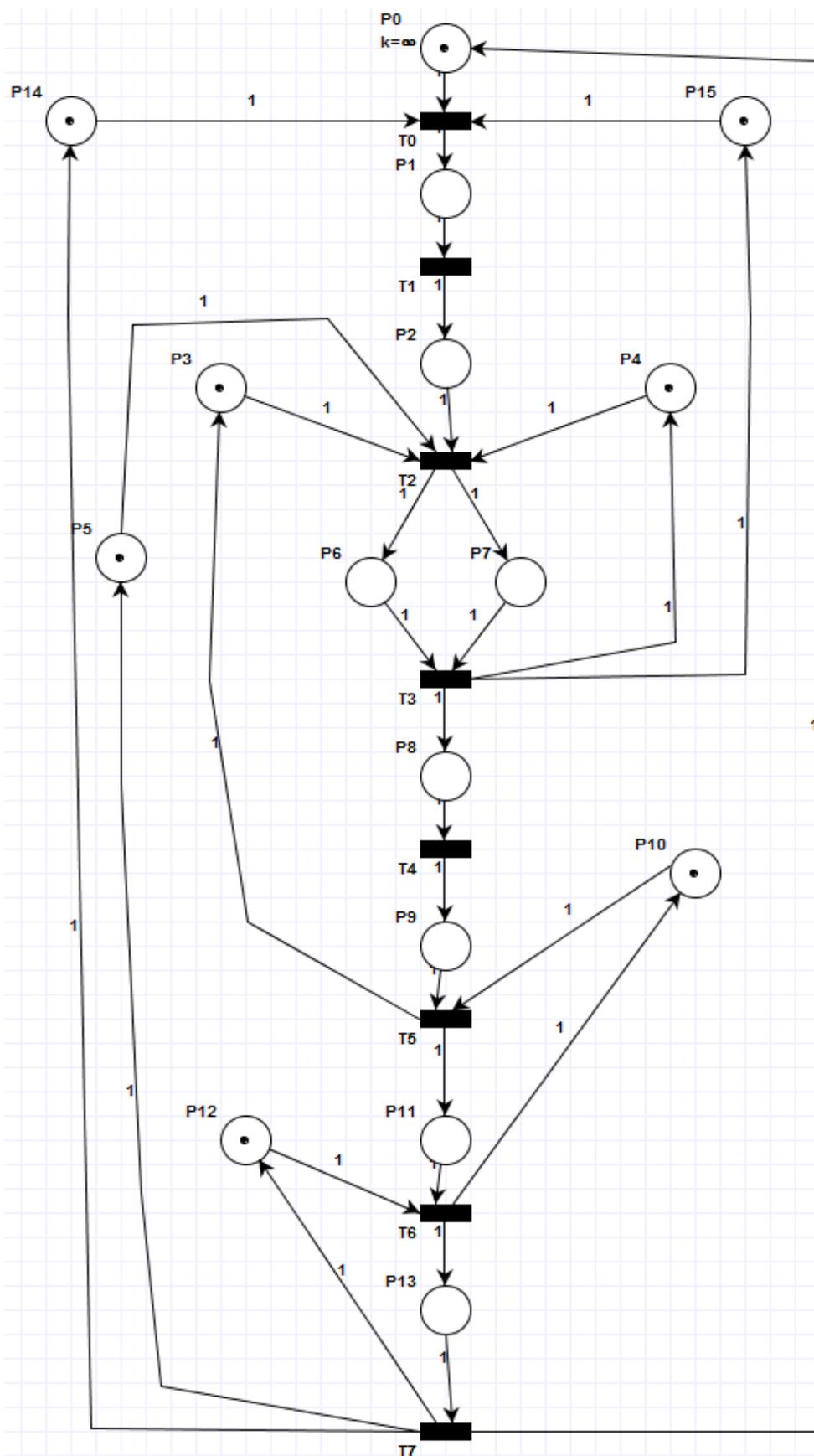


Figura 30. Modelo de Controlador Usado na Montagem de Motor (DESROCHERS; AL-JAAR, 1995).

Source net

Use current net Filename:

Results

Combined incidence matrix I

	T0	T1	T2	T3	T4	T5	T6	T7
P0	-1	0	0	0	0	0	0	1
P1	1	-1	0	0	0	0	0	0
P2	0	1	-1	0	0	0	0	0
P3	0	0	-1	0	0	1	0	0
P4	0	0	-1	1	0	0	0	0
P5	0	0	-1	0	0	0	0	1
P6	0	0	1	-1	0	0	0	0
P7	0	0	1	-1	0	0	0	0
P8	0	0	0	1	-1	0	0	0
P9	0	0	0	0	1	-1	0	0
P10	0	0	0	0	0	-1	1	0
P11	0	0	0	0	0	1	-1	0
P12	0	0	0	0	0	0	-1	1
P13	0	0	0	0	0	0	1	-1
P14	-1	0	0	0	0	0	0	1
P15	-1	0	0	1	0	0	0	0

Figura 31. Matriz de Incidência do Controlador Usado na Montagem de Motor.

Após a execução do teste para a rede mostrada na Figura 30 obteve-se uma nova rede com onze lugares, quatro transições e dezoito arcos. A função de avaliação inicial da rede era de cinquenta e seis e foi reduzida para trinta e sete, ou seja, dezenove elementos do modelo foram suprimidos, tornando a rede mais compacta e garantindo as propriedades de vivacidade, equidade e delimitação.

A Figura 32 mostra a matriz de incidência da rede reduzida.

```
-----
>> genetic_algorithm(teste_1, 30, [1 4 5 6 11 13 15 16], 5, 5, 1000);
-----
```

Tempo de execução do AG: 0.2711396233s

Melhor fitness inicial: 56

Melhor fitness final: 37

Redução percentual: 33.93

Regras aplicadas	Lugares retirados	Transições retiradas
fst	2	2
fst	9	5
fst	14	8
fpp	7	0
fst	8	4

Melhor rede resultante

-1	0	0	1
1	-1	0	0
0	-1	1	0
0	2	0	0
0	-1	0	1
0	1	-1	0
0	0	-1	1
0	0	1	-1
0	0	0	2
-1	0	0	1
-1	1	0	0

Figura 32. Matriz de Incidência do Melhor Indivíduo e Informações Gerais do Primeiro Teste.

Na Figura 32 pode-se observar uma lista que exibe os Lugares e Transições retirados do modelo e as respectivas regras de redução aplicadas pelo algoritmo genético no indivíduo. Os vetores internos “*removed_places_transitions*” e “*applied_rules*” permitem que a ferramenta disponibilize informações quanto as retiradas de elementos e as regras aplicadas.

O percentual de redução de elementos no primeiro teste realizado foi de 33,93% e o tempo de execução foi de aproximadamente 0,27 segundos.

É importante lembrar que a redução percentual de elementos para o primeiro teste foi pequena pois o algoritmo limita-se a aplicar regras somente em estruturas que não envolvam Lugares marcados, portanto, devido à existência de oito lugares marcados no modelo inicial, o potencial de redução fica reduzido.

5.2 Redução de Modelo de RP de um Sistema Flexível de Manufatura

O segundo modelo usado para testes do AGRRP é uma representação de um Sistema Flexível de Manufatura (SFM) retirado e adaptado de Desrochers (1995). A Figura 33 apresenta a ilustração (criada na ferramenta PIPE) do referido modelo e a Figura 34 apresenta a matriz de incidência gerada para o mesmo.

O SFM descrito nessa representação é o mesmo modelo utilizado na seção 4.5, porém, essa representação é uma síntese do modelo original descrita pelo autor como uma macro representação do modelo final.

O ambiente modelado é composto por seis estações de trabalho: um torno, uma máquina de corte, estação de transporte, e duas estações robóticas. São produzidos dois tipos de elementos, uma peça cilíndrica e um bloco de metal com um furo no centro. A peça final produzida após a passagem em todos os ambientes do sistema é o bloco ligado à peça cilíndrica formando uma peça.

O processo descrito entre os Lugares P4 e P6 demonstra a movimentação das peças cilíndricas até seu processamento em P8 (processamento no torno). O processo descrito entre P5, e P7 descreve a movimentação dos blocos furados até seu processamento em P9 (máquina de corte). Após a preparação das peças os blocos entre P10 e T15 e P11 e T16 levam as peças para as estações P14 e P15 onde são montadas as peças finais do sistema, usando a peça cilíndrica e o bloco furado.

O modelo RP usado no segundo teste é constituído de dezessete Lugares, dezesseis Transições e quarenta e seis Arcos, totalizando setenta e nove elementos. Os Lugares P0, P1, P2, P14, P16 estão no vetor de lugares com marcações iniciais.

A entrada de dados para a execução do algoritmo leva em consideração a matriz de incidência, o vetor de lugares marcados e os operadores genéticos definidos pelo usuário, nesse caso, os testes realizados usaram os seguintes parâmetros genéticos: população com 30 indivíduos, taxa de cruzamento de 30%, taxa de mutação de 5% e 1000 gerações.

A função de avaliação do indivíduo inicial é oitenta e um, ou seja, a soma de todos os elementos existentes na rede inicial.

A Figura 35 exibe as informações geradas após a execução do teste. A rede usada para o teste possuía oitenta e um elementos, função de avaliação inicial, e foi reduzida para quarenta e nove elementos, o que totaliza uma redução de 39,51%.

