

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**Modelo *Fuzzy* Cascata Multiatributos e Preditivo
para Despacho de AGVs em FMS**

VINICIUS FERNANDES CARIDÁ

ORIENTADOR: PROF. DR. ORIDES MORANDIN JUNIOR

São Carlos - SP

Maio/2016

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**Modelo *Fuzzy* Cascata Multiatributos e Preditivo
para Despacho de AGVs em FMS**

VINICIUS FERNANDES CARIDÁ

Tese 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 Doutor em Ciência da Computação, área
de concentração: Inteligência Artificial.

Orientador: Prof. Dr. Orides Morandin Junior

São Carlos – SP

Maio/2016

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária UFSCar
Processamento Técnico
com os dados fornecidos pelo(a) autor(a)

C277m Caridá, Vinicius Fernandes
Modelo fuzzy cascata multiatributos e preditivo
para despacho de AGVs em FMS / Vinicius Fernandes
Caridá. -- São Carlos : UFSCar, 2016.
189 p.

Tese (Doutorado) -- Universidade Federal de São
Carlos, 2016.

1. Fuzzy logic. 2. Despacho de AGVs. 3.
Inteligência artificial. 4. Sistemas flexíveis de
manufatura. I. Título.



UNIVERSIDADE FEDERAL DE SÃO CARLOS
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a defesa de Tese de Doutorado do candidato Vinicius Fernandes Caridá, realizada em 02/05/2016.

Prof. Dr. Orides Morandin Júnior
(UFSCar)

Prof. Dr. Aum Marcelo Rizzo Vincenzi
(UFSCar)

Prof. Dr. Pedro Augusto Munari Júnior
(UFSCar)

Prof. Dr. Danilo Sipoli Sanches
(UTFPR)

Profa. Dra. Verônica Oliveira de Carvalho
(UNESP)

Certifico que a sessão de defesa foi realizada com a participação à distância do membro Prof. Dr. Danilo Sipoli Sanches e, depois das arguições e deliberações realizadas, o participante à distância está de acordo com o conteúdo do parecer da comissão examinadora redigido no relatório de defesa do aluno Vinicius Fernandes Caridá.

Prof. Dr. Orides Morandin Júnior
Presidente da Comissão Examinadora
(UFSCar)

Dedico esta conquista a minha mãe Rosimeire e a minha noiva Jerusa

AGRADECIMENTO

Agradeço primeiramente a Deus pela minha vida e saúde que permitiu que eu chegasse até aqui.

A minha mãe, por todo seu amor e dedicação.

A minha noiva, por todo seu carinho e atenção e principalmente compreensão.

A toda minha família, por acreditarem em minhas capacidades e estarem sempre me incentivando e dando o suporte necessário.

Ao meu orientador, professor Dr. Orides Morandin Jr., pela confiança, ideias, incentivo e dedicação, além das conversas interessantes.

Aos meus amigos do laboratório de pesquisa, Carlos, Rogers, Rafael, Maykon, Flávio, Breno, Pedro, Renan, Paulo, Matheus, Eduardo, por toda ajuda, companheirismo e momentos de descontração.

Aos colegas e amigos da pós-graduação.

A todos os professores e funcionários de Departamento de Computação pelas gentilezas e incentivo.

A CAPES pela concessão da bolsa de estudos.

Enfim, aos amigos que fiz no doutorado e todos aqueles que diretamente ou indiretamente contribuíram para realização do doutorado.

"Qualquer coisa que você aprende se torna sua riqueza, uma riqueza que não pode ser tomada de você; seja se você aprende em um prédio chamado escola ou na escola da vida. Aprender algo novo é um prazer sem fim e um tesouro valioso. E nem todas as coisas que você aprende são ensinadas a você, mas muitas coisas que aprende você percebe ter ensinado a si mesmo.

P. Joy Bell

RESUMO

Nos últimos anos, as indústrias vêm cada vez mais aplicando técnicas de automação com o objetivo de aumentar sua eficiência para manterem-se competitivas. O manuseio de materiais é uma atividade essencial em qualquer processo de produção e sua eficácia tem impactos severos sobre os custos de produção. Sistemas de veículos autoguiados (AGVs) tornaram-se uma ferramenta estratégica importante para fábricas e armazéns automatizados. Em um cenário de negócio muito competitivo, eles podem aumentar a produtividade e reduzir os custos. O gerenciamento desses AGVs é a chave para um sistema de transporte que garanta as melhorias vislumbradas pelas indústrias. Um dos principais problemas encontrados no gerenciamento dos AGVs é a decisão de despacho. Esse trabalho propõe um modelo de despacho de veículos baseado em um sistema *fuzzy* cascata para ponderação de múltiplos atributos da fábrica e em uma estrutura baseada em espaço de estados que permita a extração de informações de estados futuros do sistema de produção fabril. O objetivo é reduzir valores de *makespan* e *tardiness* do sistema de produção. Dois cenários de fábrica são simulados e são realizados testes do modelo proposto e cinco outros métodos de despachos. É realizada uma validação estatística dos resultados em que se corrobora com 97% de confiança as hipóteses do trabalho.

Palavras-chave: Despacho de AGV. Multiatributo. Predição. Veículo Autoguiado. Lógica *Fuzzy*. Controle de AGV. Redes de Petri. Sistemas Flexíveis de Manufatura.

ABSTRACT

In recent years, manufacturers are increasingly applying automation techniques with the aim of increase their efficiency to remain competitive. The material handling is an essential activity in any process of production and its effectiveness has severe impacts on production costs. Systems of automated guided vehicles (AGVs) have become an important strategic tool for factories and automated warehouses. In a very competitive business scenario, they can increase productivity and reduce costs. The management of these AGVs is the key to a transport system that ensures the improvements envisioned by the industry. One of the main problems encountered in the management of AGVs is the dispatch decision. This paper proposes a vehicles dispatch model based on a fuzzy cascade system for consideration of multiple attributes of the factory and a structure based on state space that enables the extraction of information of future states of the industrial production system. The objective is to reduce makespan and tardiness values of the production system. Two factory scenarios are simulated and tests are performed of the model and five other methods of dispatch. A statistical validation is realized of the results in which corroborates with 97% confidence the hypotheses of the work.

Keywords: AGV Dispatching. Multiattribute. Prediction. Automated Guided Vehicle. Fuzzy Logic. AGV control. Petri nets. Flexible Manufacturing Systems.

LISTA DE FIGURAS

Figura 1 – Arranjo físico usado para validação do trabalho de Sing, Sarngadharan e Pal (2011).	41
Figura 2 – Divisão do arranjo físico para evitar <i>Deadlock</i> do trabalho de Sing, Sarngadharan e Pal (2011).....	41
Figura 3 – Arranjo físico usado para validação do trabalho de Ho, Liu e Yih (2012). 43	
Figura 4 – Arranjo físico usado para validação do trabalho de Kim e Yarlagadda (2013).	44
Figura 5 – Arranjo físico usado para validação do trabalho de Chang et al. (2013). ...	54
Figura 6 – Arranjo físico usado para validação do trabalho de Chiba, Arai e Ota (2010).	56
Figura 7 – Arranjo físico usado para validação do trabalho de Nishi e Maeno (2010).	58
Figura 8 – Arranjo físico usado para validação do trabalho de Nishi e Tanaka (2012) com o sistema de transporte contendo 115 nós e 128 arestas.....	59
Figura 9 – Proposta da tese conectada ao ambiente de manufatura em um nível macro de detalhes	66
Figura 10 – Proposta da tese conectada ao ambiente de manufatura e com o usuário em um nível maior de detalhes.....	67
Figura 11 – Esquema macro do sistema multiatributo <i>fuzzy</i> cascata.....	68
Figura 12 – Esquema detalhado do sistema multiatributo <i>fuzzy</i> cascata.	71
Figura 13 - Representação do sistema <i>fuzzy</i> Mamdani.....	71
Figura 14 – Representação esquemática macro do sistema de predição	73
Figura 15 – Exemplo de um estado no ambiente de manufatura e roteiro de produção	74
Figura 16 – Possíveis estados futuros 1	75
Figura 17 – Possíveis estados futuros 2.....	75
Figura 18 – Representação esquemática detalhado do sistema de predição.....	76
Figura 19 – Fluxograma básico do funcionamento da proposta.....	79
Figura 20 – Esquema da proposta multiatributo <i>fuzzy</i> em conjunto com predição.	80
Figura 21 – Proposta da tese conectada ao ambiente de manufatura e com o usuário em um nível maior de detalhes.....	83

Figura 22 – Esquema da proposta multiatributo <i>fuzzy</i> em conjunto com predição	84
Figura 23 – Esquema dos módulos e suas comunicações para funcionamento e validação da proposta.....	85
Figura 24 – Módulo fábrica da Figura 23.....	86
Figura 25 – Arranjo físico da fábrica fictícia (que será chamada de fábrica 1)	87
Figura 26 – Roteiros de produção da fábrica 1.....	88
Figura 27 – Arranjo físico da fábrica inspirada em uma indústria moveleira em operação (que será chamada de fábrica 2)	93
Figura 28 – Módulos da Figura 23 que serão apresentados em detalhes nas subseções a seguir.	96
Figura 29 – Visualização em 2D da fábrica 1 modelada no <i>software</i> SIMIO.....	97
Figura 30 – Visualização em 3D da fábrica 1 modelada no <i>software</i> SIMIO.....	98
Figura 31 – Visualização em 2D da fábrica 2 modelada no <i>software</i> SIMIO.....	98
Figura 32 – Visualização em 3D da fábrica 2 modelada no <i>software</i> SIMIO.....	98
Figura 33 – Configuração de dados do SIMIO, cadastro dos itens que a fábrica produz.	100
Figura 34 – Configuração de dados do SIMIO, cadastro do roteiro de produção dos itens a serem produzidos e tempos de processamento e setup.....	101
Figura 35 – Configuração dos recursos da fábrica.	102
Figura 36 – Criação de processos e steps.	103
Figura 37 – Estrutura geral do sistema <i>fuzzy</i> 1.	104
Figura 38 – Variável distância do sistema <i>fuzzy</i> 1.....	105
Figura 39 – Base de regras do sistema <i>fuzzy</i> 1.	105
Figura 40 –Módulo predição da Figura 23.	106
Figura 41 – Modelo em redes de Petri dos roteiros de fabricação da fábrica 1 modelada no <i>software</i> CPN Tools.	107
Figura 42 – Parte do modelo em redes de Petri dos roteiros de fabricação da fábrica 1 modelada no <i>software</i> CPN Tools.....	107
Figura 43 – Modelo em redes de Petri dos roteiros de fabricação da fábrica 2 modelada no <i>software</i> CPN Tools.	108
Figura 44 – Parte do modelo em redes de Petri dos roteiros de fabricação da fábrica 2 modelada no <i>software</i> CPN Tools.....	108

Figura 45 – Diagrama de comunicação entre os elementos do sistema. As setas em ambos os sentidos indicam possibilidade tanto de envio como de obtenção de informações.	110
Figura 46 – Diagrama de casos de uso do sistema.	111
Figura 47 – Diagrama de classes do Projeto CPN Control. As classes em azul são do projeto CPN Control. Classes em bege são originais da biblioteca Access/CPN, e seus detalhes foram omitidos.....	112
Figura 48 – Serviço getMark. O place de nome E1P1 possui como marcação “1`P1-E0`” Isso significa que o place E1P1 possui 1 token, do tipo string, com o valor “P1-E0”. 114	
Figura 49 – Serviço getNumStates. Nesse exemplo foram explorados todos os arcos possíveis (15), uma vez que o limite especificado foi 20, e foram encontrados 16 estados distintos. Isso significa que a partir da rede na configuração atual podem-se disparar diferentes combinações de transições e serão obtidas até 15 outras configurações distintas (estados) para a rede.....	115
Figura 50 – Serviço transmiteMarcacao – transmissão de um dos estados. Nesse exemplo, o estado transmitido foi o estado atual da fábrica. Note que a marcação de cada <i>place</i> possui a seguinte notação: nome da página da rede (RoteirosFabrica), seguida de um ponto e o nome do <i>place</i> (por exemplo, E1P1), seguido de dois pontos e a marcação propriamente dita (por exemplo: “1`P1-E0`”). Observe que os <i>places</i> que possuem marcação vazia são omitidos.	116
Figura 51 – Exemplo serviço getNumVizinhos. A partir da configuração da atual existem 4 outras configurações distintas imediatamente posteriores, uma vez que existem 4 transições habilitadas na rede.	116
Figura 52 – Serviço getTransicao. Nesse exemplo, para que a rede mudasse da configuração no estado de identificador zero (estado atual) para a configuração no estado 1, seria necessária a ativação da transição Maq1_1.....	117
Figura 53 – Exemplo de execução do serviço getTransicoesHabilitadas().	118
Figura 54 –Módulo protocolo de comunicação da Figura 23.....	121
Figura 55 – Tela do <i>software</i> Simio apresentado o processo com os steps.....	123
Figura 56 – Telas de configuração do <i>deploytool</i>	125
Figura 57 – Importação dos arquivos .dll para o módulo <i>decision</i>	126
Figura 58 – Valores médios de <i>makespan</i> dos testes realizados na fábrica 1	141
Figura 59 – Valores médios de <i>tardiness</i> dos testes realizados na fábrica 1	145
Figura 60 – Valores médios de <i>makespan</i> dos testes realizados na fábrica 2	149

Figura 61 – Valores médios de <i>tardiness</i> dos testes realizados na fábrica 2.....	153
Figura 62 – Realização do Teste T, verificação de amostras independentes	154
Figura 63 – Realização do Teste T, verificação do poder do teste pelo tamanho da amostra	155
Figura 64 - Representação geral de uma função de pertinência triangular.	176
Figura 65 - Exemplo de <i>Fuzzyficação</i>	177
Figura 66 - Agregação da regra 1 utilizando operador de min.....	178
Figura 67 - Agregação da regra 2 utilizando operador de min.....	179
Figura 68 - Composição das regras 1 e 2 utilizando operador máx.	179
Figura 69 - Método de <i>defuzificação</i> CoA	180
Figura 70 – Exemplo de um ambiente de manufatura modelado em redes de Petri ..	182
Figura 71 – Espaço de estados total da rede de Petri apresentada na Figura 70	183
Figura 72 – Equação de estados instanciada para o exemplo dado na Figura 71	184

LISTA DE TABELAS

Tabela 1 – Resultados apresentados no trabalho de Dahari e Liu (2012).	60
Tabela 2 – Síntese dos trabalhos apresentados no capítulo	62
Tabela 3 – Setores e Processos.	90
Tabela 4 – Produtos e Peças.	90
Tabela 5 – Modelos de Produtos.	91
Tabela 6 – Processos, máquinas e peças.	92
Tabela 7 – Roteiros dos Produtos (Processos e Máquinas) da fábrica 2	94
Tabela 8 – Algoritmo para acionamento do serviço	113
Tabela 9 – Pseudo código do algoritmo	120
Tabela 10 – Trechos do código do módulo GetInfo	124
Tabela 11 – trechos de código do módulo ControlInterface	124
Tabela 12 – Script criado no Matlab	125
Tabela 13 – Trechos do código para uso dos sistemas <i>fuzzy</i>	127
Tabela 14 – Pedido 1 para fábrica 1	132
Tabela 15 – Pedido 1 para fábrica 2	133
Tabela 16 – Resultados de <i>makespan</i> do pedido 1	138
Tabela 17 – Resultados de <i>makespan</i> médios dos testes realizados na fábrica 1	139
Tabela 18 – Resultados dos ganhos percentuais para <i>makespan</i> dos testes realizados na fábrica 1	140
Tabela 19 – Resultados de <i>tardiness</i> do pedido 1	142
Tabela 20 – Resultados médios de <i>tardiness</i> dos testes realizados na fábrica 1	143
Tabela 21 – Resultados dos ganhos percentuais para <i>tardiness</i> dos testes realizados na fábrica 1	144
Tabela 22 – Resultados de <i>makespan</i> do pedido 1 dos testes realizados na fábrica 2	146
Tabela 23 – Resultados de <i>makespan</i> médios dos testes realizados na fábrica 2	147
Tabela 24 – Resultados dos ganhos percentuais para <i>makespan</i> dos testes realizados na fábrica 2	148
Tabela 25 – Resultados de <i>tardiness</i> do pedido 1 dos testes realizados na fábrica 2.	150
Tabela 26 – Resultados médios de <i>tardiness</i> dos testes realizados na fábrica 2.	151
Tabela 27 – Resultados dos ganhos percentuais para <i>tardiness</i> dos testes realizados na fábrica 2	152

Tabela 28 – Resultados da Análise de Teste T <i>Fuzzy2</i> +Predição > FIFO para <i>makespan</i> na fábrica 1	155
Tabela 29 – Resultados da Análise de Teste T <i>Fuzzy2</i> +Predição > FIFO para <i>makespan</i> na fábrica 1	156
Tabela 30 – Todos os resultados da Análise de Teste T para <i>makespan</i> e <i>tardiness</i> na fábrica 1	157
Tabela 31 – Todos os resultados da Análise de Teste T para <i>makespan</i> e <i>tardiness</i> na fábrica 2	158
Tabela 32 – Resumo dos resultados finais	162

LISTA DE ABREVIATURAS E SIGLAS

CR	Razão Crítica (do inglês: <i>Critical Ratio</i>)
DBR	<i>Drum Buffer Rope</i>
EDD	Data de Entrega mais Próxima (do inglês: <i>Earliest Due Date</i>)
FBPN	<i>Fuzzy Back-Propagation Network</i>
FCM	<i>Fuzzy C-Means</i>
FEFS	Encontrada Primeiro Atendida Primeiro (do inglês: <i>First Encountered First Served</i>)
FIFO	Primeiro a Entrar Primeiro a Sair (do inglês: <i>First In First Out</i>)
FMS	Sistema Flexível de Manufatura (do inglês: <i>Flexible Manufacturing System</i>)
AGV	Veículo Autoguiado (do inglês: <i>Automated Guided Vehicle</i>)
RGV	Veículo Guiado por Trilho (do inglês: <i>Rail Guided Vehicle</i>)
SF	Sistema <i>Fuzzy</i>
SFBR	Sistema <i>Fuzzy</i> Baseado em Regras
BC	Base de Conhecimento
MI	Mecanismo de Inferência
BD	Base de Dados
BR	Base de Regras
SCF	Sistema de Classificação <i>Fuzzy</i>
AG	Algoritmo Genético
STM	Sistema de Transporte de Materiais
PMI	Instituto de Gerência de Projetos (do inglês: <i>Project Management Institute</i>)

QC	Guindaste de Cais (<i>do inglês: Quay Cranes</i>)
CGA	Algoritmo Genético Canônico (<i>do inglês: Canonical Algorithm Genetic</i>)
RAF	Raciocínio para Antecipar o Futuro
AD-FMS	Autonomia Descentralizada em Sistema Flexível de Manufatura (<i>do inglês: Autonomous Decentralized in Flexible Manufacturing Systems</i>)
PMBOK	Conjunto de conhecimento do Gerenciamento de Projetos (<i>do inglês: Project Management Body of Knowledge</i>)
TAD	Tabela de Alocação Dinâmica
SFG	Sistemas <i>Fuzzy</i> Genéticos
PN	Redes de Petri (<i>do inglês: Petri Nets</i>)
FSMCT	Regra para Reduzir a Oscilação do Tempo Médio do Ciclo (<i>do inglês: Fluctuation Smoothing Rule for Mean Cycle Time</i>)
FSVCT	Regra para Reduzir a Oscilação do Tempo de Variação do Ciclo (<i>do inglês: Fluctuation Smoothing Rule for Cycle Time Variation</i>)
HVF	Maior Valor Primeiro (<i>do inglês: High-Value First</i>)
LIFO	Último a Entrar Primeiro a Sair (<i>do inglês: Last In First Out</i>)
LPT	Tempo de Processamento mais Longo (<i>do inglês: Longest Processing Time</i>)
LVF	Menor Valor Primeiro (<i>do inglês: Low-Value First</i>)
MROQS	Menor Espaço Restante no <i>Buffer</i> de Saída (<i>do inglês: Minimum Remaining Outgoing Queue Space rule</i>)
NVS	Primeiro o veículo mais próximo (<i>do inglês: Nearest Vehicle First</i>)
PID	Proporcional Integral Derivativo
PP	Programação da Produção
PRP	Programação Reativa da Produção

RMS	Sistema de Manufatura Reconfigurável
SD	Menor Distância (do inglês: <i>Shortest Distance</i>)
SPT	Menor tempo de Processamento (do inglês: <i>Shortest Process Time</i>)
SRO	Seleção em Ordem Aleatória (do inglês: <i>Select in Random Order</i>)
SRPT	Menor Tempo de Processamento Restante (do inglês: <i>Shortest Remaining Process Time</i>)
SSED	<i>Software</i> de Simulação de Eventos Discretos
STDF	Primeiro a distância mais curta de viagem (do inglês: <i>Shortest Travel Distance First</i>)
TPN	Redes de Petri Temporizadas
WBMR	Peso Modificado Baseado na RRule (do inglês: <i>Weight Biased Modified RRule</i>)

