

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**MODELAGEM DE SISTEMAS DE MANUFATURA
USANDO REDES DE PETRI COLORIDAS FUZZY
FOCANDO A SOLUÇÃO DE CONFLITOS**

MAYRA SGAVIOLI

ORIENTADOR: PROF. DR. EDILSON REIS RODRIGUES KATO

São Carlos - SP
Junho/2010

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**MODELAGEM DE SISTEMAS DE MANUFATURA
USANDO REDES DE PETRI COLORIDAS FUZZY
FOCANDO A SOLUÇÃO DE CONFLITOS**

MAYRA SGAVIOLI

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: Dr. Edilson Reis Rodrigues Kato.

São Carlos - SP
Junho/2010

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

S523ms

Sgavioli, Mayra.

Modelagem de sistemas de manufatura usando Redes de Petri Coloridas Fuzzy focando a solução de conflitos / Mayra Sgavioli. -- São Carlos : UFSCar, 2011.
90 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2010.

1. Inteligência artificial. 2. Lógica nebulosa. 3. FMS. I.
Título.

CDD: 006.3 (20ª)

Universidade Federal de São Carlos

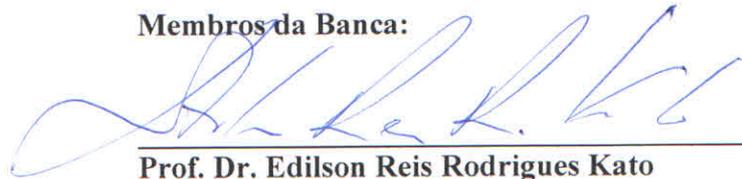
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

“Modelagem de Sistemas de Manufatura usando Redes de Petri Coloridas Fuzzy focando a Solução de Conflitos”

MAYRA SGAVIOLI

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. Orides Morandin Júnior
(DC/UFSCar)



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



Prof. Dr. Mário Lúiz Tronco
(IBILCE/UNESP/São José do Rio Preto)

São Carlos
Junho/2010

*Aos meus Pais,
Angelo e Mécia*

AGRADECIMENTOS

Agradeço a Deus por esta oportunidade, pela iluminação e por fazer desta experiência um grande aprendizado em minha vida.

Aos meus pais, Angelo e Mércia que sempre estiveram ao meu lado, me apoiando, me incentivando, me animando e me aconselhando.

Ao meu orientador, Prof. Edilson Kato, por sua dedicação e direcionamentos que tornaram possível a realização deste trabalho.

Ao professor Orides, por auxiliar o desenvolvimento deste trabalho.

Aos meus amigos Ary, Talles, Alexandre e Leonardo que me acompanharam e incentivaram e, sempre estiveram presentes, mesmo longe.

À minha amiga Élen, que me acolheu em sua casa e que tornou os meus dias muito melhores.

Aos meus amigos Flávio e Danilo, pelas conversas esclarecedoras, pelos conselhos e por me ouvirem sempre.

Aos meus companheiros de laboratório e aos colegas de mestrado, pelos agradáveis dias que passamos juntos.

A todos os professores do mestrado que de alguma forma influenciaram este trabalho.

*"O futuro pertence àqueles que
acreditam na beleza de seus sonhos"
Eleanor Roosevelt*

RESUMO

Um Sistema Flexível de Manufatura é um sistema de produção onde mais de um evento pode ocorrer ao mesmo tempo de forma paralela, assíncrona ou de forma concorrente. Estes foram projetados para atender às necessidades do mercado, que demanda por melhor qualidade dos produtos, redução de custos e prazos de entrega menores. Devido à complexidade e à flexibilidade destes sistemas, conflitos podem ocorrer quando mais de um processo requisita o mesmo recurso, como máquinas ou AGVs, ou quando um produto pode ser produzido em roteiros distintos. Assim, uma política de solução de conflito é necessária neste tipo de ambiente. Este trabalho usa Redes de Petri Coloridas Fuzzy para modelar um sistema de manufatura. O sistema de manufatura é modelado considerando os recursos, buffers, roteiros de produção, rotas dos AGV e identificando os pontos de conflitos. A resolução dos conflitos é realizada considerando tanto informações do chão de fábrica quanto da gestão da produção. Regras fuzzy são construídas para atribuir prioridades aos processos conflitantes e um sistema baseado em regras fuzzy é modelado em Redes de Petri Coloridas Fuzzy na mesma rede do sistema de manufatura, tornando o modelo do sistema independente de outros sistemas de apoio para a solução dos conflitos. Neste trabalho foram considerados os conflitos que podem ocorrer na estação de carga e descarga e nos buffer de entrada e saída. De acordo com o ponto de conflito, uma base de regras é modelada para atribuir prioridades aos processos. A partir desta modelagem é possível realizar o controle do sistema e garantir que os conflitos identificados sejam resolvidos.

Palavras-chave: Modelagem, Controle, Sistemas de Manufatura, Lógica Fuzzy, Conflitos, Redes de Petri Coloridas Fuzzy.

ABSTRACT

A Flexible Manufacturing System is a production system where more than one event can occur simultaneously in parallel, asynchronously or concurrently. These are designed to meet the needs of the market, demand for improved product quality, lower costs and shorter delivery times. Due to the complexity and flexibility of these systems, conflicts can occur when more than one process requests the same resource, such as machines or AGVs, or when a product can be produced in more than one production routing. Thus, a conflict resolution policy is needed in this type of environment. This study uses Fuzzy Coloured Petri Nets for modeling a manufacturing system. The manufacturing system is modeled considering the resources, buffers, production routings, AGV routes and identifying conflict points. The conflict resolution is performed considering both information on the shop floor and production management. Fuzzy rules are constructed to prioritize conflicting processes and a fuzzy rule-based system is modeled on Fuzzy Coloured Petri Nets in the same network as the manufacturing system, making the model of the system independent of other systems of support for the resolution of conflicts. In this work we considered the conflicts that can occur when loading and unloading station and the input and output buffer. According to the source of conflict, a rule base is shaped to assign priorities to processes. From this model it is possible to make the control system and ensure that identified conflicts are resolved.

Keywords: Modeling, Control, Manufacturing Systems, Fuzzy Logic, Conflicts, Fuzzy Coloured Petri Nets.

LISTA DE FIGURAS

Figura 2. 1 – Modelo de CPN.....	19
Figura 2. 2 – Conflito em Redes de Petri	20
Figura 2. 3 – Conflito em CPN (adaptado de Jensen, Kristensen e Wells, 2007)	20
Figura 3. 1 – Uma Fuzzy Petri Net	31
Figura 3. 2 - Representação em Redes de Petri Fuzzy da regra Tipo 1 (CHEN et al., 1990).....	32
Figura 3. 3 - Representação em Redes de Petri Fuzzy da regra Tipo 2 (CHEN et al., 1990).....	32
Figura 3. 4 - Representação em Redes de Petri Fuzzy da regra Tipo 3 (CHEN et al., 1990).....	32
Figura 3. 5 - Representação em Redes de Petri Fuzzy da regra Tipo 4 (CHEN et al., 1990).....	33
Figura 3. 6 - Outra modelagem de Rede de Petri Fuzzy	34
Figura 4.1. – Fluxograma dos critérios de desempate na estação de carga.....	42
Figura 4.2. – Fluxograma dos critérios de desempate nos buffers de entrada e saída.....	42
Figura 5. 1 - Arranjo físico do Sistema de Manufatura usado nesta proposta.....	45
Figura 5. 2 - Modelagem em CPN do arranjo físico do sistema.	46
Figura 5. 3 - Modelagem do estacionamento dos AGVs.....	48
Figura 5. 4 - Modelagem da Máquina 1.....	48
Figura 5. 5 – Modelo para seleção do roteiro.....	49
Figura 5. 6 – Modelagem da Estação de Carga e Descarga.....	50
Figura 5. 7 – Modelo da estação de carga e descarga	51
Figura 5. 8 – Modelo para leitura dos valores considerados na seleção dos produtos	52
Figura 5. 9 – Modelo para seleção do roteiro de acordo com o produto	53
Figura 5. 10 – Modelo para leitura dos valores considerados para a seleção dos roteiros do produto A.....	54
Figura 5. 11 – Modelo da estação de trabalho de M1	55
Figura 5. 12 – Modelo para leitura dos valores considerados para a seleção dos produtos nos buffers de entrada das máquinas.	56

Figura 5. 13 – Modelo para seleção do próximo destino após o processo em M1....	57
Figura 5. 14 – Conjuntos fuzzy para a variável de saída Prioridade	58
Figura 5. 15 – Conjuntos fuzzy para a variável Tamanho do Lote	58
Figura 5. 16 – Conjuntos fuzzy para a variável Prazo de Entrega.....	59
Figura 5. 17 – Modelo em FCPN para o ponto de decisão para escolha dos produtos na estação de carga.....	61
Figura 5. 18 – Modelo de defuzzificação e para ordenar os produtos de acordo com a prioridade	62
Figura 5. 19 – Conjuntos fuzzy para a variável Tempo de Processamento do Roteiro	62
Figura 5. 20 – Conjuntos fuzzy para a variável Espaço no Buffer.....	62
Figura 5. 21 – Modelo em FCPN para o ponto de decisão para escolha dos roteiros na estação de carga.....	64
Figura 5. 22 – Modelo de defuzzificação e para ordenar os roteiros de acordo com a prioridade	65
Figura 5. 23 – Modelo em FCPN para o ponto de decisão para escolha dos produtos nos buffers de entrada e de saída.....	66
Figura 5. 24 – Modelo de defuzzificação e para ordenar os produtos dos buffers de acordo com a prioridade.....	67
Figura 5. 25 – Modelo em FCPN para o ponto de decisão para escolha dos produtos nos buffers de entrada e de saída.....	68
Figura 5. 26 – Modelo em FCPN para o ponto de decisão para escolha dos produtos nos buffers de entrada e de saída.....	69
Figura 5. 27 – Modelo da transição PriorProd.....	70
Figura 5. 28 – Modelo para seleção do roteiro de acordo com o produto	71
Figura 5. 29 – Modelo para seleção do roteiro do produto A	71
Figura 5. 30 – Modelo para estação de carga	72
Figura 5. 31 – Modelo para estação de trabalho M1	72
Figura A. 1– Modelo para seleção do roteiro do produto C.....	85
Figura A. 2– Modelo para seleção do roteiro do produto D.....	86
Figura A. 3– Modelo para seleção do roteiro do produto E.....	86
Figura A. 4 – Modelo da estação de trabalho de M2.....	87
Figura A. 5 –Modelo para seleção do próximo destino após o processo em M2	87
Figura A. 6 – Modelo da estação de trabalho de M3.....	88

Figura A. 7 – Modelo para seleção do próximo destino após o processo em M3	88
Figura A. 8 – Modelo da estação de trabalho de M4.....	89
Figura A. 9 –Modelo para seleção do próximo destino após o processo em M4.	89
Figura A. 10 – Modelo da estação de trabalho de M5.....	90
Figura A. 11 – Modelo da estação de trabalho de M6.....	90

LISTA DE TABELAS

Tabela 3. 1. Funções para cálculo do conseqüente das regras	33
Tabela 3. 2. Incertezas em Manufatura (adaptado de Lee e Banerjee, 2009)	38
Tabela 4. 1. Critérios de Desempate	41
Tabela 5. 1. Roteiros de Produção.....	50
Tabela 5. 2. Regras fuzzy para a escolha do produto	59
Tabela 5. 3. Regras fuzzy para a escolha do roteiro	63
Tabela 5. 4. Regras fuzzy para a escolha dos produtos nos buffers de entrada e saída	63
Tabela 5. 5. Informações da Produção para Simulação.....	67

LISTA DE ABREVIATURAS E SIGLAS

AMS – *Automated Manufacturing System*

FMS – *Flexible Manufacturing System*

CPN – *Coloured Petri Nets*

FPN – *Fuzzy Petri Nets*

FCPN – *Fuzzy Coloured Petri Nets*

AGV – *Automated Guided Vehicle*

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO.....	14
1.1 Objetivos	15
1.2 Justificativa e Motivação.....	16
1.3 Organização do Trabalho	18
CAPÍTULO 2 - MODELAGEM DE SISTEMAS DE MANUFATURA.....	14
2.1 Considerações Iniciais.....	14
2.2 Redes de Petri e Sistemas de Manufatura.....	15
2.2.1 Modelagem em Redes de Petri	15
2.2.2 Redes de Petri Colorida	17
2.2.3 Conflitos em Redes de Petri.....	19
2.3 Abordagens e Metodologias de Programação no Controle de Sistemas de Manufatura	21
2.4 Considerações Finais	26
CAPÍTULO 3 - REDES DE PETRI FUZZY.....	29
3.1 Considerações Iniciais.....	29
3.2 Modelo de Redes de Petri Fuzzy para Representação do Conhecimento	30
3.3 Modelo de Representação para Sistema Físico Dinâmico	34
3.4 Redes de Petri Coloridas Fuzzy	35
3.5 Aplicação em Sistemas de Manufatura	36
3.6 Considerações Finais	38
CAPÍTULO 4 - MODELAGEM E CONTROLE DE SISTEMAS DE MANUFATURA USANDO REDES DE PETRI COLORIDAS FUZZY	39
4.1 Proposta.....	39
4.2 Critérios de Desempate.....	40
4.3 Método de Desenvolvimento	43
4.4 Ambiente para Validação	44
CAPÍTULO 5 - MODELAGEM E VALIDAÇÃO DA PROPOSTA	45

5.1 Considerações Iniciais.....	45
5.2 Roteiros de Produção.....	50
5.3 Modelos do FMS em Redes de Petri Colorida	51
5.4 Modelos em Redes de Petri Coloridas Fuzzy.....	57
5.4.1 Escolha dos produtos.....	58
5.4.2 Escolha do roteiro de produção.....	60
5.4.3 Buffer de Entrada e Buffer de Saída	63
5.5 Simulação e Validação do Modelo	67
5.6 Análise de Resultados.....	73
CAPÍTULO 6 - CONSIDERAÇÕES FINAIS.....	75
6.1 Conclusão	75
6.2 Trabalhos Futuros	76
REFERÊNCIAS BIBLIOGRÁFICAS	77
APÊNDICE A	82
APÊNDICE B	85

Capítulo 1

INTRODUÇÃO

As indústrias dos mais diversos segmentos têm cada vez maior necessidade de melhorias na sistemática de produção com o objetivo de aumentar a produtividade, reduzir custos, melhorar a qualidade dos produtos e as condições de atendimento à demanda. Flexibilidade e eficiência também se tornaram requisitos essenciais em sistemas de manufatura com o objetivo de manter um alto nível de produtividade e enfrentar as rápidas mudanças nas demandas do mercado.

Para tanto, várias técnicas e modelos têm sido desenvolvidos a fim de criar conceitos inovadores e soluções originais para resolução de alguns problemas complexos encontrados nestes ambientes. Sistemas de manufatura informatizados e que proporcionam maior automação foram desenvolvidos para suprir essas necessidades do mercado. Nessa categoria, encontram-se os Sistemas Automatizados de Manufatura (AMS – Automated Manufacturing System).

Os AMSs são sistemas onde os processos são executados de forma automática sem a participação direta de um trabalhador humano. A automação é implementada usando uma sequência de instruções combinado com um sistema de controle que executa estas instruções (GROOVER, 2007).

O AMSs deve atender às necessidades do planejamento da produção, rotas de produção alternativas, operação do sistema e as disponibilidades de recursos, além do controle básico (KATO *et al.*, 2001). Estes sistemas podem ser classificados em três tipos: automação fixa, automação programável e automação flexível (GROOVER, 2007).

Os sistemas com automação flexível são conhecidos como Sistemas Flexíveis de Manufatura (FMS – Flexible Manufacturing System). Os FMSs são caracterizados pela produção de uma variedade de peças com um volume médio de produção.

Neste tipo de sistema, mais de um tipo de peça pode ser produzido simultaneamente e uma variedade de operações pode ser realizada pela mesma máquina.

A flexibilidade de um sistema de manufatura pode ser definida como a habilidade de realizar mudanças devido a algumas circunstâncias ou instabilidades causadas pelo ambiente (BESKESEA *et al.*, 2004), assim como a habilidade de se ajustar as preferências dos consumidores e a agilidade do sistema na velocidade de reconfigurar-se para satisfazer a demanda (ZHOU; VENKATESH, 1999). Também é condizente com o aumento da produtividade, redução dos custos da produção e inventários e, aumento da qualidade, se administrado de maneira efetiva e eficiente (BUYURGAN; SAYGIN, 2006).

A versatilidade obtida com este tipo de sistema possibilita uma vantagem competitiva para as empresas. Entretanto, traz algumas dificuldades no âmbito do planejamento, programação, monitoramento e do controle da produção.

Dois problemas que podem ser encontrados são o compartilhamento de recursos e a escolha de roteiros de produção para a fabricação de um produto dentre os roteiros alternativos existentes. Essas situações podem resultar em situações de conflitos, que devem ser solucionados para garantir o andamento do sistema.

Sendo assim, todas as etapas da construção e gerenciamento de sistemas de manufatura devem ser elaboradas precisamente e uma política de solução de conflitos deve ser definida.

1.1 Objetivos

Neste trabalho, uma estratégia de modelagem de sistemas de flexíveis de manufatura é proposta para lidar com a complexidade destes sistemas e resolver os conflitos durante sua execução.

A modelagem do sistema de manufatura foi realizada usando Redes de Petri. Recursos como máquinas, buffers, AGVs foram modelados, assim como os roteiros de produção e rotas dos AGVs.

Uma política de solução de conflitos, como disputa de recursos e escolha dos roteiros de fabricação, foi definida para solucionar os conflitos modelados na rede de

Petri. A lógica fuzzy foi usada para a construção de regras fuzzy usando vários critérios a serem considerados para a solução dos conflitos em pontos específicos do modelo do sistema. O sistema baseado em regras fuzzy foi modelado usando Redes de Petri Coloridas Fuzzy (FCPN – Fuzzy Coloured Petri Net) e foi integrado com a modelagem do sistema de manufatura.

Para analisar os resultados obtidos com esta modelagem de resolução de conflitos foram verificados os pontos de conflitos considerados e como estes foram resolvidos ao longo da execução das tarefas, seguindo uma ordem de precedência entre os conflitos a serem resolvidos.

O método proposto baseou-se na simulação como forma de validação do modelo, além de também utilizá-la durante a concepção do próprio método de resolução de conflitos. No entanto, podem ser aplicadas outras formas analíticas de validação, já que o modelo foi desenvolvido tendo como ferramenta as Redes de Petri.

1.2 Justificativa e Motivação

Existem inúmeras técnicas de modelagem e controle na literatura para sistemas de manufatura, como modelagem matemática, cadeias de Markov, teoria de filas, grafos, redes de Petri e simulação (AGUIRE, 2007; JAIN; JAIN; SINGH, 2006).

Desde 1980, Redes de Petri têm sido vastamente utilizadas para modelar sistemas de manufatura para análise, avaliação do desempenho, simulação e controle (WU; ZHOU; HU, 2007).

Redes de Petri são uma ferramenta matemática e gráfica usada em estudos de sistemas de eventos discretos. As principais razões para justificar o uso de redes de Petri como uma ferramenta para modelagem de sistemas FMS são sua habilidade para capturar e representar sistemas com concorrência, paralelismo e sincronização. Além de compartilhamento de recursos, exclusão mútua e conflitos.

Sua habilidade em modelar sistemas hierarquicamente, a facilidade para visualização do sistema, integração entre as estruturas da rede, a habilidade em representar recursos e atividades, tamanho de buffers, restrições e precedência e o uso dos modelos para simulação e análise de desempenho são fatores que garantem sua vantagem em relação às outras técnicas de modelagem (TSINARAKIS; TSOURVELOUDIS; VALAVANIS, 2005).

Realizar o controle de um FMS é uma tarefa complexa devido aos vários subsistemas e elementos que o compõe, concorrência de eventos, múltiplos roteiros de produção, produção de uma variedade de tipos de peças, disputa por recursos, rotas dos AGVs e planos de produção. Assim, é necessário definir uma modelagem eficiente do sistema e uma política de alocação de recursos

Este trabalho é apoiado no modelo desenvolvido por Araújo (2006). O trabalho de Araújo (2006) propôs um método para modelagem de controle de AMS usando redes de Petri Virtual. No entanto, naquele trabalho, não foi adotada uma técnica de resolução de conflitos. Assim, este trabalho implementa esta condicional usando lógica fuzzy ao modelo já definido.

Regras de despacho, lógica fuzzy e sistemas baseados em agentes têm sido usados em sistemas de controle com programação dinâmica da produção. Estas abordagens são usadas para resolver conflitos na fila de entrada para o uso de um recurso compartilhado e escolha de roteiros.

A lógica fuzzy tem sido usada para construir regras combinando vários critérios a serem satisfeitos e assim atribuir prioridades aos processos conflitantes, ao contrário das outras abordagens que usam somente um critério de desempate (BILGE *et al.*, 2007).

Com esta abordagem é possível solucionar os conflitos considerando informações da gestão da produção assim como informações do chão de fábrica no momento da tomada de decisão.

A extensão de redes de Petri, Redes de Petri Coloridas (CPN – Coloured Petri Nets) foi escolhida para a modelagem do sistema de manufatura devido a sua capacidade de modelagem de sistemas complexos usando tipos de dados e funções lógicas.

Neste trabalho, o sistema baseado em regras fuzzy foi modelado em FCPN. As redes de Petri fuzzy (FPN – Fuzzy Petri Nets) vêm sendo usadas para modelagem e representação do conhecimento fuzzy, assim como o raciocínio fuzzy.

As FCPN combinam as propriedades das CPN e FPN para melhor se adaptar a resolução de problemas complexos.

A modelagem do sistema de solução de conflitos também em redes de Petri tem como objetivo integrar a decisão no mesmo modelo do sistema de manufatura e permitir que o processo de decisão seja realizado independentemente de outros sistemas de apoio.

Outros trabalhos relacionados também foram desenvolvidos no Laboratório Tear, do Departamento de Computação da Universidade Federal de São Carlos. Morandin e Kato (2005) propuseram uma modelagem em redes de Petri virtual considerando recursos compartilhados e processos de produção alternativos e Martins (2005) propôs uma arquitetura modular para controle de FMS.

1.3 Organização do Trabalho

Primeiramente, no capítulo 2, foram abordadas questões inerentes aos problemas no controle de sistemas de manufatura, os trabalhos relacionados à modelagem e controle em redes de Petri, algumas propriedades específicas das CPN e como são representados os conflitos neste tipo de modelagem.

A relação entre controle e programação da produção e as abordagens usadas na literatura para a solução de problemas entre recursos compartilhados e flexibilidade de roteiros também será abordada.

No capítulo 3, uma introdução sobre as FPN será apresentada e suas abordagens analisadas. No tópico seguinte, serão consideradas algumas características específicas das FCPN. Finalmente, serão mostrados os trabalhos relacionados às FPN e FCPN com os sistemas de manufatura.

No capítulo 4, a proposta deste trabalho será explicada, assim como o método de desenvolvimento, os critérios de desempate e o ambiente de validação.

No capítulo 5 serão apresentados os modelos do sistema de manufatura, como modelo das rotas dos AGVs, roteiros de produção e os modelos de cada máquina. Os modelos do sistema baseado em regras para cada ponto específico do sistema também serão apresentados. Neste capítulo também serão mostradas as

análises feitas durante a simulação do modelo para garantir a resolução de conflitos no sistema.

No capítulo 6 serão apresentadas as considerações finais e os trabalhos futuros.

Capítulo 2

MODELAGEM DE SISTEMAS DE MANUFATURA

Este capítulo apresenta assuntos relativos à modelagem de sistemas de manufatura. Primeiramente serão apresentadas algumas características dos sistemas de controle, depois apresentados alguns trabalhos relacionados à modelagem e controle de sistemas de manufatura em redes de Petri e, finalmente, a relação com a programação da produção e os trabalhos desenvolvidos nesta área.

2.1 Considerações Iniciais

O controle de um sistema de manufatura tem que ser modelado para satisfazer todas as restrições em tempo real da linha de produção e as mudanças dinâmicas necessárias do mercado. A metodologia de controle deve lidar primeiramente com os problemas relacionados à modelagem, simulação, análise e, finalmente com o controle em tempo real (ZHOU; VENKATESH, 1999).

A primeira etapa, composta pela modelagem, simulação e análise é onde são geralmente realizadas a construção dos modelos do AMS e determinada a estimativa das medidas de desempenho tais como utilização de robôs e máquinas, tamanho da fila de entrada das máquinas, entre outras. O propósito desta atividade é sugerir uma configuração ótima do AMS para as especificações necessárias, incluindo o arranjo físico, as rotas dos AGVs através das máquinas, as políticas de programação e sincronização das máquinas, entre outras tarefas (ZHOU; VENKATESH, 1999).

O segundo problema deve lidar com o controle em tempo real do AMS, que tipicamente envolve coordenação, programação em tempo real e monitoramento dos recursos. O objetivo do controle é manter uma alta utilização do sistema assim como satisfazer os prazos de produção (ZHOU; VENKATESH, 1999).

Segundo Groover (2007), as funções desempenhadas pelo controle de um AMS incluem as decisões para a combinação e avaliação da entrada de várias peças no sistema. São decisões baseadas de acordo com os dados do sistema, tais como índices de produção desejados e matéria-prima disponível. Decisões envolvendo o tráfego e o estado dos AGVs também devem ser regularizadas pelo controle do sistema. Como alguns caminhos no sistema permitem somente o trânsito de um veículo, o movimento destes deve ser controlado.

Outra função do controle é satisfazer algumas especificações como evitar *deadlocks*, considerar as relações de precedência de operações, limites dos *buffers*, resolução de conflitos, entre outros problemas (GOLMAKANI *et al.*, 2006).