Ao todo foram aplicadas oito regras de redução e retirados trinta e dois elementos do modelo. O tempo de execução do algoritmo foi de 0,29 segundos.

O modelo usado para o segundo teste apresenta uma boa quantidade de elementos e muitas ligações sendo representadas, porém, a diferença de tempo de execução do primeiro e segundo testes foi consideravelmente pequena.

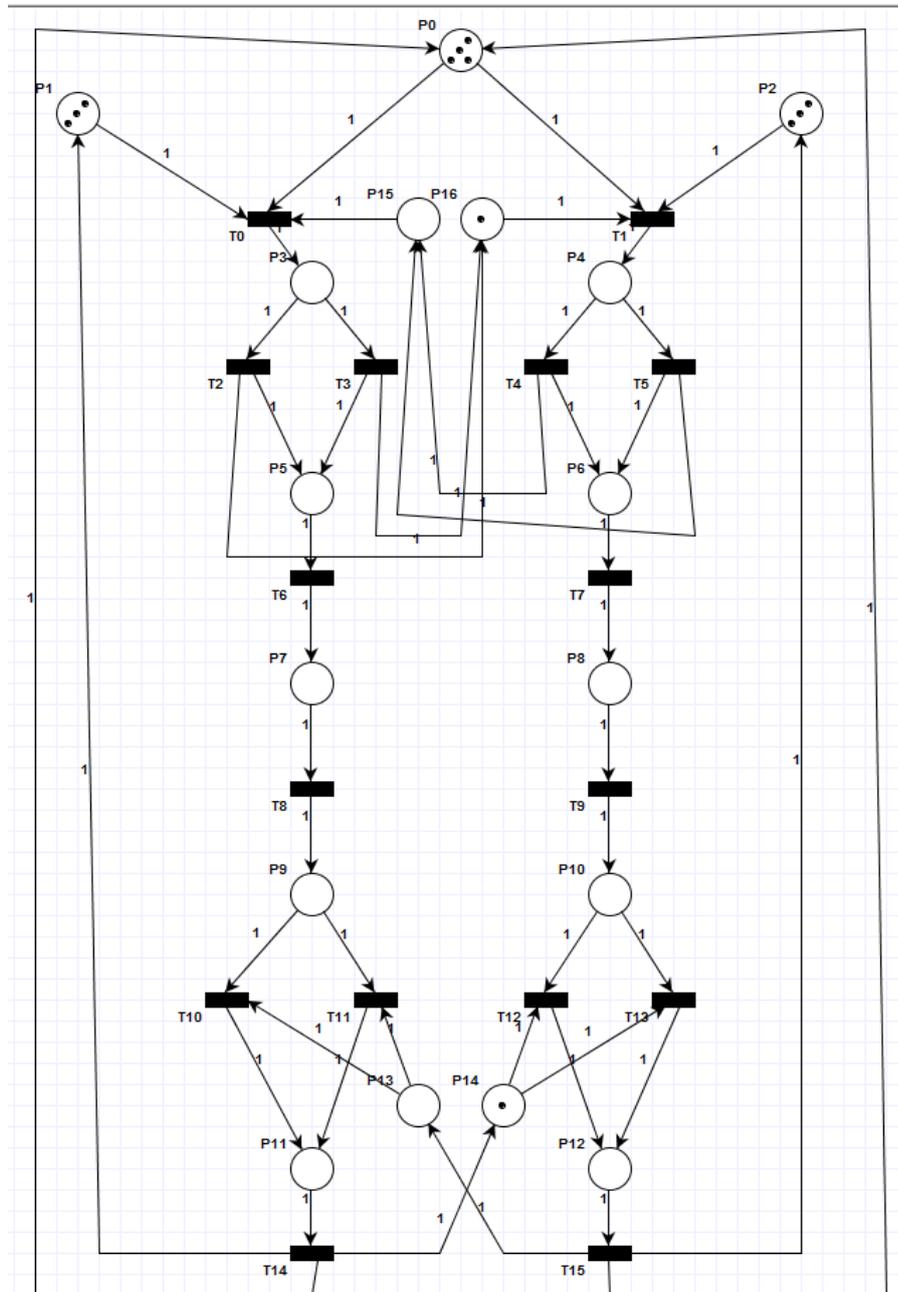


Figura 33. Modelo de Sistema Flexível de Manufatura (DESROCHERS; AL-JAAR, 1995).

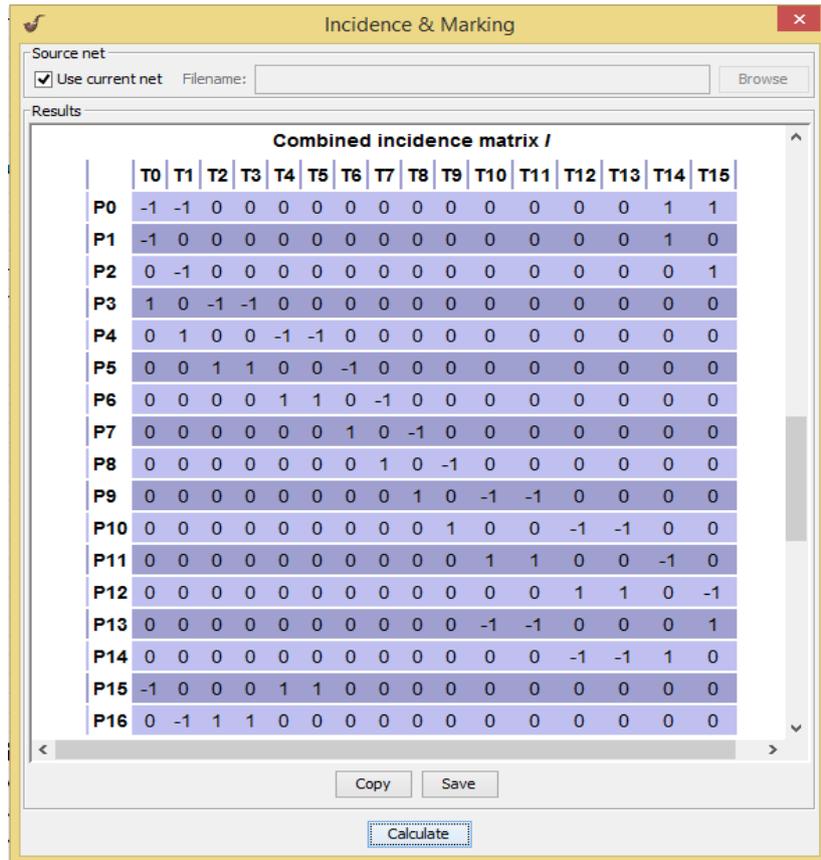


Figura 34. Matriz de Incidência para Modelo de Sistema Flexível de Manufatura.

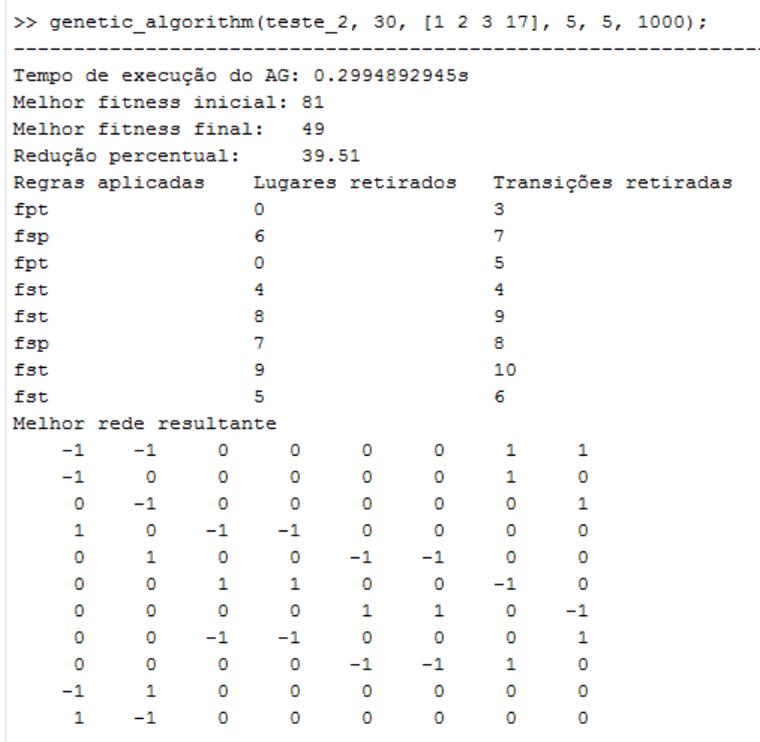


Figura 35. Matriz de Incidência do Melhor Indivíduo e Informações Gerais do SegundoTeste.

5.3 Redução de Modelo de RP Usado como Demonstração de Teoria de Síntese

O terceiro modelo usado para testes do AGRRP é uma representação de um Sistema de Manufatura arbitrário retirado e adaptado de Desrochers & Al-Jaar (1995). O referido modelo é construído pelos autores para demonstrar a construção de um modelo que mantém as propriedades de vivacidade e delimitação, criando um sistema complexo a partir de um modelo inicial mais simples, usando um processo de refinamento passo-a-passo. O processo de síntese é, em teoria, o inverso do que ocorre no processo de redução de modelos, expande-se um sistema inicialmente simples para contemplar características de representação desejadas, porém, garante-se também que a cada passo do refinamento as propriedades do sistema que está sendo expandido sejam mantidas.