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO.....	20
1.1 <i>Contextualização.....</i>	20
1.2 <i>Motivação</i>	22
1.3 <i>Justificativa</i>	25
1.4 <i>Hipóteses do trabalho.....</i>	27
1.5 <i>Objetivos</i>	28
1.6 <i>Delimitações do trabalho.....</i>	29
1.7 <i>Método de pesquisa e desenvolvimento.....</i>	29
1.8 <i>Organização do trabalho</i>	32
CAPÍTULO 2 – ESTADO DA ARTE	33
2.1 <i>Considerações iniciais.....</i>	33
2.2 <i>Problemas gerais relacionados ao controle de AGVs.....</i>	34
2.2.1 <i>Despacho e Programação.....</i>	35
2.3 <i>Trabalhos científicos correlatos à contribuição da tese.....</i>	37
2.4 <i>Considerações finais.....</i>	61
CAPÍTULO 3 – PROPOSTA DO TRABALHO	64
3.1 <i>Considerações iniciais.....</i>	64
3.2 <i>Proposta do sistema fuzzy em cascata e da estrutura baseada em espaço de estados para despacho de AGVs.....</i>	65
3.2.1 <i>Sistema Multiatributo Fuzzy Cascata.....</i>	67
3.2.2 <i>Sistema de Predição</i>	72
3.2.3 <i>Funcionamento da proposta multiatributo fuzzy em conjunto com predição</i>	78
3.3 <i>Considerações finais.....</i>	81
CAPÍTULO 4 – DESENVOLVIMENTO	82
4.1 <i>Considerações iniciais.....</i>	82
4.2 <i>Desenvolvimento do sistema de despacho de AGVs e dos modelos de fábrica</i>	82
4.2.1 <i>Desenvolvimento e características dos cenários de Fábrica.....</i>	86
4.2.2 <i>Sistema multiatributo fuzzy cascata</i>	95
4.2.3 <i>Sistema de Predição</i>	106
4.2.4 <i>Implementação do software Protocolo de comunicação</i>	121

4.3	<i>Considerações finais</i>	127
CAPÍTULO 5 – VALIDAÇÃO		128
5.1	<i>Considerações iniciais</i>	128
5.2	<i>Verificação e Validação do Software</i>	129
5.3	<i>Validação da Proposta</i>	130
5.3.1	Planejamento dos Experimentos	130
5.3.2	Análise da hipótese.....	131
5.3.3	Execução dos Experimentos	132
5.4	<i>Considerações finais</i>	159
CAPÍTULO 6 – CONCLUSÃO		161
6.1	<i>Síntese do contexto da proposta e dos objetivos</i>	161
6.2	<i>Contribuições e delimitações</i>	162
6.3	<i>Trabalhos futuros</i>	163
REFERÊNCIAS		164
APÊNDICE A – CONCEITOS TEÓRICOS DA PROPOSTA		175
7.1	<i>Explicações e equações dos sistemas fuzzy usados na Tese</i>	175
7.2	<i>Explicações sobre geração e exploração do espaço de estados</i>	181
7.3	<i>Noções de estatística</i>	185
7.3.1	Testes de Hipóteses <i>t-student</i>	185
7.3.2	Método de verificação de normalidade de uma distribuição	188
7.3.3	Método de cálculo de tamanho amostral por <i>t-student</i>	188

Capítulo 1

INTRODUÇÃO

1.1 Contextualização

As indústrias usam cada vez mais a automação para se adequar ao mercado consumidor e manterem-se competitivas. Como apresentado em Slack et al. (1997), o sucesso competitivo é decorrente de determinados objetivos de desempenho, como qualidade, custo, flexibilidade, rapidez e confiabilidade. Sistemas produtivos procuram considerar esses objetivos, que são intrinsecamente ligados, com o propósito de alcançar vantagens competitivas, na medida em que são conciliados para produzir de acordo com as expectativas dos clientes.

Segundo Aguirre (2007), a busca pela competitividade no meio industrial tornou-se umas das características mais marcantes no processo de evolução dos sistemas produtivos. Especificamente, considerando-se o âmbito deste trabalho, os sistemas de manufatura acompanham essa característica onde novas tecnologias têm sido desenvolvidas e usadas no intuito de atingir esse propósito.

Leitão (2011) afirma que as indústrias de manufatura são as principais geradoras de lucros da economia mundial e são reconhecidas por sua complexidade e imprevisibilidade, devido ao constante aumento da necessidade por produtividade, flexibilidade e reconfigurabilidade.

Os Sistemas Flexíveis de Manufatura (FMS) são sistemas de produção que atendem ao requisito de flexibilidade, fornecendo diversos tipos de mecanismos como forma de garantir uma rápida resposta às exigências de mudanças relativas ao mercado. Dentre outras características, esses sistemas possibilitam uma produção com médio grau de variedade de produtos e com médio volume de produção. Um dos principais componentes do FMS é o

Sistema de Transporte de Materiais (STM), que tem como principal função mover peças e outros materiais de um local para outro, de maneira segura, eficiente e com precisão (SLACK et al., 1997).

Dentro do ambiente de manufatura, o sistema de transporte também é responsável por realizar a integração do processo, pois transporta os materiais em processo e os produtos de um ponto ao outro da fábrica, permitindo assim a existência de um fluxo de produção. Um ponto importante destacado por Groover (2003) é que o custo de transportar os materiais pela fábrica tem grande impacto no custo total do produto.

Torres et al. (2015) e Tompkins et al. (2013) realçam a importância do manuseio de materiais, o que representa de 30% até 75% do custo de fabricação de um produto. O manuseio de materiais usa cerca de 25% do total do pessoal, 55% do total de espaço físico e 87% do tempo de todas as atividades.

A distribuição de materiais é considerada uma importante fonte de oportunidades que proporciona desafios importantes para a indústria (ANBUUDAYASANKAR; GANESH; MOHAPATRA, 2014); (VIVALDINI, K. C. T. et al., 2015).

Segundo Pradhan, Kumar e Sinha (2014), Groover (2003) e Stecke (1986) AGV é usado principalmente em aplicações industriais para mover o material em torno das instalações de fábrica e a maioria dos FMSs emprega os AGVs como transportadores. Os AGVs são veículos autoguiados usados para inúmeras aplicações, mas basicamente os objetivos gerais são: aumentar a segurança, aumentar a flexibilidade, diminuir custos e automatizar a logística de movimentação de cargas no interior das empresas, diminuindo a dependência da produção pelo operador.

Controlar corretamente as atividades dos AGVs bem como as cargas transportadas é um ponto chave para a eficiência do sistema de produção. Podem-se tomar várias decisões referentes ao controle dos AGVs, tais como:

- Despacho: que é responsável por escolher os destinos dos AGVs;
- Roteamento: que escolhe o caminho para se chegar ao destino;
- Controle de trânsito: que durante o percurso evita conflitos e *deadlock*.

Pela complexidade inerente às decisões que envolvem as atividades dos AGVs é comum a realização de estudos em áreas isoladas buscando simplificar o problema e chegar a um bom resultado.

O problema de despacho de AGVs determina a próxima tarefa a ser executada dentre um grupo de tarefas esperando para serem atendidas (KIZIL; OZBAYRAK; PAPADOPOULOU, 2006). De acordo com algum critério, o AGV escolhe qual tarefa irá transportar de um ponto a outro dentro da fábrica.

Apesar de parecer uma tarefa trivial (a escolha de uma requisição de transporte feita por uma estação de trabalho para o AGV atender), esta se torna mais complexa quando adicionamos várias estações de trabalho ou AGVs e vários produtos ao processo de decisão, que segundo Lin, *et al.* (2006) é um problema combinatorial.

Para as tomadas de decisões são considerados atributos referentes à área do controle que está sendo abordada. Porém, mesmo ao abordar o despacho, podem ser considerados no modelo de decisão atributos como recursos compartilhados ou operações que não estão diretamente relacionados com o despacho, mas sim com outros problemas de controle como, por exemplo, o roteamento.

1.2 Motivação

A movimentação interna de materiais como parte do sistema de manufatura pode assumir custos significativos em face da natureza do processo produtivo (MARTINS; ALT, 2009). Uma melhor eficiência no manuseio de materiais torna-se fundamental para o aumento da produtividade, diminuição de custos, cumprimento de prazos, etc. O uso de AGVs no sistema de transporte pode trazer inúmeras vantagens, como: segurança no transporte do material, fácil adaptação a uma planta existente, fácil integração com outros sistemas automatizados, entre outros (VIS, 2006).

No cenário acadêmico, o despacho de AGVs tem sua relevância reconhecida quando se consideram os numerosos artigos científicos que abordam esta área na atualidade.

No cenário das indústrias manufatureiras os AGVs também se mostram relevantes. De acordo com o relatório de pesquisa de mercado “*Automated Guided Vehicle Market by Type*

(Unit Load Carrier, Tow Vehicle, Pallet Truck, Assembly Line Vehicle), Industry Vertical (Automotive, & Others), Application (Transportation, Distribution, & Others), & Geography - Global Forecast to 2020", publicado pela MarketsandMarkets¹ em 2015, é esperado que o mercado de AGV chegue a US\$ 2.240,15 milhões em 2020, crescendo a uma taxa anual média de 9,8% entre 2014 e 2020. No relatório publicado em 2015 pela empresa ReportsnReports² afirma-se que o mercado de AGV deverá crescer a um ritmo rápido na maioria das regiões do mundo, devido ao crescente investimento em automação em todas as principais indústrias.

Cada vez se torna mais necessária a tomada de decisões adequadas sobre projeto, planejamento e operação em sistemas que exigem alocação eficiente de recursos escassos. Essa é a área em que atua a Pesquisa Operacional. “*Pesquisa Operacional é a aplicação de métodos científicos a problemas complexos para auxiliar no processo de tomada das decisões, tais como projetar, planejar e operar sistemas em situações que requerem alocações eficientes de recursos escassos*” (ARENALES et al., 2007).

Existem pesquisas sobre alocação de recursos partindo do viés da programação da produção e programação reativa da produção e outras partindo do viés de despacho. A principal diferença entre eles está relacionada a visão do sistema produtivo, seja ela em tempo ou em recursos.

No Laboratório de Estratégias de Automação (Tear) da UFSCar, onde este trabalho foi desenvolvido, são realizadas pesquisas voltadas à automação industrial, focando em problemas relacionados à otimização, controle e projeto de indústrias de manufatura. Para a proposta de soluções de tais problemas, no Tear desenvolvem-se pesquisas visando aprimorar técnicas e métodos usando inteligência artificial e *softwares* de simulação de eventos discretos. Tendo alguns trabalhos abordando o viés da programação reativa (MORANDIN, O. et al., 2007); (MORANDIN, O. et al., 2008); (MORANDIN, O.; KATO; TUMA, 2010); (MORANDIN, ORIDES J. et al., 2013); (TUMA; MORANDIN; CARIDÁ, 2013) e outros do despacho de AGVs (BENINCASA; MORANDIN; KATO, 2003); (MORANDIN, O. et al., 2006); (MORANDIN, O. J. et al., 2011a); (MORANDIN, O. J. et al., 2011b) (CARIDÁ et al., 2014); (CARIDÁ; MORANDIN; TUMA, 2015).

¹ <http://www.prnewswire.com/news-releases/automated-guided-vehicle-market-worth-224015-million-by-2020-501667821.html>

² <http://www.prnewswire.com/news-releases/automated-guided-vehicle-market-2019-2020-forecasts-in-new-research-reports-503715081.html>

Autores como Naso e Turchiano (2005); Sarin, Varadarajan e Wang (2011); Chang, Dong e Yang (2013); Chen e Matis (2013) apresentam algumas vantagens do despacho abseado em regras:

- Encontra soluções perto do ótimo;
- Tem um cálculo muito rápido, o que o torna ideal para programações sinuosas em sistemas de grande escala;
- Rápido em reagir às mudanças encontradas no chão de fábrica;

Segundo Gonzalez-R, Framinan e Ruiz-Usano (2010) entre as diferentes abordagens para programar tarefas em um sistema de produção, as regras de despacho são amplamente usadas na prática, particularmente em cenários no qual a fabricação é afetada por diferentes fontes de variabilidade. Ainda segundo Chen, Rajendran e Wu (2013) as regras de despacho continuam sendo uma das técnicas mais promissoras para aplicações práticas.

O despacho de AGV é uma tarefa ligada ao controle da produção em que são realizadas alocações dos recursos de fábrica. O sistema de despacho de AGV deve decidir qual das n tarefas que estão aguardando por transporte o veículo deve atender. O despacho de AGV não tem relação com o controle de movimentação do veículo.

A complexidade de realizar o despacho de AGVs está relacionada à quantidade de variáveis que podem afetar o sistema de transporte, e que, portanto, devem ser levadas em consideração. Uma variável importante é a distância, porém ao atender sempre as tarefas mais próximas é possível ficar preso em um mínimo local. Outras variáveis que podem ser analisadas são os *buffers* de entrada e saída das estações de trabalho, uso da estação de trabalho, etc. No entanto a realização do cálculo do despacho não pode despender muito tempo, pois o veículo fica estacionado até que uma decisão seja tomada.

Outro aspecto muito importante a ser considerado é que existem vários objetivos que podem ser alvos de otimização como, por exemplo, o *makespan* e o *tardiness*. Minimizar o *makespan* é diminuir o tempo total de produção e continuar produzindo a mesma quantidade de produtos com a mesma quantidade de recursos. Minimizar o *tardiness* é diminuir os atrasados levando em consideração a data de entrega de cada produto.

Portanto, ao tentar minimizar o *makespan* não há preocupação com a data de entrega, o que pode gerar maiores atrasos e ao tentar minimizar o *tardiness* não se leva em consideração o tempo total da produção.

Conseqüentemente, em um ambiente de fábrica composto por n produtos, m estações de trabalho, vários veículos, dois objetivos de minimização e tempo restrito, o despacho de veículos se torna uma tarefa altamente complexa.

Desse modo, o problema do despacho caracteriza-se por definir as melhores tarefas para os veículos, sendo que as melhores tarefas são correlacionadas ao objetivo de otimização.

1.3 Justificativa

No Tear, foram desenvolvidas pesquisas relacionadas ao despacho de AGVs que serviram como base, referência e definição para este projeto. Destacando alguns deles, tem-se o trabalho desenvolvido por Benincasa, Morandin e Kato (2003) no qual foi criado um sistema *fuzzy* ponderando três variáveis na decisão do despacho. Morandin et al. (2007) desenvolveram um algoritmo genético para otimizar a base de regras criada no trabalho anterior e melhorar o despacho de AGVs.

Também foram desenvolvidas pesquisas que serviram de testes e resultados preliminares deste projeto. Morandin, Caridá, Kato e Fonseca (2011a) desenvolveram um sistema *fuzzy* hierárquico e Morandin, Caridá, Kato e Tuma (2011b) desenvolveram um sistema *fuzzy* ponderando quatro variáveis divididas em bases de dados com alta granularidade e um algoritmo genético para realizar ajustes da base de regras. Reis et al. (2013) apresentaram uma forma de modelagem de FMS em CPN usando *fuzzy* para decisões de controle. Caridá et al. (2014) apresentaram um modelo de predição usando espaço de estados e Caridá, Morandin e Tuma (2015) apresentaram duas abordagens *fuzzy* para despacho de AGV e minimização de *makespan*.

Por meio da revisão bibliográfica verifica-se que em meados dos anos 80 iniciaram-se trabalhos para caracterização de regras de despacho para AGVs (EGBELU; TANCHOCO, 1984); (KULWIEC, 1984). Desde então o tema é pesquisado, primeiramente com propostas de regras simples de despacho (EGBELU, 1987); (BARTHOLDI III; PLATZMAN, 1989); (HAN;

MCGINNIS, 1989); (TAGHABONI, 1989). Posteriormente surgiram pesquisas abordando regras multiatributos (KLEI; KIM, 1996); (KIM, C. W.; TANCHOCO; KOO, 1999); (TAN; TANG, 2000); (YAMASHITA, 2001); (UMASHANKAR; KARTHIK, 2006), pesquisas abordando o roteamento dos AGVs (TAGHABONI-DUTTA; TANCHOCO, 1995); (SOYLU; ÖZDEMIREL; KAYALIGIL, 2000); (DUINKERKEN; OTTJES; LODEWIJKS, 2006), outros abordando *deadlock* (NAIQI; MENGCHU, 2007); (GUAN; DAI, 2009); e gestão de combustível/energia (MCHANEY, 1995).

Nos últimos cinco anos foram publicados mais de 250 artigos científicos relacionados a AGV no mundo que se dividem principalmente em roteamento (MARTÍNEZ-BARBERÁ; HERRERO-PÉREZ, 2010); (SMOLIC-ROCAK et al., 2010); (DAHARI; LIU, 2012); (UMAR et al., 2015); (VIVALDINI, K. et al., 2015); (SANTOS et al., 2016), gestão de baterias (OLIVEIRA et al., 2011); (ZHANG; ZHU; CHAN, 2014), resolução de *deadlocks* (KATO; MORANDIN; SGAVIOLI, 2010); (TAO, 2011); (NISHI; TANAKA, 2012); (FANTI et al., 2015) e despacho (CHOE; PARK; RYU, 2010); (MANIYA; BHATT, 2011); (HO; LIU; YIH, 2012); (CHEN, B.; MATIS, 2013); (AZADEH et al., 2013); (BANDARA; MAYORGA; MCLAY, 2014); (SHAHZAD; MEBARKI, 2016); (CHOE; KIM; RYU, 2016), sendo frequente o uso de técnicas de inteligência artificial para apoiar a tomada de decisão.

No que diz respeito ao despacho de AGVs, pesquisadores, (AZADEH et al., 2013); (KIM, Y.-H.; YARLAGADDA, 2013); (SU, 2014); (BANDARA; MAYORGA; MCLAY, 2014) usam técnicas de inteligência artificial como, por exemplo, lógica *fuzzy* e redes neurais, para análise de múltiplas variáveis da fábrica. Pesquisadores como, por exemplo, (NASO; TURCHIANO, 2005); (HIDEHIKO, 2008); (SMOLIC-ROCAK et al., 2010); (GHANNADPOUR et al., 2014) propõem métodos para previsão de possíveis situações futuras da fábrica.

Existem inúmeros métodos e sistemas de despacho de AGVs, porém, na revisão de literatura, ficou claro que as pesquisas que ponderam múltiplas variáveis da fábrica para tomada de decisão analisam poucas variáveis, pois, como os próprios autores justificam, a cada variável incluída, o modelo e o custo computacional crescem exponencialmente.

Todavia, ao não se incluir recursos compartilhados da fábrica, que se tornam variáveis, no sistema de decisão se pode chegar a um valor do critério de otimização maior ou errôneo. Por exemplo, ao se tentar otimizar o valor de *makespan*, que significa diminuir o tempo total de produção e continuar produzindo a mesma quantidade de produtos com a mesma quantidade

de recursos, e não se considerar o recurso “trecho de trajetória” a ser usado pelos veículos da fábrica, é passível de ocorrências de congestionamento de veículos e, portanto, ocorrência de valores maiores de *makespan* ou valores que não condizem com o que ocorrerá na fábrica. A mesma condição pode acontecer com outras variáveis, como por exemplo o tempo de *setup*. A não inclusão do tempo de *setup* no sistema de decisão pode causar maior incidência de *setup* de máquinas na fábrica no decorrer da produção, conseqüentemente, o cálculo do valor de *makespan* sem considerar o *setup* poderá ser errôneo.

Portanto, ao se desconsiderar alguns recursos compartilhados da fábrica, imagina-se que o grau de certeza sobre o valor do critério de otimização encontrado por meio do modelo é baixo em relação ao que será praticado na fábrica onde esses recursos irão incidir, isto é, decisões menos assertivas em relação à verdadeira disponibilidade desses recursos, conseqüentemente, valores de *makespan* ou *tardiness* equivocados e na média superiores.

Com relação às pesquisas que tentam prever situações futuras da fábrica, não se tem um modelo baseado em espaços de estados, no qual não se prevê, mas por meio das simulações se prediz os possíveis estados futuros da fábrica. Por fim, nota-se que os trabalhos buscam a minimização de apenas um objetivo de otimização.

1.4 Hipóteses do trabalho

A primeira hipótese desta tese é a de que incluir recursos da fábrica como, o uso das rotas do chão de fábrica e o tempo de *setup* de máquina, além das variáveis habituais por meio de um sistema *fuzzy* em cascata que permita alcançar um valor de *makespan* ou *tardiness* mais correto e que na média seja menor se comparado com propostas que não consideram os recursos em questão.

A segunda hipótese é a de que o uso de informações de estados futuros do sistema de produção fabril por meio de uma estrutura baseada em espaço de estados permita alcançar um valor de *makespan* ou *tardiness* menor se comparado com propostas que não avaliam estados futuros.

A terceira hipótese é a de que incluir outros recursos da fábrica por meio de um sistema *fuzzy* em cascata e usar informações de estados futuros do sistema de produção fabril por meio

de uma estrutura baseada em espaço de estados, isto é, trabalhar com a primeira e segunda hipótese em conjunto, permita alcançar um valor de *makespan* ou *tardiness* mais correto e menor se comparado com os valores obtidos por meio das hipóteses separadas.

1.5 Objetivos

Os objetivos gerais deste trabalho são:

- Apresentar um modelo *fuzzy* em cascata que permita ponderar múltiplos atributos da fábrica almejando a minimização de dois critérios de otimização, *makespan* e *tardiness*.
- Apresentar uma estrutura baseada em espaços de estados que permita extração de informações futuras da fábrica almejando a minimização de dois critérios de otimização, *makespan* e *tardiness*.
- Apresentar uma composição dos dois objetivos gerais anteriores, ou seja, um modelo *fuzzy* em cascata que permita ponderar múltiplos atributos da fábrica juntamente com uma estrutura baseada em espaços de estados que permita extração de informações futuras da fábrica almejando a minimização de dois critérios de otimização, *makespan* e *tardiness*.