2.2 Redes de Petri e Sistemas de Manufatura

Redes de Petri têm sido aplicadas a modelagem, análise, simulação, controle e programação de FMS (ZHOU; VENKATESH, 1999; WU *et al.*, 2007; LIU *et al.*, 2009). Na modelagem de um sistema de manufatura, conflitos, tamanho de buffers, relações de precedência, recursos compartilhados e interações estruturais podem ser modelados eficientemente (DESROCHERS; AI-JAAR, 1994; JAIN; JAIN; SINGH, 2006).

A movimentação das marcações pode ser usada para estudar o comportamento dinâmico da modelagem de sistemas de eventos discretos (ZHOU *et al.*, 1995).

2.2.1 Modelagem em Redes de Petri

Vários trabalhos têm sido desenvolvidos usando redes de Petri para modelagem e controle de sistemas de manufatura. Algumas extensões de redes de

Petri têm sido propostas para melhor se adequar a resolução dos problemas encontrados.

Tüysüz e Kahraman (2010) apresentam uma abordagem para modelagem e análise de tempo de execução de uma célula flexível de manufatura usando redes de Petri estocástica com conjuntos fuzzy para melhorar o poder de modelagem e análise de sistemas complexos. Este tipo de modelagem representa incertezas e o parâmetro de distribuição exponencial é representado por um número triangular fuzzy.

O trabalho de Morandin *et al.* (2007) propõe uma estratégia de modelagem para o planejamento, controle e inter travamento de um AMS usando Redes de Petri Virtual, considerando a comunicação com níveis superiores através de um supervisor. Redes de Petri colorida foram usadas para representar produtos e sinais. Redes de Petri temporal para representar o tempo de montagem e o de deslocamento. Rede de Petri modular também é utilizada, e para conectar os módulos, utiliza redes de Petri virtual.

O trabalho de Wu, Zhou, Hu (2007) compara os dois métodos de modelagem, o modelo em Redes de Petri Orientada a Processos (POPN – Process Oriented Petri Net) e o modelo em Redes de Petri Orientada a Recursos (ROPN – Resource Oriented Petri Nets).

Para a modelagem em POPN são modeladas as seqüências do processo de produção, os recursos, a solicitação de recursos para as operações. No primeiro passo as atividades são identificadas para o processamento de cada peça, incluindo a execução das operações e o armazenamento.

Para a modelagem em ROPN são modelados os recursos do sistema e para cada tipo de produto são modelados os processo de produção como uma sub-rede. As sub redes são unidas e cores são introduzidas para descrever os processos.

Zhang *et al.* (2007) apresenta um novo modelo de FMS em redes de Petri orientada a objetos e descreve as relações das peças do sistema. O modelo é particionado em quatro classes de objetos, peças, máquinas, transporte e armazenamento. Cada objeto é modelado em uma sub-rede que se comunica e troca informações com outros objetos.

No trabalho de Tsinarakis, Tsourveloudis, e Valavanis (2005) uma rede de Petri temporal é usada para modelar a flexibilidade de operações e de roteiros em um sistema de produção. Quatro modelos são construídos envolvendo diferentes

níveis de flexibilidade. O modelo consiste em duas partes, um que define o estado das máquinas em cada instante de tempo e o outro que descreve os processos e as peças transportadas no sistema.

Em Odrey e Mejía (2005), a abordagem desenvolvida é baseada na integração de sub modelos de redes de Petri com o modelo geral do sistema de manufatura. Os sub modelos são os planos de recuperação que são incorporados nas estações de trabalho e consistem em um sequência de passos necessários para fazer com que o sistema retorne a um estado normal.

Noureddine e Martineau (2005) apresentam uma abordagem para modelar e analisar um FMS usando redes de Petri. Primeiramente é definida a descrição física do modelo e então construído o modelo conceitual. Cada recurso é considerado como uma entidade operacional associada a uma entidade de armazenamento. Entidades adjacentes se comunicam através do modelo genérico. O modelo conceitual é obtido a partir da notação genérica e dos processos de manufatura.

Apesar do uso bem sucedido das Redes de Petri em diversos cenários, estas também possuem alguns aspectos que podem atribuir em desvantagens quando usadas em sistemas muito complexos.

Quando é requerida uma modelagem mais detalhada com uma enorme combinação de estados, o modelo tende a ser muito grande. Assim, a sua análise tornar-se difícil e complicada. Embora muitas informações possam ser obtidas através da árvore de alcançabilidade, se torna impraticável para sistemas complexos, como é o caso dos FMSs, uma vez que o número de marcações alcançáveis tende a ser muito grande (Palomino, 1995).

2.2.2 Redes de Petri Colorida

As redes de Petri de alto nível, mais especificamente as CPN, têm o mesmo poder de descrição e análise das redes de Petri de baixo nível (AGUIRE, 2007).

CPN foi introduzida por Jensen em 1988. Uma CPN consiste em três partes: a estrutura da rede, um conjunto de declarações e um conjunto de inscrições.

Como as redes de Petri, as CPNs são também grafos direcionados e bipartidos. Diferem de uma rede ordinária pela presença de marcas com tipos complexos, chamados de cores e, ao invés de pesos inteiros, os arcos são associados às inscrições que determinam dinamicamente quantas e quais marcas

devem ser removidas ou adicionadas aos lugares associados, na ocorrência de uma transição.

Formalmente, uma rede de Petri colorida é uma estrutura $(\Sigma, P, T, A, N, C, G, E, I)$, onde:

- Σ é um conjunto finito não-vazio de tipos de dados chamados conjuntos coloridos;
- P é um conjunto finito de lugares;
- T é um conjunto finito de transições;
- A é um conjunto finito de arcos onde $P \cap T = P \cap A = \emptyset$;
- N é uma função $N: A \rightarrow (P \times T) \cup (T \times P)$;
- C é uma função colorida onde $C: P \rightarrow \Sigma$;
- G é uma função de guarda booleana nas transições;
- E é uma função de expressão do arco;
- I é a função de inicialização.

O conjunto de declarações indica a natureza dos elementos citados nas diversas inscrições e é semelhante a área de declarações de uma linguagem de programação qualquer.

Na teoria de CPN usa-se a expressão conjunto de cores em substituição a tipos de dados e, por consequência, cada valor é denominado cor, que pode ser de um tipo arbitrário de dados (inteiro, lista, string, etc.). Desta forma, cada lugar na estrutura é associado a um conjunto de cores, que indica o tipo de marcas que o lugar pode conter. Isto é, para um dado lugar, todas as marcas devem ter cores que pertencem a um mesmo tipo.

A guarda de uma transição é uma expressão booleana que deve ser satisfeita para que a transição dispare. Variáveis de transição referem-se ao conjunto de variáveis presentes nas inscrições dos arcos e na guarda da referida transição. Uma ligação (do inglês *binding*) é a substituição de cada variável da transição por um valor. É requerido, entretanto, que as cores pertençam aos conjuntos de cores apropriados e que impliquem a avaliação da guarda como verdadeira.

Em cada marcação, a ocorrência de uma transição sob uma determinada ligação é dita habilitada se todos os seus lugares de entrada tiverem marcas suficientes para satisfazer as expressões dos arcos.

Cada expressão deve ser devidamente avaliada segundo as substituições determinadas pela ligação, a fim de determinar quantas e quais marcas são requeridas nos lugares de entrada. Caso a transição ocorra, então são retiradas marcas dos lugares de entrada e depositadas novas marcas nos lugares de saída. A quantidade de marcas é determinada também pela avaliação das expressões dos arcos segundo as substituições implicadas pela ligação.

A Figura 2.1 mostra um modelo genérico de CPN.

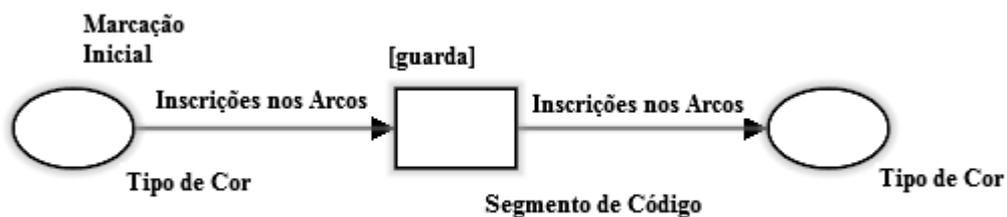


Figura 2. 1 – Modelo de CPN

2.2.3 Conflitos em Redes de Petri

Dois eventos estão em conflito se a ocorrência de um evento desabilita a ocorrência do outro. Conflitos são necessários para a modelagem de várias situações básicas, como exclusão mútua, semáforos e compartilhamento de recursos. Usualmente, conflitos precisam de tratamento especial e podem requerer intervenções (GOMES, 2005).

Alguns critérios para atribuir prioridades podem ser definidos e integrados ao controle do sistema para a resolução dos conflitos.

Em um sistema de manufatura, conflitos podem ocorrer quando uma variedade de peças tem roteiros alternativos ou uma peça deve ser selecionada em um dado conjunto quando requisitam o mesmo recurso (TUNCEL, 2007).

A modelagem de conflitos em uma rede de Petri pertence à classe das redes de livre-escolha (BRÚZ; ZHOU, 2007), onde dois ou mais arcos saindo do mesmo lugar com destino a transições distintas gera diferentes caminhos de execução, pois somente uma única, dentre as transição habilitadas, pode ser disparada em certo instante de tempo. Um conflito estrutural pode ser visto na Figura 2.2.

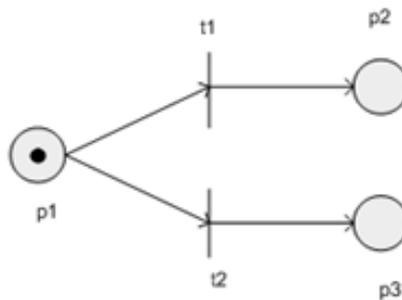


Figura 2. 2 – Conflito em Redes de Petri

Pode-se generalizar que conflitos em redes de Petri caracterizam-se pela existência de dada marcação que possua duas ou mais possibilidades, excludentes, de evolução (MARTINS MELO, 2006)

Em uma rede de Petri colorida outro tipo de conflito também pode ser encontrado. Quando vários elementos em um mesmo lugar estão individualmente habilitados, mas não concorrentemente habilitados, então esta transição está em conflito com ela mesma (JENSEN, 1997). Na Figura 2.3, a transição *EnviaPacote* tem seis possibilidades de disparo conforme as marcações no lugar *PacotesPEnviar*. Como estas marcas não podem ser disparadas concorrentemente, elas estão em conflito.

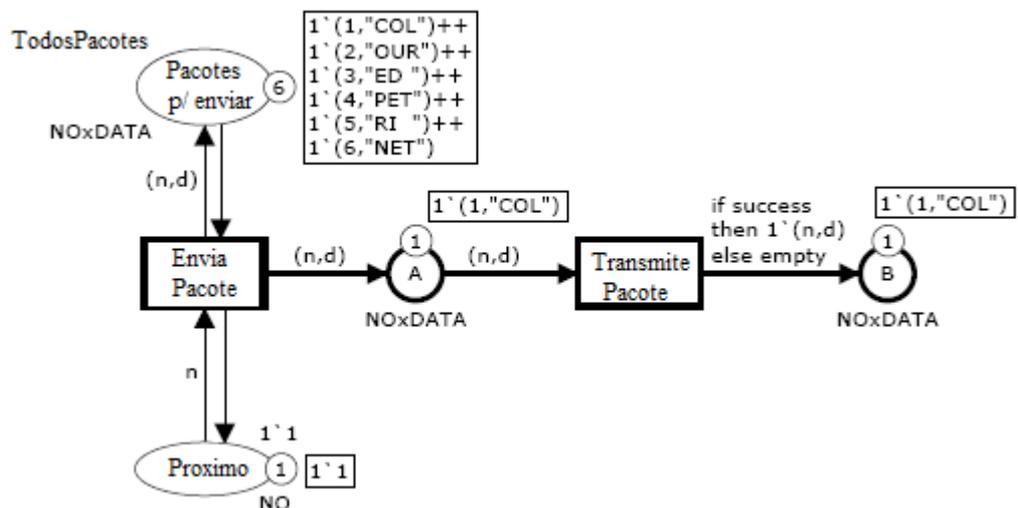


Figura 2. 3 – Conflito em CPN (adaptado de Jensen, Kristensen e Wells, 2007)

2.3 Abordagens e Metodologias de Programação no Controle de Sistemas de Manufatura

Programação da produção é um processo de tomada de decisão relacionado à alocação de recursos limitados entre as tarefas ao longo do tempo (TUNCEL, 2007). Esta tem como objetivo maximizar a taxa de produção e minimizar o tempo total do fluxo de operações através da ordenação de entrada dos lotes a serem executados na produção (AGUIRE, 2007).

Enquanto a flexibilidade de um FMS possibilita um grande número de escolha de recursos e roteiros e uma maior produtividade, isso impõe um desafio, a alocação dos recursos em diferentes processos na produção de cada produto (LEE; DICESARE, 1994)

Um sistema com flexibilidade de roteiros de produção torna possível a produção de um produto através de vários roteiros alternativos. Este tipo de flexibilidade torna possível balancear a carga de trabalho e proporciona uma melhor sincronização para o uso das máquinas, provendo ao sistema a possibilidade de expandir sua capacidade quando necessária e reduzindo a probabilidade de a linha de produção parar quando eventos inesperados ocorrem, como quebra ou manutenção das máquinas (CHANG, 2007).

O problema da produção está na alocação de recursos no tempo e na sequência correta, de forma que o resultado gerado seja a conclusão do conjunto de tarefas no menor tempo possível (AGUIRE, 2007).

O controlador tem que lidar com o ambiente dinâmico no qual um FMS opera, e assim, tem que lidar com roteiros alternativos e eventos imprevistos e inevitáveis que podem causar uma mudança nos planos de produção.

É evidente que o desempenho do sistema é altamente dependente da seleção de uma eficiente política de programação para ser usada no controle do sistema. Para utilizar todas as capacidades do sistema, o sistema de controle tem que estar habilitado a cooperar com as mudanças no estado do chão de fábrica e mudanças nos objetivos operacionais do sistema.

A maioria dos estudos para desenvolvimento de metodologias de controle de sistemas de manufatura usa programação dinâmica ou reativa. Estes tipos de programação permitem que em cada ponto de decisão seja determinada uma regra

ou política de programação apropriada em tempo real, baseada no estado corrente do chão de fábrica (SHNITS; SINREICH, 2006).

Programação dinâmica (também chamada de adaptativa ou online) não cria ou modifica uma programação da produção. Ao invés, métodos descentralizados de controle da produção despacham trabalhos quando necessários e usam informações disponíveis no momento do despacho. Estas abordagens usam regras de despacho e outras heurísticas para priorizar trabalhos esperando para serem processados por um recurso (VIEIRA; HERRMANN; LIN, 2003).

Já uma programação reativa é uma estratégia comum de reprogramação. Esta abordagem tem dois passos. Primeiramente, uma programação da produção estática é gerada e, no próximo passo, é atualizada, revisada e modificada em resposta às interrupções e outros eventos para minimizar o impacto no desempenho do sistema (VIEIRA; HERRMANN; LIN, 2003).

As abordagens que usam somente programação estática usualmente resolvem o problema por abordagens ótimas, mas podem facilmente se tornarem inviáveis em um ambiente de manufatura real, já que assumem algumas suposições não realísticas (LEE, 2008). No contexto da programação estática, uma mudança no planejamento da produção ou falha nos recursos implica uma parcial ou total reconfiguração do sistema.

Programação dinâmica tem suas vantagens sobre a programação estática. Em um sistema de manufatura, uma programação estática pode se tornar obsoleta após um pequeno período de tempo, enquanto a programação dinâmica pode se adaptar às mudanças que ocorrem no sistema (OUELHADJ; PETROVIC, 2009; SINREICH; SHNITS, 2006).

Programação dinâmica é relacionada ao controle em tempo real, já que as decisões são baseadas no estado corrente do sistema de manufatura (VIEIRA; HERRMANN; LIN, 2003).

Programação dinâmica tem sido resolvida usando as seguintes técnicas: heurísticas, meta-heurísticas, sistemas baseados em conhecimento, lógica fuzzy, redes neurais e sistemas multi-agentes (OUELHADJ; PETROVIC, 2009). Alguns trabalhos sobre programação dinâmica pertencentes a algumas categorias definidas por Ouelhadj e Petrovic, (2009) serão revisados.

Os trabalhos de Turgay (2009) e Xiang e Lee (2008) usaram sistemas baseados em agentes para resolver o problema do controle e da programação dinâmica.

No trabalho de Turgay (2009), a modelagem de um sistema de controle baseado em agentes foi representada através de redes de Petri e o desempenho foi avaliado através do tempo associado aos lugares da rede. O objetivo foi gerar uma programação eficiente para o processamento dos produtos. O controle de um sistema baseado em agentes é um mecanismo que monitora continuamente o estado e as condições do sistema e realiza decisões relacionadas à produção. Decisões como qual peça será processada e em quanto tempo, o robô ou o AGV que será usado, qual plano de processamento será implementado com qual seqüência de operações é realizada pelo agente.

Primeiramente, os dados e a base de regras são determinados considerando os componentes. Agentes independentes são considerados de acordo com o recurso do sistema como agente supervisor, agente de peças, agente das máquinas, agente do robô e agente do AGV. O agente supervisor é o mecanismo que realiza as decisões. Esse agente decide, por exemplo, qual peça será processada e qual agente será usado no processamento.

O trabalho de Xiang e Lee (2008) teve como objetivo construir uma programação dinâmica eficiente baseada em agentes para um sistema de manufatura. Um modelo baseado no algoritmo de Colônia de Formigas foi proposto para ser combinado com um agente local para otimizar o desempenho do sistema baseado em alguns critérios. Os agentes do sistema foram modelados de forma autônoma de acordo com o conhecimento relacionado às suas funções e objetivos, como agente do chão de fábrica, agente do pedido de produção, agente das máquinas, entre outros. Diferentes agentes se comunicam e coordenam suas atividades.

Os trabalhos de Lee (2008), Bilge, Firat, Albey (2008), Caprihan, Kumar, Stecke (2006), Chan, Chan, Kazerooni (2003) e Srinoi *et al.* (2002) usaram abordagens fuzzy para solução de conflitos e seleção de roteiros.

O trabalho de Lee (2008) propôs um método para extrair regras fuzzy automaticamente de uma base de dados que foi continuamente atualizada para resolver problemas de programação adaptativa. Uma base de regras foi construída

de acordo com um classificador de aprendizado dinâmico fuzzy baseado em treinamento de dados acumulados por um método de simulação.

O método engloba um esquema para aquisição automática de conhecimento para construir uma base de regras fuzzy. A simulação foi usada para avaliar o desempenho das regras candidatas. Em um ponto de decisão, cada regra de despacho foi avaliada e, a regra com melhor desempenho foi adicionada na base. A decisão de despacho foi selecionar uma tarefa entre àquelas que estão esperando para ser processada. Quando uma estação acaba de processar um peça, deve determinar qual peça será processada a seguir entre aquelas que esperam. Para a escolha do AGV o mais perto foi selecionado.

O trabalho de Bilge, Firat, Albey (2008) desenvolveu uma estratégia para para seleção de roteiros durante o controle em tempo real de um FMS. Este trabalho adotou uma abordagem em lógica fuzzy para gerar uma relação entre os vários critérios. Apresentou três algoritmos que incorporam os três critérios em diferentes maneiras. Tempo estimado de término, tempo mínimo estimado de fluxo para cada alternativa e um algoritmo fuzzy.

O trabalho de Caprihan, Kumar, Stecke (2006) apresentou uma estratégia de despacho baseado em lógica fuzzy para cooperar com os atrasos de informações. O sistema apresentava flexibilidade de roteiros. Assim, após cada operação uma decisão de controle on-line era solicitada para escalar uma máquina apropriada onde a peça deveria ser despachada. As variáveis para a decisão de despacho eram primeiramente identificadas. Para cada alternativa de despacho, o grau de verdade era avaliado e a alternativa com maior grau era executada.

Chan, Chan, Kazerooni (2003) apresentou um sistema fuzzy inteligente em tempo real para programação da produção em um FMS. A abordagem em lógica fuzzy foi proposta para melhorar o desempenho considerando várias medidas, como minimizar o número de máquinas bloqueadas, o tempo total de processamento, o número de etapas de processamento e a média de tempo de fluxo.

Sua abordagem focou no estado do sistema para atribuir prioridades as peças que estavam esperando para serem processadas. Um modelo de simulação do FMS foi construído, e em vários pontos do sistema, pontos de decisão eram considerados para verificar o seu estado. Quando um AGV estava livre, ele deveria voltar à área de estacionamento. Assim, passava por vários pontos de interseção

onde verificava solicitações. Caso existisse alguma solicitação para ele, ele iria até a estação que o requisitou.

O trabalho de Srinoi *et al.* (2002) apresentou um modelo de programação baseado em fuzzy para fazer o controle em tempo real e solucionar problemas da programação. O programador fuzzy decidia qual a melhor rota entre as alternativas seria selecionada para processar uma peça.

A seleção da rota dependia de três fatores, o número de peças esperando no buffer de entrada de cada máquina considerando seu tempo de processamento total, o tempo restante para completar a operação requisitada e o tempo de viagem da peça através da rota escolhida.

Os trabalhos de Tuncel (2007), Zhang, Jiang, Guo, (2007), Delgado e Llano (2007) e Shnits e Sinreich, (2006) usaram regras de despacho ou regras heurísticas para atribuir prioridades aos processos conflitantes.

No trabalho de Tuncel (2007), uma abordagem baseada em regras heurísticas para programação dinâmica e controle de FMS, que integra o carregamento, a entrada de peças e a escolha de roteiros foi proposta.

Para atingir uma utilização eficiente dos recursos e encontrar uma sequência melhor de operações, uma política de alocação de recursos em tempo real foi desenvolvida. As decisões consideraram as condições do chão de fábrica. A modelagem do sistema foi realizada usando redes de Petri de Alto-nível e considerou a sequência de execução das transições para análise. Um conjunto de regras de produção “se-então” foi construída baseada na heurística desenvolvida. Para construção da regra foram consideradas informações como o número de peças conflitantes, plano de processamento do produto, número de operações restantes. Assim a peça era selecionada de acordo com regras de prioridade.

Zhang, Jiang, Guo, (2007), propuseram um mecanismo para programação em tempo real baseado em simulação no qual regras de despacho e estratégias de reparo variam dinamicamente baseado em informações de tempo real. O estudo de caso examinou o impacto de desempenho do método de acordo com alguns critérios como vazão, entrega dos produtos e tempo de processamento. Um algoritmo que combina várias regras de despacho foi desenvolvido baseado na teoria de restrições. Um usuário monitorou o estado corrente do sistema e as medidas de desempenho de um intervalo prévio de produção e fornecia os valores desejados

para a produção do próximo período. O simulador avaliou as regras de despacho e estratégias de reparo e selecionava a melhor combinação.

Delgadillo e Llano (2007) propuseram uma abordagem para modelagem e programação de sistemas de manufatura usando redes de Petri e regras de despacho para resolver eventuais conflitos. A execução do algoritmo proposto corresponde aos disparos das transições até que o último estado seja alcançado. Um conflito operacional ocorre quando várias transições estão habilitadas, ou seja, existe uma competição pelo mesmo recurso. As medidas de desempenho consideradas foram minimizar o tempo total de processamento, maximizar a utilização de recursos e atingir o prazo de entrega.

Shnits e Sinreich (2006) apresentaram uma metodologia para controle multicritério de FMS. A metodologia foi baseada em um mecanismo de decisões de dois níveis para desenvolver uma programação adaptativa. O primeiro passo é selecionar um critério de decisão e um conjunto de regras de programação usando um algoritmo baseado em regras de acordo com o estado do chão de fábrica, requisitos da produção e prioridades do sistema. O segundo passo foi selecionar por simulação a regra de programação que obteve o melhor desempenho de acordo com o critério selecionado. Para selecionar o critério de operação foram considerados o estado do chão de fábrica, requisitos da produção e prioridades do sistema. A metodologia proposta expande a programação adaptativa habilitando mudanças, não somente nas regras de despacho, mas também no critério de objetivo que governa as operações do sistema e afeta a seleção de uma regra apropriada.

2.4 Considerações Finais

Várias modelagens de sistemas de manufatura em redes de Petri têm sido propostas, assim como várias extensões da rede. Este trabalho optou por usar as redes de Petri de alto nível, especificamente as redes de Petri coloridas. Nestas redes é possível modelar os recursos do sistema, inclusão de tipos na rede e funções que podem analisar a situação da rede para que uma transição seja disparada ou não.

Uma política de alocação de recursos é definida. No entanto, este trabalho não usa um sistema de apoio para realizar as decisões quando conflitos ocorrem. A tomada de decisão é incorporada na própria rede de Petri do sistema de manufatura, tornando a rede independente de outros sistemas, como sistemas fuzzy ou sistemas baseados em agentes.

Este trabalho lida com sistemas de produção que operam em ambientes altamente dinâmicos, caracterizados pela chegada aleatória de pedidos, quebra de máquinas, mudanças nas datas de entrega, entre outras. A literatura diz que para este tipo de ambiente, uma programação dinâmica parece ser mais eficiente do que outros métodos de programação (SHINITS; SINREICH, 2006).

Outra motivação para o uso da programação dinâmica é descentralizar a tomada de decisão dentro do controle sistema de manufatura, reduzir a complexidade e custo do sistema, aumentar a flexibilidade e elevar a tolerância a falhas (OUELHADJ; PETROVIC, 2009).

Como observado nos trabalhos citados, a maioria dos trabalhos baseados em agentes e em regras de despacho consideram um único critério de desempate. Regras simples geralmente não são suficientes para capturar a complexidade e o dinamismo de um ambiente de manufatura e seu controle em tempo real. O problema é tipicamente um problema multicritério com vários fatores de decisão para serem ponderados (BILGE; FIRAT; ALBEY, 2008).

Uma decisão em um ambiente de programação dinâmica é realizada somente se mais de uma peça compete pelo mesmo recurso (VIEIRA; HERRMANN; LIN, 2003). Os trabalhos que usam simulação para encontrar uma regra eficiente para atribuir prioridades gastam um tempo considerável para encontrar àquela que melhor atende os requisitos do sistema. No entanto, não avaliam se decisão é necessária.