Esse sistema foi considerado viável para teste da ferramenta devido a maneira como foi construído. Sabendo que o modelo foi construído com o intuito de manter propriedades de delimitação e vivacidade e sabendo também que as regras de redução usadas na ferramenta AGRRP também mantêm tais propriedades, pode-se verificar que o modelo é passível de ser reduzido ao máximo, voltando ao início do processo de síntese.

A Figura 34 apresenta a ilustração (criada na ferramenta PIPE) do referido modelo e a Figura 35 apresenta a matriz de incidência gerada para o mesmo. Esse mesmo modelo foi utilizado para demonstração da proposta no Capítulo 3 do presente documento.

O modelo RP usado no terceiro teste é constituído de nove Lugares, nove Transições e vinte e dois Arcos, totalizando quarenta elementos. O Lugar P1 é o único no vetor de lugares com marcações iniciais.

A entrada de dados para a execução do algoritmo leva em consideração a matriz de incidência, o vetor de lugares marcados e os operadores genéticos definidos pelo usuário, nesse caso, os testes realizados usaram os seguintes parâmetros genéticos: população com 30 indivíduos, taxa de cruzamento de 30%, taxa de mutação de 5% e 1000 gerações.

A Figura 36 mostra o resultado da redução. A função de avaliação do indivíduo inicial é quarenta, ou seja, a soma de todos os elementos existentes na rede inicial. A

rede tem apenas o primeiro Lugar marcado e foi reduzida com a aplicação de dez regras de redução. Ao final da redução tem-se apenas o Lugar 1 e a Transição 1, sendo gerado um ciclo entre os dois. A redução foi de 90% e a rede final é constituída de apenas quatro elementos. O tempo de execução foi de 0,37 segundos, aproximadamente.

O presente teste é bastante importante, pois com ele é possível verificar que o algoritmo automatiza a redução dos modelos e obtém o mesmo resultado que seria obtido caso a redução fosse realizada manualmente (como demonstrado no Capítulo 3 do presente documento) ou com o auxílio de uma ferramenta semiautomática como a proposta em (MIKOLAJCZAK; SINGH, 2009). A diferença é que o método automatizado obtém a rede reduzida de maneira mais rápida, apenas 0,37 segundos no presente caso, e permite ainda que dúvidas quanto a integridade da aplicação de reduções sejam sanadas por meio da exibição das regras aplicadas juntamente com os Lugares e Transições removidos por cada aplicação.

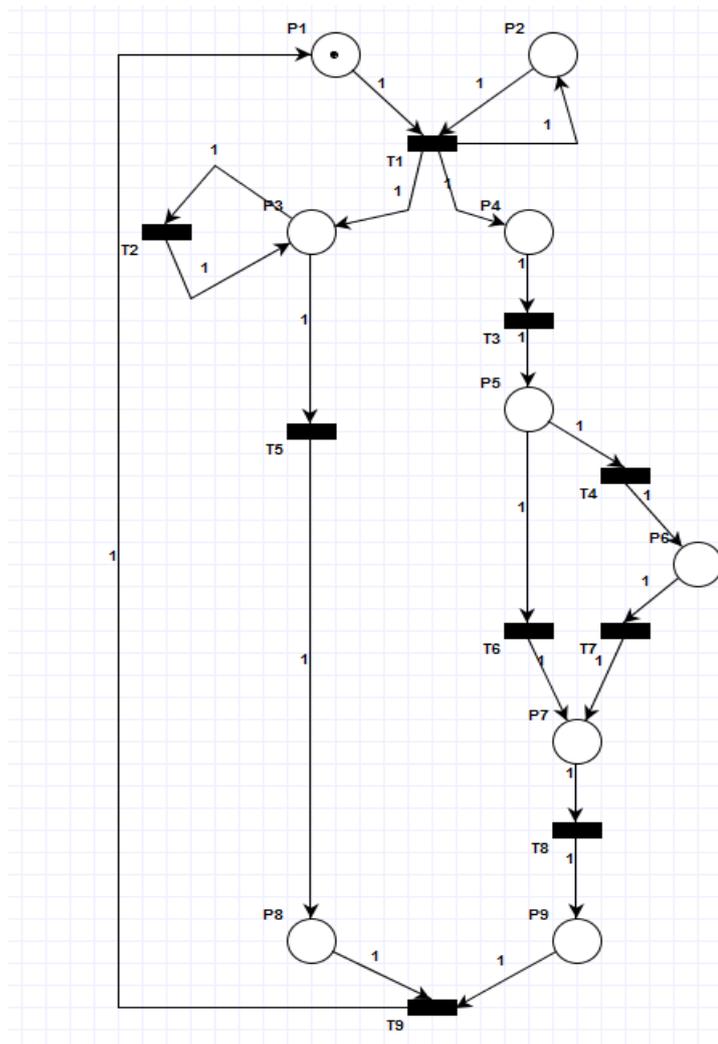


Figura 36. Modelo de Sistema Flexível de Manufatura (DESROCHERS; AL-JAAR, 1995).

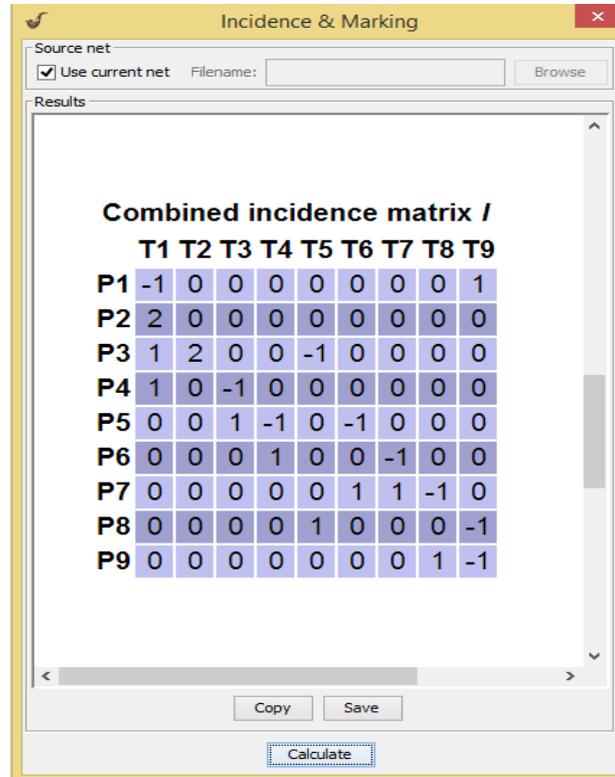


Figura 37. Matriz de Incidência para Modelo de Sistema de Manufatura.

```

-----
>> genetic_algorithm(Rede_Usada_Para_Demonstracao_De_Reducoes, 30, [1], 5, 5, 1000);
-----
Tempo de execução do AG: 0.3758646113s
Melhor fitness inicial: 40
Melhor fitness final: 4
Redução percentual: 90.00
Regras aplicadas   Lugares retirados   Transições retiradas
esp                 2                   0
est                 0                   2
fsp                 3                   5
fst                 4                   3
fst                 6                   7
fpt                 0                   4
fst                 5                   6
fst                 7                   8
fpp                 8                   0
fst                 9                   9
Melhor rede resultante
2
    
```

Figura 38. Matriz de Incidência do Melhor Indivíduo e Informações Gerais do Terceiro Teste.

5.4 Redução de Modelo de RP de uma CAVE de Realidade Virtual

O quarto modelo usado para testes do AGRRP é uma representação de uma CAVE apresentado inicialmente em (CRUZ-NEIRA et al., 1993). A Figura 39 apresenta a ilustração (criada na ferramenta PIPE) do referido modelo e a Figura 40 apresenta a matriz de incidência gerada para o mesmo.

Uma CAVE de realidade virtual é um ambiente físico com elementos que permitem simular um determinado ambiente fazendo com que o usuário se sinta imerso no ambiente virtual criado (SHEN et al., 2013).

O ambiente virtual proposto por Cruz-Neira et. al. (1993) é utilizado na pesquisa de Shen et.al. (2013) para criação de uma representação usando Redes de Petri, a fim de testar uma metodologia de macro reduções baseada em teoria de regiões.

A CAVE é composta por quatro paredes: frontal, lateral esquerda, lateral direita e chão. Fazem parte do ambiente vários projetores de imagem que são apontados para espelhos que por sua vez refletem imagens diretamente para as paredes do ambiente. Para completar a composição da CAVE o usuário conta com um bastão e um óculos de projeção 3D que permitem ao sistema interpretar a posição e orientação do usuário dinamicamente dentro do ambiente durante o tempo em que a simulação ocorre. O óculos permite também que o usuário visualize os efeitos em três dimensões no ambiente e o bastão faz com que ele consiga interagir com elementos específicos do modelo.

O modelo RP da referida CAVE é constituído de trinta e um Lugares, vinte e duas Transições e vinte e dois Arcos, totalizando quarenta elementos. Os Lugares P1, P2, P3, P5, P9, P10, P11 e P26 constituem o vetor de lugares com marcações iniciais.

Os testes realizados usaram os seguintes parâmetros genéticos: população com 30 indivíduos, taxa de cruzamento de 30%, taxa de mutação de 5% e 5000 gerações.

A Figura 41 mostra as informações gerais dos testes realizados. A função de avaliação inicial da rede é de 122 elementos e foi reduzida para 56, totalizando aproximadamente 54% de redução no número de elementos inicialmente presentes no modelo. O algoritmo foi executado em 2,43 segundos e 18 regras de redução foram aplicadas durante o processo.