Os objetivos específicos são:

- Criar um arranjo físico espelhado em um fábrica existente e em operação.
- Representar os roteiros de fabricação por meio de uma modelagem em redes de Petri.
- Modelar o arranjo físico por meio de um *software* de simulação de eventos discretos.
- Criar um *software* que gere e manipule espaços de estados de uma rede de Petri.
- Criar um *software* que permita a comunicação da simulação da fábrica com o modelo em redes de Petri.

1.6 Delimitações do trabalho

Vários fatores impactam na tarefa de transporte de materiais, alguns relacionados ao AGV, outros relacionados ao arranjo físico da fábrica, outros vinculados a outros setores que não o chão de fábrica. Por se tratar de um problema complexo, foi delimitado o escopo apenas para tratar o despacho de AGV, sem avaliar o impacto de outros elementos como, por exemplo, gerenciamento de bateria dos AGVs, melhor arranjo físico para a fábrica, controle de *deadlock*, ciclo de manutenção dos AGVs, tamanho da frota de AGVs. Além disso, são consideradas algumas delimitações aos cenários de fábrica propostos, relatadas a seguir:

- Não serão consideradas manutenções das estações de trabalho (preventiva, corretiva);
- Não serão consideradas as trocas de turno dos operadores;
- Não serão considerados os tempos de carga e descarga dos produtos;
- Os trechos de trajetória dos veículos não serão detalhados na rede de Petri;
- Cada estação de trabalho opera apenas um produto por vez e cada produto só pode ser operado por uma estação de trabalho por vez;
- Não é admitida preempção na fabricação e no transporte dos produtos;
- Os tempos de produção são determinísticos e previamente conhecidos;
- Inicia-se um produto/tarefa (*job*) transportando a matéria prima do setor de carga para o *buffer* de entrada da primeira estação de trabalho em seu roteiro;
- Termina-se um produto/tarefa (*job*) transportando-se o produto finalizado do *buffer* de saída da última estação de trabalho a operá-lo para o setor de descarga.

1.7 Método de pesquisa e desenvolvimento

Esta tese segue o método de pesquisa Hipotético-Dedutivo (POPPER, 2007); (LAKATOS; MARCONI, 2008), pois busca refutar que a hipótese formulada obtenha resultados melhores que os obtidos e publicados por outros métodos, dentro de um determinado cenário. Estatisticamente a não confirmação de que é pior ou igual aos anteriores, significa que é melhor (SILVA; MENEZES, 2005).

As hipóteses desta tese são:

- Ao se incluir recursos da fábrica como, o uso das rotas do chão de fábrica e o tempo de *setup* de máquina, além das variáveis habituais por meio de um sistema *fuzzy* em cascata alcançar-se-á um valor de *makespan* ou *tardiness* mais correto e que na média seja menor se comparado com propostas que não consideram os recursos em questão.
- Ao se usar informações de estados futuros do sistema de produção fabril por meio de uma estrutura baseada em espaço de estados alcançar-se-á um valor de *makespan* ou *tardiness* menor se comparado com propostas que não avaliam estados futuros.
- Ao se incluir outros recursos da fábrica por meio de um sistema *fuzzy* em cascata e usar informações de estados futuros do sistema de produção fabril por meio de uma estrutura baseada em espaço de estados, alcançar-se-á um valor de *makespan* ou *tardiness* mais correto e menor se comparado com os valores obtidos por meio das hipóteses separadas.

Segundo Popper (2007):

- A hipótese não é defendida para provar que está certa, mas procura-se demonstrar que é falsa. Caso não se consiga falseá-la, diz-se que “comprovou sua qualidade” ou foi “corroborada”. As hipóteses não são verificáveis, mas podem ser corroboradas.
- Os métodos estatísticos são essencialmente hipotético-dedutivos.

Orientado pelo tipo de pesquisa, foi definida uma série de atividades para desenvolver o trabalho. Iniciado por um levantamento bibliográfico da área em questão (despacho de veículos) para identificar técnicas e teorias com potencial de compor o trabalho.

Com base no levantamento bibliográfico e do histórico de trabalhos realizados no laboratório Tear, foram levantadas as técnicas e os métodos para realizar o despacho de veículos, identificando as variáveis a serem analisadas e a importância de considerar variáveis relativas às estratégias competitivas, visto que a competitividade da indústria também é guiada por suas estratégias.

Após a escolha da forma que será realizado o despacho, assim como as variáveis consideradas, a proposta foi modelada e implementada seguindo as seguintes etapas:

- Definição dos métodos de inferência, conversão escalar/*fuzzy* e conversão *fuzzy*/escalar dos sistemas *fuzzy*;
- Definição da cascata *fuzzy*;
- Definição da modelagem em rede de Petri para permitir a estrutura de espaço de estados;
- Estabelece-se o *software* SIMIO como ambiente de simulação de fábrica;
- Estabelece-se o *software* CPNtools como ambiente de simulação de redes de Petri;
- Definição do protocolo de comunicação para permitir a troca de informações entre os módulos da proposta;
- Estabelece-se como ferramenta de programação o ambiente de programação *Microsoft Visual Studio*;
- Um sistema de computador é desenvolvido para permitir a verificação da viabilidade da proposta assim como de sua comparação com outros métodos de despacho de AGVs.

Em seguida à implementação do sistema, é realizada a sua validação e a da proposta. O objetivo principal é de se avaliar as hipóteses. A validação segue as seguintes etapas:

- Um planejamento dos experimentos é feito e os diversos pedidos são executados em duas fábricas;
- A execução do sistema usando outros métodos de despacho de AGVs;
- A execução do sistema usando o método de despacho de AGVs proposto;
- Os resultados são analisados com o auxílio de ferramentas de estatística. Essa análise permite corroborar as hipóteses levantadas.

Para que se garanta o nível de significância pretendido nos testes estatísticos são gerados pedidos vários pedidos de produção e cada pedido é executado inúmeras vezes para cada método de despacho de AGVs variando-se a quantidade de AGVs atuando na fábrica. As amostras obtidas são analisadas por meio de testes de hipóteses *t-student* a fim de garantir a corroboração das hipóteses desta tese.

1.8 Organização do trabalho

Para clareza e melhor compreensão deste trabalho, a seguir é apresentada e justificada a sua divisão.

O “Capítulo 2 – Estado da arte” tem por objetivo embasar as hipóteses desta tese, por isso apresenta rapidamente algumas características sobre os problemas que incidem sobre o controle de AGVs e a diferença entre despacho e programação. Posteriormente evidenciam-se, identificam-se e avaliam-se as principais pesquisas relacionadas ao tema deste trabalho coletadas na revisão sistemática.

Após a análise dos trabalhos correlatos e os *gaps* encontrados na literatura, é apresentado o “Capítulo 3 – Proposta do trabalho” no qual são discutidos e apresentados detalhadamente todos os pontos da proposta. Alguns conceitos teóricos são omitidos com o objetivo de evitar que o capítulo 3 fique sobrecarregado, porém esses conceitos estão disponíveis no “Apêndice A – Conceitos teóricos da proposta”.

No “Capítulo 4 – Desenvolvimento” é apresentado o desenvolvimento da proposta, expondo-se modelos, trechos de códigos e detalhando como foi desenvolvido todo o sistema de despacho de AGVs proposto até se obter o produto final o qual foi validado no capítulo a seguir.

No “Capítulo 5 – Validação” são apresentados os resultados e análises estatísticas obtidos nos testes da proposta para que no “Capítulo 6 – Conclusão” seja apresentado um parecer dos resultados atingidos, considerando as contribuições e as delimitações encontradas e possíveis trabalhos futuros. Os conceitos teóricos sobre a análise estatística apresentada no capítulo 5 estão disponíveis no “Apêndice A – Conceitos teóricos da proposta”.

Com o objetivo de se ter uma ampla documentação do projeto, foi criado um relatório técnico onde estão descritos todos os testes de software, bem como todos os pedidos de fabricação usados nas simulações e igualmente todos os resultados obtidos. Esse documento encontra-se disponível para consulta no repositório do laboratório Tear (<http://www.tear.dc.ufscar.br/>) ou pode ser solicitado ao autor dessa tese.

Capítulo 2

ESTADO DA ARTE

2.1 Considerações iniciais

Veículos Autoguiados (AGVs) são sistemas de manuseio de materiais em processos automatizados, sistemas de manufatura flexível, armazéns e até em movimentação de *containers* em portos marítimos (QIU et al., 2002); (EVERS; KOPPERS, 1996); (YE; HSU; VEE, 2000); (KIM, Y.-H.; YARLAGADDA, 2013).

Muitos trabalhos relacionados a AGVs vêm sendo propostos. Existem diversos pontos inerentes ao problema de controle de transporte com AGVs e cada vez mais se busca aperfeiçoamento para uma maior flexibilidade, competitividade e qualidade.

Diferentes objetivos de otimização podem ser encontrados na literatura, como custos totais de transporte do caminho, *makespan*, *leadtime*, *tardiness*, *work in process*, entre outros. Na busca pela otimização de tais objetivos são propostos na literatura o uso de técnicas de inteligência artificial como redes neurais, algoritmos genéticos e lógica *fuzzy*.

Neste capítulo, serão apresentados os principais problemas existentes no controle de transporte com AGVs e suas dificuldades e posteriormente focando no problema do despacho de AGVs. Também serão apresentados trabalhos correlatos às hipóteses deste trabalho, exibidas no capítulo anterior, com o objetivo de fortalecer as hipóteses, bem como, consolidar a proposta desta tese.

2.2 Problemas gerais relacionados ao controle de AGVs

Existem vários problemas inerentes aos sistemas de transporte com AGVs. Peters et al. (1996) formalizam um sistema de classificação de controle, dividindo esses problemas em três vertentes:

- *Layout* do caminho;
- Capacidade de carga do veículo;
- Mecanismo de endereçamento do veículo.

Os autores Le-Anh e De Koster (2006) apresentam uma revisão sobre o projeto e controle de sistemas de veículos autoguiados. Os autores citam questões chaves que envolvem os sistemas de AGVs, como: tipo e quantidade de veículos; posição do veículo ocioso; arquitetura de controle; resolução de *deadlock*; roteamento e despacho de veículos.

Tipo e quantidade de veículos: A quantidade de veículos tem grande influência no desempenho dos sistemas de AGVs. Um sistema com quantidade de AGVs abaixo do ideal não é capaz de atender a demanda de transporte adequadamente e um sistema com quantidade de AGVs acima do ideal aumenta a chance de congestionamentos e *deadlocks* sem contar o custo dos AGVs. Assim, determinar a quantidade adequada de veículos é importante.

Posição do veículo ocioso: A posição dos veículos ociosos, também é apontada como um ponto a ser tratado em sistemas de transportes. Ao invés de fazer com que os veículos retornem ao depósito de veículos, é mais eficiente estacioná-los em locais de manutenção ou pontos que estão próximos das liberações de carga. Os locais para estacionamento dos veículos devem ser selecionados para minimizar o tempo de resposta aos pedidos ou para distribuir veículos pela rede.

Arquitetura de controle: Sendo o AGV um veículo independente de controlador humano, é necessário o uso de uma arquitetura de controle para designar tarefas aos AGVs de alguma maneira. É comum encontrar duas abordagens tradicionais: centralizada (hierárquico) e descentralizada (heterárquico), cada uma tendo suas vantagens e desvantagens. Outras abordagens de controle são propostas, como, por exemplo a arquitetura holônica, em que é possível trabalhar ora a arquitetura centralizada, ora a descentralizada, combinando as vantagens de ambas dependendo do contexto.

Deadlock: Uma das diferenças dos FMSs para outros sistemas de manufatura é o compartilhamento de recursos, que são usados em diferentes etapas do processo de produção. O compartilhamento de recursos em FMS pode dar condições para a ocorrência de *deadlocks*. Assim, todo recurso do FMS compartilhado é passível de ocorrência de *deadlock*, e esses podem ser: estações de trabalho, AGVs, rotas, cruzamentos, etc.

Roteamento: O problema de roteamento tem um papel fundamental na área de gerenciamento da distribuição de matérias em sistemas de manufatura, principalmente em arranjos físicos com rotas bidirecionais. Uma vez que a decisão do despacho é realizada, a missão do roteamento é atribuir uma rota adequada, para cada AGV desde sua origem até o destino baseado na atual situação.

2.2.1 Despacho e Programação

Existem pesquisas sobre o controle de fábrica partindo do viés da programação reativa da produção e outras partindo do viés de despacho (NGUYEN; KIM, 2009); (YANG; YAN, 2009); (KALINOVIC et al., 2011); (EL-BOURI, 2012). Na literatura existem várias visões sobre programação e despacho (QIU et al., 2002); (BRISKORN et al., 2006) (LE-ANH; DE KOSTER, 2006); (VIS, 2006); (TAY; HO, 2008); (SARIN; VARADARAJAN; WANG, 2011); (KIM, Y.-H.; YARLAGADDA, 2013); (CHANG; DONG; YANG, 2013).

No laboratório Tear (BENINCASA; MORANDIN; KATO, 2003); (MORANDIN, O. et al., 2007); (MORANDIN, O. et al., 2008); (KATO; MORANDIN; FONSECA, 2009); (MORANDIN, O. et al., 2009); (MORANDIN, O.; KATO; TUMA, 2010); (MORANDIN, O. J. et al., 2011b); (TUMA; MORANDIN; CARIDÁ, 2013); (CARIDÁ et al., 2014); (CARIDÁ; MORANDIN; TUMA, 2015) e neste trabalho de pesquisa adotamos a seguinte visão:

A programação da produção (PP) e a programação reativa da produção (PRP) são problemas de otimização combinatória que lidam com a alocação de recursos compartilhados. O despacho de AGVs é um processo de tomada de decisão com a finalidade de otimizar um ou mais objetivos. A principal diferença entre eles está relacionada à visão do sistema produtivo, seja ela em tempo ou em recursos.

A programação está ligada à atividade de planejamento da produção. A programação decide quando, onde e como deve agir um veículo para realizar as tarefas, ou seja, tem-se uma visão de toda a janela de tempo, e de todos os recursos da fábrica. A programação considera

um tempo maior que o controle, que pode variar em dias e até meses a depender das aplicações. Em uma média das empresas que o laboratório Tear tem contato, essa janela de tempo compreende a programação de uma semana de produção.

Exemplificando, admitindo-se que estamos no tempo 0, com a programação sabe-se que o AGV 1 estará na estação de trabalho 5 carregando o produto 2 no tempo 240 e indo para estação de trabalho 8. A vantagem da programação é que se tem uma perspectiva dos recursos a serem usados desde o primeiro produto que entra em fabricação, até a fabricação do último produto. A desvantagem está em relação ao tempo que se leva para gerar a programação e a falta de flexibilidade, pois com qualquer evento inesperado na produção, a programação fica inutilizável.

A programação reativa e o despacho estão ligados à atividade de controle da produção.

A programação reativa segue na mesma linha da programação, porém para avançar nas desvantagens da programação, a programação reativa refaz os cálculos toda vez que acontece um evento inesperado na produção, fazendo com que sempre se tenha uma programação válida. O agravante nesse caso diz respeito ao tempo para obtenção da resposta, pois todo o sistema produtivo depende da nova resposta para seguir. Devido à essa exigência, muitos trabalhos acabam não considerando alguns dos recursos da fábrica para tornar os cálculos mais rápidos.

O despacho decide onde e como o AGV deve agir, ou seja, o despacho decide qual tarefa o AGV deve executar, ou qual AGV deve atender uma determinada tarefa avaliando características momentâneas da fábrica e analisando somente os recursos ligados àquela tarefa, dessa forma não tendo uma visão de todo o sistema. Naso e Turchiano (2005) apresentam duas razões para que as regras de despacho sejam amplamente adotadas pelas indústrias. A primeira delas, é que as operações automatizadas da manufatura envolvem problemas combinatórios bem complexos, que não podem ser resolvidos por abordagens analíticas, exceto em casos bem simples. A segunda é justificada pelas condições das operações das fábricas, que são extremamente dinâmicas e turbulentas, e a taxa de eventos não programados torna o planejamento ineficiente.

Segundo Chen e Matis (2013), uma desvantagem do despacho é de não enxergar o todo, dessa forma não produz uma solução pensando no ótimo global. Porém, os mesmos citam que, apesar dessa característica negativa, normalmente o despacho encontra soluções perto do ótimo

e tem um cálculo muito rápido, o que o torna ideal para programações sinuosas em sistemas de grande escala.

Segundo os autores Sarin, Varadarajan e Wang (2011) e Chang, Dong e Yang (2013) entre as abordagens descritas na literatura para resolver o problema de alocação de recursos, regras de despacho, de longe, é a ferramenta mais usada para o controle de chão de fábrica. A razão por trás disso é que elas são simples de implementar, rápidas em reagir às mudanças encontradas no chão de fábrica, de fácil compreensão e exigem uma baixa carga computacional.

Como o foco deste trabalho e do laboratório em que o mesmo está inserido é o controle do chão de fábrica e devido às características vistas até aqui, a opção deste trabalho é no viés de despacho.

2.3 Trabalhos científicos correlatos à contribuição da tese

A proposta deste trabalho é pautada em duas hipóteses, que são:

- Hipótese 1: ao se incluir recursos da fábrica como, o uso das rotas do chão de fábrica e o tempo de *setup* de máquina, além das variáveis habituais por meio de um sistema *fuzzy* em cascata alcançar-se-á um valor de *makespan* ou *tardiness* mais correto e que na média seja menor se comparado com propostas que não consideram os recursos em questão.
- Hipótese 2: ao se usar informações de estados futuros do sistema de produção fabril por meio de uma estrutura baseada em espaço de estados alcançar-se-á um valor de *makespan* ou *tardiness* menor se comparado com propostas que não avaliam estados futuros.
- Hipótese 3: ao se incluir outros recursos da fábrica por meio de um sistema *fuzzy* em cascata e usar informações de estados futuros do sistema de produção fabril por meio de uma estrutura baseada em espaço de estados, alcançar-se-á um valor de *makespan* ou *tardiness* mais correto e menor se comparado com os valores obtidos por meio das hipóteses separadas.

A primeira hipótese levanta a questão de que um método que desconsidera alguns recursos do chão de fábrica pode encontrar um menor valor de *makespan* na simulação e ao ser executado na fábrica não garante o menor valor do objetivo de otimização.

Assim, para hipótese 1, buscam-se informações em trabalhos da literatura que abordam o problema de despacho de AGVs, verificando quais atributos e, quais recursos de fábrica são levados em consideração para a tomada de decisão e quais os objetivos de otimização alvo.

Despacho é uma questão importante para gestão e controle de AGV (KIM, Y.-H.; YARLAGADDA, 2013); (CHOE; PARK; RYU, 2010) (LE-ANH; DE KOSTER, 2006); (VIS, 2006); (QIU et al., 2002). Os primeiros trabalhos relacionados à regra de despacho de veículos foram motivados pela criação de regras simples de despacho que avaliavam somente um atributo do sistema.

Em seu trabalho, Egbelu e Tanchoco (1984) apresentaram regras de despacho divididas em dois grupos: estação de trabalho escolhe um veículo para o trabalho ou veículo nomeia qual estação de trabalho será atendida. Uma das regras proposta pelos autores foi “menor espaço restante no *buffer* de saída” (*Minimum Remaining Outgoing Queue Space rule – MROQS*). A regra MROQS é baseada no atributo do espaço restante do *buffer* de saída de cada estação de trabalho. O objetivo desta regra é reduzir a possibilidade de estações de trabalho serem bloqueadas. Outra regra proposta pelos autores foi “tempo de chegada da unidade de carregamento no chão de fábrica” (*Unit Load Shop Arrival Time rule – ULSAT*). Essa regra é baseada somente no atributo de tempo de chegada e o objetivo é reduzir o tempo de *makespan*.

Bartholdi III e Platzman (1989) apresentaram a regra “Encontrada Primeiro Atendida Primeiro” (*First Encountered First Served – FEFS*) tendo como objetivo entregar cargas tão rápido quanto possível para vários AGVs operando em um caminho simples de *loop* fechado. Na regra FEFS, os AGVs de carga simples ou múltipla carga circulam continuamente em *loop*. Sempre que o AGV está disponível, ele carrega a primeira carga que encontra. As cargas carregadas são entregues sempre que seus destinos são encontrados. O objetivo da regra é diminuir o tempo de *makespan*.

Outras pesquisas também surgiram na mesma linha como Egbelu (1987); Han e McGinnis (1989); Taghaboni (1989), ou seja, considerando apenas um atributo do sistema para otimização de um objetivo.

Com o passar do tempo alguns pesquisadores apostaram na hipótese que ponderando mais de um atributo do sistema alcançar-se-ia melhores resultados no objetivo de otimização abordado.

Klein e Kim (1996) apresentaram um estudo comparativo entre regras de despacho de simples e múltiplos atributos. Os atributos considerados foram: menor distância, *buffer* de saída da estação de trabalho e tempo de espera da solicitação. Usaram-se três objetivos para medida de desempenho: *makespan*, tempo total de viagem do veículo e tempo de espera da solicitação. Foram simuladas em um arranjo físico as regras com os atributos isolados e posteriormente combinando esses atributos, em que uma regra que pondera os pesos dos três atributos em conjunto usando A*. Na média dos testes, a regra multiatributo obteve melhores resultados nos três objetivos de otimização.