De acordo com a revisão da literatura, três critérios são mais usados para as decisões: como tempo de processamento, tempo de espera e tempo de deslocamento (BILGE; FIRAT; ALBEY, 2008). Neste trabalho, as variáveis que serão consideradas são o tamanho do lote, o prazo de entrega, o nível dos buffers dos roteiros e o tempo de processamento do roteiro. E esses fatores serão ponderados em regras fuzzy. Dos critérios mais usados na literatura, este trabalho só considerou o tempo de processamento do produto, mais especificamente o tempo deste em roteiros específicos. Ao invés de utilizar informações individuais do produto, como

tempo de espera no buffer e tempo de deslocamento, foram utilizadas informações da gestão da produção, como o tamanho do lote e prazo de entrega.

Capítulo 3

REDES DE PETRI FUZZY

Este capítulo apresenta uma introdução sobre Redes de Petri Fuzzy e suas abordagens. Depois decorre sobre as Redes de Petri Coloridas Fuzzy e a aplicação destas em sistemas de manufatura.

3.1 Considerações Iniciais

Rede de Petri ordinária é um grafo que permite a representação de diferentes interações entre eventos discretos, como concorrência, sincronização, precedência, escolhas, entre outras. Ela também possui aspectos matemáticos que permitem a modelagem formal destas interações e a análise das propriedades do sistema modelado. No entanto, são inadequadas para lidar com incertezas (CARDOSO; CHEZÁLVIEL, 1997).

A modelagem de regras fuzzy, conceitos de raciocínio fuzzy e lógica proposicional foram combinadas com a representação de redes de Petri. Assim, Rede de Petri Fuzzy (FPN – Fuzzy Petri Net) foi introduzida para processar dados fuzzy e realizar o raciocínio fuzzy automaticamente.

Conjuntos fuzzy, introduzidos por Zadeh em 1965, permitem a representação de classes onde os limites não são bem determinados. Isto permite considerar categorias de informações vagas ou flexíveis que podem ser usadas por especialistas em regras fuzzy (CARDOSO; CAMARGO, 1999).

O trabalho de Cardoso e Camargo (1999) divide em duas abordagens os modelos de FPN. A primeira abrange modelos de sistema de raciocínio, os componentes compreendem representação do conhecimento. Na maioria dos casos

as transições são associadas a regras fuzzy e toda a rede descreve um sistema especialista fuzzy ou um sistema de controle fuzzy. Na segunda abordagem, sistemas dinâmicos físicos são modelados e as transições denotam possíveis mudanças de estado.

A principal diferença entre as abordagens é que, enquanto na primeira uma FPN descreve o mecanismo encadeado de um conjunto de regras e a marcação corresponde a uma parte do processo de raciocínio, na segunda, esta representa o estado dinâmico e a marcação denota o conhecimento sobre o estado do sistema em certo tempo.

Looney (1988) é considerado o primeiro trabalho publicado relacionado à FPN (CARDOSO; CAMARGO, 1999). Looney modificou um modelo em redes de Petri para permitir raciocínio baseado em regras usando lógica proposicional.

Sob o mesmo nome, diferentes modelos foram propostos ao longo dos anos para atender os mais diversos propósitos em várias áreas de pesquisa. Alguns trabalhos sobre as duas abordagens sobre FPN serão revisados, assim como a extensão Redes de Petri Fuzzy Colorida e os trabalhos relacionados a sistemas de manufatura.

3.2 Modelo de Redes de Petri Fuzzy para Representação do Conhecimento

O modelo para representação de regras de produção fuzzy de um sistema baseado em regras foi formalizado por Chen *et al.* (1990). Este permite uma representação estruturada do conhecimento. Neste modelo, lugares são associados às proposições e as transições, com seus respectivos lugares de entrada e de saída, representam regras de produção.

O modelo de Chen *et al.* (1990) é definido por:

$$FPN = (P, T, D, I, O, f, \alpha, \beta),$$

onde

- $P = \{p_1, p_2, \dots, p_n\}$ é o conjunto finito de lugares,
- $T = \{t_1, t_2, \dots, t_n\}$ é o conjunto finito de transições,
- $D = \{d_1, d_2, \dots, d_n\}$ é o conjunto finito de proposições,

- $I: P \rightarrow T$ é a função de entrada, um mapeamento de lugares para transições.
- $O: T \rightarrow P$ é a função de saída, um mapeamento de transições para lugares.
- $f: T \rightarrow [0, 1]$ associa as transições a valores reais entre zero e um.
- $\alpha: P \rightarrow [0, 1]$ associa os lugares aos seus valores reais entre zero e um.
- $\beta: P \rightarrow D$ associa os lugares às suas proposições.

Uma regra de produção fuzzy é uma declaração condicional e é usada para determinar se um antecedente infere em uma consequência. A fórmula geral de uma regra fuzzy é dada por:

$$R_i: \text{SE } d_j \text{ ENTÃO } d_k \text{ (GC} = \mu_i\text{),}$$

onde d_j e d_k são proposições que podem conter algumas variáveis fuzzy e são associadas aos lugares p_j e p_k da rede. A Figura 3.1 mostra o modelo em FPN para a fórmula geral de regra fuzzy mostrada acima.

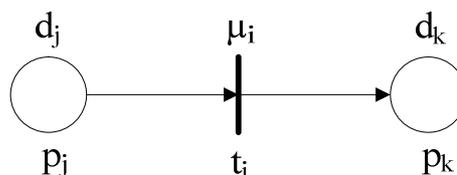


Figura 3. 1 – Uma Fuzzy Petri Net

A parte antecedente da regra representa a premissa e a parte consequente representa a conclusão ou decisão. Cada lugar pode ou não conter uma marca associada a um valor verdade. O grau de verdade da proposição associada a um lugar é definida como y_j e y_k e é um valor real entre $[0,1]$, calculado pela função de pertinência do conjunto correspondente. O valor $\mu_i = f(t_i)$ é o grau de certeza (GC) entre $[0,1]$ que representa a confiança na regra. As relações entre lugares a transições e transições a lugares são representadas por arcos direcionados e definidas nas funções I e O , respectivamente.

Se a parte antecedente ou a consequência de uma regra de produção fuzzy contém os conectores “e” ou “ou”, então é chamada de regra de produção fuzzy composta. De acordo com Chen *et al.* (1990), as regras compostas podem ser distinguidas em tipos:

- Tipo 1: SE d_{j1} E d_{j2} E ... E d_{jn} ENTÃO d_k (GC = μ_i)
- Tipo 2: SE d_{j1} ENTÃO d_{k1} E d_{k2} E ... E d_{kn} (GC = μ_i)
- Tipo 3: SE d_{j1} OU d_{j2} OU ... OU d_{jn} ENTÃO d_k (GC = μ_i)
- Tipo 4: SE d_{j1} ENTÃO d_{k1} OU d_{k2} OU ... O d_{kn} (GC = μ_i)

As regras dos Tipo 1, Tipo 2, Tipo 3 e Tipo 4 podem ser modeladas por uma FPN como mostrado nas Figura 3.2, Figura 3.3, Figura 3.4 e Figura 3.5, respectivamente. No entanto, como o Tipo 4 não realiza implicações específicas, não é permitido a presença deste tipo na base de conhecimento.

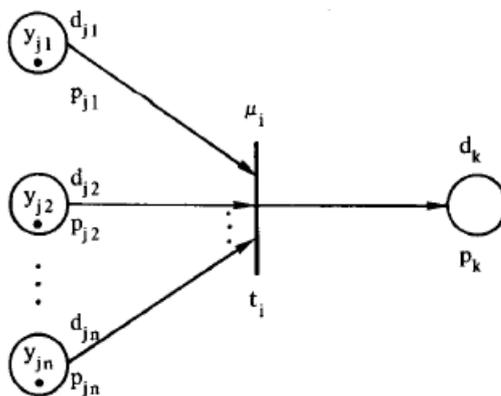


Figura 3. 2 - Representação em Redes de Petri Fuzzy da regra Tipo 1 (CHEN et al., 1990).

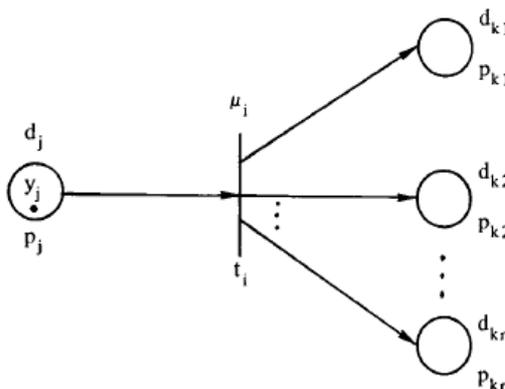


Figura 3. 3 - Representação em Redes de Petri Fuzzy da regra Tipo 2 (CHEN et al., 1990).

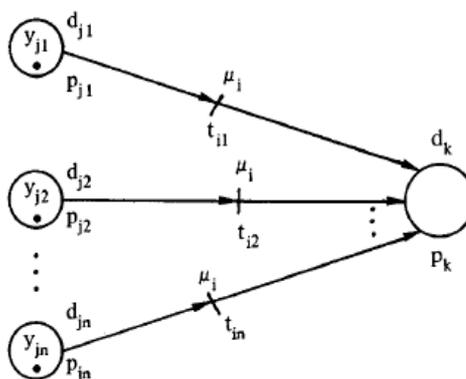


Figura 3. 4 - Representação em Redes de Petri Fuzzy da regra Tipo 3 (CHEN et al., 1990).

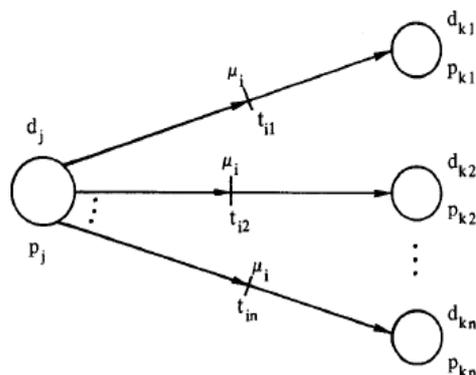


Figura 3. 5 - Representação em Redes de Petri Fuzzy da regra Tipo 4 (CHEN et al., 1990).

Em uma rede marcada, após o disparo de uma transição, o valor do conseqüente da regra é determinado de acordo com a Tabela 1

Tabela 3. 1. Funções para cálculo do conseqüente das regras

Tipo	Regra	Grau do Consequente
1	SE d_{j1} E d_{j2} E...E d_{jn} ENTÃO d_k (GC = μ_i)	$y_k = \mu_i * \min(y_{i1}, \dots, y_{in})$
2	SE d_{j1} ENTÃO d_{k1} E d_{k2} E...E d_{kn} (GC = μ_i)	$y_{km} = \mu_i * y_{j1}, m = 1 .. n$
3	SE d_{j1} OU d_{j2} OU...OU d_{jn} ENTÃO d_k (GC = μ_i)	$y_k = \max(y_{i1} * \mu_{i1}, \dots, y_{in} * \mu_{in})$

Um algoritmo também é proposto para realizar o raciocínio automático fuzzy e para determinar onde existe uma relação antecedente-consequente entre duas proposições. Assim, se o grau de verdade de uma proposição d_i é dada, é possível avaliar o grau da proposição d_j . No algoritmo, as funções para o cálculo do conseqüente das regras são usadas e se um nó final recebe duas ou mais marcações, o valor máximo é considerado.

Segundo Cardoso *et al.* (1996), o benefício desta abordagem é que a concorrência (varias regras podem ser aplicadas em qualquer ordem) e as escolhas (seleção de uma regra em um conjunto de regras conflitantes) podem ser formalmente capturadas. Assim, este tipo de FPN tem o propósito de integrar o controle fuzzy com baixo custo a um controle de eventos discretos baseado em rede de Petri.

Em outros trabalhos, as proposições das regras não são representadas pelos lugares, mas toda a regra de produção é representada por transições e a rede modela o processo de inferência e defuzzificação, ao invés de um modelo de raciocínio como em Chen *et al.* (1990), como mostrado na Figura 3.6.

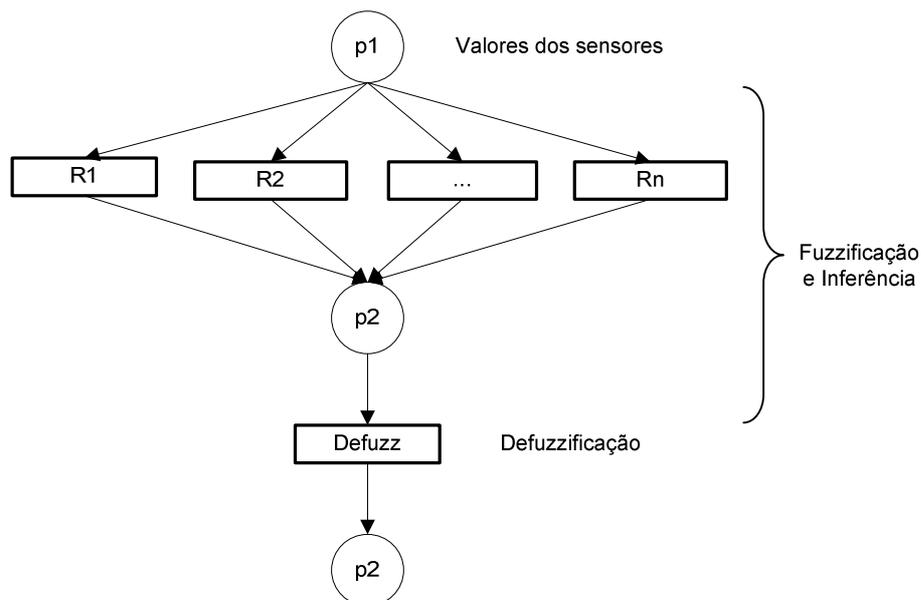


Figura 3. 6 - Outra modelagem de Rede de Petri Fuzzy (Adaptado de Lundell, Tang e Nygard, 2005).

Os trabalhos de Hanna (1994), Hanna, Buck e Smith (1994), Yeung *et al.* (1999), Lundell, Tang e Nygard (2005), Ting *et al.* (2008) usam esta modelagem para descrever o sistema baseado em regras fuzzy.

3.3 Modelo de Representação para Sistema Físico

A segunda abordagem do trabalho de Cardoso e Camargo (1999) sobre FPN pode ser representada pelo trabalho de Cardoso (1999). Neste trabalho, uma rede de Petri fuzzy é um modelo de um sistema físico. Um lugar representa um possível estado parcial, uma transição denota uma possível mudança de estado e, uma seqüência de disparos representa um possível comportamento. O modelo apresentado neste trabalho associa uma distribuição de possibilidade à localização de uma marca. Uma marcação fuzzy diz que cada localização para a marca tem uma distribuição de possibilidade ao longo do conjunto de lugares.

O trabalho de Cai *et al.* (2008) apresenta um modelo de controle de tráfego de trens baseado neste tipo FPN. Intervalos temporais fuzzy são adotados para lidar com a incerteza no sistema.

3.4 Redes de Petri Coloridas Fuzzy

Algumas variações das Redes de Petri Fuzzy foram propostas para se adequar melhor ao domínio dos problemas, como as Redes de Petri Coloridas Fuzzy (FCPN - Fuzzy Coloured Petri Net).

As FCPN combinam os recursos das CPN com os recursos das FPN. Em uma FCPN, o valor verdade de uma proposição é associado a um atributo da marca colorida e o raciocínio é descrito pelas inscrições dos arcos e das transições, pois é necessário representar as funções de pertinência.

O modelo de Yeung *et al.* (1996) define uma FCPN como uma junção da FPN proposta por Chen *et al.* (1990) e a CPN proposta por Jensen (1992). Assim, é definida como a tupla $(\Sigma, P, T, D, A, N, C, G, E, \beta, f, I)$, onde

- Σ é um conjunto finito não-vazio de tipos de dados chamados conjuntos coloridos;
- $P = \{ P_C, P_F \}$ é um conjunto finito de lugares;
 - $P_C = \{ pc_1, pc_2, \dots, pc_m \}$ é um conjunto finito de lugares que modelam o controle dinâmico do sistema, chamado de lugares de controle;
 - $P_F = \{ pf_1, pf_2, \dots, pf_n \}$ é um conjunto finito de lugares que modelam as regras fuzzy, chamado de lugares fuzzy;
 - $P_C \cap P_F = \emptyset$,
- $T = \{ T_C, T_F \}$ é um conjunto finito de transições;
 - $T_C = \{ tc_1, tc_2, \dots, tc_i \}$ é um conjunto finito de transições que conectam lugares de controle, chamadas de transições de controle;
 - $T_F = \{ tf_1, tf_2, \dots, tf_j \}$ é um conjunto finito de transições que conectam lugares fuzzy, chamadas de transições fuzzy;
 - $T_C \cap T_F = \emptyset$,
- $D = \{ d_1, d_2, \dots, d_h \}$ é um conjunto finito de proposições, $| P_F | = | D |$;
- A é um conjunto finito de arcos onde $P \cap T = P \cap A = \emptyset$;
- N é uma função $N: A \rightarrow (P \times T) \cup (T \times P)$;
- C é uma função colorida onde $C: P \rightarrow \Sigma$;
- G é uma função de guarda booleana nas transições;

- E é uma função de expressão do arco
- $\beta : P_F \rightarrow D$ é uma função bijetiva que conecta lugares fuzzy à proposições;
- $f : T \rightarrow [0,1]$ é uma função de associação que associa um grau de verdade a cada cor usada em cada transição fuzzy;
- I é a função de inicialização.

3.5 Aplicação em Sistemas de Manufatura

Redes de Petri Fuzzy e Redes de Petri Coloridas Fuzzy têm sido usadas para modelagem, planejamento e controle de sistemas de manufatura.

O trabalho de Fengler *et. al* (1996), propôs uma FCPN para determinar a sequência de produção e os roteiros dos produtos de um sistema de manufatura. Os roteiros foram definidos a partir da evolução da prioridade dos trabalhos ou dos tempos de processamento por uma função fuzzy associada às transições representando regras fuzzy. A cor associada às marcas nos lugares de saída das transições é o grau de pertinência de cada trabalho. O trabalho que será executado é escolhido de acordo com o valor da marca.

Já no trabalho de Yeung (1999), uma FCPN foi proposta para modelar e controlar a rotação do braço de um robô. Nesta FCPN, o sistema foi dividido em dois subsistemas. A modelagem em CPN do sequenciamento das operações de um robô e outro o sistema de controle fuzzy modelado em FPN. Na FPN, as regras de produção foram representadas por transições e os processos de fuzzificação, inferência e defuzzificação foram modelados.

O trabalho de Racoceanu *et al.* (2003) propôs uma ferramenta de modelagem e análise para monitoramento de sistemas de eventos discretos. A modelagem consiste em um conjunto colaborativo de duas *Fuzzy Reasoning Petri Nets* (FRPN). Uma *Monitoring Fuzzy Petri Nets* (MFPN) representa a árvore de falhas. O segundo modelo usou *Recovery Fuzzy Petri Nets* (RFPN) para as atividades de recuperação.

Para modelar o monitoramento das tarefas foi usada a MFPN com uma abordagem Fuzzy temporal, como o instante da ocorrência da falha. Os lugares representam as possíveis faltas e as transições no modelo foram associadas a

regras fuzzy. A marca no lugar de saída da transição em um lugar foi associada a um número fuzzy. A recuperação é consequência do raciocínio baseado em uma base de regras fuzzy, modelada por outra ferramenta de FPN com topologia similar.

A comunicação entre os modelos foi realizada por um protocolo de emissão/recepção, usando uma representação inspirada em redes de Petri sincronizada, adaptada a transmissão de variáveis fuzzy.

O trabalho de Hanafi *et al.* (2007) apresentou um modelo em FCPN para logística reversa. O modelo consiste em três principais componentes, fuzzificação, avaliação das regras e defuzzificação.

O problema abordado foi a incerteza para quantificar os produtos retornados e assim determinar e planejar transporte e alocação dos recursos e planejar a quantidade matéria-prima para manufaturar novamente os produtos retornados. O objetivo foi prover um modelo para prever a taxa de produtos retornados em várias localidades ao longo de um período específico de tempo.

Qiao *et al.* (2008) usou FPN para resolver o problema de reprogramação. O modelo descreveu a estratégia de reprogramação composto por duas partes, uma que determina quando começar o processo e o método usado. Um modelo de raciocínio também é proposto.

O trabalho de Liu *et al.* (2008) usou um modelo de FPN para predição de qualidade de produtos durante o processo de manufatura. O raciocínio baseado em FPN foi usado pra prever perigos potenciais, falhas iminentes e defeitos do sistema, dos produtos ou dos equipamentos

O trabalho de Lee and Banerjee (2009) usou uma FCPN com tempo de espera estocástico como uma metodologia para incorporar informações incertas e estocásticas em um sistema de manufatura. O modelo combinou FCPN e Redes de Petri estocástica (*Stochastic Petri Nets* – SPN). O modelo incorporando informação incerta e estocástica foi representado usando a arquitetura de linguagem de marcação de rede de Petri estendida (*extended Petri Net Markup Language* – xPNML). Os níveis de incerteza encontrados no problema foram o Tipo 2 da Tabela 3.2 para múltiplos tipos de produtos modelado em CPN, o tipo 1 para o tempo de operação estocástico modelado em SPN e o tipo 3 modelado em FPN.

Tabela 3. 2. Incertezas em Manufatura (adaptado de Lee e Banerjee, 2009)

Tipos de incerteza	Exemplo de incerteza
Processos instáveis/ Tempo de operação instável	Tempo de execução de uma tarefa é sempre mudado. (Min 1.5 hr ~ 4 hrs)
Programação incerta	Se a tarefa A estiver <i>quase</i> completa, preparar a tarefa B.
Regra de operação fuzzy	Se a temperatura é <i>alta</i> e a pressão é <i>baixa</i> então controle da taxa de abastecimento é <i>alta</i> .
Planejamento de recurso ambíguo	Se a utilização do recurso é <i>alta</i> , enviar mais dois recursos.
Evento não-discreto	Se a tarefa <i>não estiver terminada</i> , executar pra próxima operação.
Ambiguidade no conhecimento	Se ocorrer uma emergência, parar o sistema <i>ou</i> chamar um técnico.

3.6 Considerações Finais

FPN têm sido aplicadas em várias áreas da literatura, assim como em sistemas de manufatura. A abordagem usada neste trabalho é a abordagem de FPN para modelagem do conhecimento. As regras de um sistema baseado em regras serão modeladas e um algoritmo de raciocínio será executado.

A abordagem relacionada à modelagem de sistemas físicos dinâmicos é usada para tempos incertos representados por distribuições de possibilidade. No entanto, para realizar o controle de um sistema, como o desta proposta, a primeira abordagem será usada. As informações temporais da rede são determinísticas e os eventos devem ocorrer quando uma transição é disparada, pois na modelagem de um controle de um sistema de manufatura, não é viável avaliar a possibilidade de que os disparos ocorram, e sim a sua real execução.

Os trabalhos desenvolvidos em sistemas de manufatura consideram problemas específicos, como seleção de roteiros (FENGLER *et al.*, 1996), controle de qualidade (HANNA, 1994) (LIU *et al.*, 2008) manipulação de robôs (YEUNG *et al.*, 1996), sistemas de visão (HANNA; BUCK; SMITH, 1994).

Neste trabalho, serão considerados basicamente vários problemas desses sistemas, como alocação de recursos, seleção de roteiros e seleção de AGVs, englobando todo o modelo do sistema de controle.

Capítulo 4

MODELAGEM E CONTROLE DE SISTEMAS DE MANUFATURA USANDO REDES DE PETRI COLORIDAS FUZZY

Neste capítulo será apresentada a proposta deste trabalho, os critérios usados, bem como as etapas de desenvolvimento.

4.1 Proposta

Este trabalho tem o objetivo de desenvolver uma estratégia modelagem do controle em Sistemas Automáticos de Manufatura (AMS). Redes de Petri foram usadas para a modelagem do sistema e para resolução dos conflitos foi usada a lógica fuzzy.

Controlar um FMS é uma tarefa complexa devido aos vários subsistemas e elementos que o compõe, múltiplos roteiros de produção e concorrência de eventos. Desta forma, conflitos podem ocorrer entre os processos.

Realizar o controle, considerando as informações do estado corrente do sistema no momento da tomada de decisão é uma forma eficiente de garantir que a programação adapte-se às mudanças que ocorrem no sistema. Se houver mudanças na gestão da produção, quebra de máquinas ou indisponibilidade de AGV, o planejamento não precisará ser refeito, pois a decisão, a partir do momento da ocorrência dos eventos imprevistos, considerará o estado atual do sistema. As alternativas envolvidas no conflito serão somente aquelas que são viáveis para

serem executadas naquele momento. Portanto, será considerada uma programação da produção dinâmica ao invés de uma programação estática.

Nesta abordagem, os conflitos como disputa por recursos e escolha dos roteiros de fabricação serão solucionados durante a execução do sistema. O sistema de decisão será modelado em FCPN e a partir desta modelagem será possível simular e analisar o controle, determinar sua execução e garantir a solução de conflitos no sistema.

A incerteza sobre a continuação do disparo das transições em conflito necessita de uma decisão. Assim, lógica fuzzy foi escolhida para atribuir prioridades aos processos e realizar a solução dos conflitos usando um sistema baseado em regras.

O propósito desta modelagem é sugerir uma configuração do FMS para as especificações necessárias bem como a resolução de conflitos integrada no próprio modelo do sistema. Com o uso das FCPN é possível modelar um sistema baseado em regras fuzzy para a tomada de decisão. O sistema de manufatura e o sistema fuzzy serão modelados na mesma rede e assim, realizar a solução de conflitos sem a necessidade de outros sistemas de apoio.