A Rede de Petri usada no presente teste foi também utilizada por (SHEN et al., 2013) para demonstração de uma metodologia de redução. O diferencial apresentado

na pesquisa é o foco da redução proposta que, ao contrário das metodologias tradicionais de redução, não se baseia em micro reduções do modelo. De acordo com o autor, para redes com muitos elementos as micro reduções não são suficientemente eficientes e por esse motivo é importante aplicar uma macro redução que condense o modelo e permita aplicar posteriormente micro reduções no mesmo modelo.

Ao aplicar a metodologia de macro reduções no modelo apresentado na Figura 39, foi gerado um modelo condensado com 11 Lugares, 21 Transições e 41 arcos. A tabela 3 mostra a comparação da redução obtida na pesquisa desenvolvida por Shen et. al. (2013) e a da presente pesquisa.

Tabela 3. Comparação entre Metodologias Baseadas em Macro e Micro Reduções.

Metodologia/ Informações	Número de Lugares: Início – Fim	Número de Transições: Início – Fim	Número de Arcos: Início – Fim	Tipo de Reduções Aplicadas
(SHEN et. al. 2013)	31 - 11	22 - 21	68 - 41	Macro Reduções
AGRRP	31 - 13	22 - 10	68 - 33	Micro Reduções

É importante observar que a metodologia de macro reduções não impede uma posterior aplicação da metodologia de micro reduções, porém, o comparativo da Tabela 3 dá a noção do potencial de redução das duas abordagens, sendo importante frisar que a ferramenta AGRRP pode ter seu potencial de redução aumentado com a inserção de mais regras para micro reduções na base de regras.

A dificuldade de se aplicar várias micro reduções em grandes modelos, citada por Shen et. al. (2013), é contornada pela automatização das reduções conseguida por meio do AGRRP que, aproveitando-se do paralelismo intrínseco à heurística, é capaz de reduzir grandes modelos por meio da seleção e aplicação automatizada de um grande número de micro reduções.

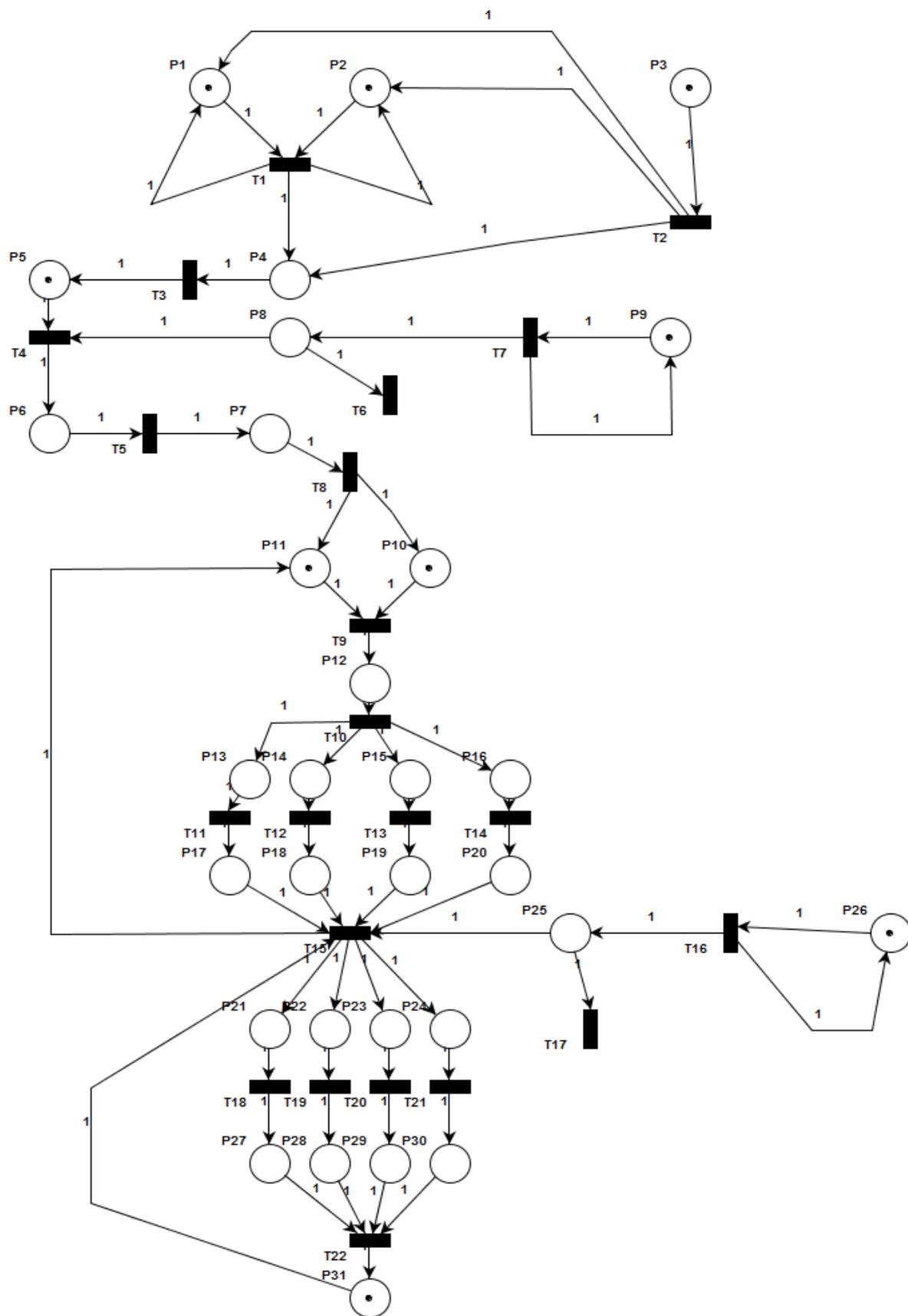


Figura 39. Modelo de CAVE de Realidade Virtual (CRUZ-NEIRA et. al. 1993) .

Combined incidence matrix I																						
	T01	T02	T03	T04	T05	T06	T07	T08	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22
P01	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P02	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P03	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P04	1	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P05	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P06	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P07	0	0	0	0	1	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P08	0	0	0	-1	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P09	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P10	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
P11	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	1	0	0	0	0	0	0	0
P12	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0
P13	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0
P14	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0
P15	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0	0	0	0	0	0
P16	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	0	0	0	0	0	0	0
P17	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	0	0	0	0	0	0
P18	0	0	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0	0	0	0
P19	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0
P20	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0
P21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0
P22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	0	0
P23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	-1	0	0
P24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	-1	0
P25	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	-1	0	0	0	0	0	0
P26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
P27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1
P28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1
P29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1
P30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1
P31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	1

Figura 40. Matriz de Incidência CAVE de Realidade Virtual.

```
>> genetic_algorithm(teste_4, 30, [1 2 3 5 9 10 11 26], 5, 5, 5000);
```

```
-----
Tempo de execução do AG: 2.4321174133s
```

```
Melhor fitness inicial: 122
```

```
Melhor fitness final: 56
```

```
Redução percentual: 54.10
```

```
Regras aplicadas Lugares retirados Transições retiradas
```

```
fst 6 5
fst 7 8
fsp 13 11
fst 12 10
fsp 14 12
fsp 21 18
fpp 17 0
fst 15 13
fsp 22 19
fsp 23 20
fst 16 14
fpp 18 0
fpp 19 0
fsp 24 21
fpp 27 0
fpp 28 0
fpp 29 0
fsp 30 22
```

```
Melhor rede resultante
```

```
  2  2  0  0  0  0  0  0  0  0
  2  2  0  0  0  0  0  0  0  0
  0 -1  0  0  0  0  0  0  0  0
  1  1 -1  0  0  0  0  0  0  0
  0  0  1 -1  0  0  0  0  0  0
  0  0  0 -1 -1  1  0  0  0  0
  0  0  0  0  0  2  0  0  0  0
  0  0  0  1  0  0 -1  0  0  0
  0  0  0  1  0  0 -1  1  0  0
  0  0  0  0  0  0  1 -1  0  0
  0  0  0  0  0  0  0 -1  1 -1
  0  0  0  0  0  0  0  0  2  0
  0  0  0  0  0  0  0  2  0  0
```

p-----

Figura 41. Matriz de Incidência do Melhor Indivíduo e Informações Gerais do Quarto Teste.

5.5 Redução de um Modelo de RP de um Sistema Flexível de Manufatura

O quinto modelo usado para testes do AGRRP é uma representação de um Sistema Flexível de Manufatura retirado e adaptado de (DESROCHERS; AL-JAAR, 1995).

Como dito anteriormente, o modelo de SFM utilizado no presente teste é o mesmo do utilizado na seção 4.2. A diferença dos dois modelos não é relativa ao produto para o qual o sistema de manufatura foi desenvolvido. O maior motivo pelo qual o sistema usado no presente teste tem uma quantidade muito maior de elementos é pela necessidade de inserir diversos elementos e caminhos das peças para garantir exclusão mútua devido à grande quantidade de recursos compartilhados que são descritos no sistema.

Portanto, o processo mostrado nesse sistema é o de produção das peças cilíndricas e do bloco furado e a posterior ligação das duas peças, mas nesse modelo contempla-se todo o transporte de cargas, relações entre o acesso à recursos compartilhados por diversas partes do sistema e muitas outras complexidades relativas a transporte e organização dos produtos finais.