Kim et al. (1999) propuseram uma regra de despacho de AGV, tendo como finalidade principal o balanceamento de cargas entre as estações de trabalho e otimização do *throughput*. Propuseram para isso uma equação composta de vários atributos, envolvendo o estado corrente do “chão de fábrica”. Os atributos usados na equação foram os seguintes:

- Quantidade máxima de trabalhos suportada pela estação de trabalho, incluindo o trabalho em processamento;
- Número de trabalhos existentes em cada estação de trabalho;
- Tempo restante até que o *buffer* de entrada fique vazio;
- Tempo restante até que o *buffer* de saída fique cheio;
- Tempo de viagem do AGV até a estação que está esperando por transporte;
- Tempo de processamento de máquina por trabalho;
- Número de AGVs envolvidos no sistema.

Tan e Tang (2000) propuseram uma regra de despacho composta por múltiplos atributos, usando o método de inferência *fuzzy* denominado Takagi e Sugeno e também um Algoritmo Genético para a seleção de pesos entre notas obtidas. O principal propósito do trabalho foi de associar as variáveis *fuzzy* do sistema e reduzir ociosidade e bloqueios nas estações de trabalho. Os atributos envolvidos na regra foram: tempo restante para que o *buffer* da entrada da estação de trabalho de destino fique vazio; menor distância de viagem do veículo para a estação de trabalho de origem para a estação de trabalho de destino; menor tempo antes que o *buffer* de saída da estação de trabalho de origem fique cheio; número de peças já completas pela estação de trabalho.

Jeong e Randhawa (2001) propuseram uma regra de despacho de AGV do tipo “veículo iniciado” composta por múltiplos atributos, conferindo pesos para cada atributo por meio de uma Rede Neural Artificial. Os atributos considerados foram os seguintes: espaço restante no *buffer* de saída da estação de trabalho; distância entre o veículo disponível e a estação de trabalho esperando pelo serviço de transporte e espaço restante no *buffer* de entrada da estação de trabalho. A validação da regra multiatributo proposta se deu por meio de comparações de regras de atributo único avaliando os seguintes objetivos: tempo de viagem do veículo vazio, tempo total de bloqueio das estações de trabalho e *work-in-process*.

Benincasa et al. (2003) apresentaram um modelo para a definição de uma regra para despacho de AGV baseado em sistemas *fuzzy*. Foram considerados três atributos de entrada: distância entre o AGV e a estação de trabalho, número de cruzamentos entre o AGV e a estação de trabalho e espaço restante no *buffer* de saída da estação de trabalho requisitante. Cada atributo foi dividido em 3 conjuntos triangulares *fuzzy*. Foi criada uma base de regras manualmente com todas as combinações possíveis, ou seja, três atributos com três conjuntos por atributo são iguais a 27 regras. Como validação, o modelo foi aplicado em um FMS com seis estações de trabalho, variando entre 1 e 2 AGVs e comparado com a regra FIFO (primeiro a solicitar, primeiro a ser atendido). O trabalho proposto apresentou menores *makespans* em todas as simulações.

Umashankar e Karthik (2006) propuseram uma abordagem inteligente de despacho de AGVs com base em múltiplos atributos do controlador de lógica *fuzzy*, que, simultaneamente leva em conta vários aspectos em cada decisão de despacho. O controlador opera em duas fases, sendo a primeira fase para determinar qual AGV será escolhido ponderando: o uso do AGV, a distância do AGV ao centro de trabalho e o *buffer* de saída. Caso haja empate, para resolver o conflito, a segunda fase pondera o índice de agendamento e a prioridade do processo. A validação foi por uma abordagem empírica. A abordagem consiste em testar um conjunto representativo das condições de entrada e verificar a exatidão do sistema com base em decisões de um especialista. De acordo com os autores, o sistema obteve 100% de acertos.

Morandin et al. (2007) abordaram as mesmas características do trabalho de Benincasa et al. (2003), porém, ao invés de usar todas as possíveis regras geradas manualmente, a partir do conjunto de dados numéricos que representa amostras ou exemplos do problema e com funções de decisões previamente definidas, o método proposto usa um algoritmo genético (AG) para gerar uma base de regras *fuzzy* apropriada. O arranjo físico usado na validação tinha apenas

6 máquinas e 5 produtos. A validação se deu na comparação com o trabalho citado anteriormente, apresentando melhores resultados de *makespan*.

A proposta de Sing, Sarngadharan e Pal (2011) usa veículos de múltiplas cargas em uma indústria de automóveis. Os autores propõem cinco regras de despacho, sendo uma de único destino, em que o veículo carrega toda sua capacidade com peças a serem entregues em uma única estação de trabalho, e outras quatro com múltiplos destinos com variações. A primeira elenca 3 destinos pela prioridade, a segunda elenca 3 destinos, sendo o primeiro o destino com maior prioridade e os outros 2 os destinos mais próximos do primeiro. A terceira segue o raciocínio da segunda, porém com uma restrição que o segundo e terceiro destinos são os mais pertos do primeiro, mas que estão entre os 6 de maior prioridade geral. E a quarta regra de despacho elenca os seis pontos de entrega com maior prioridade e faz uma média de distância de viagem para ponderar quem será atendido.

As cinco regras foram testadas em um ambiente simulado, conforme apresentado na Figura 1. Foram realizados testes com um e com dois AGVs no ambiente e o desempenho medido foi por meio do *throughput* e tempo de espera do AGV. A regra número 5 foi a que obteve melhor resultado. Os autores também citam que o sistema evita *deadlocks*, porém esse conceito foi resolvido nas simulações com dois AGV dividindo o ambiente em dois, conforme ilustrado na Figura 2, em que cada AGV só opera em uma área específica que não ocasiona *deadlocks*.

Um ponto negativo é que as regras propostas foram comparadas somente entre elas mesmas e apontada a melhor, porém não foram comparadas com abordagens de outros autores.

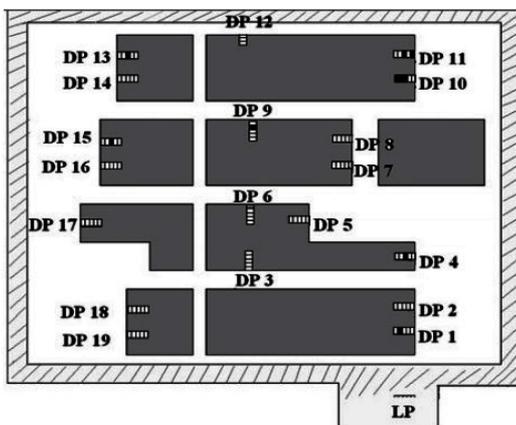


Figura 1 – Arranjo físico usado para validação do trabalho de Sing, Sarngadharan e Pal (2011).

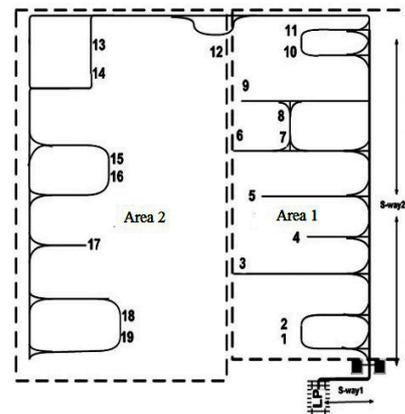


Figura 2 – Divisão do arranjo físico para evitar *Deadlock* do trabalho de Sing, Sarngadharan e Pal (2011).

Morandin et al. (2011a) apresentaram um sistema *fuzzy* hierárquico para realizar despacho de veículos considerando múltiplos atributos. Foram usados os mesmos atributos e a base de regras do trabalho de Benincasa et al. (2003). Usando o método de Wang Mendel para ponderar os atributos e regras criou-se uma hierarquia de decisão *fuzzy*. A proposta foi comparada com a regra FIFO e com o trabalho de Benincasa et al. (2003), no qual a proposta alcançou melhores valores de *makespan* e também um tempo de resposta menor devido à hierarquia.

Um ponto negativo da proposta é o arranjo físico usado para validação, pois o mesmo tem apenas 6 máquinas e 5 produtos. Porém a proposta gerou resultados preliminares que ajudaram a formular as hipóteses deste trabalho.

A pesquisa de Ho, Liu e Yih (2012) destaca que existem dois problemas ligados ao despacho de AGVs de múltiplas cargas. O primeiro é chamado de *pickup-dispatching* (despacho para coleta) e o segundo *load-selection* (seleção de carga). Basicamente, na definição dos autores, o *pickup-dispatching* ocorre quando um AGV tem uma tarefa de entrega ou coleta de produtos e o mesmo deve decidir que ponto atender. O problema de *load-selection* se dá quando existe uma tarefa de coleta de produtos e chegando ao ponto de coleta escolhido e é necessário decidir qual/quais produtos serão carregados. Dessa forma, a proposta do artigo é abordar os dois problemas em conjunto. Para isso foi proposto um método no qual:

- Quando é uma tarefa de entrega, a regra usada é escolher a mais próxima (SD-*Shortest Distance*).
- Quando a tarefa é uma coleta, são executadas quatro fases: preparação, *clustering*, avaliação e execução. Na fase de preparação, são calculados os pesos para três atributos (inatividade da estação de trabalho, tempo de espera da tarefa e distância). Esses pesos são usados na segunda e terceira etapas. Na etapa de *clustering*, as peças que necessitam de transporte são agrupadas em grupos de peças com base na semelhança dos três atributos. Na fase de avaliação, grupos de peça são avaliados considerando esses três atributos. O grupo com o maior valor de avaliação será atendido pelo AGV.

Foram realizadas simulações em um sistema de manufatura hipotético, apresentado na Figura 3, e a proposta foi comparada com simulações usando somente um atributo e com outra abordagem da literatura, sendo melhor em ambos os testes no que diz respeito aos objetivos de otimização de *flow time*, e *tardiness*.

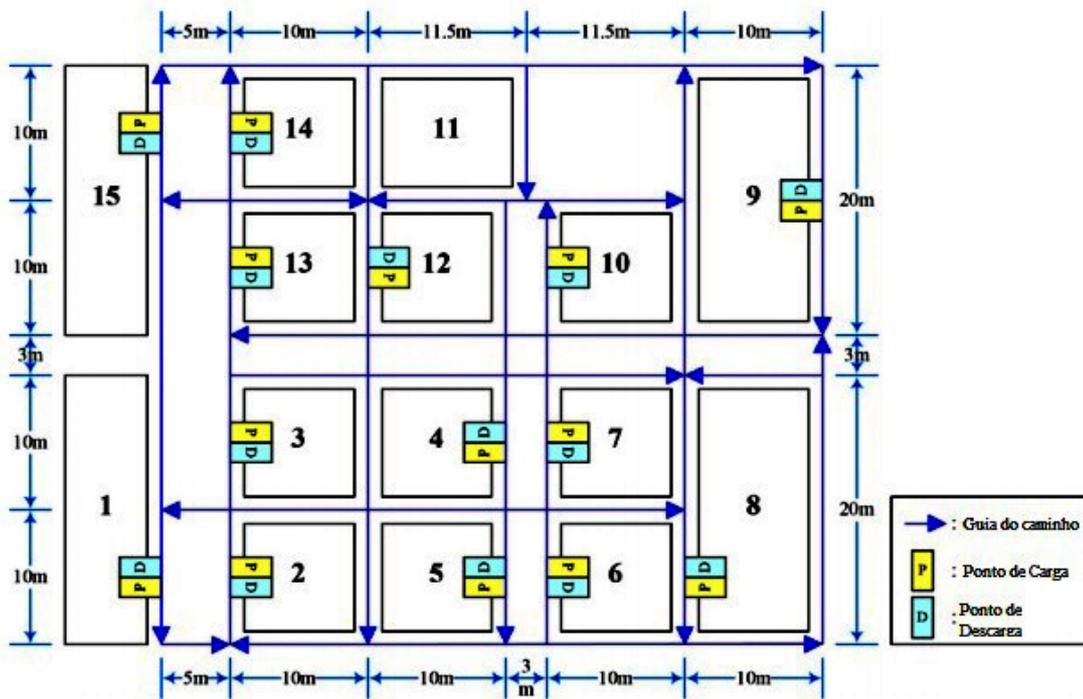


Figura 3 – Arranjo físico usado para validação do trabalho de Ho, Liu e Yih (2012).

Apesar dos autores definirem uma forma diferente de atribuição de tarefas para cada caso, *pickup-dispatching* e *load-selection*, os dois estão contidos dentro do problema de despacho, assim não se tem transporte se tratado somente um dos casos.

Os autores, e outros trabalhos citados anteriormente, não deixam claro os parâmetros do arranjo, como, por exemplo, roteiros dos produtos, tempos de processamento, etc., tornando mais difícil que outros comparem suas abordagens nas mesmas condições de ambientes.

Outro ponto que pode ser verificado na maioria dos trabalhos apresentados é que, apesar de testarem a abordagem para mais de um objetivo de otimização, as regras de despacho não foram criadas para tratá-los especificamente, e sim para análise, sem considerar atributos que de fato tem impacto em cada objetivo.

Kim e Yarlalagadda (2013) propuseram uma regra de despacho multiatributo baseada em uma função matemática que pondera dois parâmetros: tempo de espera da carga e a distância de viagem a partir da posição atual do veículo com a carga correspondente. O veículo vazio verifica se existem cargas para serem transportadas e escolhe a que tem menor valor de saída da função de decisão. No caso de não existirem cargas, o AGV é enviado para o estacionamento. A regra proposta foi aplicada no arranjo físico apresentado na Figura 4 e comparado com duas

regras de simples atributos: primeiro a distância mais curta de viagem (STDF - *Shortest Travel Distance First*) e primeiro o veículo mais próximo (NVS - *Nearest Vehicle First*).

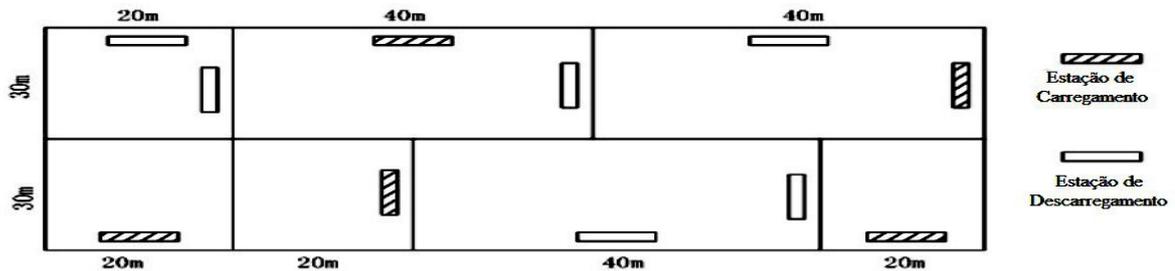


Figura 4 – Arranjo físico usado para validação do trabalho de Kim e Yarlagadda (2013).

Cada regra de despacho foi testada com 4 e com 6 AGVs no arranjo físico proposto e foram avaliados os tempos médios de espera de carga e os tempos máximos de espera de carga. A regra multiatributo proposta pelos autores apresentou melhores resultados em ambos os objetivos.

Um ponto negativo da abordagem é que, enviando o AGV para o estacionamento quando não se tem demanda, pode-se perder muito tempo. Enquanto o AGV está indo para o estacionamento podem surgir novas demandas, e talvez avaliar informações futuras e direcionar o AGV para elas de forma antecipada possa trazer ganhos para a abordagem.

Outro ponto recorrente na maioria dos trabalhos apresentados até aqui é que eles provam que abordagens multiatributos são melhores do que regras simples, porém não provam se a forma que se pondera múltiplos atributos é melhor pois não fazem comparações com abordagens correlatas.

Os autores Chen e Matis (2013) propuseram a regra de despacho WBMR (*Weight Biased Modified RRrule*). Essa regra é uma variação da regra RRrule proposta em 1993 por Raghu e Rajendran. Raghu e Rajendran fizeram uma combinação linear de duas regras simples SPRT (*slack per remaining process time*) e SPT (*shortest process time*) adicionando ainda uma estimativa de tempo de espera do produto na próxima máquina (W_{nxt}) para minimização do *tardiness*.

Porém, segundo os autores Chen e Matis (2013), a regra RRrule tem dois problemas principais que as tornam impraticável para uso. O primeiro é na estimativa do W_{nxt} , em que a mesma é feita examinando apenas o próximo produto da fila na lista de tarefas fazendo com

que não se tenha precisão. O segundo é que, na prática, nem todo produto tem a mesma importância relativa e pode ser vantajoso mover alguns produtos de peso superior no cronograma.

Dessa forma a regra WBMR resolve estas deficiências, eliminando o W_{nxt} e incluindo na equação o ω_i (peso do produto) e o β (peso do parâmetro de polarização) permitindo a polarização do cronograma para enfrentar as prioridades de produção e cumprir os prazos por meio do ajuste de um único parâmetro, no qual tal efeito é quantificado por meio da avaliação de atraso em diferentes limiares de truncamento.

No teste numérico apresentado pelos autores, a regra WBMR foi comparada com regras simples (FIFO, SPRT e SPT) e com a regra RRrule e demonstrou o desempenho superior para minimizar o *tardiness*.

É citado várias vezes pelos autores, até mesmo como justificativa para o trabalho, que o maior problema da regra RRrule é a inviabilidade para aplicações práticas. Logo os autores deveriam ter feito os testes no mínimo em um cenário simulado espelhado em um ambiente real, pois o teste numérico apresentado não comprova essa eficiência prática.

Azadeh et al. (2013) propuseram uma abordagem integrada baseada em simulação de eventos discretos (DES) e redes neurais artificiais (RNAs) visando priorizar as regras de despacho e minimizar o *makespan*. A proposta usa como base quatro regras de despacho descritas na literatura: FIFO (*First-In, First-Out*), LIFO (*Last-In, First-Out*), LVF (*Low-Value First*) e HVF (*High-Value First*) no tempo de processamento das tarefas.

Foram simulados no *software* Visual SLAM três cenários de *job shop* sendo eles com 6x4 (seis produtos e quatro máquinas), 5x5 e 20x5. Os autores geraram N sequências aleatórias de prioridade das regras de despacho para cada máquina (PDR). Posteriormente foram executados os cenários para todas as alternativas geradas e obtidos N valores de *makespan*. Com os valores obtidos foi treinada uma BPNN (*back-propagation neural network*) usando o algoritmo de *feed-forward* (alimentação direta) para aprender e testar o BPNN de acordo com o valor do percentual de erro médio absoluto. Esse ciclo foi simulado até que se encontrou a prioridade ideal das regras para cada máquina.

A proposta foi comparada com as regras FIFO, LIFO, LVF e HVF isoladamente, e apresentou menores valores de *makespan* em todos os testes. No entanto, segundo os próprios autores, dependendo do tamanho do problema, isto é, número de máquinas e produtos, fazer

uso da proposta para encontrar soluções ótimas ou perto do ideal fica mais difícil ou até mesmo impossível.

Shahzad e Mebarki (2016), com o objetivo de diminuir o atraso máximo, propuseram uma abordagem baseada em aprendizado de máquina e heurística de busca. A proposta é usar árvores de decisão para gerar regras de despacho que ponderam múltiplos atributos por meio da mineração de dados de uma base existente que armazena todos os resultados de simulações anteriores. Posteriormente os autores usam busca Tabu para refinar a base de regras inicial. As regras foram testadas em um ambiente de fábrica 6x6 simulado e diminuíram o *tardiness* em relação abordagens anteriores.

Analisando esses trabalhos é possível chegar à conclusão que a hipótese de que considerando múltiplos atributos alcançam-se melhores resultados no objetivo de otimização buscado é favorável. Assim como afirmado pelos autores, o desempenho das regras de despacho em tempo real é totalmente dependente do contexto, e no geral, uma regra simples de despacho não pode garantir um desempenho satisfatório para todas as possíveis condição de operações da fábrica (NASO; TURCHIANO, 2005). Para aumentar o desempenho, tendo em vista o ambiente dinâmico e os vários objetivos a serem alcançados, são usadas as regras de despacho multiatributos (GUAN; DAI, 2009).

Porém, ponderar múltiplos atributos em sistema de produção significa lidar com fatores como ambiguidade, incerteza e informações vagas. Em problemas de difícil solução, em que se faz necessário o auxílio matemático/computacional, modelar tais fatores se torna uma tarefa difícil. Entretanto, sistemas *fuzzy* são apontados na literatura por ser uma técnica eficaz para modelar informações imprecisas e por meio da base de regras *fuzzy* atuar como mecanismo formal para representação do conhecimento (LUGER, 2004); (MUKAIDONO, 2001); (KLIR; YUAN, 1995).

Assim apoia-se a primeira hipótese desse estudo, que usando regras multiatributos se obtém melhores valores do objetivo de otimização. Em outras palavras, ao desconsiderar elementos de fábrica que são recursos compartilhados pode-se chegar a decisões menos assertivas em relação à verdadeira disponibilidade desses recursos, portanto, chegar a decisões de atribuição de tarefas que são aplicadas na prática não encontram boas soluções no que diz respeito ao objetivo de otimização.

Também se sabe que existem muitos problemas que afetam o sistema de transporte, como colocado na seção anterior, e também muitas variáveis que afetam o sistema de produção como um todo. Sabe-se ainda que, a cada nova variável considerada, o problema cresce exponencialmente, assim considerar todos os elementos tornaria o problema intratável.