4.2 Critérios de Desempate

Alguns critérios de desempate entre os processos serão utilizados para solucionar os conflitos. As informações como tempos de processamento total, nível dos buffers de entrada, prazo de entrega e tamanho do lote serão consideradas.

De acordo com os pontos de conflitos no sistema, regras fuzzy foram construídas para atribuir prioridades aos processos.

As decisões são realizadas a partir das ordens de produção. Na estação de carga é decidido qual produto deverá ser fabricado primeiro. As variáveis usadas nesta solução são o tamanho do lote e o prazo de entrega. Assim, uma sequência é determinada para que os produtos entrem em processamento.

Se o produto tiver dois ou mais roteiros distintos para ser processado, o roteiro no qual o produto seguirá será escolhido considerando o tempo de processamento do roteiro e o nível dos buffers de entrada das máquinas de cada

roteiro. Quando um roteiro é selecionado para um produto, a peça será processada neste roteiro até que ela seja finalizada. Isto será realizado para todos os produtos na estação de entrada.

Um AGV é requisitado para levar a peça com a prioridade mais alta da estação de carga para o buffer de entrada da primeira máquina do roteiro determinado. As outras podem requisitar um AGV após a peça anterior deixar a estação. Se a máquina estiver desocupada e o buffer vazio, o processamento começará logo após a peça chegar ao buffer. Se a máquina estiver ocupada e tiver mais de uma peça no buffer, uma peça é escolhida baseada no tempo total de processamento do roteiro e no prazo de entrega do produto.

Quando cada peça terminar uma etapa do processamento, este seguirá para o buffer de saída da máquina. Se tiver mais de uma peça no buffer, uma peça é selecionada para requisitar um AGV usando os mesmos critérios usados no buffer de entrada. Assim, um AGV é selecionado para transportar esta peça.

Se o produto estiver finalizado, então o produto irá para a estação de saída. Se não, a peça seguirá para o buffer de entrada da próxima máquina do roteiro. A Tabela 4.1 mostra a relação entre os pontos de decisão, o tipo de escolha e os critérios de desempate usados na a decisão.

Tabela 4. 1. Critérios de Desempate

Ponto de Decisão	Tipo de Escolha	Critério de Desempate
Estação de carga (Conflito entre os produtos)	Escolha dos produtos	Tamanho do lote
		Prazo de entrega
Estação de carga (Conflito entre os roteiros de produção dos produtos)	Escolha do roteiro de produção	Tempo de processamento do roteiro
		Nível dos buffers
Buffer de entrada	Escolha da peça que será processada primeiro	Tempo de processamento do roteiro
		Prazo de entrega
Buffer de saída	Escolha da peça que irá requisitar um AGV	Tempo de processamento do roteiro
		Prazo de entrega

As Figura 4.1 e 4.2 mostram os fluxogramas para os critérios de desempate na estação de entrada e nos buffer de entrada de saída, respectivamente.

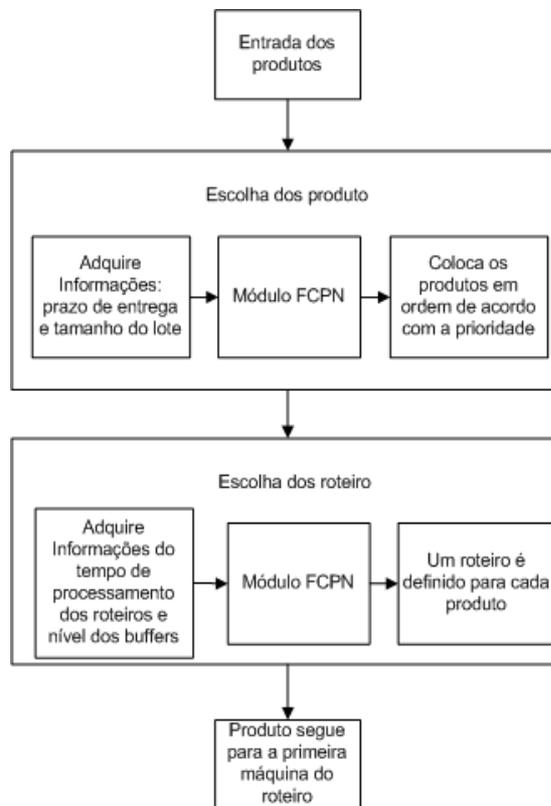


Figura 4.1. – Fluxograma dos critérios de desempate na estação de carga.

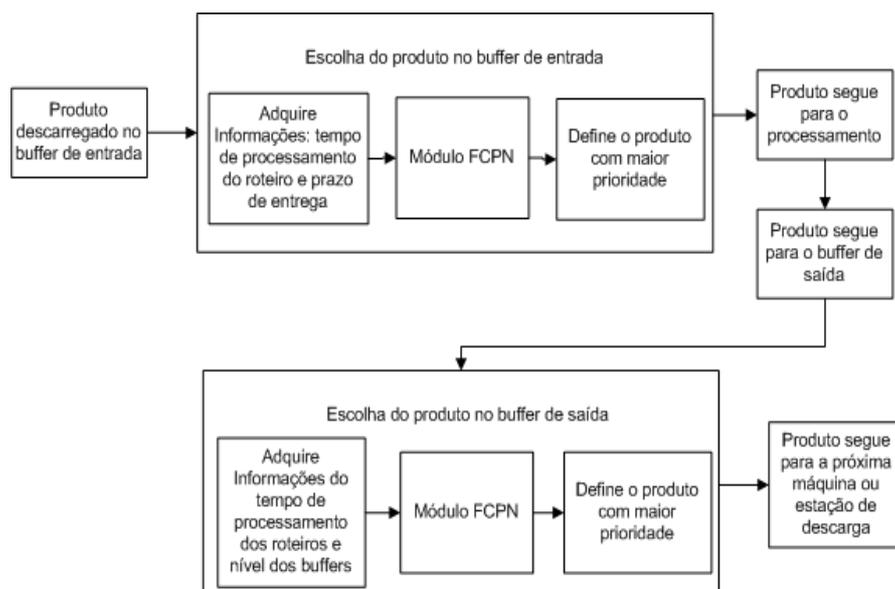


Figura 4.2. – Fluxograma dos critérios de desempate nos buffers de entrada e saída.

Quando dois pontos conflitantes distintos requisitam um AGV, a prioridade é dada para aquele ponto que requisitou primeiro. As outras aguardam até que tenha AGVs livres no estacionamento. Quando um AGV finaliza o atendimento em um ponto, ele volta para o estacionamento para verificar as novas solicitações de transporte.

4.3 Método de Desenvolvimento

A modelagem do sistema de manufatura foi dividida em módulos. A estação de carga e descarga, as estações de trabalho e o estacionamento foram modelados separadamente. Os módulos para aquisição das informações também foram construídos, como aquisição das informações sobre o prazo de entrega, o tamanho do lote, tempo de processamento, roteiros alternativos dos produtos e informações sobre o nível dos buffers das máquinas.

Em pontos específicos no modelo do sistema de manufatura foram introduzidos pontos para realizar a comunicação com o módulo de aquisição de informação e com os módulos em FCPN para a atribuição de prioridades.

Para cada ponto de conflitos, os conjuntos fuzzy relacionados às variáveis foram construídos e uma base de regras foi especificada por um especialista. A modelagem foi realizada de acordo com as especificações da FCPN de Yeung *et al.* (1996). Os graus de verdade das proposições são avaliados e, se a regra é ativada, a transição é disparada. O algoritmo de raciocínio usado foi o algoritmo de Chen *et al.* (1990). Para o processo de defuzzificação foi usado o método de centro de máximos.

Os processos conflitantes receberão prioridades e entrarão em uma fila de acordo com a sua prioridade. O processo com a prioridade mais alta ocupará o primeiro lugar na fila e será executado primeiro. Se o estado do ambiente não mudar, os outros serão executados de acordo com a ordem. Caso contrário, o cálculo das prioridades é refeito para determinar uma nova sequência. Depois, o sistema continuará sua execução até que o próximo conflito ocorra.

A ferramenta de modelagem *CPN Tools* será utilizada para a construção dos modelos. Esta ferramenta foi desenvolvida pela Universidade de Aarhus, Dinamarca,

e possui uma grande comunidade de pesquisadores que se dedicam ao seu uso e aperfeiçoamento. Ela vislumbra todas as possibilidades de modelagem necessárias para a implementação da FCPN. A implementação de redes hierárquicas é possível através do uso de *transições substitutas e lugares de fusão*.

Nesta ferramenta é possível realizar a execução da modelagem e análise das movimentações das marcas e assim, simular o andamento e desempenho do sistema.

4.4 Ambiente para Validação

O ambiente de manufatura no qual será aplicada está proposta é composto de máquinas com buffers de entrada e saída limitados, um sistema de transporte, um estacionamento e uma estação de carga e descarga. Roteiros de produção alternativos também foram propostos.

Os modelos do sistema serão validados por simulação para garantir a ausência de conflitos nos pontos identificados.

Capítulo 5

MODELAGEM E VALIDAÇÃO DA PROPOSTA

5.1 Considerações Iniciais

O modelo de FMS adotado para a modelagem e validação desta proposta é composto por seis máquinas – M1, M2, M3, M4, M5, M6 com buffers de entrada e saída com capacidade finita de quatro elementos, estação de carga e descarga, três AGVs e uma área de estacionamento dos veículos. Este ambiente foi escolhido, pois tem sido usado em vários outros trabalhos pelo grupo TEAR. A Figura 5.1 mostra o arranjo físico do sistema.

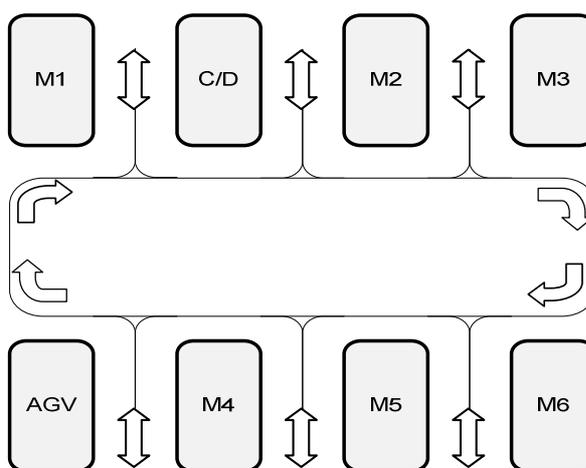


Figura 5. 1 - Arranjo físico do Sistema de Manufatura usado nesta proposta.

A modelagem em CPN Tools do ambiente da Figura 5.1 foi desenvolvida por Araújo (2006). Portanto, alguns modelos já existentes serão modificados para atender as especificidades do modelo proposto. Nesta sessão, os modelos propostos por Araújo serão apresentados e brevemente explicados. Nas próximas sessões, os modelos construídos e usados nesta proposta serão explicados mais detalhadamente.

A técnica *top-down* foi utilizada para a construção do modelo no trabalho de Araújo (2006). Nesta técnica uma visão geral do sistema é modelada primeiramente, mas sem entrar em detalhes sobre o nível dos subsistemas. Neste tipo de modelagem é considerada a disponibilidade dos recursos dentro do sistema.

O método *top-down* foi escolhido frente ao método *bottom-up*, pois confere a possibilidade de se abordar primeiramente o arranjo físico do sistema e então continuar a modelagem pelos elementos mais específicos. Cada nível hierárquico do método apresenta o modelo de partes do sistema com níveis de detalhamento diferentes. Mesmo em módulos distintos, os elementos ficam ligados entre si.

Os componentes do sistema de manufatura foram modelados em Redes de Petri Colorida (CPN), como as rotas do AGVs, máquinas e buffers. A Figura 5.2 mostra o modelo após o primeiro passo da modelagem.

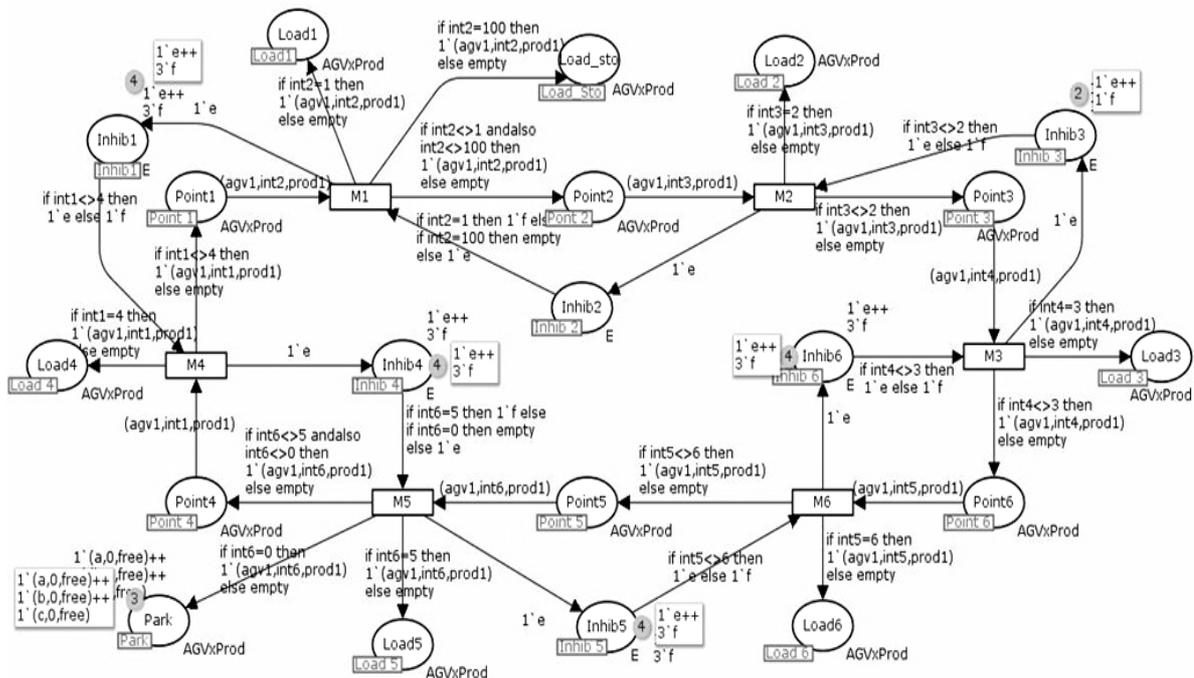


Figura 5. 2 - Modelagem em CPN do arranjo físico do sistema (ARAÚJO, 2006).

Os caminhos possíveis para o AGV são determinados pelo arranjo físico do sistema. Pontos de controle são usados para restringir o número de AGVs no caminho.

O lugar *Load i* representa a área de entrada e saída da máquina *i*, *Load_sto* representa a estação de carga e descarga e *Park* representa o estacionamento dos AGVs.

Neste modelo, restrições referentes ao transporte são consideradas. Cada lugar de controle nomeado *Point i* tem o seu correspondente lugar *Inhib i*. *Inhib i* é um lugar correspondente (conhecido em inglês como *anti-place*) ao lugar original, onde para cada arco de/para o lugar original existe um arco para/de o lugar correspondente movendo o mesmo número de marcas. Essa estrutura simula um arco inibidor e é usada para limitar a capacidade dos lugares. Assim, a presença de um único AGV em um ponto de controle e a presença de, no máximo, três AGVs na estação de entrada de cada máquina são garantidas por essa estrutura. Marcas coloridas *e* são usadas para o lugar de controle e marcas da cor *f* na estação de entrada das máquinas.

Tipos de cores são definidos. *AGVxProd* é uma tripla com o nome de um AGV, um número inteiro e um tipo de produto. O número inteiro representa o destino do AGV. Inscrições nos arcos são adicionadas no modelo para avaliar a situação e realizar ações específicas. As marcas iniciais referentes aos AGVs são adicionadas no lugar referente ao estacionamento. As marcas são do tipo *AGVxProd*, mas neste estágio o tipo de produto é indicado por *free*, indicando que o AGV está livre.

Na próxima etapa, o sistema foi dividido em outros modelos. *Lugares de fusão* foram usados para a construção do modelo em vários níveis por uma rede hierárquica. Um *lugar de fusão* representa o mesmo lugar em módulos diferentes. Este é indicado pelo retângulo com o nome da fusão logo abaixo do círculo que representa o lugar. Como nem todos os caminhos possíveis foram modelados no modelo da Figura 5.2, alguns caminhos também foram representados nos modelos específicos. Modelos separados foram criados para cada máquina, para a estação de carga e descarga e estacionamento dos AGVs.

Já com a quantidade, o tipo e ordem de peças definidas, os arcos ao longo do sistema são definidos para que possam apontar o caminho de uma peça ou veículo de transporte de acordo com os planos definidos previamente. As rotas dos produtos são definidas através das inscrições nos arcos.

No modelo do estacionamento na Figura 5.3, a primeira solicitação é que um AGV vá até a estação de carga, pois é lá que ele será carregado com uma peça a fim de levá-la para o processamento em alguma máquina. O número inteiro que caracteriza essa solicitação é o número 100.

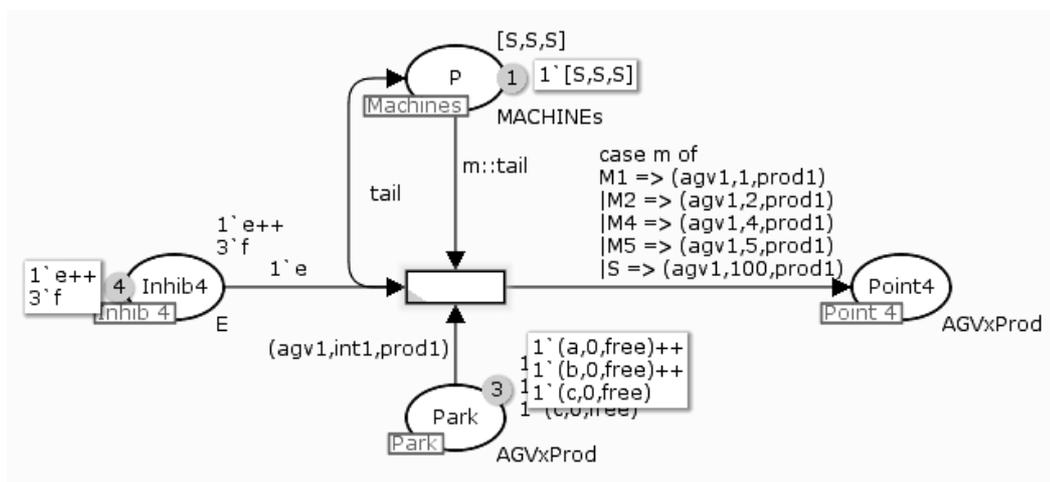


Figura 5. 3 - Modelagem do estacionamento dos AGVs (ARAÚJO, 2006)

No modelo das máquinas, as estações de entrada e saída de peças e os buffers de entrada e saída foram modelados. A Figura 5.4 mostra o modelo para a máquina 1. Este modelo também foi usado para as outras máquinas.

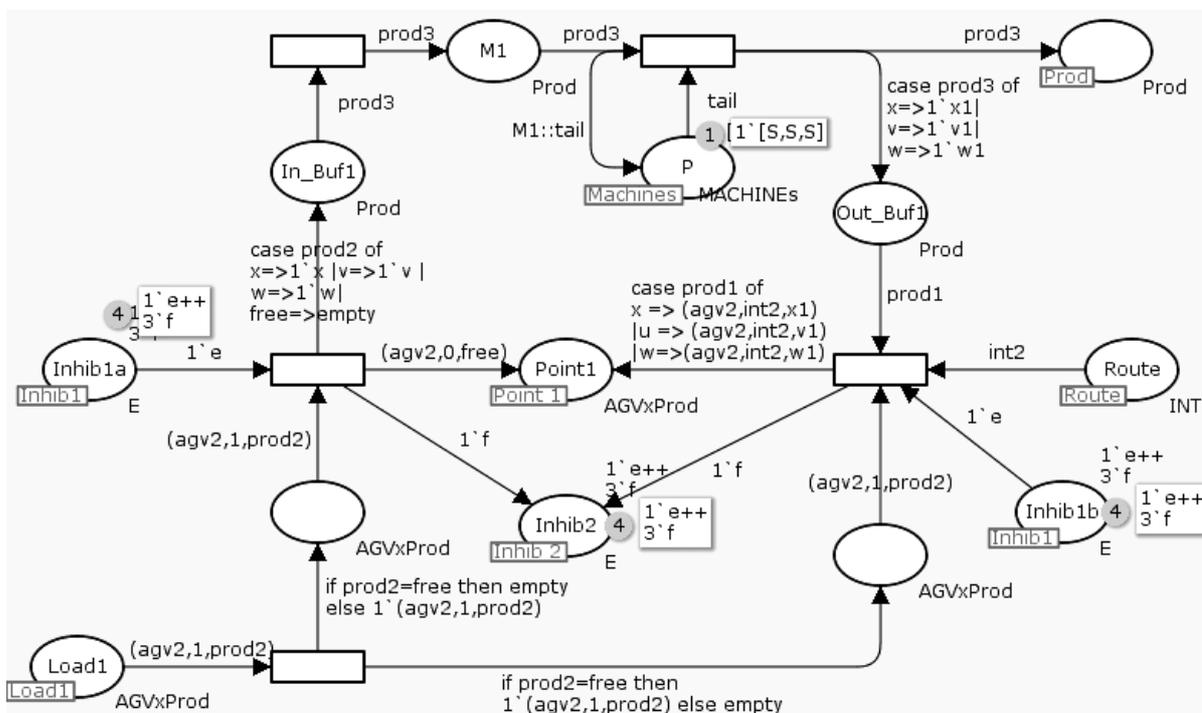


Figura 5. 4 - Modelagem da Máquina 1(ARAÚJO, 2006).

acordo com a prioridade calculada. O lugar do tipo *Rec_LP* possui uma lista dos produtos já ordenados. Para cada produto será avaliados os roteiros alternativos existentes na transição substituta *EscolhaRoteiro*. Sendo assim, de acordo com o roteiro selecionado para cada produto, a primeira máquina do roteiro também é determinada e os elementos compõem a tripla *AGVxProd*. Assim, a marca segue para o *Point2*. Se o AGV estiver carregado, o produto é descarregado em *Storage2* e o AGV livre vai para *Point2* também.

A transição substituta *EscolheProduto* é representada pelo modelo da Figura 5.8. Neste modelo, as informações como o prazo de entrega de cada produto e o tamanho do lote são usados para compor o registro do tipo *Rec_P*. Neste registro, informações como o produto, o tamanho do lote, o prazo de entrega, prioridade do produto e o conjunto fuzzy da variável de saída são armazenadas. Inicialmente, todos os produtos possuem prioridade zero. A quantidade de produtos e os tipos de produtos que vão ser selecionados nesta etapa deve ser previamente determinado pelo usuário.

Assim que todos os registros são compostos, a transição substituta *FuzzyProduto* é chamada para executar o raciocínio fuzzy e atribuir prioridades aos produtos. Os modelos fuzzy para todos os pontos de decisão serão apresentados na próxima sessão. Sendo assim, o retorno desta transição é uma lista de *Rec_P*, a *Rec_LP*, onde os produtos já estão ordenados de acordo com a sua prioridade.

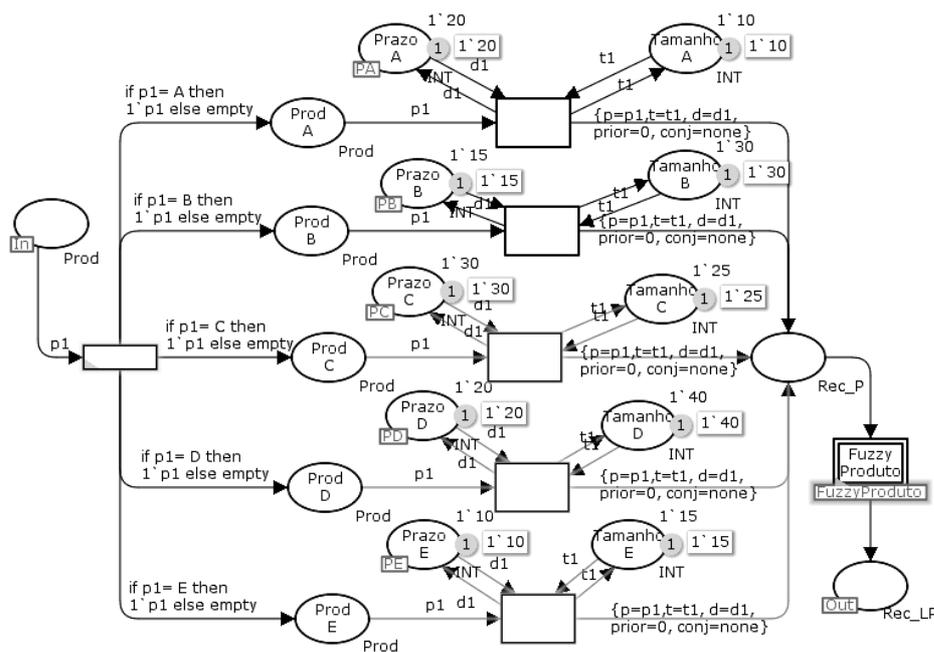


Figura 5. 8 – Modelo para leitura dos valores considerados na seleção dos produtos

O modelo para a transição substituta *EscolhaRoteiro* é mostrado na Figura 5.9. Neste modelo, somente os produtos são adicionados em uma lista, as informações como a prioridade, o tamanho do lote e o prazo de entrega serão descartadas. Os produtos são ordenados na ordem previamente determinada. Para cada produto, existe uma transição substituta onde serão determinadas as informações necessárias para o cálculo da prioridade de cada roteiro alternativo para cada produto. No ambiente escolhido, o produto B não possui roteiros alternativos. Portanto, não é necessária a construção de um modelo específico para este produto.