A RP, criada na ferramenta PIPE, do referido modelo é mostrada nas Figuras 42 e 43, na primeira são mostrados os Lugares entre P1 e P43 e na segunda mostra-se os Lugares entre P38 e P80. Para facilitar a visualização os lugares presentes na Figura 42 que também são mostrados na Figura 43 estão destacados, em ambas as ilustrações, por meio de marcações na cor vermelha. Dessa forma, espera-se simplificar a visualização do modelo como um todo por meio da referência de continuidade entre as duas ilustrações.

A Figura 44 mostra a matriz de incidência do modelo utilizado no presente teste.

O modelo RP usado no quinto teste é constituído de oitenta Lugares, sessenta e oito transições e duzentos e cinquenta e um Arcos, totalizando trezentos e noventa e nove elementos. Os Lugares P1, P2, P3, P6, P7, P16, P17, P18, P19, P28, P31, P37, P39, P41 e P60 são os lugares que possuem marcações iniciais.

Os testes realizados usaram os seguintes parâmetros genéticos: população com 30 indivíduos, taxa de cruzamento de 30%, taxa de mutação de 5% e 10000 gerações.

Números inferiores de geração foram testados, porém, devido ao tamanho da rede não apresentaram resultados suficientemente bons de redução.

A Figura 46 apresenta os resultados gerais obtidos com a redução e a Figura 47 mostra a melhor rede obtida por meio das reduções. Para melhorar a visualização da matriz da Figura 47 colocou-se a mesma na horizontal. A leitura da referida matriz deve ser feita começando no final da página.

O principal objetivo do presente teste é verificar o comportamento do algoritmo ao receber como entrada um modelo real de RP que apresenta grande profusão de elementos.

Sistemas Flexíveis de manufatura, como é o caso do modelo usado no presente teste, são frequentemente modelados por meio de Redes de Petri, como pode ser observado na seção Trabalhos Relacionados do Capítulo 2, e devido à grande quantidade de características a serem modeladas, frequentemente cita-se a existência de problemas relacionados a verificação de propriedades e simulação dos modelos devido ao elevado número de elementos. Portanto, o algoritmo AGRRP é uma ferramenta que se mostra importante no presente contexto.

Foram aplicadas quarenta e uma regras de redução no modelo. A função de avaliação da rede inicial foi de 399 e a avaliação do melhor indivíduo ao final foi de 224, o que representa uma redução de 43.86%. O tempo de execução do algoritmo foi de aproximadamente 11,2 segundos.

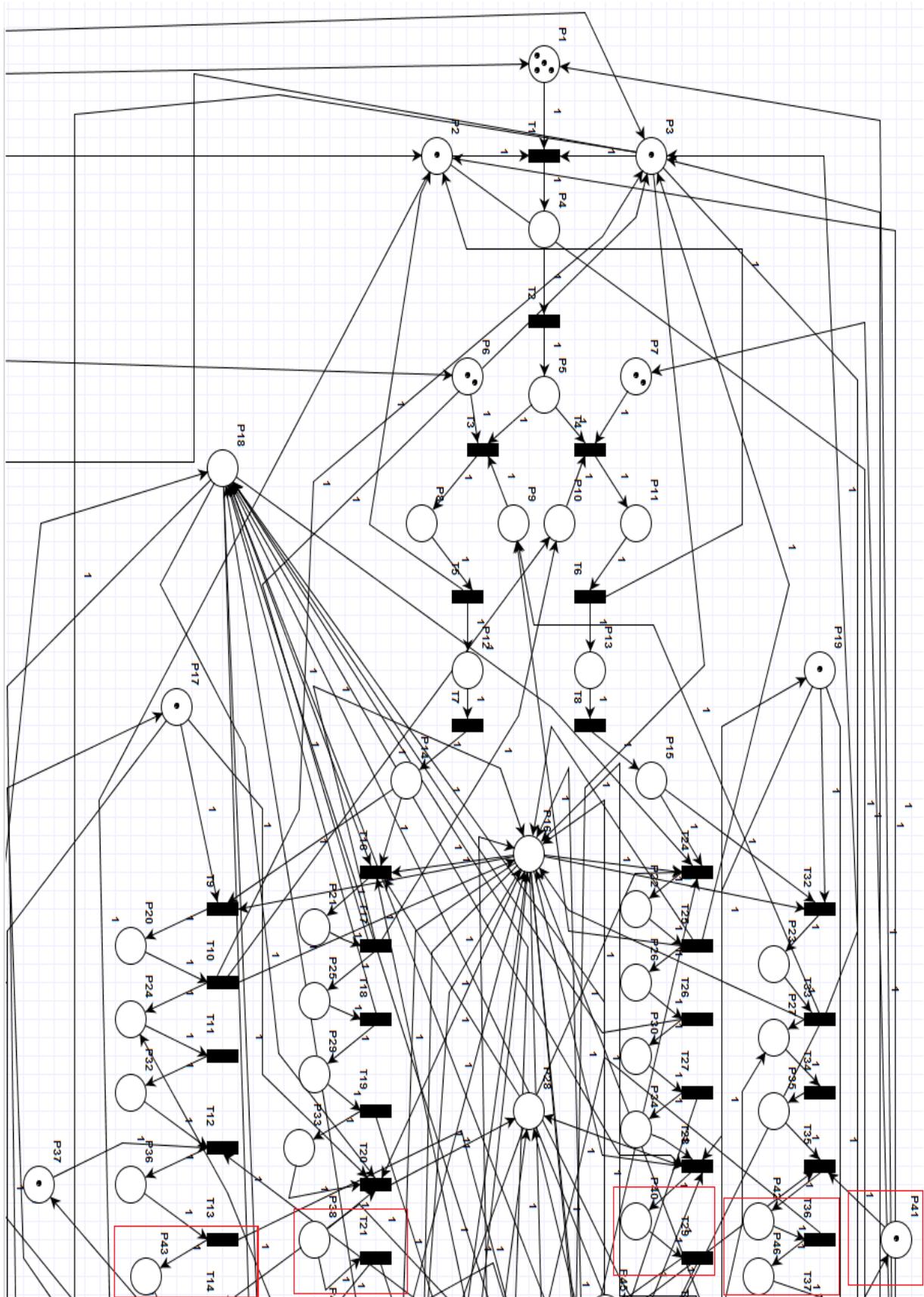


Figura 42. Primeira Parte Modelo de FMS Usado no Quinto Teste


```
>> genetic_algorithm(teste_5, 30, [1 2 3 6 7 16 17 18 19 28 31 37 39 41 60], 5, 5, 10000);
-----
Tempo de execução do AG: 11.2665370590s
Melhor fitness inicial: 399
Melhor fitness final: 224
Redução percentual: 43.86
Regras aplicadas   Lugares retirados   Transições retiradas
fst                4                   2
fst                8                   5
fsp                12                  7
fsp                13                  8
fst                11                  6
fsp                27                  34
fst                20                  10
fsp                45                  30
fst                36                  13
fst                47                  15
fst                21                  17
fsp                57                  41
fst                25                  18
fst                29                  19
fsp                44                  22
fst                48                  23
fst                22                  25
fsp                61                  65
fst                26                  26
fst                30                  27
fsp                66                  43
fsp                69                  67
fst                40                  29
fsp                70                  44
fsp                74                  45
fst                49                  31
fst                23                  33
fsp                75                  53
fst                42                  36
fsp                76                  61
fst                50                  38
fst                78                  46
fst                54                  48
fst                58                  49
fst                63                  50
fst                71                  52
fst                79                  54
fst                55                  56
fst                59                  57
fst                64                  58
fst                72                  60
fst                80                  62
fst                73                  68
```

Figura 45. Informações Gerais do Quarto Teste.

5.6 Contribuições e Limitação

A presente seção é destinada a apresentar algumas das contribuições identificadas na ferramenta AGRRP. Os detalhes observados na seção Trabalhos Relacionados apresentada no Capítulo 3 do presente documento servem como base teórica para a identificação das contribuições a seguir.

É importante frisar que nas buscas, realizadas nas bases de dados “*ACM Digital Library*”, “*IEEE Xplore Digital Library*” e “*SciVerse Scopus*”, utilizando as *strings* de busca “*Genetic Algorithm Petri Net Reduction*”, “*Heuristic Petri Net Reduction*”, “*Petri Net Reduction Methodology*”, “*Petri Net Reduction Strategy*”, “*Novel Petri Net Reduction*” e “*Artificial Intelligence Petri Net Reduction*”, não foram encontradas pesquisas que abordassem a utilização de estratégias de inteligência artificial aliadas à metodologias de redução de Redes de Petri. Esse é um fator bastante importante no que diz respeito às contribuições da presente pesquisa, tendo em vista que pode-se considerar esse fato como um indicativo de inovação na área de pesquisa.

5.6.1 Contribuições

Automatização:

O método proposto tem como contribuição permitir que o pesquisador que deseja reduzir um modelo de RP (seja para propósito de teste de propriedades ou simulação), não precise se preocupar em como e quando cada regra será aplicada. Os cruzamentos fazem com que os indivíduos do AG evoluam por meio da aplicação de regras, sendo que o indivíduo mais apto será o indivíduo que obteve a melhor diminuição percentual de elementos.