Dessa forma a primeira hipótese é incluir mais recursos da fábrica como, o uso das rotas do chão de fábrica e o tempo de *setup* de máquina, por meio de um sistema *fuzzy* em cascata. Espera-se alcançar valores dos objetivos de otimização, *makespan* e *tardiness*, menores quando comparados com outras abordagens.

Para fundamentar a segunda hipótese, de que o uso de informações de estados futuros do sistema de produção fabril por meio de uma estrutura baseada em espaço de estados permita alcançar um valor de *makespan* ou *tardiness* menor se comparado com propostas que não avaliam estados futuros, buscaram-se trabalhos que fizeram alguma tentativa de incorporar no despacho de AGVs informações futuras da fábrica, verificando como foram suas abordagens e quais os resultados alcançados.

Algumas pesquisas apontam que um ponto negativo das regras de despacho é a visão momentânea para tomada de decisão. As regras de despacho, mesmo as multiatributos, são consideradas “míopes” pelo fato de considerar apenas o atual momento para a tomada de decisão (SEIFERT; MORITO, 2001); (NASO; TURCHIANO, 2005). Avaliando apenas o presente, nem sempre é possível obter a melhor decisão de despacho, sendo que, caso fosse considerado o estado futuro da fábrica, seria possível tomar uma decisão mais assertiva (HIDEHIKO, 2008); (SMOLIC-ROCAK et al., 2010).

Naso e Turchiano (2005) adotaram uma estratégia de decisão multiatributo para considerar múltiplos aspectos em cada decisão de despacho. Se existir solicitação crítica, como *buffer* de saída cheio, causando um travamento na produção, essa solicitação tem prioridade. Para outras solicitações é usado *fuzzy* hierárquico para tomadas de decisões em conjunto com algoritmos genéticos para ponderar as variáveis de maior impacto e gerar a hierarquia. Segundo os autores, regras de despacho se baseiam no curto prazo, informações locais, e negligenciam expressamente qualquer consequência futura da decisão real. Então, foi incorporada no trabalho uma variável que os autores chamam de encadeamento de tarefas, na qual, são verificados quais destinos levam a outros serviços, formando uma cadeia de tarefas. O estudo é aplicado a dois arranjos físicos simulados (sendo um com quatro estações de trabalho e outro com nove estações de trabalho), e comparado com regras simples descritas na literatura no que diz respeito

a *flow time*, *makespan* e *lateness*. Os autores foram os primeiros a incluir informações futuras da fábrica na decisão de despacho, porém a informação futura considerada só colabora para o objetivo *flow-time* devido às suas características. Mesmo com a variável de encadeamento de tarefas não tendo correlação direta com os objetivos de *makespan* e *lateness*, ao ser comparada com regras simples (como: STD_OUT (*shortest travel distance to source workcenter*)) apresentou melhores resultados devido ao uso de múltiplos atributos.

Hidehiko (2008) trabalha com o despacho de veículos em sistemas de manufatura autônomos, descentralizados e flexíveis (AD-FMS – *Autonomous Decentralized Flexible Manufacturing Systems*) por meio de simulação de hipóteses. Para definir a próxima tarefa do AGV, foi definido um algoritmo de raciocínio para a antecipação do futuro (RAF). Esse algoritmo prevê as possíveis próximas ações para o AGV e por meio de uma simulação das hipóteses e verificação de inconsistências, em que se verifica a prioridade de cada uma delas definindo a próxima ação a ser tomada. Para aumentar a eficiência dessa verificação de hipóteses, foi proposta uma avaliação baseada em casos passados, consultando o histórico de fabricação, possibilitando uma escolha baseada na experiência. Para a validação da proposta, foram realizadas simulações em três diferentes arranjos físicos de fábrica e comparado o resultado com uma abordagem simples de multiagentes verificando o desempenho em relação ao número de seleção de hipóteses e volume de produção (*throughput*).

Apesar de que, o foco principal da proposta foi a abordagem multiagentes, sendo o AGV um agente, a previsão de tarefas por meio da memória do agente e no histórico de fabricação, garantiu-se um melhor desempenho no volume de produção.

Os autores Lu e Hsieh (2011) afirmam que o uso de regras de despacho é um dos vários métodos para determinar as sequências de trabalho e, geralmente, são aplicadas nas células robóticas de manufatura. No entanto, devido às incertezas no ambiente de produção, tais como as peças defeituosas e as mudanças de demanda, a política de despacho deve ser considerada para garantir operações eficientes. Abordando essas questões com o objetivo de maximizar a taxa de produção ou o de minimizar o tempo de ciclo, os autores propuseram uma abordagem preditiva para realizar o despacho em uma célula de manufatura.

Ainda segundo os autores Lu e Hsieh, em geral, uma vez que as tarefas de carregamento são geradas, essas podem ser executadas imediatamente. No entanto, para as tarefas de descarregamento, o robô precisa esperar no local atual até que a máquina termine o processamento. O tempo gasto em espera para a descarga das peças (robô em repouso) aumenta

o tempo de ciclo celular. Assim, o algoritmo preditivo proposto pelos autores usa o conhecimento do processo para prever o tempo de processamento restante para cada máquina e determinar a sequência de descarga apropriada para o robô e dessa forma reduzir o tempo ocioso.

Após os testes, foi concluído que todos os fatores, como regra de despacho, capacidade dos *buffers*, demanda de produção, entre outros, têm efeito significativo sobre as medidas de desempenho. E ao comparar a abordagem preditiva com as regras FCFS e SD a preditiva mostrou ganho superior a 30% em relação a FCFS e a 15% em relação a SD, fornecendo flexibilidade para se adaptar as variações do sistema de produção e incertezas.

Morandin et al. (2011b) apresentaram um sistema *fuzzy*-genético para realizar despacho de veículos considerando múltiplos atributos e, sendo um deles em destaque, referente à previsão de tarefas o que permite ao sistema realizar uma predição simples e o encadeamento de atividades. Os outros atributos considerados no trabalho foram divididos em duas categorias, sendo elas para o atendimento do *makespan* e para o atendimento do *tardiness*. Para o *makespan* foram consideradas: distância, número de cruzamentos, tamanho do *buffer* de entrada e de saída das estações de trabalho. Para o *tardiness* foram consideradas: distância, número de cruzamentos, encadeamento de tarefas e a data devida dos produtos. Para a validação, foi comparado em termos de *makespan* com o trabalho desenvolvido por Benicasa, Morandin e Kato (2003) e com o trabalho apresentado por Morandin et al. (2011a) demonstrando um melhor resultado e o objetivo de desempenho *tardiness* foi avaliado realizando um tratamento estatístico o que mostrou pouco atraso das entregas na abordagem de produção por demanda.

Dessa forma, conseguiram-se resultados preliminares que ajudam a sustentar a segunda hipótese deste trabalho em que, incluindo predição, se alcançam melhores resultados no objetivo de otimização. E que é possível tratar mais de um objetivo de otimização de forma individualizada no mesmo sistema de decisão.

Segundo Dang (2013) problemas multiobjetivos são geralmente difíceis de resolver em uma quantidade limitada de tempo devido ao grande número de atividades e recursos. Restrições adicionais, como calendário de recursos e rotas alternativas também fazem a resolução desse tipo de problema mais complexos. Quando ocorre algum evento inesperado, por exemplo, algum recurso está em manutenção, alguma atividade leva mais tempo para a execução do que o planejado, ou uma nova ordem de produção chega, o processo de decisão precisa ser tão rápido quanto possível, de modo a reagir às mudanças. Usando regras de

despacho é capaz de encontrar soluções para uma quantidade relativamente aceitável de tempo, porém nem sempre com qualidade ideal.

Para resolver essa deficiência, Dang (2013) propôs um AG para fazer a programação da produção e assim ter algumas estimativas, como menor *makespan* encontrado, *makespan* médio, tempo previsto para realização da tarefa X . A partir daí eles propuseram novas regras de despacho que incluem em seu cálculo algumas dessas estimativas, assim fazendo uma forma de predição para ajudar na tomada de decisão, como por exemplo as regras:

- Primeiro a ser ponderada pela data de vencimento prevista (WEPD - *Weighted Earliest Predicted Due Date*), selecionando a tarefa com menor valor na expressão: $(\text{data de vencimento} - \text{mínimo makespan}) * \text{prioridade da tarefa}$.
- Primeiro pela previsão do tempo de folga (EPST - *Earliest Predicted Slack Time*), selecionando a tarefa o menor valor na expressão: $(\text{data de vencimento}) - (\text{tempo corrente} + \text{duração da atividade} + ((\text{mínimo makespan} + \text{makespan médio}) / 2))$.

Em um ambiente, que foi construído na linguagem C++, com 90 recursos diferentes, 535 tarefas com diferentes datas de entrega, em que cada tarefa representa um componente a ser fabricado. Cada fluxo de trabalho contém pelo menos duas rotas alternativas e cada rota tem no mínimo quatro atividades. Cada atividade tem pelo menos dois recursos alternativos com desempenho diferente. Comparando as regras propostas com regras tradicionais da literatura, como FIFO, SPT, WSPT, para os objetivos de entrega no prazo e minimização do custo de uso de recursos, a abordagem alcançou melhores resultados.

Marvizadeh e Choobineh (2014) propuseram uma abordagem de despacho de AGV com olhar para frente, que considera um índice de contribuição de um potencial movimento de material ao fluxo interno da fábrica antes do veículo ser despachado com o objetivo de minimizar o WIP (*Work in Progress*). Os autores usam o princípio divergência Kullback-Leibler para medir o índice de contribuição antes do despacho acontecer e também por meio de uma simulação medem qual seria o índice de contribuição para o fluxo da fábrica depois da tarefa realizada. A tarefa com maior média entre os dois índices de contribuição para o fluxo da fábrica é atendida pelo veículo.

Miyamoto e Inoue (2016) apresentaram um método de despacho e roteamento de veículos com objetivo de evitar *deadlocks* em uma fábrica de semicondutores. Os autores

usaram uma estratégia de modelo de controle preditivo em que se define um pequeno horizonte de tempo de programação para realizar a otimização. Os autores apresentam uma função de custo com comutação multiobjetivo e consideram o controle preditivo acrescentando novas tarefas em tempo real. Os mesmos afirmam ainda que alguns dos problemas da programação da produção são a complexidade e custo computacional, e quando se usa um olhar a frente em um pequeno horizonte de tempo diminui-se a complexidade e custo computacional e o despacho se torna mais eficaz.

Choe, Kim e Ryu (2016) apresentaram um algoritmo de aprendizagem de preferência on-line chamado OnPL que pode se adaptar dinamicamente a política de despacho de AGVs às mudanças no ambiente em um terminal de containers automatizado. A adaptação da política é alcançada por meio da atualização da função de preferência aos pares depois de cada decisão de despacho. Para esta atualização, OnPL usa redes neurais, gera novos exemplos de treinamento e renova o banco de dados, substituindo alguns exemplos antigos por novos. A função de preferência é então reaprendida a partir dos exemplos do banco de dados renovado. Os exemplos de formação de preferências são gerados após cada decisão de despacho. Para isso, cada tarefa candidata é avaliada por meio de uma simulação de despacho continuada para os trabalhos futuros em um curto olhar para frente. Após a avaliação, o melhor trabalho está emparelhado com cada um dos trabalhos restantes para fazer exemplos positivos, e as inversões destes pares são gerados para fazer exemplos negativos.

Tomando como exemplo prático para entender melhor como a predição pode ajudar na decisão do despacho, um AGV que finalizou uma tarefa na estação de trabalho 1 verifica quais são as tarefas que aguardam transportes. Existem solicitações de transportes para as estações de trabalho 4 e 7. De acordo com as circunstâncias o AGV é despachado para a estação de trabalho 4. Porém após poucos segundos da saída do AGV a estação de trabalho 1 termina o processamento de um produto e que a próxima etapa de fabricação desse produto seria exatamente na estação de trabalho 4, ou ainda, esse produto poderia ter prioridade de produção. Esse exemplo é um caso particular, mas deixa claro que se fosse levado em consideração o tempo restante de processamento das estações de trabalho próximas ao AGV, poderia ter-se um ganho no objetivo de otimização.

Seguindo essa linha de raciocínio espera-se conseguir melhores resultados no objetivo de otimização incluindo na decisão de despacho informações futuras do sistema de produção. Como o despacho é feito em tempo de execução, avaliar todo o sistema de produção seria muito

custoso em tempo, assim a proposta é um módulo de predição que avalie algumas informações dos recursos relacionadas a solicitação em um futuro próximo.

Como também pode ser visto, muitos trabalhos apresentados até aqui testam seus métodos de despacho para mais de um objetivo de otimização, porém não existe um tratamento específico para cada objetivo ou uma possível alternância dos mesmos.

Sabe-se que em problemas de controle no dia a dia da fábrica é desejável a otimização de múltiplas funções objetivo, que são mutuamente concorrentes. Por exemplo entre *makespan* e *tardiness*. O *makespan* busca produzir mais com os mesmos recursos, sem se importar quem será produzido primeiro. O *tardiness*, por sua vez, busca produzir conforme a demanda e entregar os produtos no prazo. Assim mesmo garantindo um menor *makespan* não se garante prazo de entrega.

Dessa forma, pretende-se abordar mais de um objetivo de otimização e que eles possam se alternar de acordo com o desejo do usuário.

O trabalho de Gonzalez-R, Framinan e Ruiz-Usano (2010) centrou-se em analisar a influência das regras de despacho no desempenho de um sistema DBR (*drum-buffer-robe*). Baseado em um cenário real de fábrica de caixas de engrenagens em forma de linha (*flow-line*) com 5 máquinas e cada uma delas com *buffer* de entrada, três tipos de produtos e usando *kanban* para controle da produção. Existe uma máquina central identificada como gargalo e do total de peças produzidas 20% são do tipo 1, 30% do tipo 2 e 30% do tipo 3. As simulações foram feitas no *software* Arena 12.0 em que foram adicionados em alguns testes quebras de máquinas e em outros tempos de *setup* das máquinas. Os objetivos de otimização avaliados separadamente foram: *due date*, valor máximo de *tardiness* e *work-in-process*.

Foram avaliadas 10 regras de despacho simples descritas na literatura: SPT (*Shortest Processing Time*), SRPT (*Shortest Remaining Processing Time*), LPT (*Longest Processing Time*), EDD (*Earliest Due Date*), CR (*Critical Ratio*), SI (*shortest imminent truncated*), LS (*Least Slack*), NSUT (*No Set-Up Time*), FCFS (*First Come First Served*) e SRO (*Select in Random Order*).

Para os testes sem tempo de *setup* as regras que apresentaram melhores resultados foram: EDD e SI. Para os testes com tempo de *setup* as regras que apresentaram melhores resultados foram FCFS e SI. Fazendo uma análise geral dos resultados, os autores afirmam que para cada tipo de ambiente de produção uma regra pode ser melhor e que para múltiplos

objetivos, a regra que é melhor para um ou outro objetivo pode não ser a melhor na média. O que reforça ainda mais a primeira hipótese deste trabalho, uma vez que incluindo mais informações, por exemplo tempo de *setup*, alteram-se os resultados, e também reforça-se a possibilidade de tratar múltiplos objetivos de forma integrada.

Segundo Chen (2013), na prática, a maioria dos problemas de decisão envolve múltiplos objetivos. Muitos estudos teóricos consideram apenas um objetivo, que em diversas vezes leva ao irreal e a decisões mal balanceadas. Ainda segundo o autor, embora exista um consenso que o despacho é um dos problemas de tomada de decisão mais importantes em uma fábrica e que deve ser multiobjetivo, pesquisas nessa área são insuficientes, especialmente para sistemas de produção grandes e complexos.

Para abordar as questões levantadas, Chen (2013) propôs uma nova regra despacho que une duas regras simples de despacho, no qual cada uma tem um objetivo de otimização diferente, sendo eles: tempo de ciclo médio e o atraso máximo. A proposta usa FCM (*fuzzy c-means*) para classificar as tarefas. Depois de classificadas, é usado FBPN (*fuzzy back-propagation network*) para estimar o tempo de ciclo restante de cada tarefa. Para gerar a nova regra bi-objetivo, duas regras de despacho tradicionais, FSMCT (*Fluctuation Smoothing Rule for Mean Cycle Time*) e EDD (*Earliest Due Date*) são fundidas em uma forma não-linear.

Para avaliar a eficácia da proposta foi simulado, implementando em visual C++, uma fábrica de *wafers* com produção de 25.000 unidades/mês com mais de dez tipos de produtos e mais de 500 estações de trabalho para a realização de operações de lotes de 58 até 110 *wafers*. A proposta foi comparada com algumas regras existentes na literatura, como: FIFO, EDD, SRPT (*Shortest Remaining Processing Time*), CR (*Critical Ratio*), FSVCT (*Fluctuation Smoothing Rule for Cycle Time Variation*), FSMCT e NFS (*Nonlinear Fluctuation Smoothing Rule*). Em todos os testes, a regra proposta atingiu melhores resultados nos dois objetivos.

Apesar de usar um arranjo físico de grande porte, os autores não dão detalhes dos parâmetros usados, como roteiros dos produtos, roteiros alternativos, tempos de processamento, entre outros. A proposta também não foi comparada com abordagens multiatributos, ou multiobjetivos, isto é, a validação se deu apenas em comparações com regras simples. Outro ponto a ser ressaltado é que os objetivos abordados possuem características similares, não existindo alto grau de conflito entre eles.

Chang et al. (2013) propuseram uma abordagem baseada em AG (Algoritmo Genético) para resolver o problema de decisão de despacho multiobjetivo em tempo real. Os autores definiram o despacho como planejador *on-line* e a programação com planejador *off-line*. No estudo, quatro medidas de desempenho foram consideradas para avaliar a função de *fitness* do AG, incluindo nível *de work in process*, tempo de ciclo médio, atraso médio e prazo médio de produtos com urgência, sendo que, segundo os autores, essas medidas de desempenho são intimamente relacionadas com o rendimento, nível de satisfação dos clientes e capacidade para lidar com tarefas urgentes.

A ideia principal do trabalho foi integrar o despacho e a programação por meio da função de *fitness* do AG. Existe uma base de dados com várias regras de despacho simples descritas na literatura, como: FCFS, FEFS, STTF, etc. O AG cria a população inicial e em cima da simulação da fábrica vai evoluindo essa população e testando as regras de despacho que estão na base de dados. Quando o critério de parada é satisfeito, nesse caso os autores usaram o número de gerações, é armazenada a melhor programação e a regra de despacho usada para atingir esse melhor resultado. Dessa forma a produção segue conforme o planejamento e quando ocorrer situações não previstas, como quebra de máquinas, falta de matéria prima, pedidos urgentes, etc., ao invés de reprogramar, a regra de despacho selecionada é ativada para fazer o controle.

A proposta foi aplicada no arranjo físico mostrado na Figura 5 e os valores obtidos nas 4 variáveis de desempenho foram comparados com os métodos tradicionais de despacho de única regra, no qual obteve melhores resultados.

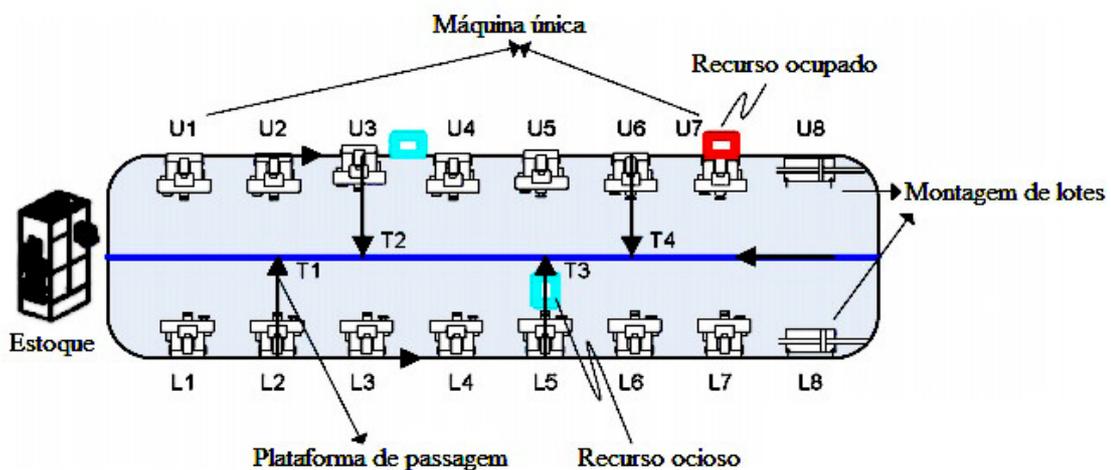


Figura 5 – Arranjo físico usado para validação do trabalho de Chang et al. (2013).

Os autores citaram que uma das vantagens de integrar o despacho com a programação é principalmente no início da produção, momento no qual se tem todo o sistema vazio, acarretando em empates quando se prioriza, por exemplo, a regra MROQS (menor espaço restante no *buffer* de saída). Dessa forma, fundamenta-se a configuração que pretendemos desenvolver a cascata *fuzzy* deste trabalho, em que o sistema terá atributos genéricos e atributos intrinsecamente ligados a cada objetivo de otimização, como por exemplo data de entrega do produto para o objetivo *tardiness*. Portanto, mesmo que os atributos genéricos estejam em situações de empates ou conflitos, o atributo usado será o que tem mais aderência para o objetivo de otimização em uso.