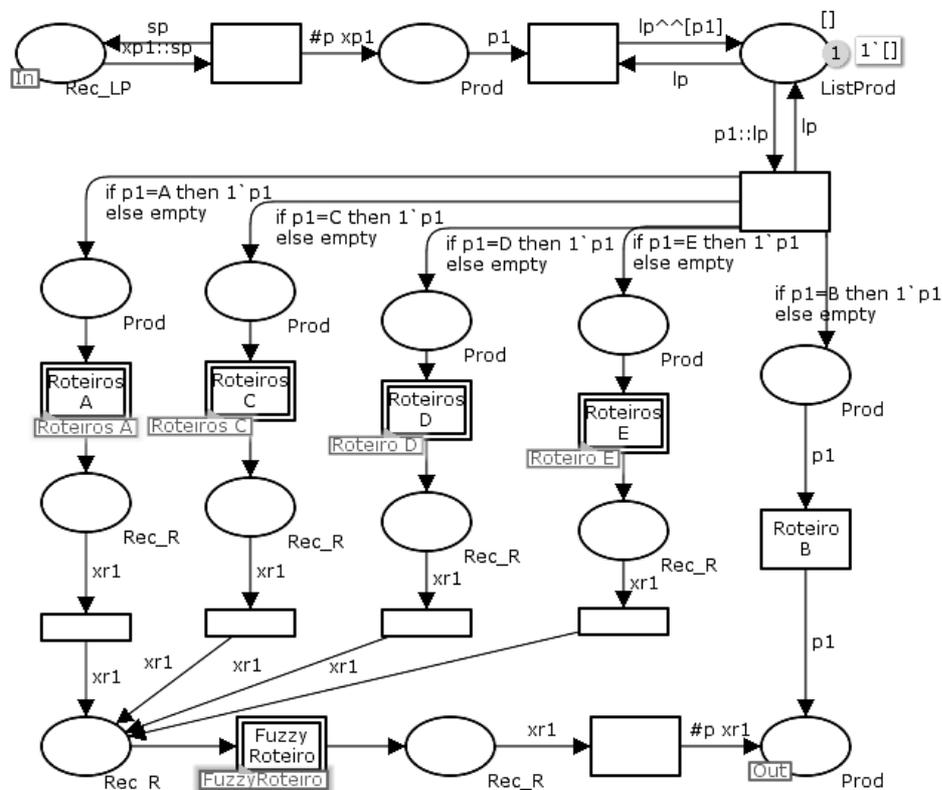


Figura 5. 9 – Modelo para seleção do roteiro de acordo com o produto

Após a aquisição das informações, cada roteiro alternativo de cada produto é calculado pela transição substituta *FuzzyRoteiro*. Após essa etapa, um produto é retornado e é adicionado na tripla do modelo da estação de carga.

No modelo da transição substituta *RoteirosA*, mostrado na Figura 5.10, dois roteiros são considerados, o roteiro 1 e o roteiro 2 para o produto A. A partir deste modelos os produtos não serão somente chamados de A, B, C, D, E. Eles serão compostos pelas siglas R_i , onde i é o número do roteiro. Assim, é possível identificar ao longo da execução do sistema em qual roteiro o produto está sendo produzido. Os lugares *TPAR1* e *TPAR2* contém informações sobre o tempo de processamento

dos roteiros 1 e 2, respectivamente. Os lugares chamados *BuffersM1*, *BuffersM2*, *BuffersM3* e *BuffersM5* são lugares de fusão. Estes lugares também são encontrados nos modelos das estações das máquinas. Estes lugares calculam o número de peças que estão nos buffers de entrada de cada máquina do roteiro de produção.

Essas duas informações são usadas posteriormente para atribuir prioridades aos roteiros pelo modelo em FCPN. Elas são adicionados ao registro do tipo *Rec_R* que também contém informações sobre o produto e a prioridade do roteiro.

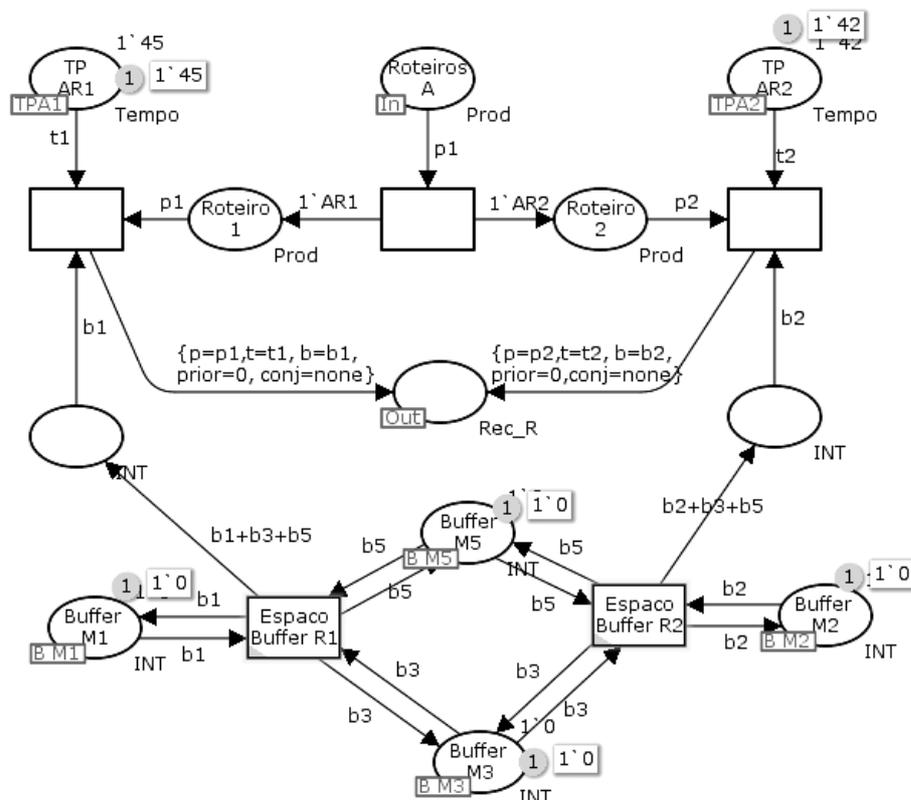


Figura 5. 10 – Modelo para leitura dos valores considerados para a seleção dos roteiros do produto A.

O modelo da Figura 5.11 apresenta o modelo da estação da máquina M1. Os modelos para as estações das outras máquinas podem ser encontrados no Apêndice B.

Neste modelo, o AGV chega pelo lugar *Load1* e pode seguir um dos dois arcos existentes conforme sua marca. No entanto, as duas ações serão realizadas somente se o segundo parâmetro da tripla de *AGVxProd* for igual ao número da máquina, neste caso, igual a um. Caso contrário, a marca vai direto para o próximo *Point* e não entra em *Load1*. A ação no arco vertical será executada se o AGV

estiver carregado com uma peça para ser processada em M1. Sendo assim, o AGV descarrega o produto do tipo *Prod* no lugar que indica o buffer de entrada, *In_Buf1*, e o AGV livre, vai para o *Point1*. O AGV seguirá o outro arco se estiver livre. Então irá coletar uma peça do buffer de saída, *Out_Buf1* e ir para o *Point1*.

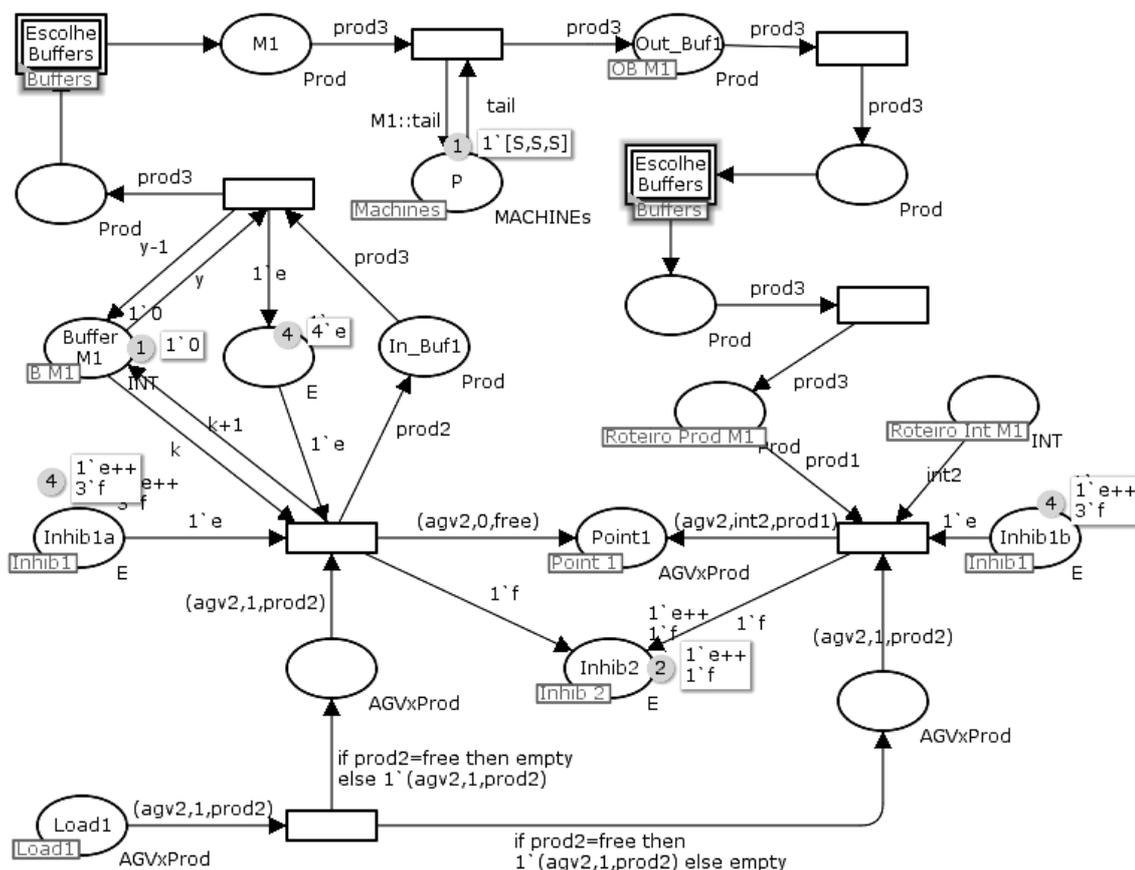


Figura 5. 11 – Modelo da estação de trabalho de M1

Quando uma peça chega ao buffer de entrada, o lugar corresponde do tipo *E* é usado para garantir a presença de somente quatro peças no buffer. O lugar de fusão *BufferM1* é usado para calcular o número de peças dentro do buffer e retorna um número inteiro para ser usado nas transições substitutas para cálculo das variáveis do roteiro, como visto no modelo da Figura 5.10 para o produto A. As peças que estão no buffer passam pela transição substituta *EscolheBuffers*, mostrada na Figura 5.12 para a leitura dos valores das variáveis que serão consideradas na escolha de uma peça no buffer e o modelo em FCPN é usado para a atribuição das prioridades.

Quando uma peça é escolhida, esta será processada por M1 e depois irá para o *Out_Buf1*. A mesma transição substituta *EscolheBuffers* é chamada para escolher qual produto tem maior prioridade para requisitar um AGV.

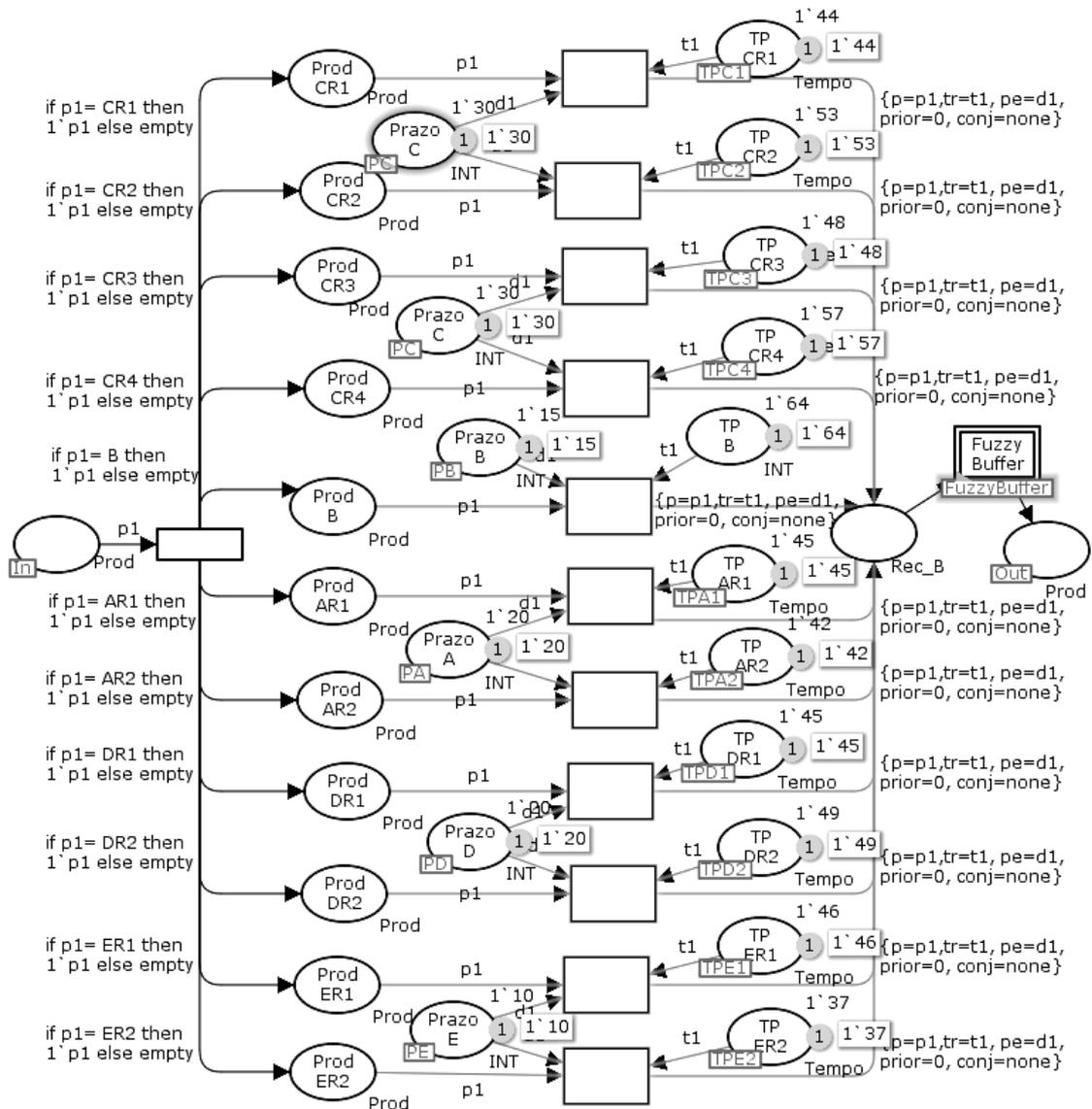


Figura 5. 12 – Modelo para leitura dos valores considerados para a seleção dos produtos nos buffers de entrada das máquinas.

Após o produto ser escolhido, o lugar de fusão *RoteiroProdM1* no modelo das máquinas é chamado. Este lugar é representado em outro modelo para a verificação de qual a próxima máquina do roteiro de acordo com os produtos que são processados em M1. O modelo é mostrado na Figura 5.13. De acordo com o produto no lugar *RoteiroProdM1*, a próxima máquina do roteiro é definida e um número inteiro é resultado e colocado no lugar de fusão *RoteiroIntM1*. Este número é colocado na tripla *AGVxProd*, assim como o produto, quando um AGV livre chega na estação para buscar uma peça no buffer de saída da máquina.

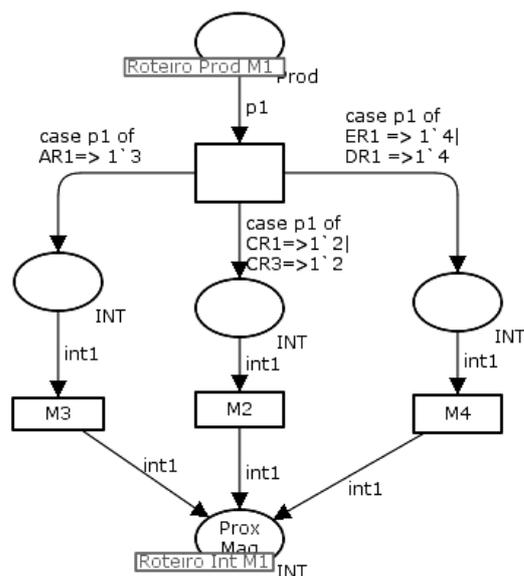


Figura 5. 13 – Modelo para seleção do próximo destino após o processo em M1.

5.4 Modelos em Redes de Petri Coloridas Fuzzy

Os modelos em FCPN foram separados de acordo com o ponto de decisão. Para cada ponto foram definidos conjuntos fuzzy para as variáveis usadas e uma base de regras que será modelada em redes de Petri. O modelo de representação usado foi o modelo de Yeung *et al.* (1996) e o modelo de raciocínio é baseado no modelo de representação de regras de Chen *et al.* (1990). Neste trabalho, os graus de confiabilidade das regras proposto por Chen *et al.* (1990) serão iguais a um.

Conjuntos fuzzy foram construídos para as variáveis usadas em cada ponto de decisão e uma base de regras fuzzy foi construída. As bases de regras foram construídas considerando a opinião de um especialista.

A variável de saída é a prioridade entre os elementos conflitantes e é mostrada na Figura 5.14. O método de defuzzificação usado é o método de Média de Máximos. Neste método, o valor defuzzificado $d_{cm}(C)$ é definido como a média do menor valor e o maior valor do conjunto de saída que tem o maior grau de pertinência (KLIR; YUAN, 1995). Formalmente, é definido como,

$$d_{cm}(C) = \frac{\inf M + \sup M}{2}$$

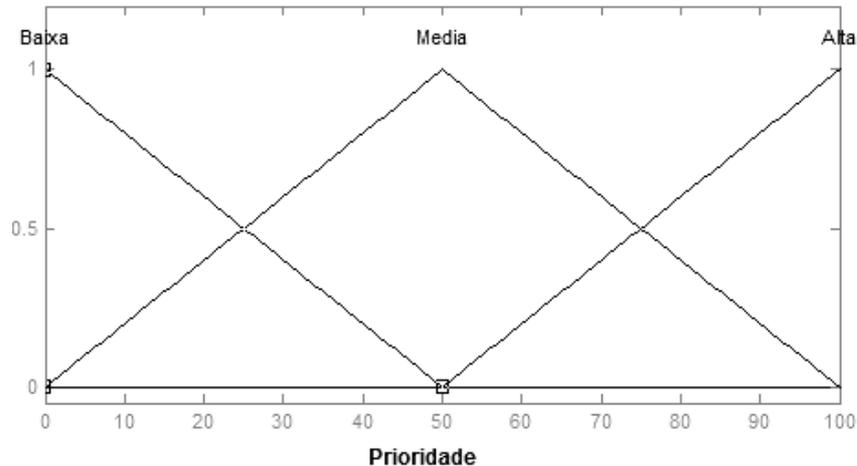


Figura 5. 14 – Conjuntos fuzzy para a variável de saída Prioridade

5.4.1 Escolha dos produtos

Na estação de carga, a sequência em que os produtos devem entrar no sistema é determinada. As ordens de produção são especificadas e a produção dinâmica deve ser determinada. Primeiramente, nenhum produto tem prioridade sobre outro e eles estão esperando para serem processados. Nesta etapa, um sistema baseado em regras fuzzy é modelado para atribuir prioridades e colocá-los em uma lista para produção.

As variáveis consideradas para esta decisão foram o tamanho do lote e o prazo de entrega. Conjuntos fuzzy foram especificados para essas variáveis. Nas Figuras 5.15 e 5.16 são mostrados os conjuntos.

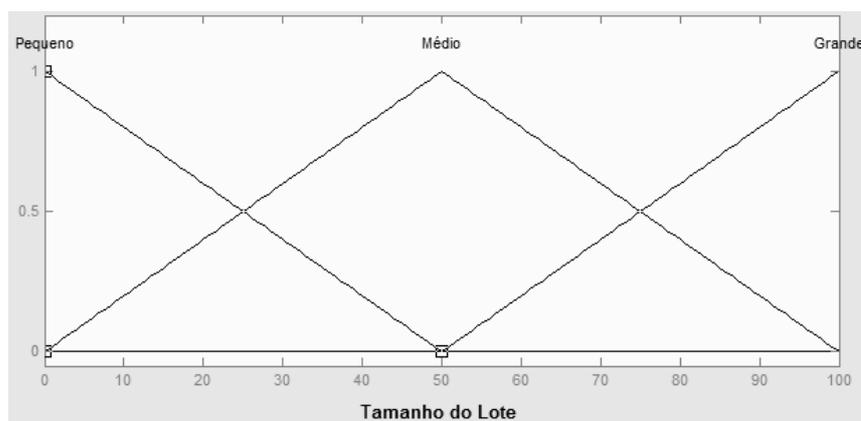


Figura 5. 15 – Conjuntos fuzzy para a variável Tamanho do Lote

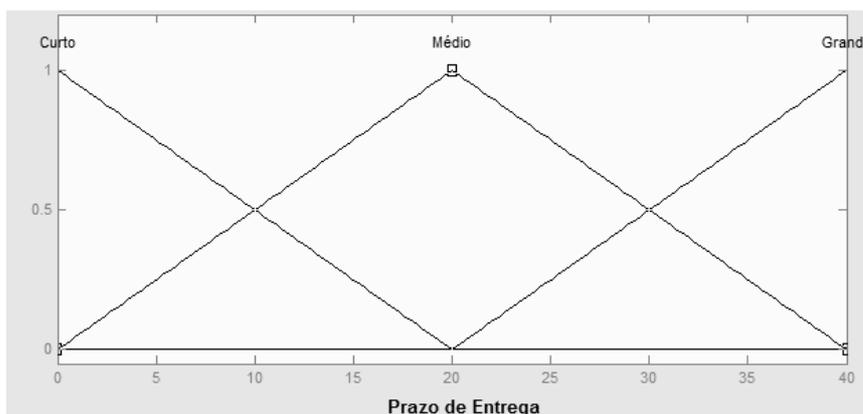


Figura 5. 16 – Conjuntos fuzzy para a variável Prazo de Entrega

A base de regras fuzzy foi construída considerando as duas variáveis e a variável de saída usada é a prioridade do produto. A Tabela 5.2 mostra a base de regras.

Tabela 5. 2. Regras fuzzy para a escolha do produto

R1	SE <i>Tamanho do lote</i> é pequeno E <i>Prazo de entrega</i> é curto ENTÃO prioridade é média
R2	SE <i>Tamanho do lote</i> é pequeno E <i>Prazo de entrega</i> é médio ENTÃO prioridade é baixa
R3	SE <i>Tamanho do lote</i> é médio E <i>Prazo de entrega</i> é curto ENTÃO prioridade é média
R4	SE <i>Tamanho do lote</i> é médio E <i>Prazo de entrega</i> é médio ENTÃO prioridade é baixa
R5	SE <i>Tamanho do lote</i> é grande E <i>Prazo de entrega</i> é curto ENTÃO prioridade é alta
R6	SE <i>Tamanho do lote</i> é grande E <i>Prazo de entrega</i> é médio ENTÃO prioridade alta

O modelo da Figura 5.17 recebe um registro do tipo *Rec_P*. Neste registro as informações do produto, tamanho do lote, prazo de entrega, prioridade e o conjunto da variável de saída são armazenados.

As inscrições nos arcos verificam se as informações pertencem aos conjuntos definidos nos lugares. O parâmetro *#t* verifica os conjuntos do tamanho do lote e o *#d* verifica o prazo de entrega. Se as informações do produto pertencer a algum conjunto, uma marcação é adicionada no lugar apropriado.

Se dois lugares de entrada de uma mesma transição receber marcas, a transição é habilitada, indicando que aquela regra foi ativada. Quando a transição for habilitada, ela pode ser disparada. No disparo da transição, o segmento de código desta realiza o cálculo da função de pertinência das proposições das regras. Segundo o modelo de raciocínio de Chen *et al.* (1990), o valor mínimo entre os graus de verdades é o valor resultante da regra. Se duas ou mais marcas pertencerem ao mesmo conjunto de saída, o valor máximo é considerado.

Esta etapa é realizada para cada produto separadamente. Após cada produto receber sua prioridade, a transição substituta *PriorProd* é chamada realizar o

processo de defuzzificação e ordenar os produtos de acordo com a prioridade. Na figura 5.18 é mostrado o modelo para ordenar os produtos.

A função *insereprod* é uma função recursiva que recebe um produto e uma lista. Primeiramente, um produto qualquer é adicionado na lista. A partir disto, as prioridades são comparadas e os outros elementos são adicionados. A lista é retornada como um tipo *Rec_LP*. De volta ao modelo de FCPN, uma lista é retornada ao modelo de *Storage* já visto.

5.4.2 Escolha do roteiro de produção

Para a seleção do roteiro, a FCPN é usada para avaliar os roteiros alternativos de cada produto. Os produtos seguem a ordem determinada no passo anterior, assim, um produto por vez é analisado. Somente quando a análise dos roteiros de um produto acabar, outro poderá começar.

Para determinar a prioridade dos roteiros, o tempo de processamento do roteiro e o nível dos buffers de entrada das máquinas de cada roteiro são considerados. Conjuntos fuzzy foram definidos para estas variáveis como mostrados na figura 5.19 e 5.20.