Visualização da Redução:

Além de retirar esta necessidade de profundo estudo relacionado a sequência de regras a serem aplicadas, ao final, o usuário tem a possibilidade de verificar a sequência de regras aplicadas e com isso, pode fazer inferências que relacionem a sequência com os resultados obtidos em seu problema.

Combinação de Reduções:

Outra contribuição relacionada está no fato de que mudanças estruturais na rede, por exemplo, a fusão de lugares em um determinado ponto da rede pode criar

uma estrutura alterada que permite uma nova redução que não poderia ocorrer antes da alteração. Essa possibilidade é extremamente difícil de ser avaliada somente estudando a rede e mesmo que seja possível, ainda assim, para cada novo modelo seria necessário um estudo em relação a possíveis combinações mais ou menos favoráveis. Com a utilização do AG, a própria evolução dos indivíduos buscando um melhor valor de avaliação, faz com que esse espaço de busca (relacionado as combinações) possa ser avaliado fazendo uso do paralelismo intrínseco da heurística e ao final, verificações podem ser realizadas por meio da visualização das regras aplicadas e elementos retirados.

Escalabilidade:

Levando em consideração a existência de vários tipos de RP's e a grande quantidade de regras de redução propostas na literatura (como pode ser observado no Capítulo 2), pode-se considerar que a escalabilidade do sistema proposto é um fator positivo.

Cada regra criada para a base de regras da ferramenta é, por si só, uma parte independente do processamento do sistema. Todas as regras compõe o processamento final por meio dos mecanismos genéticos do algoritmo. Portanto, a adaptação do presente algoritmo para contemplar a redução de diferentes modelos de RP requisitaria apenas a inserção de novas regras na base e adaptação da estrutura do indivíduo, sendo o resto dos mecanismos do algoritmo reaproveitáveis sem grandes transtornos. Isso faz com que o sistema tenha como característica uma boa escalabilidade.

Potencial de Redução e Propriedades Mantidas:

A base de regras de redução, usada como ferramenta para que seja possível identificar um padrão de redução na estrutura da rede, faz com que o algoritmo tenha melhor aproveitamento se contiver muitas regras. Quanto mais regras disponíveis, maior a chance de que alterações possam ser feitas na estrutura da rede e quanto mais alterações feitas, mais possibilidades de novas alterações se tornarem possíveis, ou seja, é um processo que se expande por meio de combinações.

Inicialmente, as regras usadas na presente pesquisa mantém vivacidade, equidade e delimitação. O trabalho proposto por Uzam, (2004) trata de uma metodologia para prevenção de *deadlocks* em Sistemas Flexíveis de Manufatura e o autor cita que uma rede reduzida que mantenha a propriedade de vivacidade é sinônimo de uma rede que está livre de *deadlocks*, portanto, a utilização da ferramenta

AGRRP como auxílio para condução de pesquisas de prevenção de *deadlocks* baseando-se em grafos de alcançabilidade, como é o caso da pesquisa previamente citada, é uma estratégia viável e importante.

Finalizando, é importante também citar que o real potencial do algoritmo é aproveitado quanto maior for o modelo de RP a ser reduzido. Uma grande quantidade de elementos, juntamente com uma base com muitas regras de redução permite que diversas combinações relacionadas à estrutura ocorram, fazendo com que o indivíduo seja bastante reduzido e que assim, a verificação de propriedades, análise e simulação dos modelos seja mais simples e eficiente.

5.6.2 Limitação

A maior limitação identificada na presente ferramenta é não possuir estruturas de indivíduo e regras de redução que permitam receber como entrada outros tipos de redes como, por exemplo, redes coloridas, temporizadas, entre outras.

A referida limitação faz com que várias pesquisas que poderiam ser usadas como comparativo para validar a proposta não possam ser avaliadas, porém, adaptar o algoritmo para superar essa limitação é o próximo passo (elencado nos trabalhos futuros) e a escalabilidade do sistema favorece o processo de adaptação da ferramenta.

Capítulo 6

CONCLUSÃO

A presente pesquisa foi desenvolvida com foco nas Rede de Petri e sua utilização para modelar e avaliar sistemas, principalmente os de manufatura.

As pesquisas usadas como referência, e norte, do presente estudo indicam que a grande profusão de elementos em modelos de RP é um problema frequentemente enfrentado. Para lidar com o referido problema pesquisadores focam seus estudos em metodologias e técnicas de redução, bem como estratégias algorítmicas que permitam aplicar, de maneira semiautomática ou não, regras de redução em modelos, fazendo com que elementos sejam eliminados do modelo preservando propriedades estruturais da rede. Após a redução de uma rede, pode-se aplicar metodologias para verificação de propriedades e simulação do modelo sem que a grande profusão de elementos seja um limitante das metodologias.

Com base na situação apresentada, foi proposto um algoritmo que recebe como entrada um modelo de RP e retorna um modelo reduzido (usando seis regras de redução propostas e provadas previamente em estudos científicos) procurando solucionar problemas relacionados à grande profusão de elementos.

Durante o desenvolvimento buscou-se criar uma ferramenta que necessita de menos intervenção do usuário do que a ferramenta semiautomática proposta por Mikolajczak (2009). Portanto, o projeto do algoritmo focou em retirar do usuário a necessidade de qualquer intervenção sobre onde e como uma regra de redução deve ser aplicada. As únicas intervenções diretas são relacionadas a definir qual o percentual de redução buscado para a rede e as informações iniciais sobre a rede a ser reduzida.

A linguagem de programação Matlab foi escolhida para implementar a estratégia algorítmica por ser naturalmente eficiente para lidar com matrizes (forma de representação usada para expressar as Redes de Petri). A escolha da referida linguagem também foi importante por permitir a criação e disponibilização da ferramenta como uma *toolbox*, o que é feito para que pesquisadores possam conhecer e melhorar estratégias algorítmicas de áreas específicas de conhecimento.

Para realizar o processamento necessário na ferramenta criou-se uma estratégia heurística baseada em Algoritmo Genético que varre o espaço de busca (estrutura da Rede usada como entrada) aplicando regras de redução para que a quantidade total de elementos seja reduzida até um número limite de gerações. As aplicações de regras são executadas por meio dos operadores genéticos de Cruzamento e Mutação e os resultados são avaliados por uma função de avaliação que leva em consideração a quantidade de elementos (Lugares, Transições e Arcos) presentes no modelo.

Após o desenvolvimento da ferramenta de redução automatizada de Redes de Petri, a qual foi denominada Algoritmo Genético para Redução de Redes de Petri (AGRRP), cinco redes de tamanhos variados foram selecionadas na literatura para validar e testar a estratégia criada.

Os testes mostraram que a ferramenta permite a redução automatizada de modelos de RP e retorna ao usuário o modelo reduzido e informações sobre: regras aplicadas, lugares e transições retirados, percentual de redução de elementos e o tempo de execução do algoritmo.

A redução de modelos de RP automaticamente usando a ferramenta permite que metodologias de análise das redes como grafo de alcançabilidade e análise baseada em espaço de estados possam ser executadas mais rapidamente (devido a menor quantidade de elementos no modelo) preservando as propriedades de vivacidade, equidade e delimitação apresentadas pela rede, como mostrado nos Capítulos 3 e 4 do presente documento.

Portanto, conclui-se que a nova metodologia, baseada em algoritmos genéticos, é a maior contribuição da presente pesquisa. A ferramenta baseada na metodologia pode auxiliar na aplicação de metodologias de análise como grafo de alcançabilidade, análise baseada espaço de estados ou ainda outras metodologias de análise de RP's.

É importante ressaltar que a ferramenta pode também ser adaptada para contar com mais regras de redução apenas inserindo códigos específicos das regras na base de regras usada pelo algoritmo genético, sem necessidade de adaptação do

processamento geral do algoritmo. Para permitir que o algoritmo receba como entrada diferentes tipos de RP como, por exemplo, redes coloridas, temporizadas e demais classificações de redes, serão necessárias modificações na representação do indivíduo e inserção de regras específicas de redução, porém, os demais componentes do algoritmo podem ser reutilizados, sendo essa uma importante característica da estratégia.

Por fim, após os testes e avaliações, percebe-se que o objetivo principal de criar um sistema inteligente baseado em algoritmo genético para automatizar aplicação de regras de redução em modelos de Rede de Petri foi atingido com sucesso. Além disso, a ferramenta que automatiza a redução tem associada a si uma base de regras de redução, se mostrou aplicável a diferentes tipos de sistemas modelados como RP e traz como resultado um condensado de informações relativas ao percentual de redução, regras aplicadas, elementos retirados e a rede reduzida representada por uma matriz de incidência.

6.1 Trabalhos Futuros

Após o desenvolvimento da primeira versão da ferramenta podem ser elencados alguns trabalhos futuros com o objetivo de promover melhorias e aumentar o escopo de atuação da estratégia. Alguns trabalhos futuros para o decorrer da pesquisa são:

Aumentar o número de Regras de Redução para Redes de Petri Genéricas e incluir Regras de Redução específicas para outras categorias de RP.

Criar estruturas de indivíduo alternativas que contemplem o recebimento de outras categorias de RP como entrada para o presente algoritmo.

Otimizar o processo de escolha e aplicação de regras de redução a fim de melhorar o desempenho do algoritmo.

Realizar testes da ferramenta em conjunto com metodologias de avaliação de propriedades estruturais e simulação de modelos.