Na revisão bibliográfica encontram-se trabalhos de outros domínios de aplicação, porém abordando problemas cuja a natureza ou comportamento são similares aos aqui estudados. Logo técnicas ou ferramentas usadas por alguns autores servem de inspiração e/ou reforço de que essas têm usabilidade para este trabalho. A seguir são apresentados alguns trabalhos que reforçam ainda mais a eficácia de modelagens em redes de Petri e ponderação de múltiplos atributos com sistemas *fuzzy*.

Os autores Chiba, Arai e Ota (2010) propõem um processo para sistemas de AGV integrando dois problemas inerentes: rede de caminhos e roteamento. A proposta defende que ambos os problemas devem ser resolvidos ao mesmo tempo, porque um depende do outro. Dessa forma a proposta do trabalho foi de um sistema cooperativo de co-evolução, em que cada um dos problemas tem um AG para encontrar uma boa solução e para cada nova iteração os problemas interagem por meio do *feedback* do ambiente de simulação. Para validação é usado um arranjo físico de fábrica sem rotas, apresentado na Figura 6, em que a abordagem proposta é responsável pela criação dos caminhos bidirecionais.

A proposta foi comparada com uma geração de caminhos aleatórios e com uma abordagem usando algoritmo genético canônico. Obtendo melhores resultados criando caminhos com menos intersecções e mais eficientes.

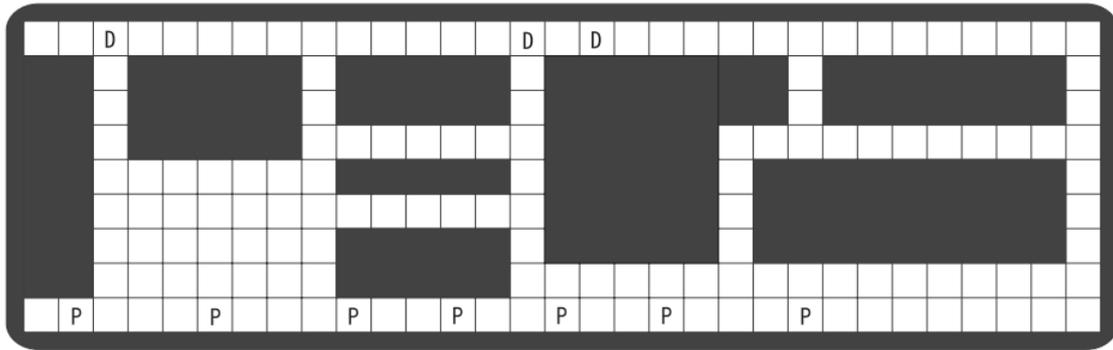


Figura 6 – Arranjo físico usado para validação do trabalho de Chiba, Arai e Ota (2010).

Na linha da terceira hipótese deste trabalho, porém em outro domínio de problema, a proposta de Wu, Lou e Tang (2011) foi o uso de um algoritmo genético multiobjetivo, que os autores chamaram de MOGA, para encontrar os melhores parâmetros de um sistema de controle PID (Proporcional Integral Derivativo) para modelo de condução do AGV na otimização do sistema de servo controle. Segundo os autores a contribuição está na criação de múltiplas populações em cada geração, sendo que essas populações passam por uma função ótima de Pareto, uma seleção elitista e uma função de *fitness* modificada. Dessa forma foi possível balancear no PID múltiplos objetivos, e não focar em somente um. A validação se deu por meio da comparação da proposta por uma abordagem de objetivo único, no qual a proposta apresentou melhores resultados.

O trabalho de Long Lv et al. (2011) aponta que a programação de tarefas e a programação de AGV em FMS são considerados como dois problemas independentes pela maioria dos pesquisadores e que muitas vezes o isolamento ignora conflitos de uso do AGV na programação de tarefas. Segundo os autores, as pesquisas sobre programação de AGV estão voltadas principalmente para otimizar o caminho de movimentação do AGV para melhorar a eficiência do sistema de produção, por sua vez o caminho a ser seguido pelo AGV depende da sequência de processamento das tarefas. Assim, considerando simultaneamente os dois problemas pode-se melhorar o desempenho do FMS.

Para isso os autores propõem um AG para fazer a programação das tarefas considerando os AGVs em um FMS com quatro tipos de produtos e quatro estações de trabalho. Não são feitas comparações com outras abordagens, porém, segundo Long Lv et al. (2011) os ganhos da abordagem se dão em ter uma programação mais próxima do ambiente real.

Kalinovic et al. (2011) propõem uma modificação no algoritmo do banqueiro para roteamento de AGVs. Em um ambiente de teste com 8 estações de trabalhos, 4 pontos de abastecimento, 34 rotas e 4 AGVs. Apesar dos autores citarem no início do trabalho que a abordagem trata também a programação, nos experimentos fica claro que nesse ambiente proposto são pré-definidas missões (ou despachos) para os AGVs, e o algoritmo do banqueiro modificado proposto tem por objetivo avaliar quais rotas os AGVs devem seguir para completar suas missões previamente definidas. A validação se deu por meio da comparação com o algoritmo do banqueiro original e com outra variação da literatura, analisando-se três pontos: distância viajada pelos AGVs, tempo de execução da missão, tempo de espera da missão. Na média entre os objetivos a proposta obteve melhores resultados.

Os autores Nishi e Maeno (2010) propõem uma abordagem de modelagem em PNs (redes de Petri) para minimizar o tempo total de transporte em baias de fabricação de semicondutores. A abordagem se baseia na criação de um planejamento de rotas livres de colisão com múltiplos AGVs de forma eficiente para minimizar o tempo de viagem total.

Pensando em reduzir a complexidade computacional, os autores propõem uma abordagem de decomposição da PN para resolver problemas de otimização de grande porte. O modelo PN é decomposto em várias sub-redes que compartilham um conjunto de lugares correspondentes aos recursos compartilhados. As sub-redes são criadas de forma independente, removendo os lugares compartilhados originais e criando para cada sub-rede o seu próprio conjunto de lugares de recursos com as conexões apropriadas. Os espaços de estados das sub-redes não são muito grandes, o primeiro passo do método consiste em calcular uma solução ótima para cada sub-rede usando o algoritmo do caminho mais curto. Porém segundo os autores a solução individual de cada sub-rede não é a ideal, assim o segundo passo é resolver o problema de otimização restrita no qual uma função penalidade é adicionada à função objetivo. A medida que se resolve o subproblema, o passo é repetido para cada sub-rede até que a condição de disparo para toda a PN esteja satisfeito. As soluções locais das sub-redes são integradas pelo algoritmo de otimização Novel. O menor tempo de cálculo pelo método proposto permite usar a metodologia indicada para o planejamento rápido e otimização dinâmica para sistemas de transporte AGVs na fabricação de semicondutores.

Os autores aplicaram a abordagem proposta no arranjo físico apresentado na Figura 7 e compararam com uma abordagem convencional no que diz respeito ao tempo de processamento

para se chegar ao planejamento da rota. A abordagem se mostrou eficaz, porém, não foi comparada ao desempenho do sistema de produção.

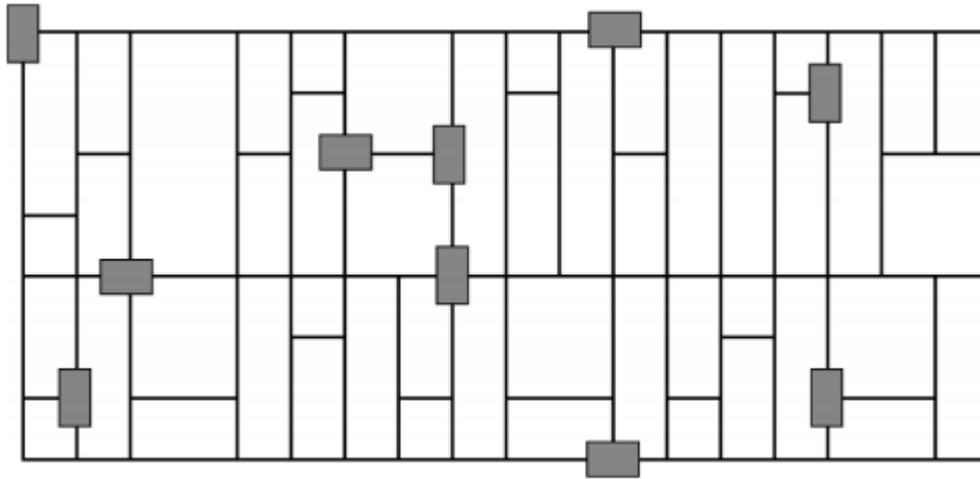


Figura 7 – Arranjo físico usado para validação do trabalho de Nishi e Maeno (2010).

Smolic-Rocak et al. (2010) apresentaram um método de roteamento dinâmico de supervisão e controle de vários AGVs. A fim de resolver o problema do caminho mais curto dinamicamente, o método de despacho proposto usa janelas de tempo em um vetor de formulário. O objetivo almejado pelos autores foi aumentar a eficiência da fábrica, em relação à taxa de produção, e também diminuir o número de *deadlocks*. Assim, a proposta leva em conta o número de tarefas ativas e suas prioridades. Por meio da técnica de janelas de tempo, é feita a previsão do caminho a ser traçado por um AGV e verifica-se a possibilidade de ocorrer *deadlocks* por essa trajetória. Para conseguir obter essa resposta, o arranjo físico da fábrica foi dividido em zonas. O método proposto usa janelas de tempo em uma forma vetorial para encontrar os caminhos candidatos para realizar a tarefa na menor distância. A viabilidade de um determinado caminho é avaliada realizando os testes de janelas de tempo sobrepostos, ou seja, é avaliado se em condições futuras da fábrica o caminho mais curto escolhido será viável para o despacho do AGV, e caso não o algoritmo busca alternativas para evitar o *deadlock*.

Os autores Nishi e Tanaka (2012) propõem uma decomposição de redes de Petri temporizadas, para despacho de AGVs livres de *deadlock* no roteamento em caminhos bidirecionais. Foi aplicado o método de decomposição de Redes de Petri com função de penalidade para reduzir a complexidade computacional, dividindo as redes de Petri temporizadas (TPNs) em sub-redes. Para que as sub-redes não ficassem “mortas” também foi

usado pelos autores um algoritmo de otimização sem marcação final. O método foi aplicado em um cenário hipotético, conforme ilustrado na Figura 8.

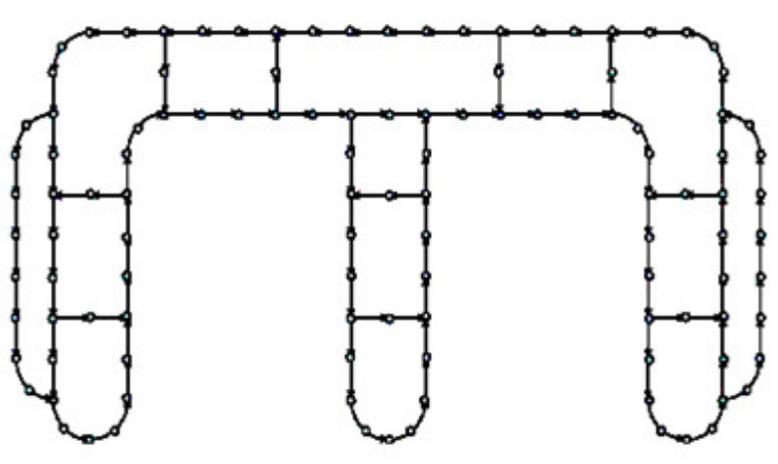


Figura 8 – Arranjo físico usado para validação do trabalho de Nishi e Tanaka (2012) com o sistema de transporte contendo 115 nós e 128 arestas.

Para validação os autores incorporaram no método proposto o algoritmo BA (*blocking avoidance algorithm*) e o desempenho do método é comparado com o método NN (*nearest neighborhood method*).

Em seu trabalho, Khanmohammadi et al. (2010), propõem um método de planejamento de caminho para AGVs. A proposta usa técnicas de controle *fuzzy* para guiar vários AGVs em um ambiente desconhecido e chegar a certos destinos. Segundo os autores o maior problema encontrado é ficar preso a um mínimo local. O sistema *fuzzy* proposto é fundamentado em uma base de dados que é dividida em 4 conjuntos gaussianos. O controle *fuzzy* consiste em duas partes: prevenção contra colisões com obstáculos imóveis e prevenção contra colisões com obstáculos móveis. Para isso foram consideradas quatro variáveis: velocidade do movimento, direção do movimento, trabalho preferencial para o AGV e distância do AGV para o destino. Nas simulações apresentadas, os AGVs chegaram ao destino sem apresentar colisões.

Dahari e Liu (2012) testaram os principais algoritmos descritos na literatura para realizar o roteamento de AGVs. Os autores dividem o problema de roteamento em dois: roteamento com fio-guia (*Wire-Guided AGV routing*) e roteamento sem fio (*Wireless-Guided AGV routing*).

Ainda segundo os autores o roteamento com fio-guia é normalmente representado graficamente. As rotas de um AGV podem ser descritas como um conjunto de arestas, ou um

conjunto de nós. A aplicação mais comum de métodos de roteamento AGV sem fio é o controle da zona. Ou seja, são divididas manualmente áreas de trabalho dos AGVs em várias praças (zonas), cada AGV trabalhando em diferentes praças (zonas) para evitar colisões.

Assim o trabalho apresentou os principais algoritmos de roteamento e suas formas básicas, sendo eles: Algoritmo Dijkstra, Algoritmo Floyd-Washal, Algoritmo Genético (GA), Algoritmo A*, Curva de Bezier, Lógica *Fuzzy*, Algoritmo de Campo Potencial Artificial (APF) e Otimização por Enxame de Partículas (PSO).

No final do trabalho é apresentada uma tabela (Tabela 1) com resultados do teste realizado com os algoritmos citados, porém os autores não mencionam as características do teste.

Tabela 1 – Resultados apresentados no trabalho de Dahari e Liu (2012).

Algoritmo	Dijkstra / Floyd	Curva Bezier	Lógica <i>Fuzzy</i>	APF	GA	PSO	Redes Neurais
Complexidade de projeto	Baixo	Baixo	Baixo	Baixo	Médio	Médio	Médio
Complexidade de Algoritmo	Baixo	Baixo	Baixo	Baixo	Médio	Médio	Médio
Pré tratamento do mapa	Não	Não	Não	Não	Não	Não	Não
Velocidade em mapas pequenos	Alto	Alto	Alto	Alto	Baixo	Baixo	Alto
Velocidade em mapas grandes	Baixo	Baixo	Alto	Baixo	Médio	Médio	Alto
Complexidade da Programação	Baixo	Médio	Baixo	Baixo	Alto	Alto	Baixo
Precisão	100%	Médio	Baixo	Baixo	Alto<100%	Alto<100%	Alto
Adaptabilidade	Fio Condutor	Rota livre	Universal	Rota Livre	Universal	Universal	Universal

As conclusões são que o algoritmo curva de Bezier é o mais instável, não sendo recomendado para mapas complexos ou sistemas de multiAGVs. A lógica *fuzzy* é um dos algoritmos mais fáceis de programação, mas a precisão é altamente dependente da estrutura do algoritmo e dos parâmetros usados. O algoritmo “Campo Potencial Artificial” mostrou a capacidade de adaptação em mapas complexos e sistemas multiAGV. GA e PSO tiveram desempenho um pouco instável no teste, especialmente quando as instalações são complexas. Por fim, os autores alertam que dependendo do arranjo físico de teste esses números podem ser diferentes.

2.4 Considerações finais

Pesquisas relacionadas ao controle de AGVs tiveram início na década de 80 e estão em evidência até a atualidade. As pesquisas iniciaram com decisões simples para um problema isolado do controle de AGVs, posteriormente evoluíram para decisões com múltiplos atributos e atualmente buscam-se novas informações para melhorar as decisões. Um exemplo é o uso de informações futuras para apoiar o sistema de decisão em tempo de operação.

Pesquisas recentes vêm apontando que a inclusão de mais recursos ao controle de AGVs pode ser uma vantagem. Por exemplo, muitos trabalhos encontrados na revisão usam a distância entre os recursos de fábrica para tomada de decisão de despacho, porém, para o AGV chegar ao destino é necessário o uso do recurso rota. Assim, se o despacho despreza as rotas, o mesmo pode não alcançar resultados correlatos aos de ambientes com fatores reais.

Outro ponto importante a ser destacado é que trabalhos que abordam regras multiatributos conseguem melhores valores do objetivo de otimização. Porém, na grande maioria dos trabalhos da literatura que aborda o tema, as validações são realizadas por meio de comparações com regras de único atributo. Logo é possível afirmar que regras multiatributos são melhores que regras simples, porém, não se sabe qual a melhor abordagem multiatributos. Também é verificado na literatura que, mesmo em abordagens multiatributos, quando se acrescenta alguma informação futura do sistema de produção, alcançam-se melhores resultados.

Voltando ao quesito validação, verifica-se que não existe uma padronização de formas para validação e, quando se usa simulações, não existe um *benchmark*, ou seja, cada autor propõe seu próprio arranjo físico de fábrica, e que as informações/parâmetros como: produtos, roteiros, tempos de processamentos, tempo de viagem do veículo, não são disponibilizadas. Esse fato dificulta as comparações entre as abordagens. Continuando nessa linha de raciocínio, outro ponto negativo é que esses arranjos físicos normalmente têm poucos recursos compartilhados, dessa forma, com pouca complexidade, não refletindo uma fábrica de porte real. Qiu et al. (2002) reforçam essa ideia afirmando que muitos algoritmos para a programação e roteamento dos AGVs têm sido propostos. No entanto, a maior parte dos resultados existentes são aplicáveis em sistemas com um pequeno número de AGVs, oferecendo um baixo grau de simultaneidade.

Vale ressaltar ainda que muitos dos trabalhos testam suas abordagens para mais de um objetivo de otimização, porém não existe um tratamento específico ou até uma junção desses objetivos. E, por fim, devido os bons resultados alcançados em alguns artigos da literatura, bem como os resultados promissores alcançados em testes preliminares desta tese, é conclusivo que o uso de sistemas *fuzzy*, bem como redes de Petri, têm grande aderência na resolução dos problemas de controle de AGVs.

Na Tabela 2 se apresenta de forma condensada os artigos referenciados nesse capítulo, explicitando-se os autores, critério de otimização buscado, variáveis consideradas e técnicas usadas.

Tabela 2 – Síntese dos trabalhos apresentados no capítulo

Autor e data	Critério de otimização	Variáveis	Técnica
Egbelu e Tanchoco (1984)	<i>Makespan</i>	Buffer de saída Tempo de chegada do produto	Regra único atributo
Bartholdi Iii e Platzman (1989)	<i>Makespan</i>	Primeiro encontrado	Regra único atributo
Klein e Kim (1996)	<i>Makespan</i>	Distância Buffer de saída Tempo de espera da solicitação	A*
Kim et al. (1999)	<i>Throughput</i>	Buffer de entrada Buffer de saída Tempo para o AGV percorrer a distância	Equação composta
Tan e Tang (2000)	Ociosidade do sistema	Distância Buffer de entrada Buffer de saída	<i>Fuzzy</i>
Jeong e Randhawa (2001)	<i>Work-in-process</i>	Distância Buffer de entrada Buffer de saída	Rede neural artificial
Benincasa et al. (2003)	<i>Makespan</i>	Distância Entroncamento Buffer de saída	<i>Fuzzy</i>
Umashankar e Karthik (2006)	<i>Tardiness</i>	Distância Buffer de saída Prioridade do processo	<i>Fuzzy</i>
Morandin et al. (2007)	<i>Makespan</i>	Distância Entroncamento Buffer de saída	<i>Fuzzy</i> e Algoritmos genéticos
Morandin et al. (2011a)	<i>Makespan</i>	Distância Entroncamento Buffer de saída	<i>Fuzzy</i> hierárquico

Ho, Liu e Yih (2012)	<i>Tardiness</i>	Distância Tempo de espera da tarefa	
Kim e Yarlagadda (2013)	<i>Makespan</i>	Distância Tempo de espera da carga	Equação composta
Chen e Matis (2013)	<i>Tardiness</i>	Menor tempo de processamento Tempo de processamento restante	
Azadeh et al. (2013)	<i>Makespan</i>	Ordem de pedido Menor valor no tempo de processamento	Redes neurais artificiais
Shahzad e Mebarki (2016)	<i>Tardiness</i>	Histórico de produção	Árvores de decisão e busca tabu
Naso e Turchiano (2005)	<i>Makespan</i>	Distância Buffer de entrada Encadeamento de tarefas	<i>Fuzzy</i> hierárquico e Algoritmos genéticos
Hidehiko (2008)	<i>Throughput</i>	Histórico de produção	
Morandin et al. (2011b)	<i>Makespan</i>	Distância Buffer de entrada Buffer de saída Entroncamentos	<i>Fuzzy</i>
	<i>Tardiness</i>	Distância Entroncamentos Data de entrega Encadeamento de tarefas	<i>Fuzzy</i>
Chen (2013)	<i>Tardiness</i>	Histórico de produção	<i>Fuzzy c-means</i>

PROPOSTA DO TRABALHO

3.1 Considerações iniciais

De acordo com os temas discutidos nos capítulos anteriores, bem como baseando-se nos trabalhos descritos na literatura, derivam-se importantes hipóteses no desafio de desenvolver uma abordagem eficaz para o problema de despacho de veículos em sistemas de manufatura.