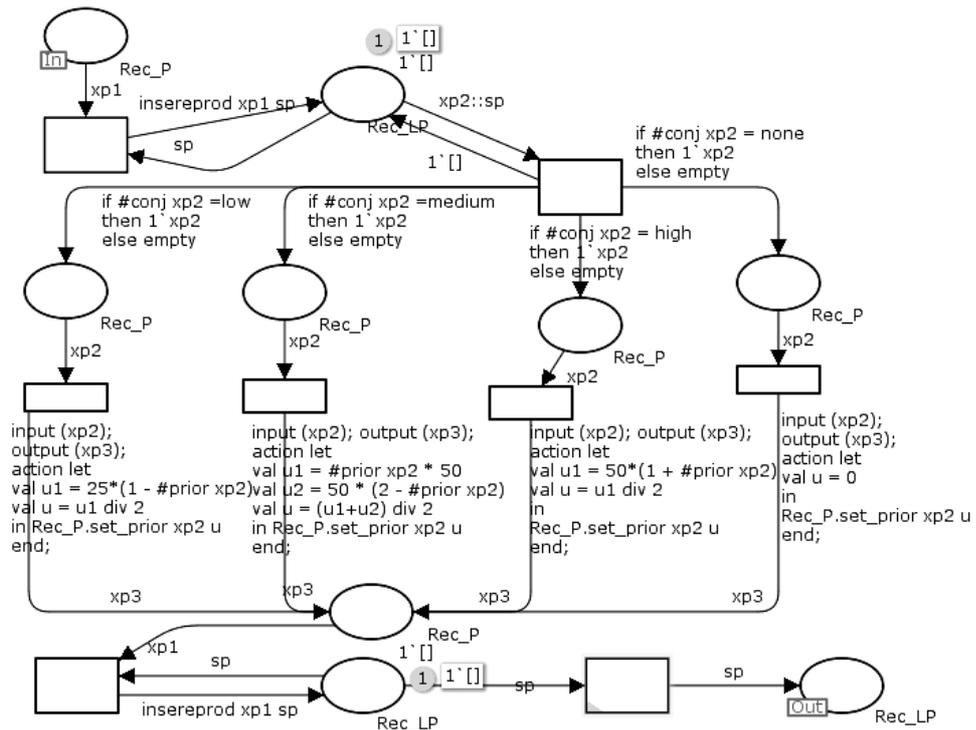


Figura 5. 18 – Modelo de defuzzificação e para ordenar os produtos de acordo com a prioridade

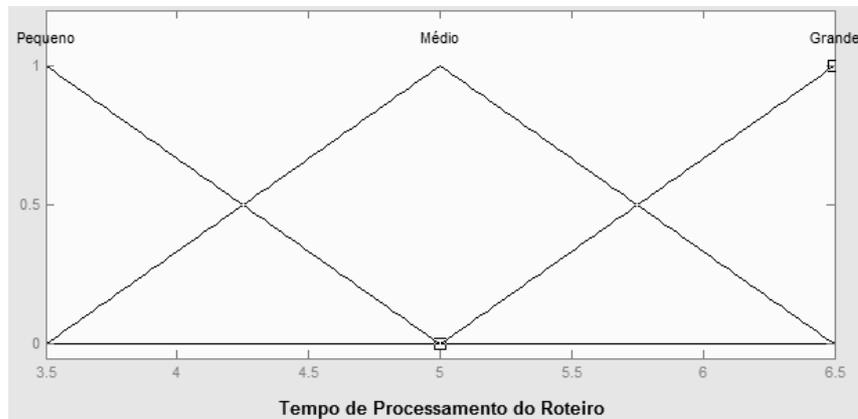


Figura 5. 19 – Conjuntos fuzzy para a variável Tempo de Processamento do Roteiro

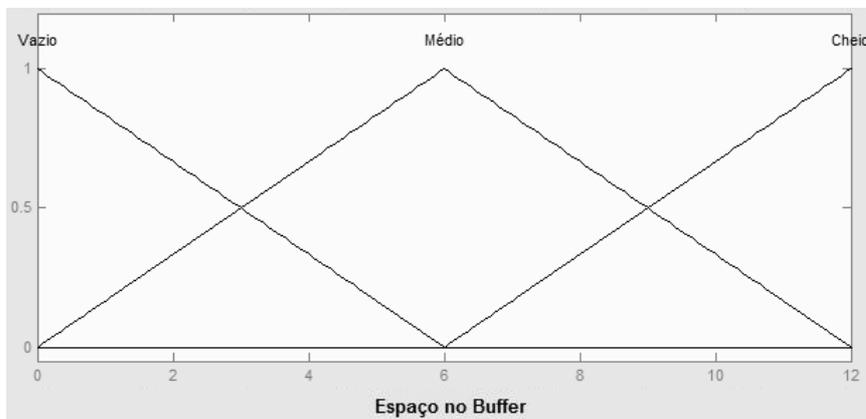


Figura 5. 20 – Conjuntos fuzzy para a variável Espaço no Buffer

Uma base de regras para foi definida considerando as duas variáveis. A variável de saída é a prioridade do roteiro. A Tabela 5.3 mostra a base de regras.

Tabela 5. 3. Regras fuzzy para a escolha do roteiro

R1	SE <i>Tempo de Processamento</i> é pequeno E <i>Nível dos Buffers</i> é vazio ENTÃO prioridade é alta
R2	SE <i>Tempo de Processamento</i> é pequeno E <i>Nível dos Buffers</i> é médio ENTÃO prioridade é média
R3	SE <i>Tempo de Processamento</i> é médio E <i>Nível dos Buffers</i> é médio ENTÃO prioridade é alta
R4	SE <i>Tempo de Processamento</i> é médio E <i>Nível dos Buffers</i> é cheio ENTÃO prioridade é baixa
R5	SE <i>Tempo de Processamento</i> é grande E <i>Nível dos Buffers</i> é vazio ENTÃO prioridade é média
R6	SE <i>Tempo de Processamento</i> é grande E <i>Nível dos Buffers</i> é médio ENTÃO prioridade baixa

O modelo em FCPN da figura 5.21 mostra a modelagem da base de regras. As inscrições são usadas para verificar se as informações do roteiro pertencem aos conjuntos e algumas regras serão ativadas. Cada roteiro recebe uma prioridade e na transição substituta *PriorRoteiro*, o processo de defuzzificação é realizado e os roteiros são colocados em ordem, mas somente aquele com a maior prioridade é retornado e o outro é desconsiderado. Este processo é mostrado na Figura 5.22. O roteiro em que o produto será processado é definido e assim, os roteiros dos outros produtos podem ser avaliados.

5.4.3 Buffer de Entrada e Buffer de Saída

Quando mais de um produto está no buffer de entrada de uma máquina para ser processado, uma decisão deve ser tomada para escolher qual deles será executado primeiro. Todos os produtos que estão no buffer de entrada serão avaliados pelas regras fuzzy e prioridades serão atribuídas a cada produto. Isto também acontece quando mais de um produto está no buffer de saída de uma máquina e estes competem por um AGV para serem transportados até o próximo destino.

As variáveis usadas para esta escolha foram o tempo de processamento do produto e o prazo de entrega. Os conjuntos fuzzy para estas variáveis foram os mesmo já definidos e a base de regras pode ser vista na Tabela 5.4.

Tabela 5. 4. Regras fuzzy para a escolha dos produtos nos buffers de entrada e saída

R1	SE <i>Tempo de Processamento</i> é médio E <i>Prazo de entrega</i> é curto ENTÃO prioridade é alta
R2	SE <i>Tempo de Processamento</i> é grande E <i>Prazo de entrega</i> é curto ENTÃO prioridade é alta
R3	SE <i>Tempo de Processamento</i> é grande E <i>Prazo de entrega</i> é médio ENTÃO prioridade é média
R4	SE <i>Tempo de Processamento</i> é médio E <i>Prazo de entrega</i> é médio ENTÃO prioridade é média
R5	SE <i>Tempo de Processamento</i> é pequeno E <i>Prazo de entrega</i> é médio ENTÃO prioridade é baixa
R6	SE <i>Tempo de Processamento</i> é médio E <i>Prazo de entrega</i> é grande ENTÃO prioridade baixa

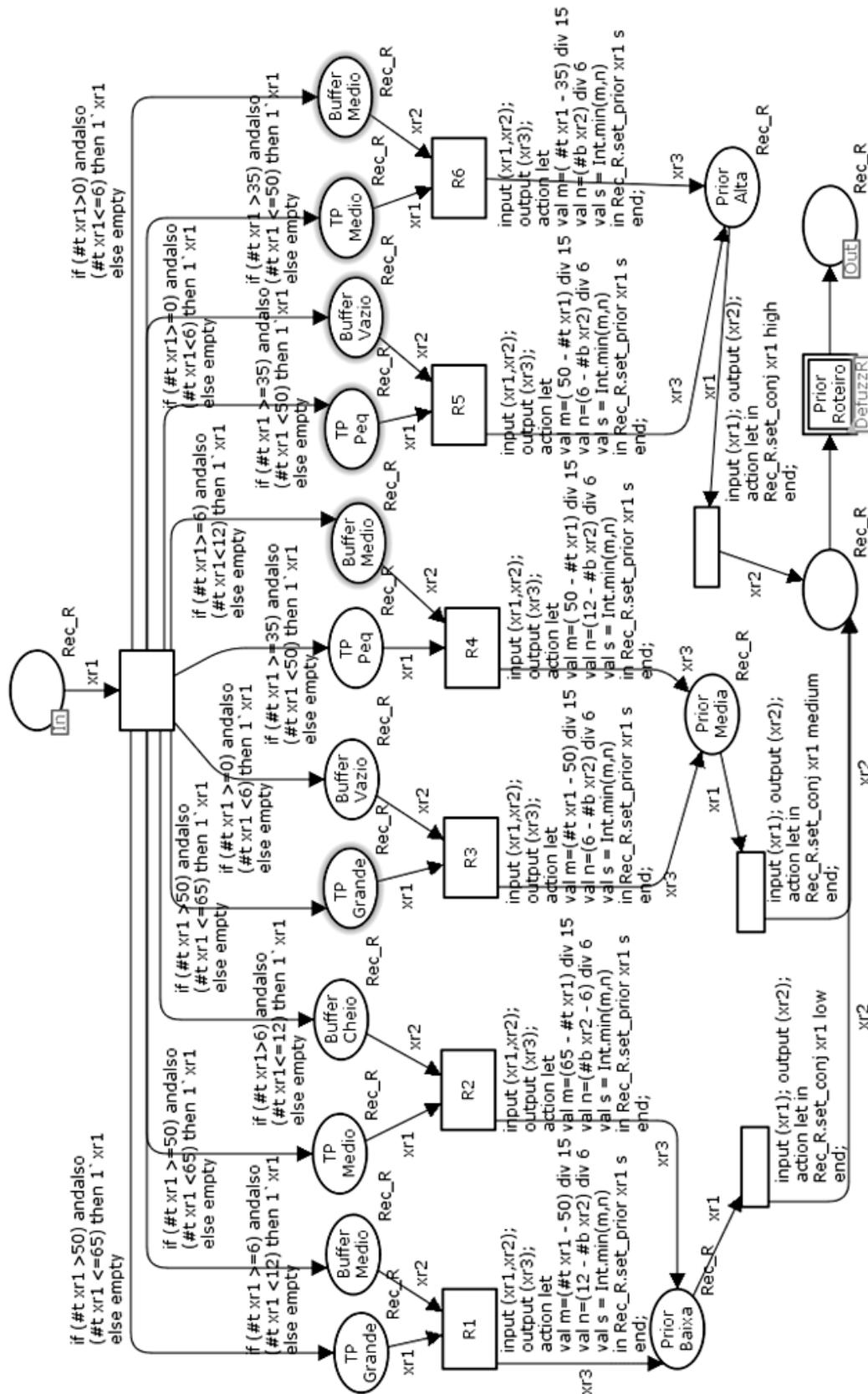


Figura 5. 21 – Modelo em FCPN para o ponto de decisão para escolha dos roteiros na estação de carga.

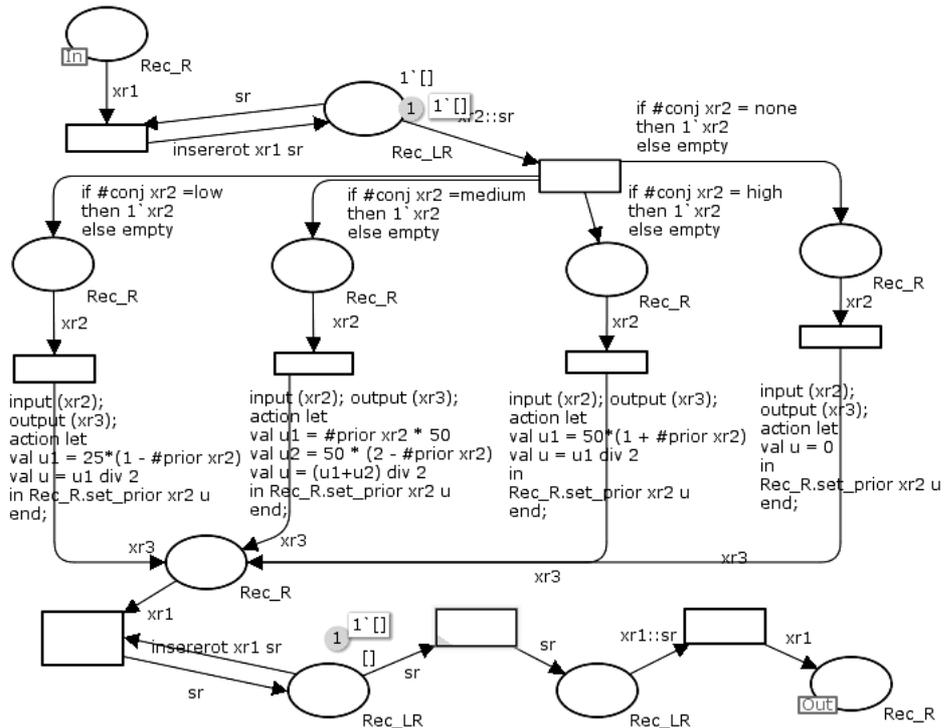


Figura 5. 22 – Modelo de defuzzificação e para ordenar os roteiros de acordo com a prioridade

O modelo em FCPN para esta base de regras é mostrado na Figura 5.23. O modelo também usa as inscrições dos arcos para verificar se as informações pertencem aos conjuntos e as transições realizam o cálculo das funções de pertinência.

Os produtos são colocados depois em uma lista de acordo com a prioridade e o primeiro produto da lista é escolhido para ser processado. O processo de defuzzificação e ordenação da lista é mostrado na Figura 5.24. Os outros produtos voltam ao buffer de entrada e quando a máquina estiver liberada, as prioridades dos produtos são recalculadas.

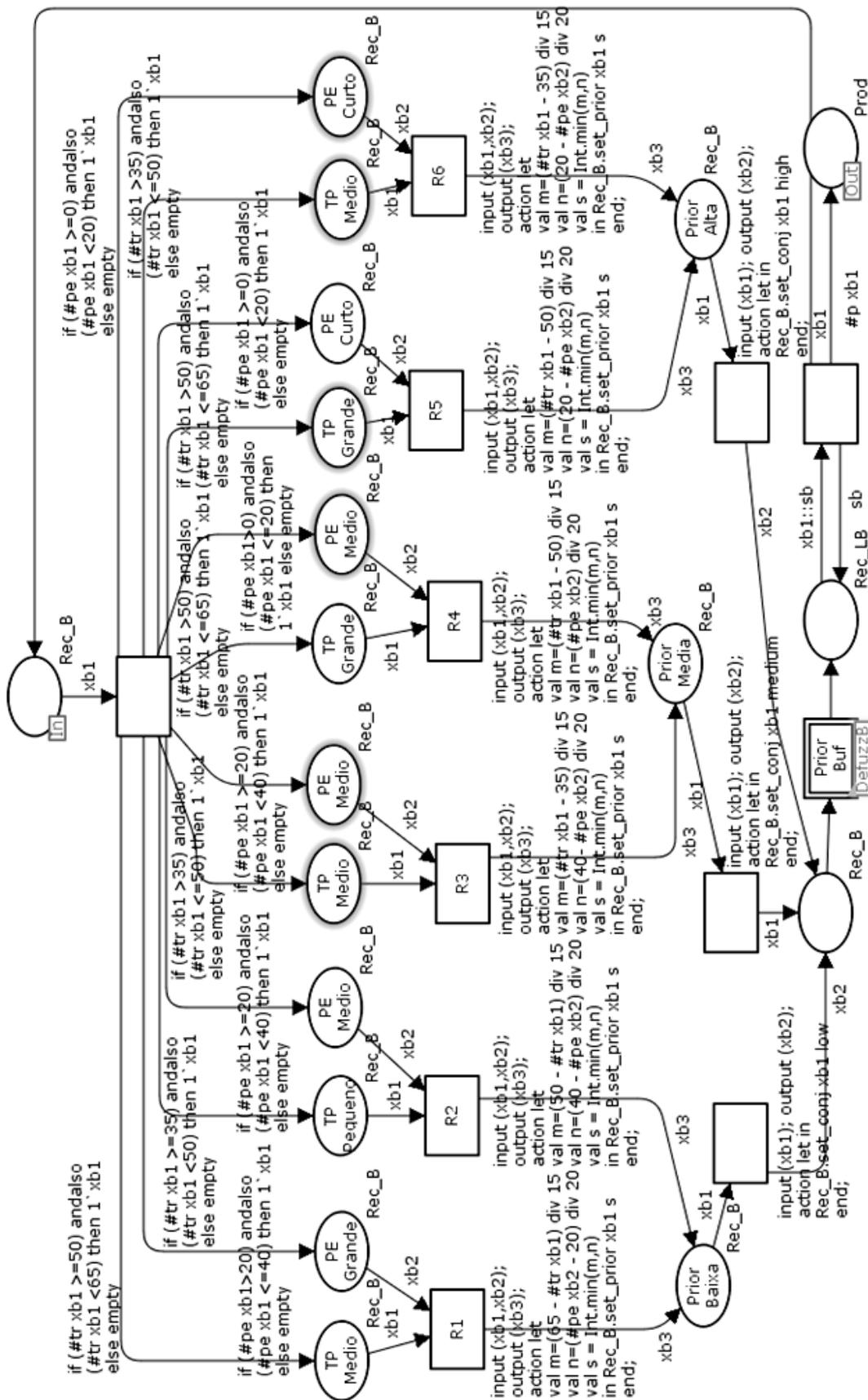


Figura 5. 23 – Modelo em FCPN para o ponto de decisão para escolha dos produtos nos buffers de entrada e de saída

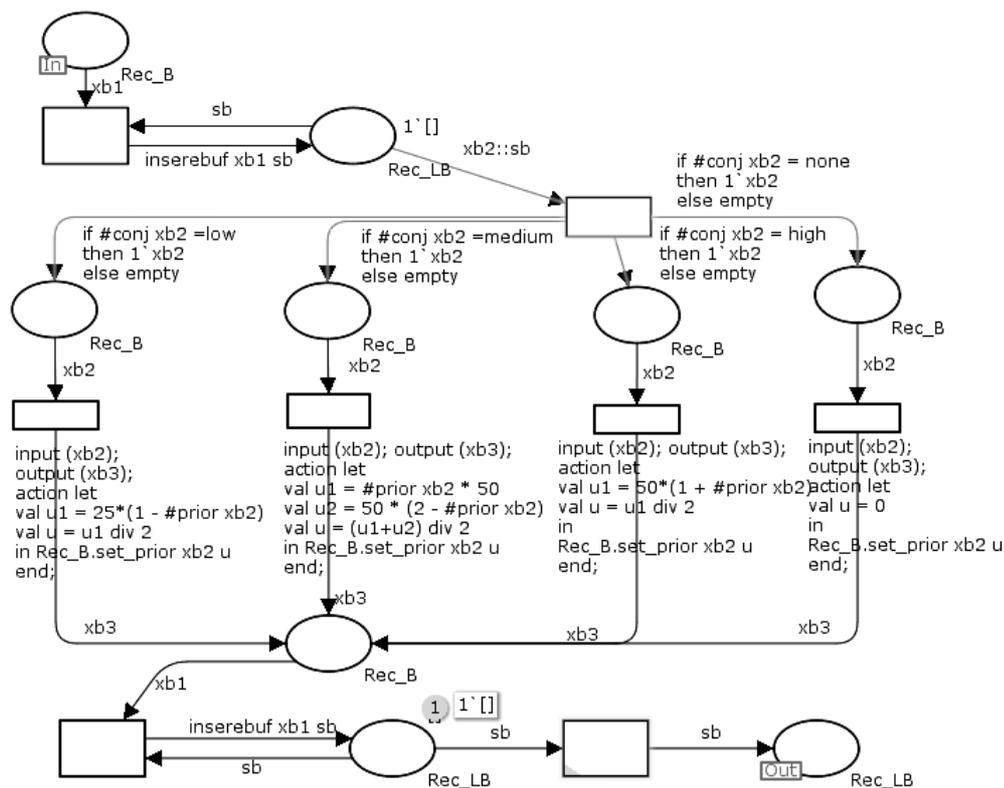


Figura 5. 24 – Modelo de defuzzificação e para ordenar os produtos dos buffers de acordo com a prioridade

5.5 Simulação e Validação do Modelo

O modelo foi simulado para verificar as propriedades do modelo e verificar a solução dos conflitos. As informações da gestão da produção foram adicionadas nas marcações do modelo para a simulação. As informações dos tamanhos dos lotes e os prazos de entrega foram definidas conforme mostradas na tabela 5.5 .

Tabela 5. 5. Informações da Produção para Simulação

Produto	Tamanho do Lote (und.)	Prazo de Entrega (dias)
A	10	20
B	30	15
C	25	30
D	40	20
E	15	10

No modelo da transição *EscolhaProduto*, as informações são adicionadas em um registro, então a decisão fuzzy é realizada para atribuir prioridades aos produtos.

Na figura 5.25 é mostrada as marcações dos lugares como prazo de entrega e tamanho do lote e, finalmente, as informações em um registro, já com a prioridade inicial de cada um.

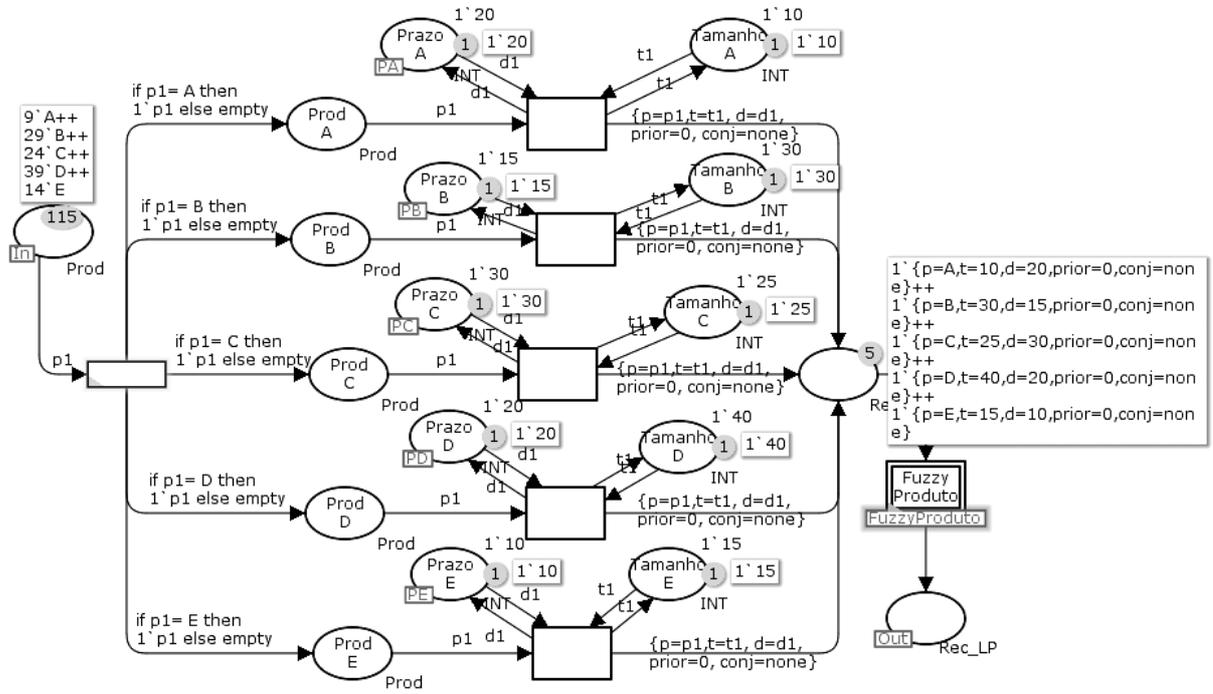


Figura 5. 25 – Modelo em FCPN para o ponto de decisão para escolha dos produtos nos buffers de entrada e de saída

A modelagem da base de regras fuzzy com marcações é mostrada na Figura 5.26. Para o produto E, as regras 1 e 3 são ativadas e então, a prioridade do produto E é calculada. Depois a prioridade dos outros produtos também é calculada e, então, os produtos aguardam para serem colocados em uma lista de acordo com a sua prioridade.

Para garantir a solução dos conflitos após o cálculo das prioridades, a estrutura de dados *lista* é usada nos pontos de decisão.

Em uma rede de Petri Colorida uma transição pode estar em conflito com ela mesma quando existe mais de uma marcação no lugar de entrada desta transição e elas não podem ser disparadas concorrentemente. Sendo assim, uma forma eficiente de garantir que uma transição não fique em conflito com ela mesma é o uso de listas de prioridade, ou seja, garantir que o disparo das transições seja sequencial. Com este tipo de estrutura as marcações respeitam uma ordem de disparo e não ficam mais conflitantes. Após a defuzzificação, os produtos são colocados em uma lista como mostrado na Figura 5.27.