Por fim, considera-se importante que a ferramenta AGRRP seja primeiramente adaptada para receber e reduzir Redes de Petri Temporizadas (RPT). A partir dessa adaptação podem ser conduzidos testes com objetivo de comparar o tempo de execução do AGRRP com o tempo de execução do algoritmo proposto por (YANG, H.

Y. H.; LI, 2009). A referida pesquisa tem por objetivo aplicar cinco regras de redução em dois modelos de RPT criados com o intuito de validar primordialmente o desempenho de algoritmos de redução. Para isso os modelos são avaliados e replicados de dez até duzentas vezes a fim de testar o comportamento da estratégia algorítmica ao receber como entrada modelos com enorme profusão de elementos.

REFERÊNCIAS

BALLARINI, P.; DJAFRI, H.; DUFLOT, M.; HADDAD, S.; PEKERGIN, N. Petri nets compositional modeling and verification of Flexible Manufacturing Systems. 2011 IEEE Conference on Automation Science and Engineering (CASE). **Anais...** p.588–593, 2011.

BARROS, J. P. M. P. R. **CpPNeTS: uma Classe de Redes de Petri de Alto-nível**, 1996. UNIVERSIDADE NOVA DE LISBOA.

BERNARDINELLO, L.; FIORELLA, C. A survey of basic net models and modular net classes. *Advances in Petri Nets*. **Anais...** v. 609, p.304–351, 1992. Springer Berlin Heidelberg. Disponível em: <http://dx.doi.org/10.1007/3-540-55610-9_177>. .

BERTHELOT, G.; LRI-IIE. Checking properties of nets using transformations. (G. Rozenberg, Ed.)**Advances in Petri Nets**, v. 222, p. 19–40, 1985. Springer Berlin Heidelberg. Disponível em: <<http://dx.doi.org/10.1007/BFb0016204>>. .

C. A. PETRI. Concepts of Net Theory. **Mathematical Foundations of Computer Science**. p.137–146, 1973. Mathematical Institute of the Slovak Academy of Sciences. Disponível em: <<http://dblp.uni-trier.de>>. .

CHUNG, E.; KIMBER, T.; KIRBY, B.; MASTER, T.; WORTHINGTON, M. **PETRI NETS GROUP PROJECT FINAL REPORT**. 2007.

CRUZ-NEIRA, C.; SANDIN, D. J.; DEFANTI, T. A. Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE. , p. 135–142, 1993.

DESROCHERS, A. A.; AL-JAAR, R. Y. **Applications of petri nets in manufacturing systems: modeling, control, and performance analysis**. IEEE Press, 1995.

DICESARE, F. A review of synthesis techniques for Petri nets with applications to automated manufacturing systems. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 23, n. 1, p. 301–312, 1993. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=214792>>. Acesso em: 27/5/2014.

DRISCOLL, A. T. **Learning MATLAB**. The MathWorks, 2009.

FENG, G.; YI, W.; HUIQUN, Z. Deciding Substitutability of Web Services with Petri Net Reduction Techniques. **2011 10th IEEE/ACIS International Conference on Computer and Information Science**, p. 323–327, 2011. Ieee. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6086489>>. Acesso em: 13/5/2014.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization and Machine Learning**. 1st ed. Addison-Wesley Longman Publishing Co., Inc., 1989.

GROOVER, M. Fundamentals of modern manufacturing: materials processes, and systems. ,2007. Disponível em: <<http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Fundamentals+of+Modern+Manufacturing+:+Materials,+Processes,+and+Systems#0>>. Acesso em: 22/5/2014.

HOLLAND, J. H. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**. Michigan: University of Michigan Press, 1975.

HU, H.; ZHOU, M.; LI, Z.; MEMBER, S. Liveness and Ratio-Enforcing Supervision of Automated Manufacturing Systems Using Petri Nets. , v. 42, n. 2, p. 392–403, 2012.

HUANG, Y.-S.; PAN, Y.-L.; ZHOU, M. Computationally Improved Optimal Deadlock Control Policy for Flexible Manufacturing Systems. **IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans**, v. 42, n. 2, p. 404–415, 2012.

JENSEN, K. AND KRISTENSEN, L. M. **Coloured Petri Nets: Modelling and Validation of Concurrent Systems**. Springer, 2009.

LEE, D.; TU, J.-F.; LEE, H.-J.; CHEN, K. P. Verifying the Produce of Car-Bind through Timed Petri Net. 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA). **Anais...** p.1332–1336, 2012.

LEE-KWANG, H.; FAVREL, J.; BAPTISTE, P. Generalized Petri Net Reduction Method. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 17, n. 2, p. 297–303, 1987. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4309041>>. Acesso em: 27/5/2014.

LETTERS, P. P.; SCIENTIFIC, W.; COMPANY, P.; et al. New Efficient Petri Nets Reductions for Parallel Programs Verification. ,2006.

LU, L.; ZHANG, C. Reduction rules and deadlock detecting methods in Object Oriented Petri Net models. **2010 International Conference on Computer Application and System Modeling (ICCSM 2010)**, , n. Iccasm, p. V12–389–V12–394, 2010. Ieee. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5622322>>. .

LV, Y. Q.; LEE, C. K. M.; WU, Z.; CHAN, H. K.; IP, W. H. Priority-Based Distributed Manufacturing Process Modeling via Hierarchical Timed Color Petri Net. **IEEE Transactions on Industrial Informatics**, v. 9, p. 1836–1846, 2013.

MIKOLAJCZAK, B.; SINGH, A. TransCPN - Software Tool for Transformation of Colored Petri Nets. **2009 Sixth International Conference on Information Technology: New Generations**, p. 211–216, 2009. Ieee. Disponível em:

<<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5070619>>. Acesso em: 13/5/2014.

MURATA, T. Petri nets: Properties, analysis and applications. **Proceedings of the IEEE**, v. 77, p. 541–580, 1989.

MURATA, T. AND KOH, J. Reduction and expansion of live and safe marked graphs. **IEEE Transactions on Circuits and Systems**, v. 27, p. 68–71, 1980.

PAN, Y.-L.; HUANG, Y.-S.; JENG, M.-D. An efficient deadlock prevention policy for FMSs using reduction method and theory of regions. **2011 IEEE International Conference on Systems, Man, and Cybernetics**, p. 568–573, 2011. Ieee. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6083771>>. .

PETERSON, J. L. **Petri Net Theory and the Modeling of Systems**. Prentice Hall PTR, 1981.

QING-XIU, L.; BAO-XIANG, C.; YI-WEI, Z. An improved verification method for workflow model based on Petri net reduction. **2010 2nd IEEE International Conference on Information Management and Engineering**, p. 252–256, 2010. Ieee. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5477436>>. .

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. Prentice Hall, 2009.

SHEN, V. R. L.; CHUNG, Y. F.; CHEN, S.-M.; GUO, J.-Y. A novel reduction approach for Petri net systems based on matching theory. **Expert Systems with Applications**, v. 40, n. 11, p. 4562–4576, 2013. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0957417413000894>>. Acesso em: 13/5/2014.

SILVA, M. **Petri Nets in Automation and Computer Engineering**. Madrid: Editorial AC, 1985.

SLOAN, R. H.; BUY, U. Reduction rules for time Petri nets. **Acta Informatica**, v. 33, n. 5, p. 687–706, 1996. Disponível em: <<http://link.springer.com/10.1007/BF03036471>>. .

TARNAUCA, B.; PUIU, D.; COMNAC, V.; SUCIU, C. Modelling a flexible manufacturing system using reconfigurable finite capacity Petri nets. 13th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM). **Anais...** p.1079–1084, 2012.

UZAM, M. An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions. **The International Journal of Advanced Manufacturing Technology**, v. 19, n. 3, p. 192–208, 2002.

UZAM, M. The use of the Petri net reduction approach for an optimal deadlock prevention policy for flexible manufacturing systems. **The International Journal of Advanced Manufacturing Technology**, v. 23, n. 3-4, p. 204–219, 2004. Disponível em: <<http://link.springer.com/10.1007/s00170-002-1526-5>>. Acesso em: 13/5/2014.

WU, S.; FU, J.; LI, H.; PETZOLD, L. Automatic identification of model reductions for discrete stochastic simulation. **The Journal of chemical physics**, v. 137, n. 3, p. 034106, 2012. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/22830682>>. Acesso em: 13/5/2014.

XIA, C. Property Analysis of Petri Net Reduction. , v. 1, p. 250–253, 2010.

XIA, C.; LIU, Z. Property Preservation of Time Petri Net Reduction. **2010 Third International Symposium on Intelligent Information Technology and Security Informatics**, p. 154–159, 2010. Ieee. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5453628>>. Acesso em: 13/5/2014.

YANG, H.; LI, Z. Extended GSPN modeling and reduction algorithms for rapid performance analysis of service composition system. **Intelligent Computing and Intelligent Systems**, ..., 2009. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5357914>. Acesso em: 13/5/2014.

YANG, H. Y. H.; LI, Z. L. Z. Extended GSPN modeling and reduction algorithms for rapid performance analysis of service composition system. **2009 IEEE International Conference on Intelligent Computing and Intelligent Systems**, v. 1, 2009.

YASUDA, G.; BIN, G. Petri net model based specification and distributed control of robotic manufacturing systems. 2010 IEEE International Conference on Information and Automation (ICIA). **Anais...** p.699–705, 2010.

Apêndice A*

EXEMPLO DE USO DA FERRAMENTA AGRRP

Para demonstrar a utilização da ferramenta AGRRP será utilizado o modelo da “CAVE” de realidade virtual utilizado demonstrado no Capítulo 4.