A delas é a de que incluir recursos da fábrica como, o uso das rotas do chão de fábrica e o tempo de *setup* de máquina, além das variáveis habituais por meio de um sistema *fuzzy* em cascata que permita alcançar um valor de *makespan* ou *tardiness* mais correto e que na média seja menor se comparado com propostas que não consideram os recursos em questão.

A segunda hipótese é a de que o uso de informações de estados futuros do sistema de produção fabril por meio de uma estrutura baseada em espaço de estados permita alcançar um valor de *makespan* ou *tardiness* menor se comparado com propostas que não avaliam estados futuros.

A terceira hipótese é a de que incluir outros recursos da fábrica por meio de um sistema *fuzzy* em cascata e usar informações de estados futuros do sistema de produção fabril por meio de uma estrutura baseada em espaço de estados, isto é, trabalhar com a primeira e segunda hipótese em conjunto, permita alcançar um valor de *makespan* ou *tardiness* mais correto e menor se comparado com os valores obtidos por meio das hipóteses separadas.

A proposta é o uso do sistema *fuzzy* em cascata e uma estrutura baseada em espaço de estados aplicados no despacho de AGVs.

Neste capítulo, a teoria de funcionamento do sistema *fuzzy* em cascata bem como da estrutura de espaço de estados serão descritas e instanciadas. Completando a descrição, são apresentados diagramas, pseudocódigos e parâmetros que elucidam a proposta.

3.2 Proposta do sistema *fuzzy* em cascata e da estrutura baseada em espaço de estados para despacho de AGVs.

Conforme apresentado anteriormente, o ambiente de manufatura é caracterizado por ser não linear, complexo e caótico. Os sistemas de manufatura são geralmente imprevisíveis e de difícil controle, além de sofrerem constantes pressões provindas da demanda de mercado por produtos customizados com um curto prazo de entrega, o que exige uma resposta rápida e uma adaptação eficiente às mudanças.

Dentro do FMS, o sistema de transporte possui um papel fundamental, pois ele é o responsável por transportar todo o material em processamento de um ponto a outro, permitindo assim o fluxo da produção.

O problema de realizar o despacho de veículos em um ambiente de manufatura, em outras palavras, atribuir a melhor tarefa para o veículo de forma a otimizar o critério objetivo, se torna complexo em virtude das características do ambiente de manufatura. Quando se tem um ambiente de manufatura composto de várias máquinas, AGVs, produtos e rotas o processo de decisão torna-se um problema do tipo *NP-hard*.

Conforme levantamento realizado na fase de revisão da literatura, constatou-se que a busca por minimização de *makespan* e *tardiness* em problemas de despacho de AGVs tem sua relevância e atualidade e que a ponderação de múltiplos atributos para tomada de decisão e inclusive o uso de sistemas *fuzzy* são bem referenciados.

Outra constatação, em se tratando de problemas de despacho de veículos, foi a de que não é trivial o uso de estruturas que permitem extrair informações de estados futuros do ambiente de manufatura.

Em resumo, a proposta é um sistema de despacho que atua diretamente no controle dos AGVs baseada em espaço de estados para extração de informações de estados futuros do

ambiente de manufatura e empregando sistemas *fuzzy* em cascata na tomada de decisão com o objetivo de minimizar ora *makespan*, ora *tardiness*, dependendo da escolha do usuário.

Para que a proposta seja capaz de controlar os AGVs se faz necessário um fluxo de informação entre ambos, no qual o sistema proposto seja capaz de coletar informações do ambiente de manufatura como, por exemplo, localização dos AGVs, status das estações de trabalho, status dos *buffers*, etc, e também enviar dados que se traduzem em ações a serem executadas no ambiente de manufatura.

Em ambientes de manufatura automatizados se faz o uso de controladores lógicos programáveis (CLPs), sistemas supervisórios e protocolos de comunicação como, por exemplo, OPC, para coletar, transmitir e monitorar as informações do ambiente de manufatura.

Devido ao fato de não se dispor de um ambiente de manufatura integralmente automatizado para implementar e validar a proposta desta tese, são usados ambientes de manufaturas virtuais que são simulados por meio de um *software* de simulação de eventos discretos (SSED).

Na Figura 9 é ilustrada a proposta da tese conectada ao ambiente de manufatura em um nível macro de detalhes.

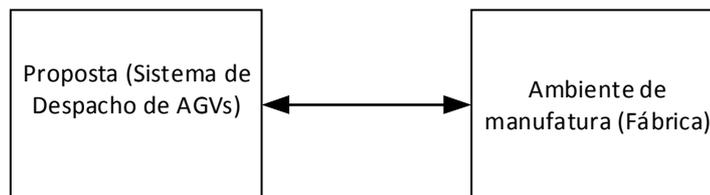


Figura 9 – Proposta da tese conectada ao ambiente de manufatura em um nível macro de detalhes

Embora não se tenha uma fábrica real para implementação e validação, a proposta é construída com baixo acoplamento visando uma fácil adaptação. Portanto, se uma fábrica estiver preparada fisicamente, ou seja, possuir um nível de automação requerido, somente alterando o protocolo de comunicação a proposta pode ser implementada em uma fábrica real.

Conforme citado anteriormente, constatou-se durante a revisão da literatura que a busca por minimização de *makespan* e *tardiness* em problemas de despacho de AGVs tem sua relevância e atualidade. Contudo, os trabalhos encontrados buscam a minimização de somente

um dos critérios. Portanto uma contribuição adicional da proposta é permitir que o usuário escolha qual critério deseja-se minimizar.

Na Figura 10 é ilustrada a proposta no nível de maior detalhe, em que se tem a proposta dividida em dois subsistemas se comunicando com o usuário e com a fábrica.

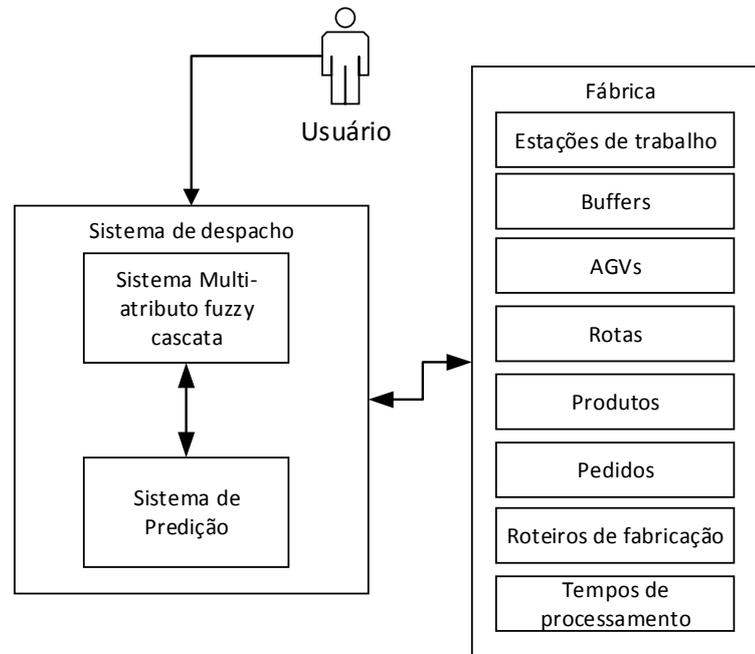


Figura 10 – Proposta da tese conectada ao ambiente de manufatura e com o usuário em um nível maior de detalhes.

3.2.1 Sistema Multiatributo *Fuzzy* Cascata

A proposta baseia-se na possibilidade de ponderar múltiplos atributos, sendo eles de níveis organizacionais diferentes, e também considerar recursos compartilhados do sistema que afetam diretamente ou indiretamente o despacho de AGVs.

Dado que a proposta desta tese se traduzirá em um sistema computacional capaz de tomar decisões racionais em um ambiente de incerteza e imprecisão, optou-se pelo uso de sistemas *fuzzy*. Entretanto a proposta é ponderar múltiplos atributos e sabe-se que a cada novo atributo de entrada de um sistema *fuzzy* faz com que a base de regras *fuzzy* cresça exponencialmente, o que pode acarretar em um custo computacional alto. Esse fator pode causar lentidão no processamento, o que faria com que o sistema de despacho se torne inviável, pois a decisão precisa ser em tempo real.

Com a finalidade de contornar esse obstáculo e conjuntamente permitir que a proposta possa considerar dois critérios de otimização, faz-se o uso de sistemas *fuzzy* em cascata. Na Figura 11 apresenta-se o sistema *fuzzy* em cascata, em que o mesmo é formado por quatro sistemas *fuzzy* e um sistema de pesos.

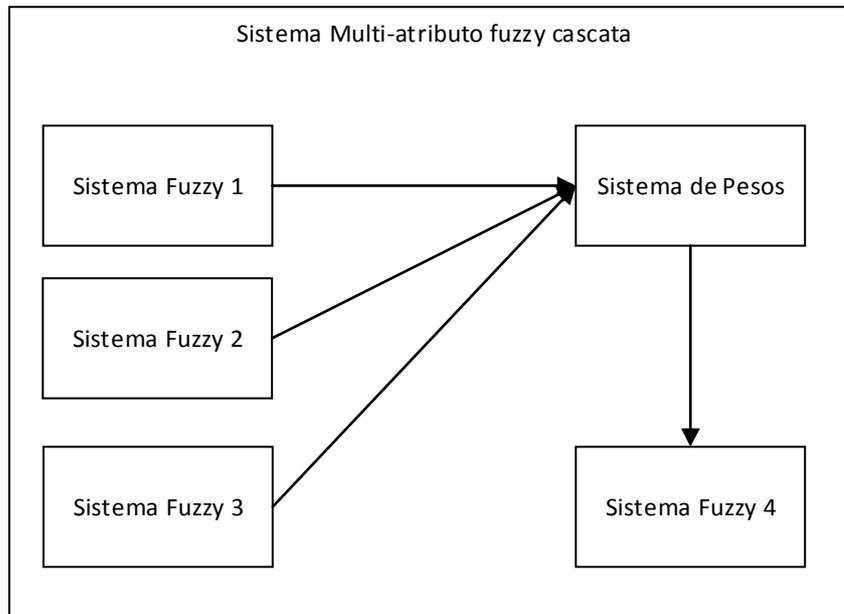


Figura 11 – Esquema macro do sistema multiatributo *fuzzy* cascata.

Essa configuração vai ao encontro com o desenvolvimento com baixo acoplamento, pois na hipótese de se incluir um novo critério de otimização, pode-se fazê-lo somente incluindo um novo sistema *fuzzy* no primeiro nível da cascata.

Os principais atributos do ambiente de manufatura apontados na base de conhecimento do laboratório Tear e na literatura são: distância, número de entroncamentos, *buffer* de entrada e de saída das estações de trabalho, encadeamento de tarefas e data de entrega. As rotas a serem usadas pelos AGVs e os *setups* das estações de trabalho também são fatores importantes para o controle de AGVs. Esses atributos são divididos nos sistemas *fuzzy* 1, 2 e 3 de acordo com a correlação com o objetivo de otimização

O sistema *fuzzy* 1 recebe como variáveis de entrada atributos do ambiente de manufatura que tem relação com os dois critérios de otimização, sendo eles:

- **Distância:** é a distância que o AGV tem que percorrer até a estação de trabalho que está requerendo transporte somada a distância entre a estação de trabalho

requisitante e o destino final. Considerando esse atributo, dando prioridade para os veículos mais próximos, se economiza tempo de produção e uso dos AGVs;

- **Número de entroncamentos:** os entroncamentos ou cruzamentos existentes no arranjo físico do ambiente quando caminhos se cruzam. Considerando a quantidade de entroncamentos pode-se diminuir a possibilidade de ocorrências de *deadlock* e também minimizar o tempo de transporte;
- **Rotas:** as rotas são espaços no arranjo físico que permitem a circulação dos AGVs. A rota é um recurso que só permite o uso de um AGV em um determinado tempo. Assim ao desconsiderar rotas no despacho de AGVs, dois ou mais AGVs podem alocar o mesmo recurso ao mesmo tempo, o que não é possível de ser executado.

O sistema *fuzzy* 2 recebe como variáveis de entrada atributos do ambiente de manufatura que tem maior relação com o critério de *makespan*, sendo eles:

- **Quantidade de peças no buffer de entrada e de buffer de saída:** O *buffer* é um espaço de armazenamento temporário de peças a serem redirecionadas. Normalmente cada estação de trabalhos possui um *buffer* de entrada e um *buffer* de saída. Se o *buffer* de entrada esvaziar por completo, a estação de trabalho fica sem tarefa e interrompe a produção local. Se o *buffer* de saída encher por completo, a estação de trabalho fica sem espaço para alocar seus produtos acabados, causando também interrupção na produção local. Pelo controle dos níveis do *buffer* de entrada e do *buffer* de saída pode-se garantir um menor número de interrupções na produção, ganhando-se tempo. Outro benefício que se pode produzir ao incluir esses atributos na decisão é não permitir que os AGVs fiquem presos em mínimos locais, pois na ótica da variável distância, os AGVs podem ficar atendendo somente uma região de estações de trabalho próximas.
- **Tempo de setup:** o termo *setup* é usado para definir o tempo de preparação da máquina para iniciar a produção de um produto localmente. Ao desconsiderar o tempo de *setup* se tem um tempo de produção menor do que o real.

O sistema *fuzzy* 3 recebe como variáveis de entrada atributos do ambiente de manufatura que tem maior relação com o critério de *tardiness*, sendo eles:

- **Encadeamento de tarefas:** é a quantidade de tarefas que podem ser encadeadas, ou seja, o encadeamento de tarefas acontece quando uma tarefa pode iniciar onde a anterior terminou, sem interrupções. Por exemplo, o AGV carrega uma peça A na estação de trabalho 1 e descarrega na estação de trabalho 3 e imediatamente carrega a peça B para levar para estação de trabalho 5, onde existe uma peça C aguardando para ser levada para estação de trabalho 2. Considerando essa variável diminui-se a quantidade de quilômetros rodados pelos AGV, diminuindo a quantidade de energia gasta e diminuindo o tráfego de veículos no arranjo físico, consequentemente diminuindo a possibilidade de ocorrências de congestionamentos e *deadlocks*.
- **Data de entrega:** é a data que a empresa se comprometeu para entregar o pedido ao seu cliente. Considerando esse atributo, alinha-se a produção com a demanda, cumprem-se os prazos com baixo desperdício melhorando a confiabilidade de entrega de bens e serviços.

O sistema de pesos recebe a informação de qual critério de otimização foi escolhido pelo usuário e a partir dessa informação atribui pesos para as saídas dos sistemas *fuzzy* 1, 2 e 3. As variáveis de saída dos sistemas *fuzzy* 1, 2 e 3, chamadas de prioridade são multiplicadas pelo peso atribuído de acordo com o critério de otimização escolhido pelo usuário e se tornam variáveis de entrada para o sistema *fuzzy* 4. O sistema *fuzzy* 4, por sua vez, realiza novos cálculos e fornece uma nova prioridade. Na Figura 12 é exemplificado esse funcionamento.

Os quatro sistemas *fuzzy* da proposta são do tipo Mamdani. O tipo Mamdani usa conjuntos *fuzzy* nos consequentes das regras *fuzzy* (MAMDANI; ASSILIAN, 1975). A saída final é representada por um conjunto *fuzzy* resultante da agregação da saída inferida de cada regra. Para se obter uma saída final não *fuzzy* (escalar) adota-se um dos métodos de transformação da saída *fuzzy* em não *fuzzy* (JANG; SUN; MIZUTANI, 1997), (PEDRYCZ; GOMIDE, 1998), (CORDÓN et al., 2001). Na Figura 13 são ilustrados os componentes de um sistema *fuzzy* do tipo Mamdani.

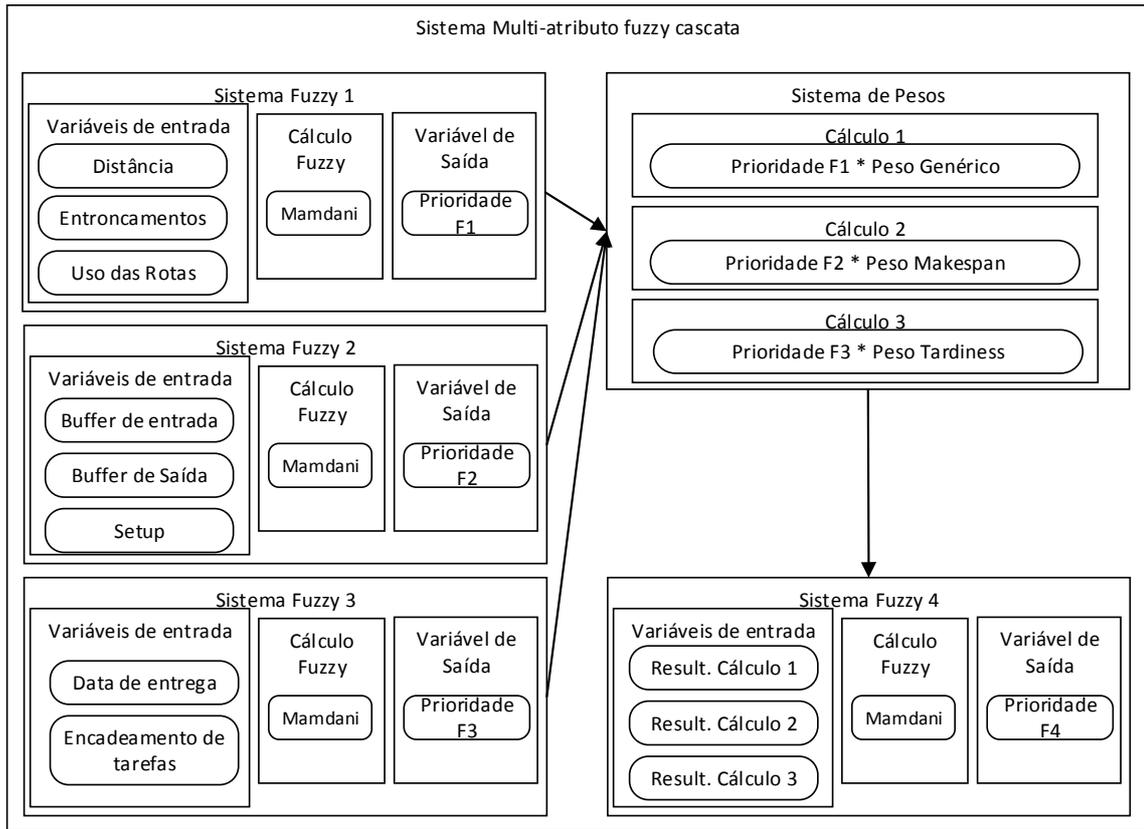


Figura 12 – Esquema detalhado do sistema multiatributo *fuzzy* cascata.

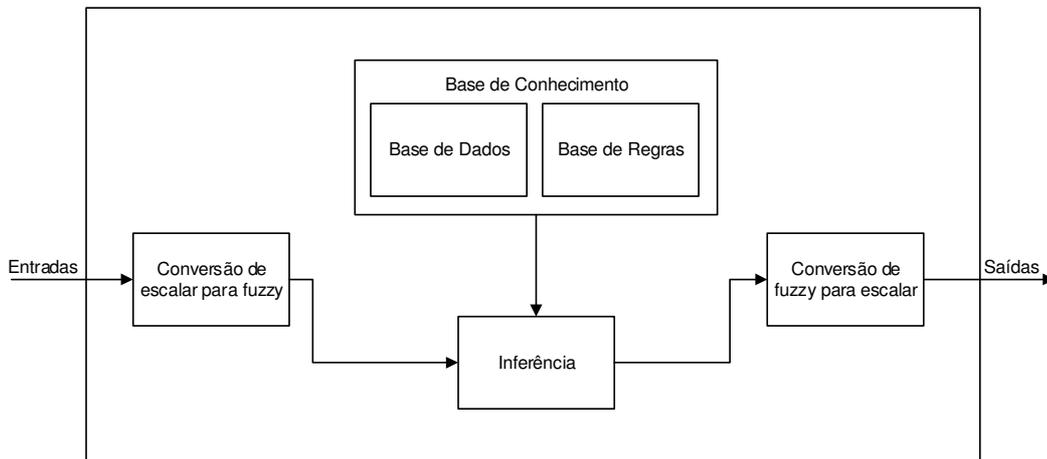


Figura 13 - Representação do sistema *fuzzy* Mamdani.

A base da teoria de conjuntos *fuzzy* é o conceito de função de pertinência. Na construção dos sistemas *fuzzy* da proposta todas as variáveis são representadas por funções triangulares, pois a ideia de definição de regiões de pertinência total, média e nula é mais intuitiva que a especificação de valor modal e dispersão associada ao projeto de funções Gaussianas. A

conversão de um escalar para *fuzzy* (*Fuzzificação*) usa as funções de pertinência pré-estabelecidas, mapeia cada variável de entrada do sistema em graus de pertinência do *fuzzy* que representa a variável em questão.

A proposta usa o método de inferência composicional simplificada com entradas numéricas, usando os operadores max-min para a inferência (conjunção pelo mínimo e agregação pelo máximo). A conjunção é usada nos antecedentes da regra encontrando o mínimo deles e a agregação é usada no consequente das regras encontrando o máximo delas.

Os sistemas do tipo Mamdani obtêm como resultado do seu processamento de inferência um conjunto *fuzzy* que deve ser convertido de *fuzzy* para escalar para a obtenção de uma saída numérica exata. A conversão de *fuzzy* para escalar (*Defuzzificação*) usada nesse trabalho é centro de área (CoA).