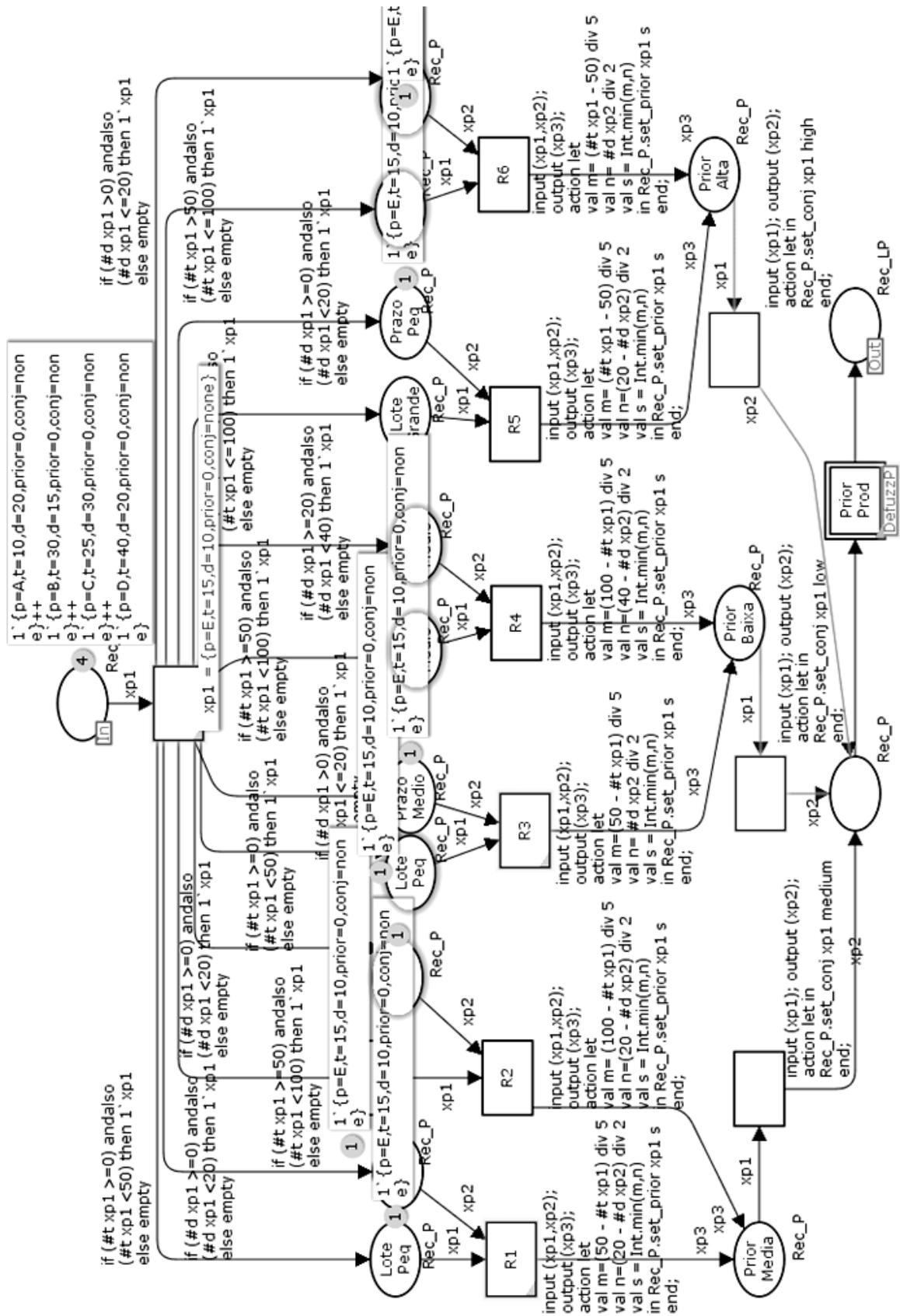


Figura 5. 26 – Modelo em FCPN para o ponto de decisão para escolha dos produtos nos buffers de entrada e de saída

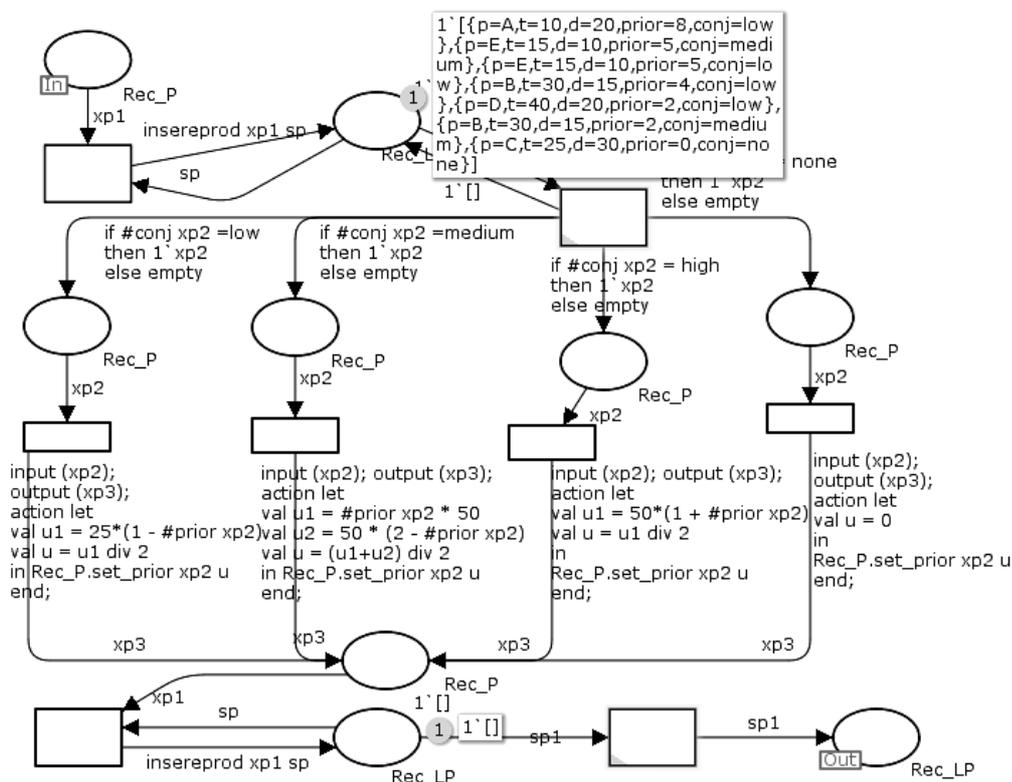


Figura 5. 27 – Modelo da transição PriorProd

A Figura 5.28 mostra o modelo da transição *EscolheRoteiro* quando uma lista de produtos já foi definida e os roteiros do primeiro produto da lista é avaliado.

O número de peças nos buffers de entrada das máquinas pertencentes aos roteiros é calculado como mostrado na Figura 5.29. O tempo de processamento do roteiro também é colocado em uma marcação e estas informações compõem o registro que será usado no modelo fuzzy para a escolha do roteiro.

Após a escolha do roteiro, o produto deve seguir para a primeira máquina deste. Um AGV é requisitado para realizar o transporte. O modelo do *Storage* nesta etapa é mostrado na Figura 5.30. Assim que o produto vai para a máquina, os roteiros do próximo produto da lista são avaliados e este pode requisitar um AGV para levá-lo a primeira máquina para ser processado.

Na estação de trabalho onde o produto vai ser processado, os produtos que chegam são colocados no buffer de entrada. Se a peça chega no buffer e a máquina está livre, a peça pode ir direto para o processamento. Se as peças chegam e a máquina está ocupada, está deve esperar até que a máquina desocupe. Quando duas ou mais peças estiverem aguardando no buffers, uma decisão é realizada para

Na decisão no buffer de entrada será considerada as variáveis prazo de entrega e tempo de processamento. A decisão para atribuir as prioridades é realizada e um produto é definido para entrar em processamento. Os outros produtos voltam para o buffer e aguardam até a máquina ser liberada.

Para a decisão no buffer de saída, as mesmas variáveis são usadas. Neste ponto, uma decisão é realizada quando mais de um produto requisita um AGV. A peça que será transportada primeira é aquela com maior prioridade.

Todas as etapas do processo devem ser repetidas até o produto ser finalizado. Quando o produto é finalizado, este vai para o *Storage*, onde é armazenado.

5.6 Análise de Resultados

Na estação de carga e descarga existiam dois pontos de conflitos. O primeiro, onde os produtos das ordens de produção competiam para entrar primeiro em produção e que requisitaria o AGV primeiro para levá-lo até a primeira máquina de processamento.

O outro conflito na estação de carga e descarga é caracterizado pela escolha de roteiros alternativos para a produção de um produto.

O primeiro caso foi resolvido determinando uma sequência de execução de acordo com prioridades atribuídas aos processos. Com o auxílio de listas de dados, o conflito é resolvido.

Na escolha do roteiro, as opções disponíveis são avaliadas para cada produto e somente aquele com a maior prioridade é escolhido.

Outros dois pontos de conflitos foram identificados no modelo de representação das máquinas. Nestes modelos, o buffer de entrada e de saída possuía capacidade para quatro peças. Quando a máquina estava ocupada e tinha mais de uma peça esperando para usar a máquina, as peças entravam em conflito para a utilização do recurso. Neste caso, uma prioridade era dada aos processos e aquele com prioridade mais alta era executado. Os outros processos voltavam para o buffer de entrada e quando a máquina fosse liberada, suas prioridades seriam calculadas novamente. No buffer de saída, quando uma peça é finalizada, esta

requisita um AGV para transporte. No entanto, quando mais de uma peça estão aguardando por transporte, aquela com maior prioridade têm preferência para requisitar o AGV.

Capítulo 6

CONSIDERAÇÕES FINAIS

6.1 Conclusão

No intuito de definir uma estratégia de modelagem e controle de sistemas de manufatura para solução de conflitos usando Redes de Petri Coloridas Fuzzy, este trabalho apresentou um modelo que considera informações relevantes para a tomada de decisão ao longo da execução do sistema.

Foi implementada a estratégia de modelagem que incorpora a resolução de conflitos utilizando variáveis tanto do planejamento como do controle da produção. A resolução de conflitos usando lógica fuzzy se mostrou bastante eficiente devido ao fato de incorporar variáveis de níveis de informação diferentes dentro de um único sistema e realizar o controle ponderando essas variáveis no momento da tomada de decisão.

A programação dinâmica, baseada em lógica fuzzy, foi usada com sucesso em quatro pontos de conflitos, na estação de carga para escolha dos produtos, na estação de carga para a seleção do roteiro e na estação de trabalho das máquinas nos buffers de entrada e de saída. Com uma política de programação dinâmica, as decisões são realizadas de acordo com o estado corrente do sistema e, caso algum imprevisto ocorra, o planejamento não precisará ser refeito.

O modelo foi implementado e simulado com êxito usando a ferramenta CPNTools, uma ferramenta disponível e acessível que trata extensões de redes de Petri de alto nível. Nela também foi possível a implementação da Rede de Petri Colorida Fuzzy.

A partir dos estudos realizados foi proposta uma modelagem em FCPN das bases de regras usadas em quatro pontos de conflito. A modelagem do sistema baseado em regras fuzzy em redes de Petri possibilitou a modelagem deste no mesmo modelo do sistema de manufatura desenvolvido em redes de Petri colorida.

Os trabalhos realizados para solução de conflitos geralmente usam sistemas de apoio para a tomada de decisão. No entanto, nesta proposta o modelo de decisão é acoplado no mesmo modelo, não necessitando de outro sistema para esta função.

O modelo foi desenvolvido baseado em um FMS com máquinas, buffers e AGV. A maioria dos produtos neste ambiente de validação possuía roteiros de fabricação alternativos. O modelo foi simulado para garantir a solução de conflitos nos pontos do sistema e os resultados foram satisfatórios.

Para a garantia da resolução do conflito, a estrutura de dado *lista* foi usada para que as marcações da rede fossem organizadas de acordo com as prioridades definidas pelo sistema fuzzy e garantir o comportamento do sistema como planejado.

6.2 Trabalhos Futuros

Com base neste trabalho, diversas inclusões e extensões podem ser desenvolvidas. Entre as quais se encontram descritas a seguir.

- A inclusão de mais variáveis para o critério de seleção nos pontos de conflito, incluindo outras informações do sistema de produção.
- Incluir outros pontos ao redor do modelo do sistema onde o AGV possa verificar as solicitações das peças e não somente no estacionamento.
- Comparar medidas de desempenho com outras propostas de programação dinâmica e com outras variáveis nos mesmos pontos de conflito.
- Desenvolver um método de implantação da estratégia em controladores reais de chão de fábrica.

REFERÊNCIAS BIBLIOGRÁFICAS

AGUIRRE, L. A. **Enciclopédia de Automática Controle e Automação**. São Paulo: Editora Blucker, 2007. Volume 1.

ARAÚJO, R. G., **Um Método para Modelagem de Controle de AMS Usando Redes de Petri Virtuais**. 2006. 81 f. Dissertação (Mestrado em Ciência da Computação), Universidade Federal de São Carlos, São Carlos, 2006.

BESKESEA, A.; KAHRAMANA, C.; IRANI, Z. **Quantification of flexibility in advanced manufacturing systems using fuzzy concept**. International Journal of Production Economics, 89, 45-56, 2004.

BILGE, U.; FIRAT, M.; ALBEY, E. **A parametric fuzzy logic approach to dynamic part routing under full routing flexibility**. Computers & Industrial Engineering 55, 15–33, 2008.

BUYURGAN, N.; SAYGIN, C. **An integrated control framework for flexible manufacturing systems**. International Journal of Advanced Manufacturing Technology, 27: 1248–1259, 2006.

CAI, G.; ZHAO, S.; JIA, L.; YE, Y.; LI, X. **Approach of Train Group Operation Conflict Prevention Reasoning Based on Fuzzy Time Petri Nets**, International Conference on Networked Computing and Advanced Information Management, 2008.

CAPRIHAN, R.; KUMAR, A.; STECKE, K. E. **A fuzzy dispatching strategy for due-date scheduling of FMSs with information delays**. International Journal of Flexible Manufacturing Systems, 18:29–53, 2006.

CARDOSO, J. Time Fuzzy Petri Nets, in: J. Cardoso, H. Camargo (Eds.), **Fuzziness in Petri Nets**, Studies in Fuzziness and Soft Computing, Vol. 22, Physica-Verlag, Heidelberg, pp. 115-145, 1999.

CARDOSO, J.; CAMARGO, H. **Fuzziness in Petri Nets**. Studies in Fuzziness and Soft Computing, Physica-Verlag, New York, 1999.

CARDOSO, J.; PRADIN-CHÉZALVIEL, B. **Logic and Fuzzy Petri nets**. In **Workshop on Manufacturing and Petri nets**, Toulouse, France, p. 17-34, June 1997.

CARDOSO, J.; VALETTE, R.; DUBOIS, D. **Fuzzy Petri Nets: an overview**. 13th IFAC World Congress, San Francisco, USA, June 30-July 5, 1996

CHAN, F. T. S.; CHAN., H. K.; KAZEROONI, A. **Real time fuzzy scheduling rules in FMS**. **Journal of Intelligent Manufacturing**, 14,341-350, 2003.

CHANG, A. Y. **On the measurement of routing flexibility: A multiple attribute approach.** International Journal of Production Economics, 109, 122–136, 2007.

CHEN, S. M.; KE, J. S.; CHANG, J. F. **Knowledge Representation Using Fuzzy Petri Nets.** IEEE Transactions on Knowledge And Data Engineering, September, 1990.

DELGADILLO, G. M.; LLANO, S. P. **Scheduling Application using Petri Nets: A Case Study: Intergráficas S.A.** 19th International Conference on Production Research, 2007.

DESROCHERS, A. A.; AL-JAAR, R. Y. **Applications of Petri Net in Manufacturing Systems—Modeling, Control, and Performance Analysis.** IEEE Press, New York, 1994.

FENGLER, W.; WENDT, A.; BOGOLJUBOW, J.; DÄNE, B. **The Use of Fuzzy Coloured Petri Nets for Modelling and Simulation in Manufacturing.** 17th International Conference on Application and Theorie of Petri Nets: Workshop for Manufacturing and Petri Nets, Osaka June 24-28, 1996.

GOLMAKANI, H. R.; MILLS, J. K.; BENHABIB, B. **Deadlock-free optimal routing in flexible manufacturing cells via supervisory control theory.** IEEE International Conference on Systems, Man and Cybernetics, vol 4, 3390- 3395, 2003.

GOMES, L. **On conflict resolution in Petri nets models through model structuring and composition.** IEEE International Conference on Industrial Informatics, 2005.

GROOVER, M. P. **Automation, production systems, and computer integrated manufacturing.** Prentice Hall, 3rd ed., 2007.

HANAFI, J. KARA, S. KAEBERNICK, H. **Generating Fuzzy Coloured Petri Net Forecasting Model to Predict the Return of Products,** International Symposium on Electronics & the Environment, 2007

HANNA, M. **Determination of product quality from an FMS cell using fuzzy Petri nets.** IEEE International Conference on Systems, Man, and Cybernetics, vol.2, no., pp.2002-2007 vol.2, 2-5, Oct 1994.

HANNA, M.; BUCK, A.; SMITH, R. **Fuzzy Petri nets to Control Vision System and Robot Behaviour under Uncertain Situations within an FMS cell.** IEEE International Conference on Fuzzy Systems, v 3, 1994.

HRÚZ B.; ZHOU, M. C. **Modeling and control of discrete systems: with Petri nets and other tool.** Springer, 2007

JAIN, A.; JAIN, P. K.; SINGH, I. P. **Performance Modeling of FMS with flexible process plans – A petri net approach**. International Journal of Simulating Modeling, 5:3, 101 – 113, 2006.

JENSEN, K. **Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use**, Volume 1, Basic Concepts. Monographs in Theoretical Computer Science, Springer-Verlag, 1997.

JENSEN, K.; KRISTENSEN, L. M.; WELLS, L. **Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems**. Springer, 2007.

KATO, E. R. R.; MORANDIN JR, O.; POLITANO, P. R.; CAMARGO, H. A. A. **Control Design Approach for Complex Automatic Manufacturing Systems**. IEEE International Conference on Systems, Man and Cybernetics, 2475-2480, vol.4, 2001.

KLIR, G. J. and YUAN, B. **Fuzzy sets and fuzzy logic: theory and applications**. Upper Saddle River, NJ: Prentice Hall, 1995.

LEE, D. Y.; DICESARE, F. **Scheduling Flexible Manufacturing Systems Using Petri Nets and Heuristic Search**. IEEE Transactions on Robotics and Automation, vol. 10, 123–132, 1994.

LEE, H.; BANERJEE, A. **Modeling and Representation of Manufacturing Process with uncertainties Using Parametric Modular Design and xPNML**. International Symposium on Collaborative Technologies and Systems, 2009

LEE, K. K. **Fuzzy rule generation for adaptive scheduling in a dynamic manufacturing environment**. Applied Soft Computing, 8, 1295–1304, 2008.

Liu, C.; Shi, H.; Yuan, J. **Quality prediction of complex manufacturing processes based on fuzzy Petri Nets**. IEEE International Conference on Fuzzy Systems, pp.1004-1011, 1-6 June 2008.

LOONEY, C.G. **Fuzzy Petri nets for rule-based decision making**. IEEE Transactions on Systems, Man and Cybernetics, vol.18, no.1, pp.178-183, Jan/Feb 1988.

LUNDELL, M.; TANG, J.; NYGARD, K. **Fuzzy Petri Net for UAV Decision Making**. International Symposium on Collaborative Technologies and Systems, 2005.

MARTINS MELO, M. SimRP - **Simulador de Redes de Petri Flexível com Geração de Código VHDL**. 2006. 171 f. Dissertação (Mestrado em Sistemas Mecatrônicos). Universidade de Brasília, Brasília, 2006

MARTINS, R. W. C. **Uma arquitetura modular para controle de FMS**. 2005. 92 f. Dissertação (Mestrado em Ciência da Computação). Universidade Federal de São Carlos, São Carlos, 2005.

MORANDIN JR, O.; KATO, E. R. R. **Virtual PetriNets as a Modular Modeling Method for Planning and Control Tasks of FMS**. International Journal of Computer Integrated Manufacturing, v. 18, 100-106, 2005.

MORANDIN JR., O.; KATO, E. R. R.; ARAUJO, R. G.; SASSO, L. V. **A Modeling Strategy for Control and Interlocking of an AMS using Virtual Petri Nets**. IEEE International Conference on Systems, Man and Cybernetics, 2007.

NOUREDDINE, M.; MARTINEAU, P. **Modelling and Analysis Flexible Manufacturing Systems**. Information Technology Journal, 4 (3): 233-238, 2005.

ODREY, N. C.; MEJÍA, G. **An augmented Petri Net approach for error recovery in manufacturing systems control**. Robotics and Computer-Integrated Manufacturing, 21, 346–354, 2005.

OUELHADJ, D.; PETROVIC, S. **A survey of dynamic scheduling in manufacturing systems**. Journal of Scheduling, 12: 417–431, 2009.

PALOMINO, R. C. **Uma Abordagem para a Modelagem, Análise e Controle de Sistemas de Produção Utilizando Redes de Petri**. 1995. Florianópolis. Dissertação (Mestrado em Engenharia de Produção). Universidade Federal de Santa Catarina, Florianópolis, 1995.

RACOCEANU, D.; MINCA, E.; ZERHOUNI, N. **Fuzzy Petri Nets for monitoring and recovery**. IEEE International Conference on Robotics and Automation, v 3, 2003.

SHNITS, B.; SINREICH, D. **Controlling flexible manufacturing systems based on a dynamic selection of the appropriate operational criteria and scheduling policy**. International Journal of Flexible Manufacturing Systems, 18:1–27, 2006.

SINREICH, D.; SHNITS, B. **A Robust FMS Control Architecture with an Embedded Adaptive Scheduling Mechanism**. Journal of Manufacturing Systems, Vol. 25/No. 4, 2006.

SRINOI, P.; SHAYAN, E.; GHOTB, F. **Scheduling of Flexible Manufacturing Systems Using Fuzzy Logic**. Profiles in Industrial Research: Knowledge and Innovation, 94-102, 2002.

TING, Y.; LU, W. B.; CHEN, C. H.; WANG, G. K. **A fuzzy reasoning design for fault detection and diagnosis of a computer-controlled system**. Engineering Applications of Artificial Intelligence, 2008.

TSINARAKIS, G. J.; TSOURVELOUDIS, N. C.; VALAVANIS, K. P. **Petri Net Modeling of Routing and Operation Flexibility in Production Systems**. Proceedings of the 13th Mediterranean Conference on Control and Automation, Limassol, Cyprus, June 27-29, 2005.

TÜYSÜZ, F.; KAHRAMAN, C. **Modeling a flexible manufacturing cell using stochastic Petri nets with fuzzy parameters**. Expert Systems with Applications 37, 3910–3920, 2010.

TUNCEL, G. A. **Heuristic Rule-Based Approach for Dynamic Scheduling of Flexible Manufacturing Systems Multiprocessor Scheduling: Theory and Applications**, Itech Education and Publishing, Vienna, Austria, pp.436, December, 2007.

TURGAY, S. **Agent-based FMS control**. Robotics and Computer-Integrated Manufacturing, 25, 470– 480, 2009.

VIEIRA, G. E.; HERRMANN, J. W.; LIN, E. **Rescheduling manufacturing systems: a framework of strategies, policies, and methods**. Journal of Scheduling, v. 6, n. 1, p. 35-58, 2003.

WU, N.; ZHOU, M. C.; HU, G. **On Petri Net Modeling of Automated Manufacturing Systems**. Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control, London, UK, 15-17, April 2007.

XIANG, W.; LEE, H. P. **Ant colony intelligence in multi-agent dynamic manufacturing scheduling**. Engineering Applications of Artificial Intelligence, 21, 73–85, 2008.

YEUNG, D.S., SHIU, S.C.K., TSANG, E.C.C. **Modelling flexible manufacturing systems using weighted Fuzzy Coloured Petri Nets**. Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology (1999).

Yeung, D.S.; Liu, J. N. K.; Shiu, S. C. K.; Fung, G. S. K. **Fuzzy coloured Petri nets in modelling flexible manufacturing systems**. Proceedings of ISAI/IFIS 1996: Mexico-USA Collaboration in Intelligent Systems Technologies, pp.100-7, 1996.

ZHANG, H.; JIANG, Z.; GUO, C. **Simulation Based Real-time Scheduling Method for Dispatching and Rework Control of Semiconductor Manufacturing System**. IEEE International Conference on Systems, Man and Cybernetics, 2901-2905, 2007.

ZHANG, H.; LI, D.; YANG, S.; WANG, W. **A New Model of Flexible Manufacturing, System Based on Petri Nets**. Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation August 5 - 8, 2007.

ZHOU, M.; VENKATESH, K. **Modeling, Simulation, and Control of Flexible Manufacturing Systems: A Petri Net Approach**. World Scientific, 1999.

Apêndice A

DECLARAÇÃO DOS TIPOS, VARIÁVEIS E FUNÇÕES DA MODELAGEM EM *CPN TOOLS*

```
colset E = with e|f;
```

```
colset INT = int;
```

```
colset Prod = with A|B|C|D|E|AR1|AR2|CR1|CR2|CR3|CR4|DR1|DR2|ER1|ER2|free;
```

```
colset MACHINE = with M1|M2|M3|M4|M5|M6|S;
```

```
colset AGV = with a|b|c;
```

```
colset CONJ = with low|medium|high|none;
```

```
colset AGVxProd = product AGV*INT*Prod;
```

```
colset Rec_P = record p:Prod * t:INT * d:INT * prior:INT*conj: CONJ;
```

```
colset Rec_R = record p: Prod*t:INT*b:INT* prior:INT*conj: CONJ;
```

```
colset Rec_B = record p: Prod*tr:INT*pe:INT* prior:INT*conj: CONJ;
```

```
colset Rec_LP = list Rec_P;
```

```
colset Rec_LR = list Rec_R;
```

```
colset Rec_LB = list Rec_B;
```

```
colset Tempo = INT;
```

```
colset ListProd = list Prod;
```

```
var lp: ListProd;
```

```
var p1, p2,p3,p4: Prod;
```

```
var tail1: Prod;

var sp,sp1: Rec_LP;
var sr: Rec_LR;
var sb: Rec_LB;

colset MACHINES = list MACHINE;
var m1:MACHINE;
var tail:MACHINES;

var agv1,agv2,agv3,agv4: AGV;
var int1,int2,int3,int4,int5,int6:INT;
var prod1,prod2, prod3, prod4:Prod;

var k, y: INT;
var d1: INT;

var b1,b2,b3,b4,b5,b6: INT;
var t1,t2,t3,t4: Tempo;

var xp1, xp2, xp3: Rec_P;
var xr1, xr2,xr3: Rec_R;
var xb1, xb2,xb3: Rec_B;

fun higPriorProd (p1: Rec_P, p2: Rec_P) =(#prior p1> #prior p2);

fun insereprod elm [] = [elm] | insereprod elm (q::queue) =
  if higPriorProd (elm,q)
  then elm::q::queue
  else q::(insereprod elm queue);

fun higPriorRot (p1: Rec_R, p2: Rec_R) =(#prior p1> #prior p2);

fun insererot elm [] = [elm] | insererot elm (q::queue) =
```

```
if higPriorRot (elm,q)
  then elm::q::queue
  else q::(insererot elm queue);
```

```
fun higPriorBuf (p1: Rec_B, p2: Rec_B) =(#prior p1> #prior p2);
```

```
fun inserebuf elm [] = [elm] | inserebuf elm (q::queue) =
  if higPriorBuf (elm,q)
  then elm::q::queue
  else q::(inserebuf elm queue);
```

Apêndice B

MODELOS EM *CPN TOOLS*

Neste apêndice, encontram-se os modelos do sistema de manufatura similares aos já explicados no Capítulo 5.