O primeiro passo para utilizar a ferramenta de redução automatizada é carregar todos os *scripts* (codificação das funções usadas pela ferramenta) para o *workspace* (espaço de trabalho) do software Matlab. A Figura 48 mostra o ícone que permite carregar os *scripts* de uma pasta para montar seu *workspace* e começar a utilização da ferramenta. Na parte esquerda da imagem pode-se observar a existência de todos os *scripts* que compõe a ferramenta AGRRP, exibidos no Apêndice A do presente documento.

Na Figura 48 pode-se notar também a existência de matrizes com a extensão “.mat”. Esse é o formato de matriz que é aceito pelo algoritmo para executar as reduções e é nesse formato que devem ser construídas as matrizes de incidência das redes que desejarmos reduzir.

Se o usuário desejar carregar uma matriz vinda de outra ferramenta, existe a possibilidade de carregar matrizes no formato .csv (valores separados por vírgula do inglês, *Comma-separated values*). Para isso basta que se utilize o comando “variável = csvread(‘arquivo.csv’)”, dessa forma, matrizes criadas em outros softwares podem ser exportadas com a extensão .csv e carregadas para utilização dentro da interface do Matlab.

A Figura 49 exhibe o carregamento da matriz “teste_4.mat” (matriz de incidência da CAVE de realidade virtual apresentada no Capítulo 4) para a variável “teste_modelo_CAVE” que será usado durante a demonstração da ferramenta. Para

realizar o carregamento de qualquer matriz de incidência para utilização na ferramenta deve-se:

1. Salvar a matriz de incidência com a extensão “.mat”.
2. Criar uma variável dentro do ambiente do Matlab.
3. Executar a ação “load”, especificando dentro dos parênteses o nome da matriz que se deseja carregar (lembre-se que o nome deve estar entre aspas simples).

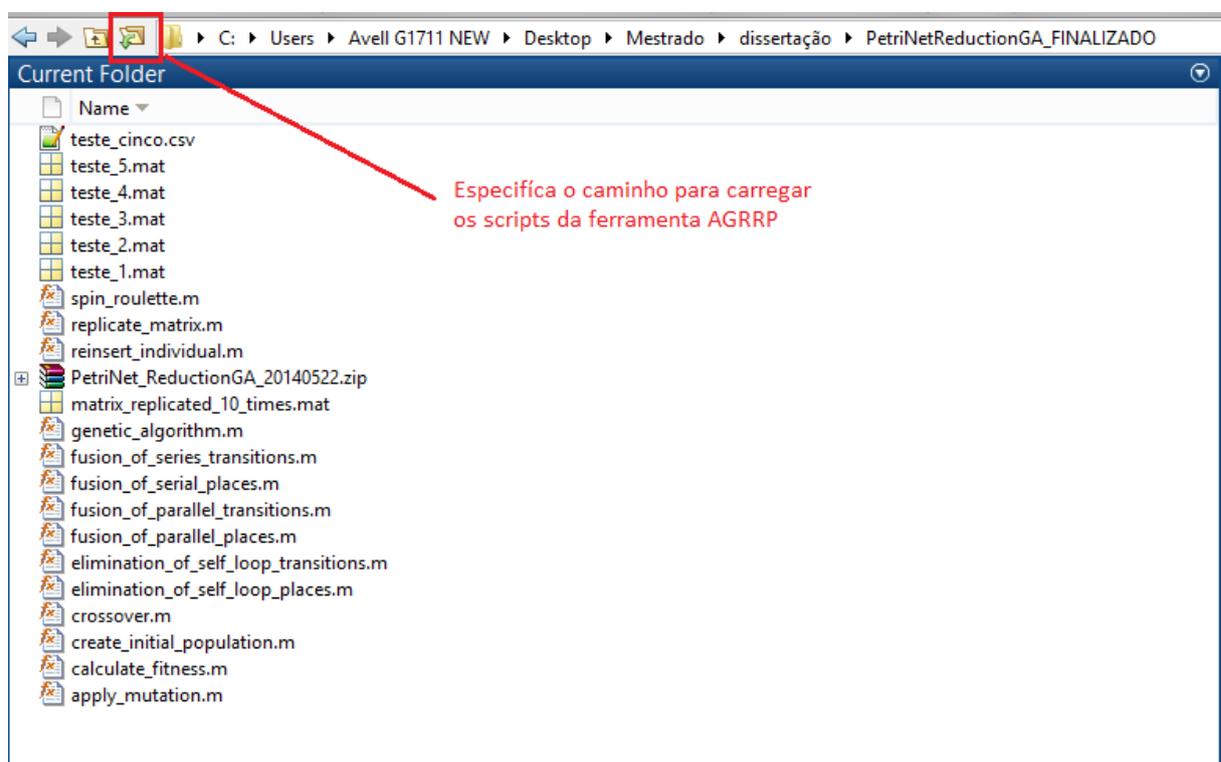


Figura 47. Carregando Scripts para Workspace do Matlab.

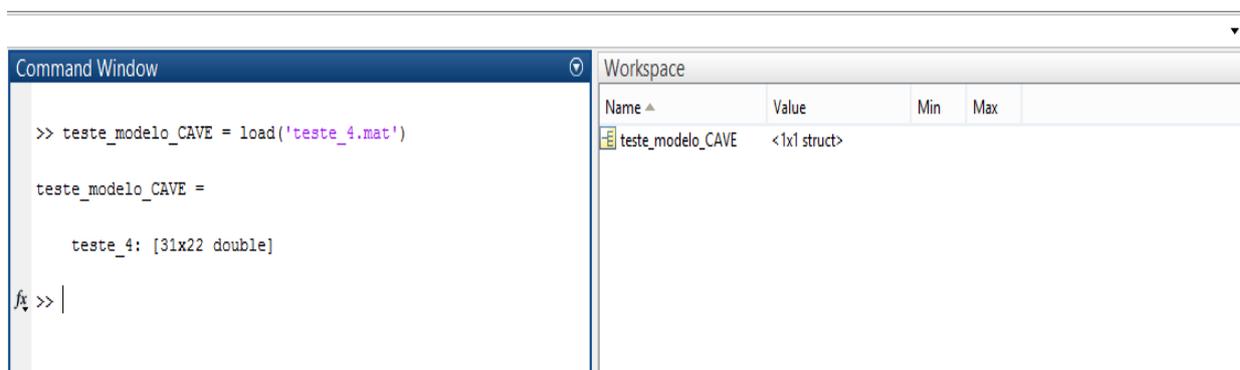


Figura 48. Carregando Matriz de Incidência para Variável "teste_modelo_CAVE".

Após carregar os *scripts* e ter disponível a matriz de incidência, pode-se requisitar a execução do algoritmo genético usando como entrada a matriz de incidência e definindo os parâmetros genéticos desejados.

A Figura 50 mostra a execução do teste por meio da chamada da função (*script*) *genetic_algorithm*. O primeiro parâmetro passado “teste_modelo_CAVE” é a matriz de incidência do modelo que será reduzido. O segundo parâmetro “30” é referente ao número de indivíduos escolhidos para compor a população inicial do algoritmo genético. O terceiro parâmetro “[1 2 3 5 9 10 11 26]” é um vetor composto pelos índices dos Lugares com marcações iniciais na rede. O quarto parâmetro “5” é relativo à variabilidade inicial dos indivíduos, ou seja, define o número de regras que serão aplicadas aleatoriamente nos indivíduos da população inicial para gerar variabilidade inicial no algoritmo. O quinto parâmetro “5” é a porcentagem de mutação que ocorrerá durante as gerações de execução do algoritmo. Por fim, o sexto parâmetro “5000” é o número de gerações que o usuário define como ponto de parada do algoritmo.

Pode-se observar também na Figura 46 que o algoritmo exibe informações sobre o tempo de execução, o melhor *fitness* (função de avaliação inicial), o melhor *fitness* final, a redução percentual, as regras aplicadas e a matriz de incidência do melhor indivíduo gerado durante a execução do algoritmo.

```
genetic_algorithm(teste_modelo_CAVE, 30, [1 2 3 5 9 10 11 26], 5, 5, 5000);
```

Tempo de execução do AG: 2.1833006249s

Melhor fitness inicial: 122

Melhor fitness final: 56

Redução percentual: 54.10

Regras aplicadas	Lugares retirados	Transições retiradas
fst	6	5
fst	7	8
fsp	13	11
fsp	14	12
fsp	15	13
fsp	16	14
fst	12	10
fpp	17	0
fpp	18	0
fpp	19	0
fst	21	18
fst	22	19
fsp	23	20
fpp	27	0
fst	24	21
fpp	28	0
fpp	29	0
fsp	30	22

Melhor rede resultante

2	2	0	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0	0
0	-1	0	0	0	0	0	0	0	0
1	1	-1	0	0	0	0	0	0	0
0	0	1	-1	0	0	0	0	0	0
0	0	0	-1	-1	1	0	0	0	0
0	0	0	0	0	2	0	0	0	0
0	0	0	1	0	0	-1	0	0	0
0	0	0	1	0	0	-1	1	0	0
0	0	0	0	0	0	1	-1	0	0
0	0	0	0	0	0	0	-1	1	-1
0	0	0	0	0	0	0	0	2	0
0	0	0	0	0	0	0	2	0	0

Figura 49. Execução de Teste na Ferramenta AGRRP.