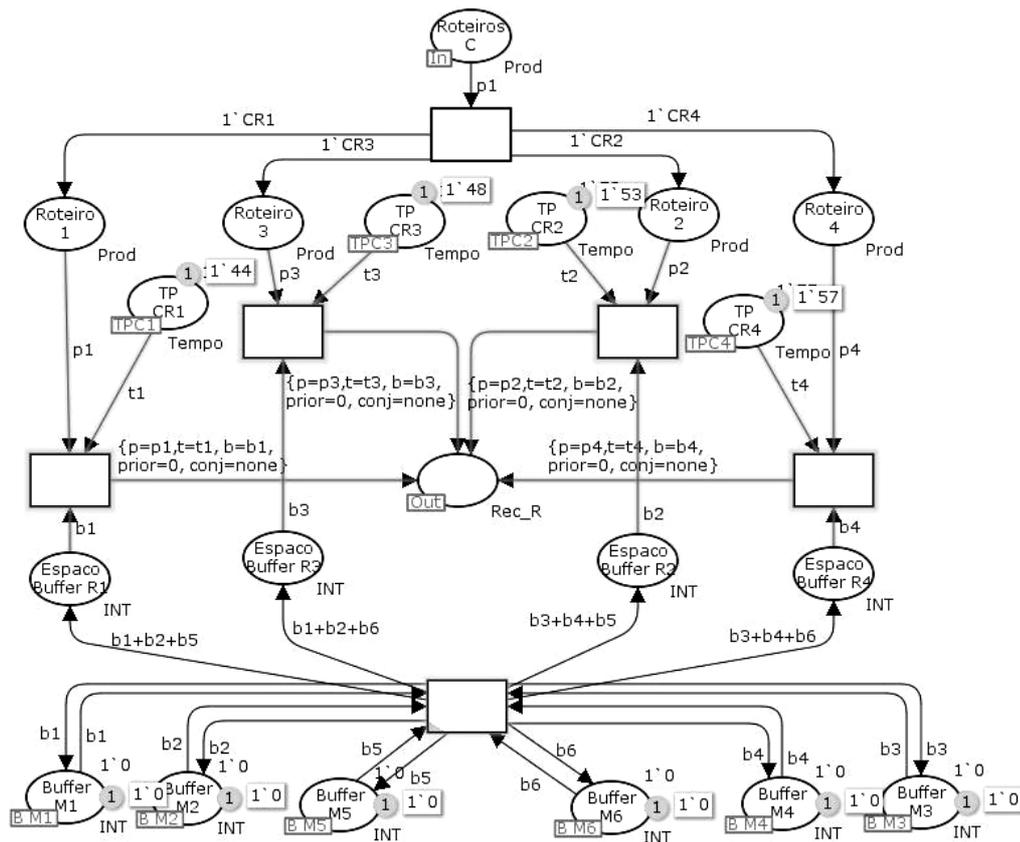


Figura A. 1– Modelo para seleção do roteiro do produto C

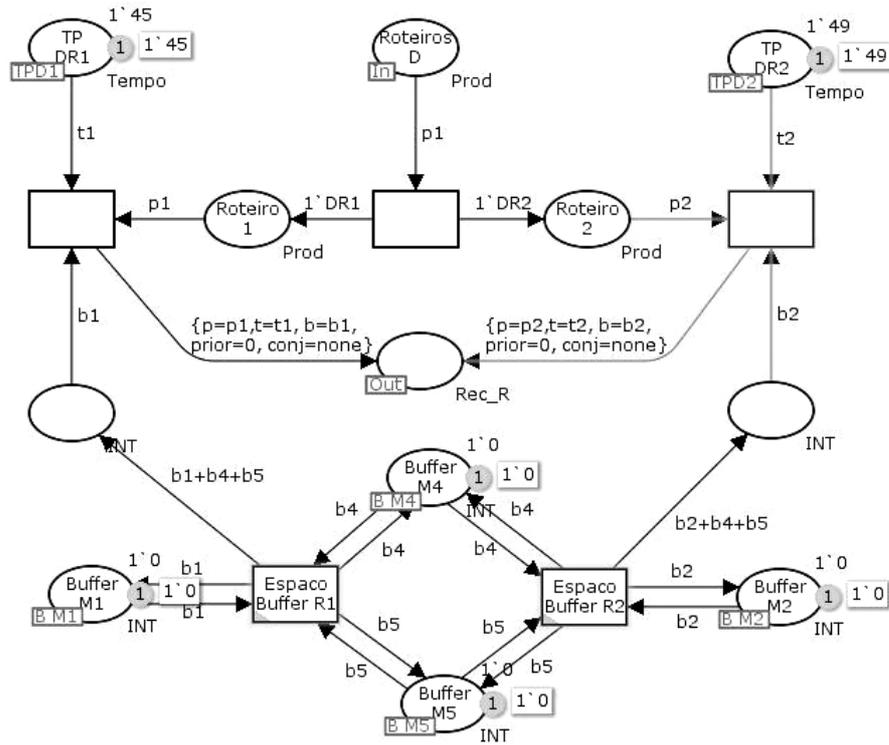


Figura A. 2– Modelo para seleção do roteiro do produto D

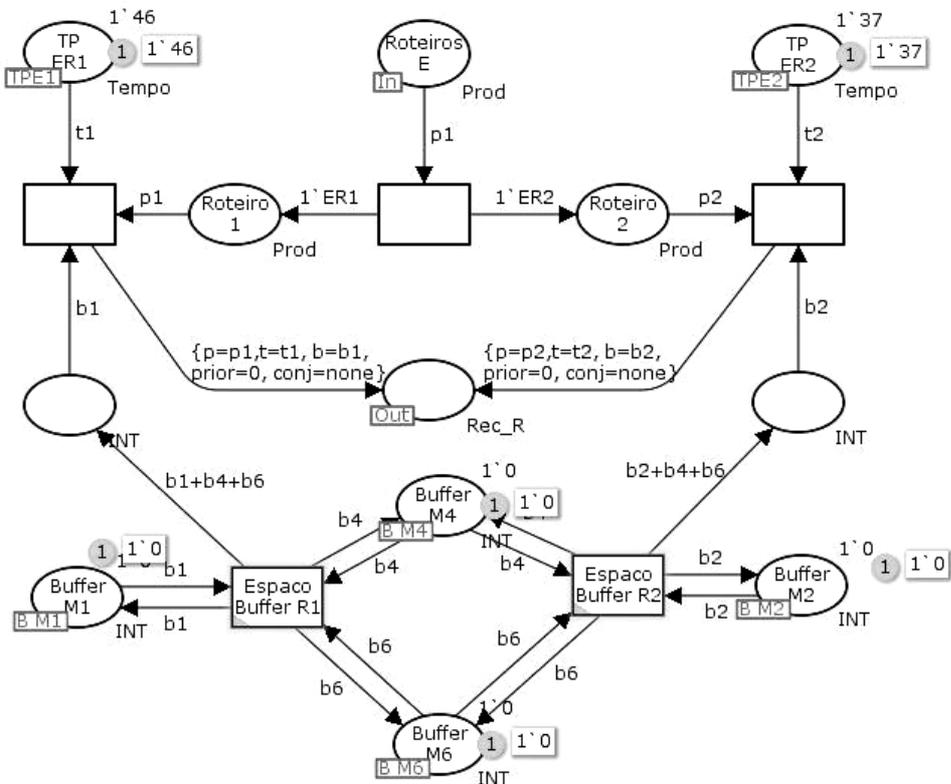


Figura A. 3– Modelo para seleção do roteiro do produto E

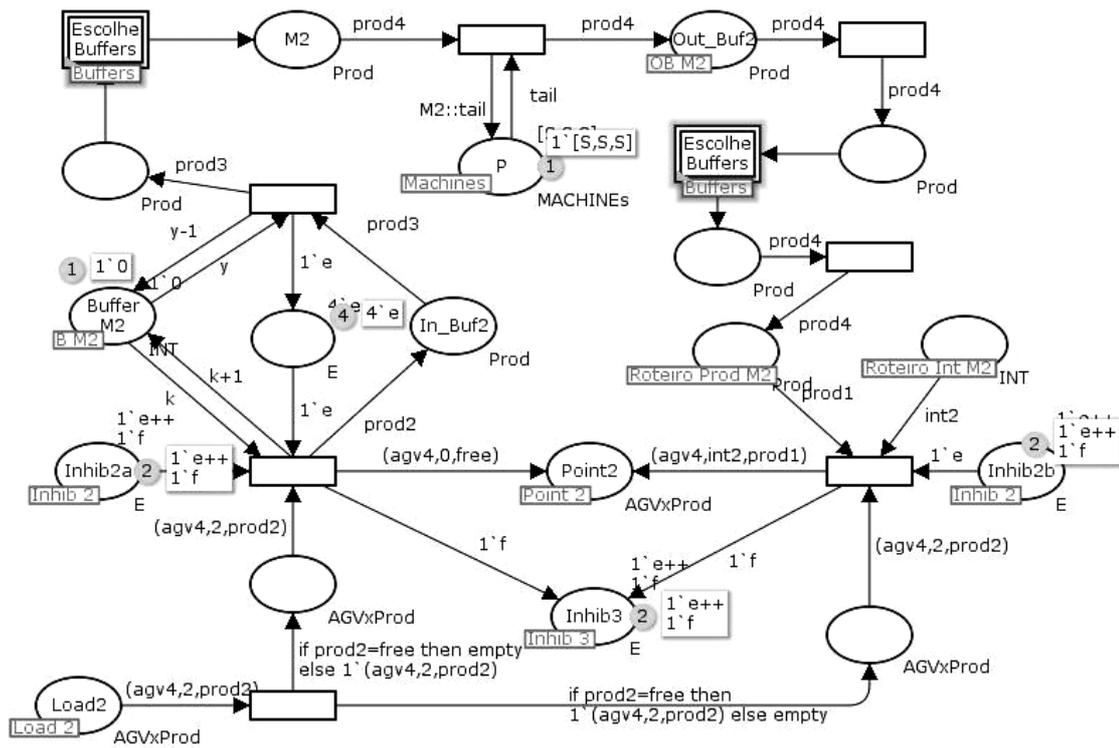


Figura A. 4 – Modelo da estação de trabalho de M2

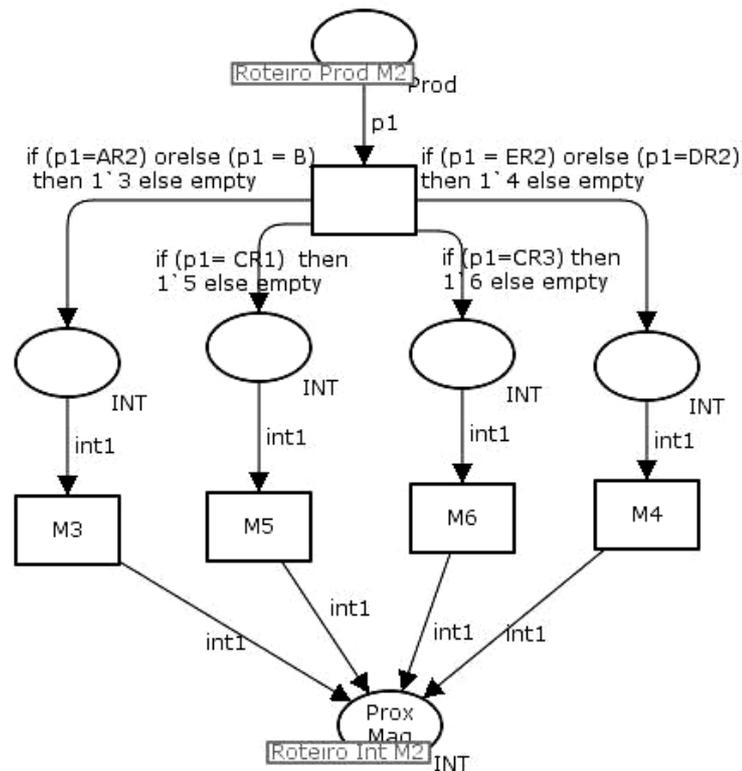


Figura A. 5 –Modelo para seleção do próximo destino após o processo em M2

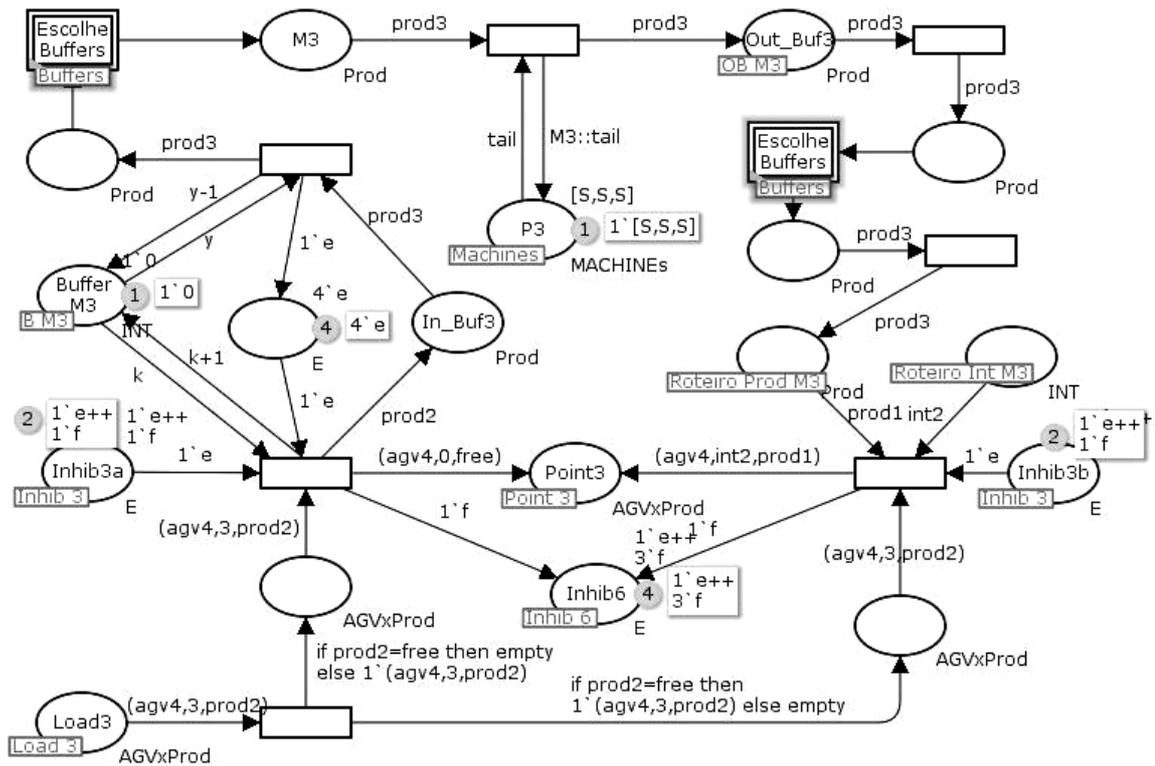


Figura A. 6 – Modelo da estação de trabalho de M3

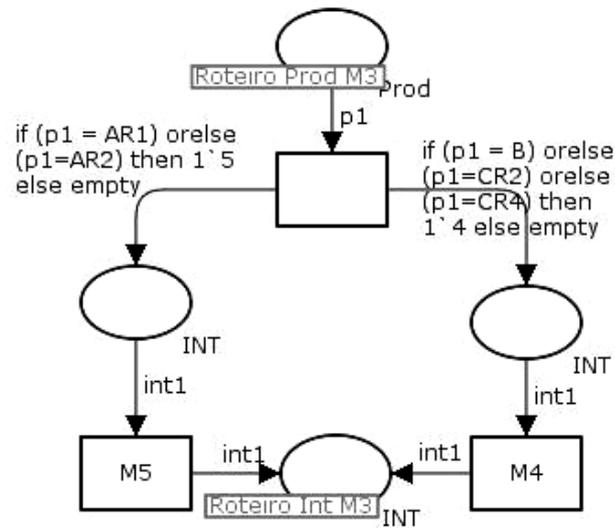


Figura A. 7 – Modelo para seleção do próximo destino após o processo em M3

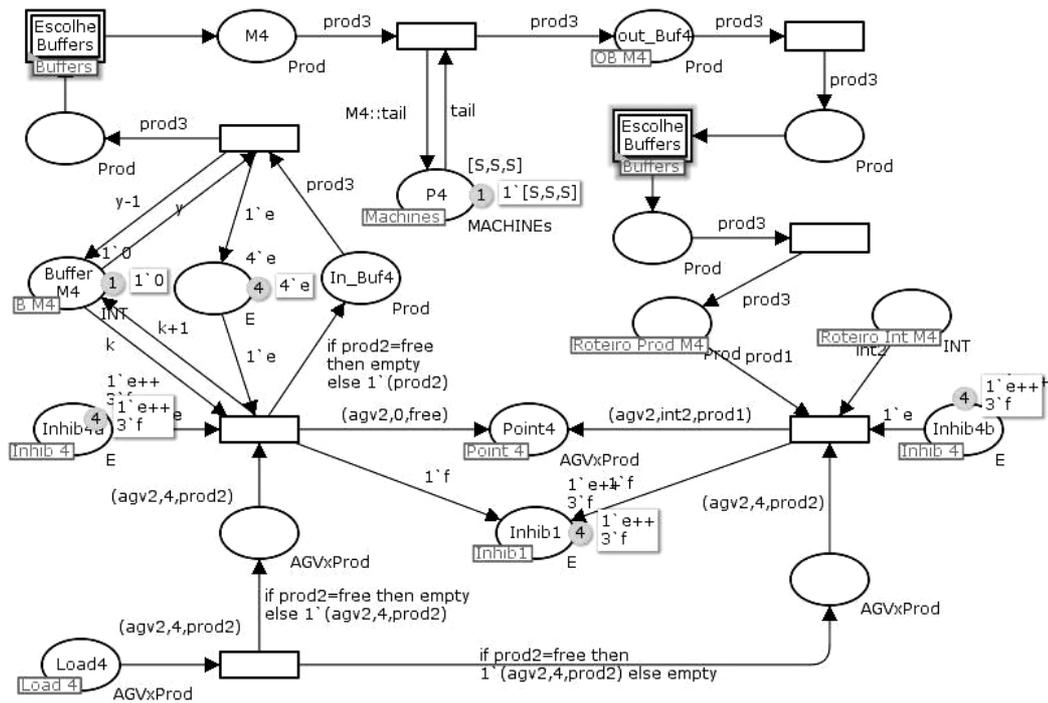


Figura A. 8 – Modelo da estação de trabalho de M4

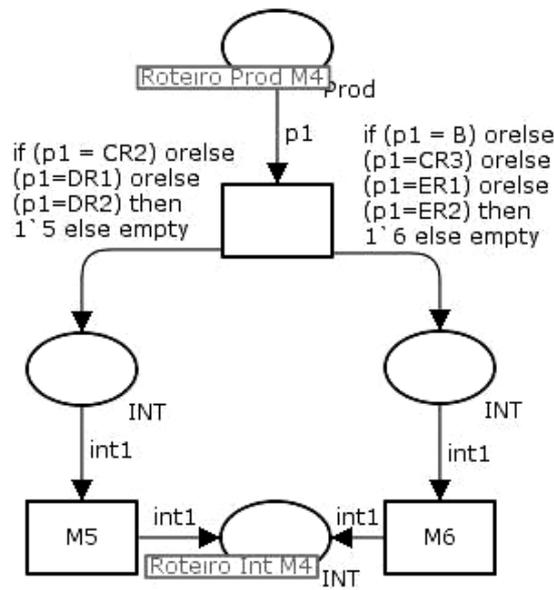


Figura A. 9 –Modelo para seleção do próximo destino após o processo em M4.

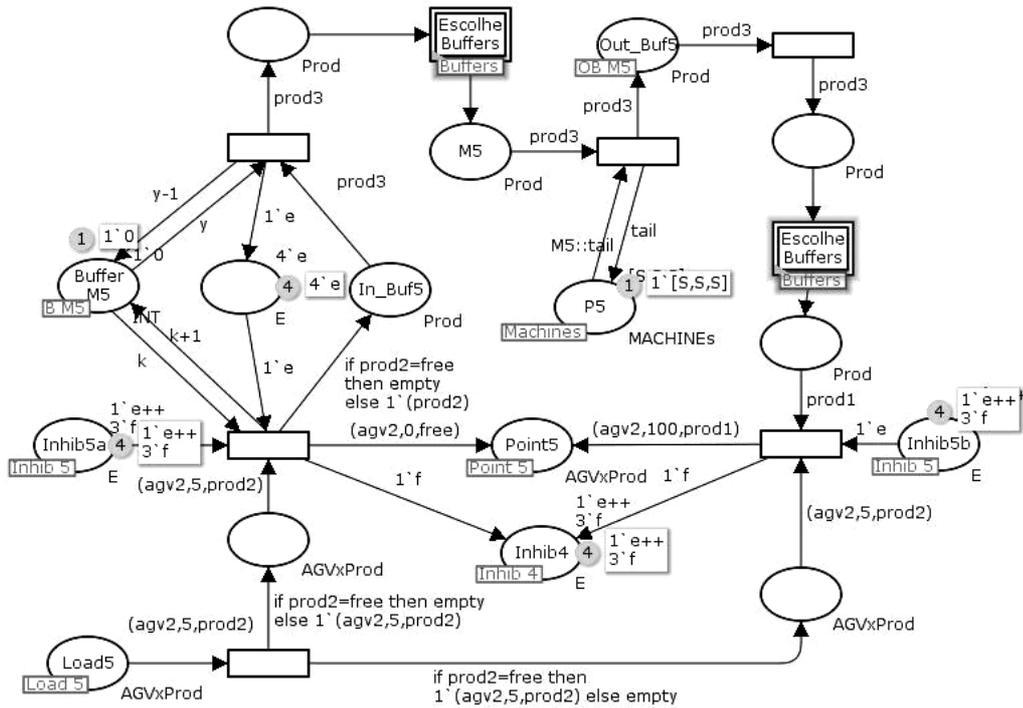


Figura A. 10 – Modelo da estação de trabalho de M5

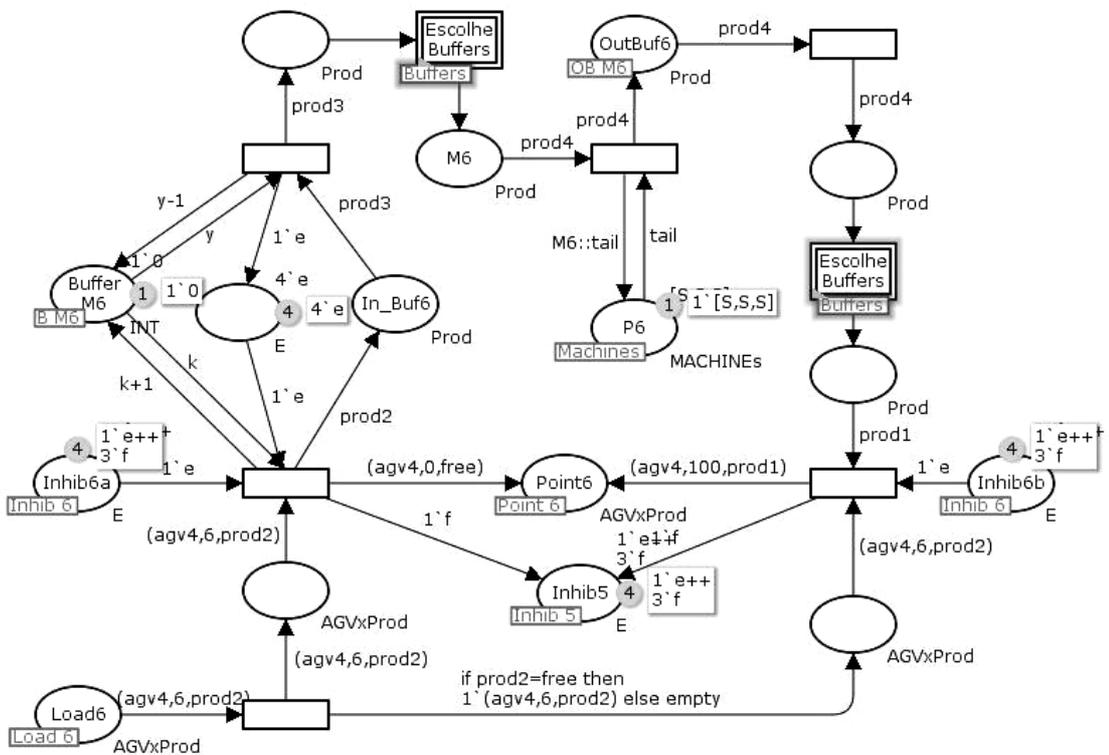


Figura A. 11 – Modelo da estação de trabalho de M6