Detalhes sobre os conceitos de função de pertinência, representação matemática e gráfica das funções triangulares, exemplos e equações para o cálculo da conversão de escalar para *fuzzy*, inferência e conversão de *fuzzy* para escalar estão descritos no Apêndice A.

3.2.2 Sistema de Predição

A proposta baseia-se na possibilidade de extrair informações de estados futuros do ambiente de produção e usar essas informações para melhorar a decisão de despacho de AGVs.

A proposta de predição é baseada na exploração de espaço de estados gerados a partir de um grafo. A Rede de Petri possui características favoráveis para a modelagem de sistemas que possuem concorrência por recursos, permitindo diversas abordagens, indo desde uma resolução de conflito por regra padrão, decisões nebulosas ou mesmo interferência de operador (MURATA, 1989). Por estes motivos a modelagem da fábrica em redes de Petri colorida (CPN) é usada nesta proposta.

Conjuntamente com os roteiros de fabricação dos produtos da fábrica modelados em CPN, o sistema de predição inclui os módulos: ranking, predição de AGV e sistema *fuzzy* 5. A representação esquemática é apresentada na Figura 14.

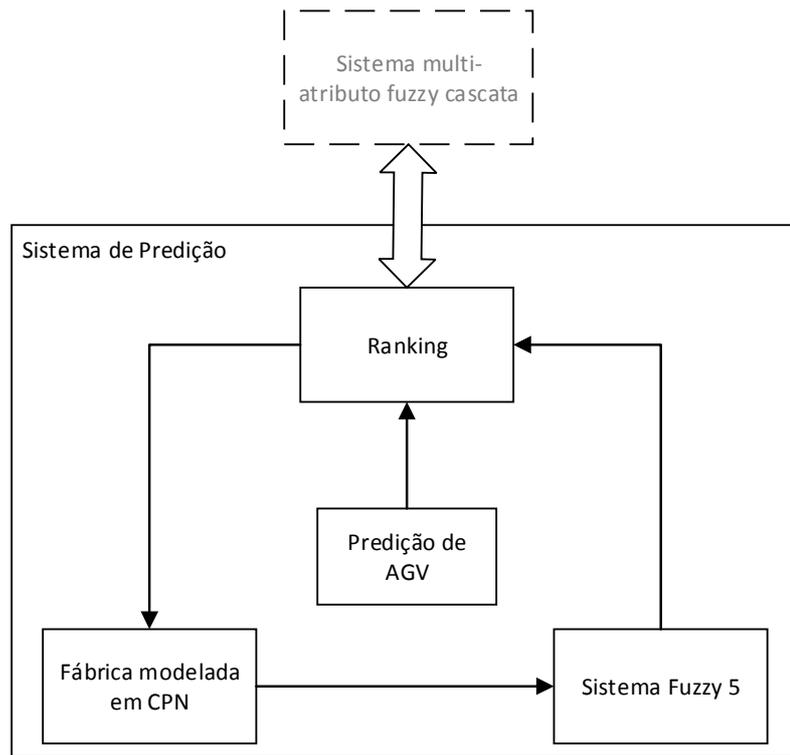


Figura 14 – Representação esquemática macro do sistema de predição

O módulo *ranking* recebe a lista das tarefas que estão aguardando para serem atendidas e suas respectivas prioridades calculadas pelo sistema multiatributo *fuzzy cascata*. Então o *ranking* realiza a ordenação hierárquica decrescente com base nos valores de prioridade.

O ambiente de fábrica modelado em redes de Petri replica o estado atual do ambiente de manufatura por meio de marcas que representam os produtos e os AGVs. A partir dessa rede é gerado e explorado o espaço de estados. Detalhes sobre a técnica de exploração do espaço de estados são descritas no capítulo 4 e no Apêndice A.

Dependendo da profundidade da exploração do espaço de estados, essa pode ser uma tarefa computacionalmente custosa e, como enfatizado ao longo desta tese, o despacho precisa ser realizado em tempo real. Portanto a proposta é de explorar os próximos estados somente das cinco primeiras tarefas do ranking e destes são investigados os possíveis próximos estados aprofundando somente três níveis no espaço de estados.

Na Figura 15 é ilustrado um exemplo de um estado que pode ocorrer em um ambiente de manufatura. Nesse exemplo têm-se cinco estações de trabalho (M1, M2, M3, M4 e M5), um AGV, que está parado na M5 e dois tipos de produtos (P1 e P2), cujo roteiro de produção

é apresentado na mesma figura. Cada produto tem três etapas de produção (E1, E2 e E3). Existem um produto 1 que acabou de ser processado por M3 e um produto 2 que acabou de ser processado por M4, ambos necessitam de transporte para seguir para próxima etapa de produção.

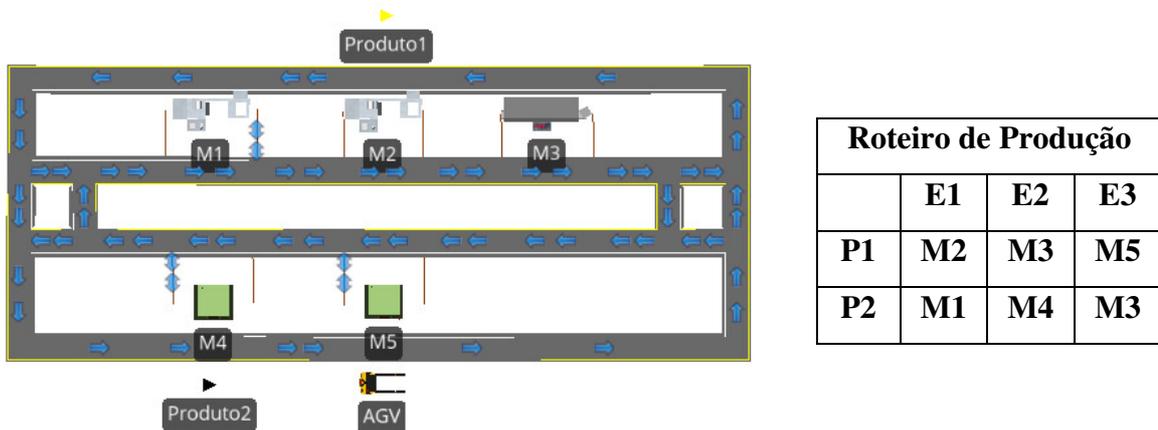


Figura 15 – Exemplo de um estado no ambiente de manufatura e roteiro de produção

Existem vários próximos estados possíveis. Porém nas figuras Figura 16 e Figura 17 são apresentados dois que exemplificam perfeitamente como o espaço de estados pode ajudar na decisão de despacho de AGVs.

Na Figura 16 o AGV atende primeiramente o P1 levando-o para M3 e posteriormente P2 levando-o para M3. P1 é processado enquanto P2 aguarda no *buffer* de entrada. Em seguida P1 é levado para M5. Simultaneamente P1 é processado pela M5 e P2 é processado pela M3. Terminados os processamentos, os dois produtos estão finalizados.

Na Figura 17 o AGV atende primeiramente o P2 levando-o para M3 e posteriormente P1 levando-o para M3. P2 é processado enquanto P1 aguarda no *buffer* de entrada. Terminado o processamento, P2 está pronto e é levado para o estoque. P1 é processado pela M3 e terminado seu processamento é levado para M5 para que depois de processado seja finalizado.

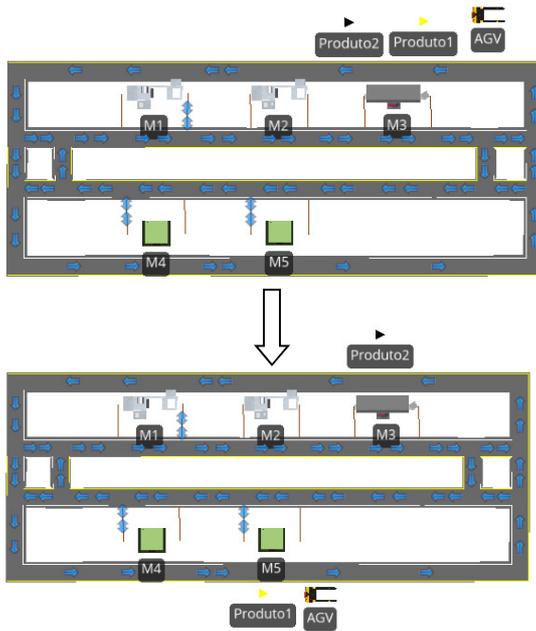


Figura 16 – Possíveis estados futuros 1

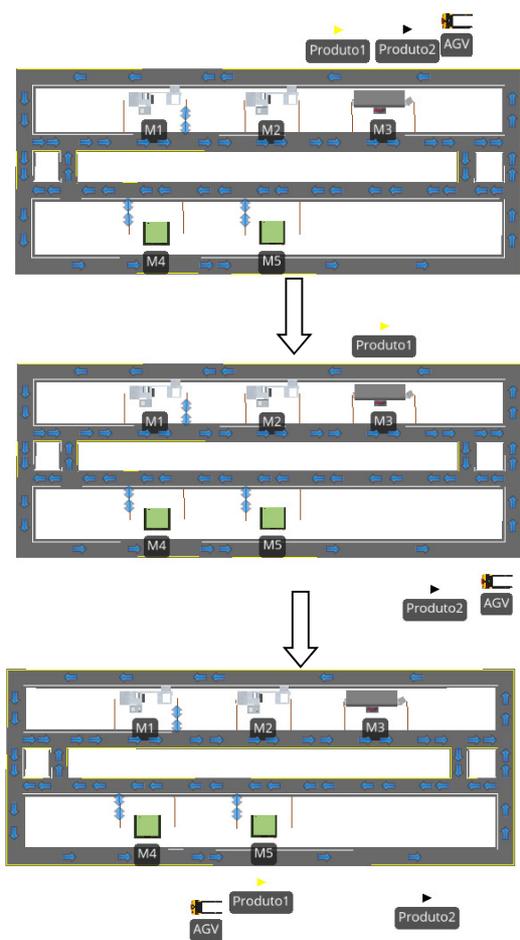


Figura 17 – Possíveis estados futuros 2

No estado 1 o tempo para produção de todos os produtos (*makespan*) é menor. Porém no estado 2 o produto 2 fica pronto mais rapidamente, e a depender o prazo de entrega desse produto, pode ser um fator para minimizar o *tardiness*.

Portanto, ao gerar o espaço de estados, são recuperadas as seguintes informações dos possíveis estados futuros:

- Quantidade de peças no *buffer* de entrada da estação de trabalho nos próximos estados;
- Tempo para alcançar o terceiro nível seguindo um determinado ramo;
- Quantidade de transições habilitadas, que significam quantidade de tarefas que podem ser iniciadas.

Essas informações são enviadas para o sistema *fuzzy 5* que retorna uma nova prioridade. A nova prioridade atualiza o *ranking*, isto é, reordenando as tarefas. A tarefa na primeira

posição do *ranking* é a que será atendida. A representação esquemática detalhada do sistema de predição é apresentada na Figura 18.

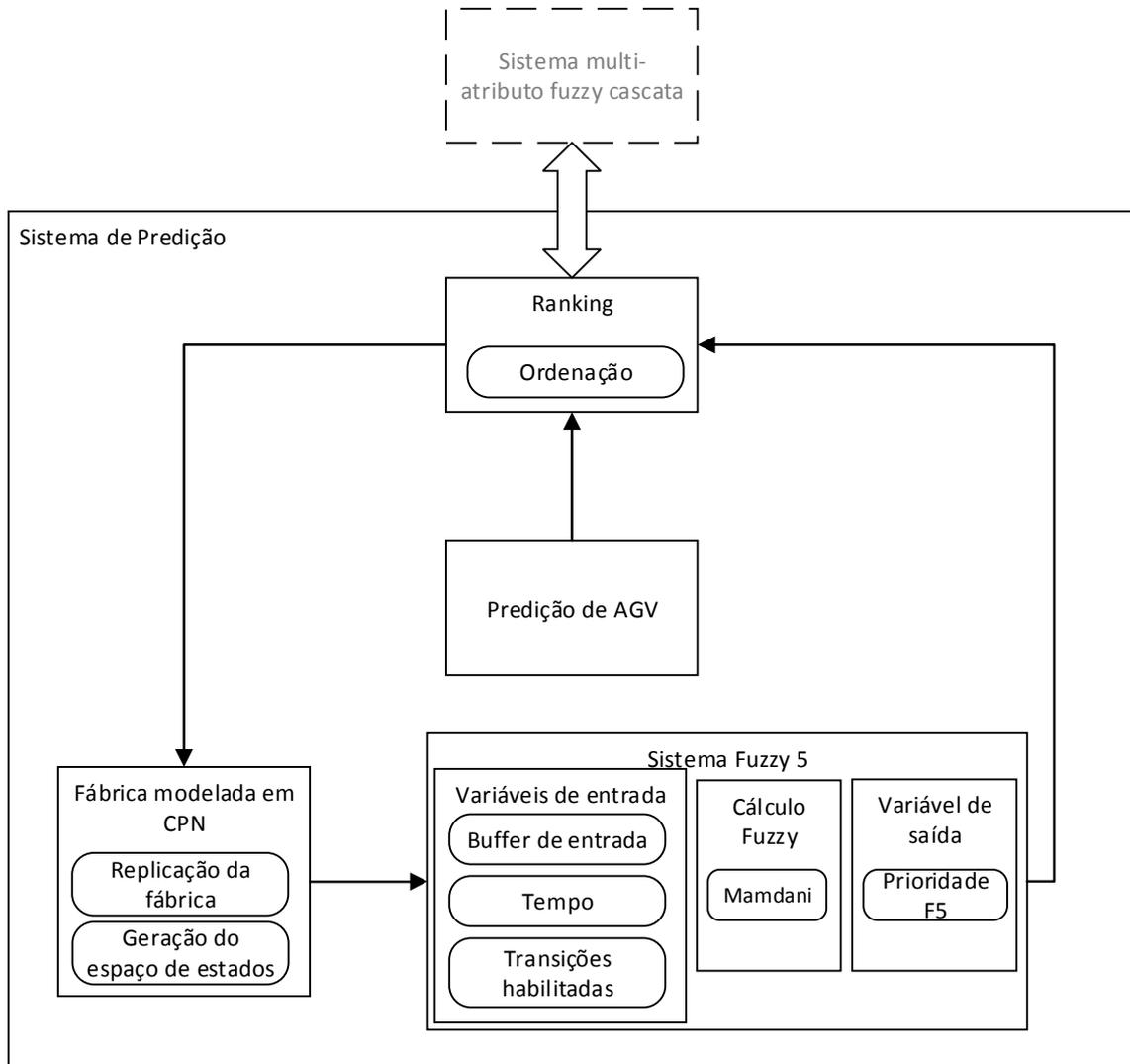


Figura 18 – Representação esquemática detalhado do sistema de predição

O sistema *fuzzy* 5 segue as mesmas características (conversão escalar/*fuzzy*, conversão *fuzzy*/escalar, método de inferência, função de pertinência) dos quatro sistemas *fuzzy* do sistema multiatributo *fuzzy* cascata descrito na seção anterior. Descrições sobre o módulo predição de AGV são apresentadas na próxima subseção deste capítulo.

3.2.2.1 Proposta do módulo de predição AGV

Como explicado em momentos anteriores, o problema do despacho de AGVs é atribuir as melhores tarefas aos veículos de maneira a minimizar o critério de otimização. Nos trabalhos encontrados na literatura bem como nesta tese, essa atribuição acontece quando um veículo se encontra na situação de livre. Porém pode ocorrer de um veículo em trânsito ter melhores condições de atender uma determinada tarefa.

Considere como exemplo a seguinte situação: existe um produto para ser retirado na estação de trabalho 2. O veículo 1 está livre, pois acabou de descarregar um produto na estação de trabalho 5. O veículo 2 está ocupado, todavia o veículo 2 está levando um produto para ser processado na estação de trabalho 2. Portanto, a depender do tempo que falta para o veículo 2 chegar ao seu destino, ele pode ser uma melhor opção para colaborar com a minimização do critério de otimização.

Sendo assim, uma contribuição adicional desta tese é realizar uma verificação dos AGVs que estão em trânsito, seguindo a seguinte inequação:

$$(A + B) < C \quad (1)$$

Onde:

- A é o tempo que o AGV que está em trânsito (AGVt) tem para concluir a tarefa atual.
- B é o tempo que o AGVt leva para ir de onde ele terminará sua tarefa atual até o local da tarefa a ser atendida.
- C é o tempo que o AGV livre (AGVl) leva para ir de onde ele está até o local da tarefa a ser atendida.

O AGV em trânsito pode acumular somente uma tarefa. Se a condição descrita na inequação 1 for verdadeira, então:

- AGVt acumula uma nova tarefa a ser realizada imediatamente após a conclusão da tarefa atual;
- AGVl atende a tarefa que se encontra na segunda posição do ranking.

Senão, isto é, a condição descrita na inequação 1 for falsa, então:

- AGV_l atende a tarefa;
- AGV_t termina sua tarefa atual sem acumular uma nova tarefa.

3.2.3 Funcionamento da proposta multiatributo *fuzzy* em conjunto com predição

Recapitulando, a proposta desta tese é um sistema de despacho que atua diretamente no controle dos AGVs baseada em espaço de estados para extração de informações de estados futuros do ambiente de manufatura e empregando sistemas *fuzzy* em cascata na tomada de decisão com o objetivo de minimizar ora *makespan*, ora *tardiness*, dependendo da escolha do usuário. Na Figura 10, apresentada anteriormente, é ilustrada a proposta, em que se tem a proposta dividida em três subsistemas se comunicando com o usuário e com a fábrica virtual.

O funcionamento da proposta se dá com a execução da simulação da fábrica, da execução da simulação da rede de Petri e dos sistemas multiatributo *fuzzy* cascata e predição. Quando todos os softwares estão em execução é solicitado que o usuário escolha qual o critério de otimização que se deseja minimizar, *makespan* ou *tardiness*. Realizada a escolha, a informação é transmitida para o sistema de pesos, que pertence ao sistema multiatributo *fuzzy* cascata, e para fábrica. A fábrica por sua vez inicia a produção. Na Figura 19 é apresentado o fluxograma que ilustra o funcionamento básico da proposta.

No momento em que um AGV da fábrica se encontra com o status de livre, a informação é transmitida da fábrica para o sistema multiatributo *fuzzy* cascata e inicia-se o processo de decisão. Primeiramente são mapeados todos os produtos que se encontram nas seguintes áreas da fábrica:

- Nos *buffers* da saída das estações de trabalho.
- No estoque de matéria prima, aguardando o início da sua produção.
- Produtos em processamento nas estações de trabalho que tenham mais de 50% do processo concluído.

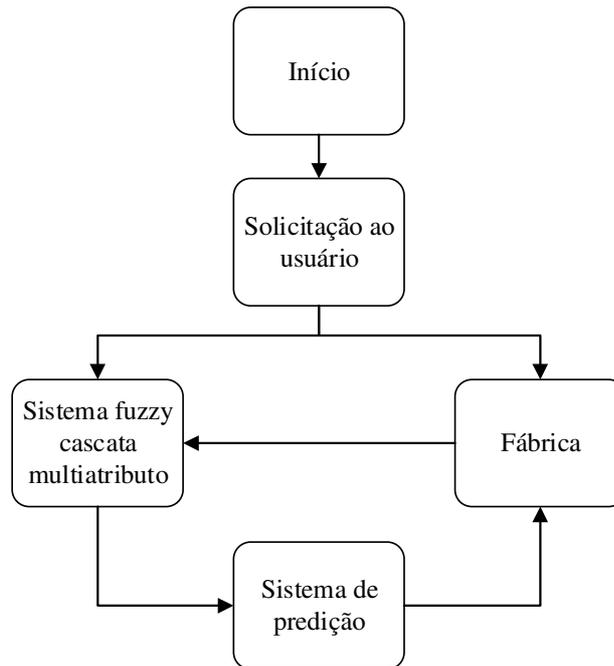


Figura 19 – Fluxograma básico do funcionamento da proposta.

Realizado o mapeamento, são coletadas da fábrica todas as informações referentes a cada produto (distância, entroncamentos, uso de rotas, *buffer* de entrada, *buffer* de saída, *setup*, data de entrega e encadeamento de tarefas). Essas informações são distribuídas como variáveis de entrada nos sistemas *fuzzy* 1, 2 e 3. Cada sistema *fuzzy* fornece uma saída que é ajustada pelo sistema de pesos e se tornam variáveis de entrada no sistema *fuzzy* 4.

O sistema *fuzzy* 4, por sua vez, fornece uma nova saída que é enviada para o sistema de predição que inicia ordenando as tarefas seguindo pela ordem de maior prioridade. Replica-se o estado atual da fábrica no modelo em redes de Petri e é gerado o espaço de estados. São coletadas do espaço de estados as informações de *buffer* de entrada, tempo e transições habilitadas das cinco tarefas melhores posicionadas no ranking em três níveis do espaço de estados.

O sistema *fuzzy* 5 recebe essas novas informações como variáveis de entrada e fornece uma saída que é usada para atualizar o ranking. Em paralelo o sistema predição de AGV realiza a verificação dos AGVs em trânsito sem uma segunda tarefa alocada. Finalizando esses processos, tem-se a decisão de quais tarefas serão atendidas.

Na Figura 20 é ilustrada a representação esquemática do sistema de despacho completo em detalhes.

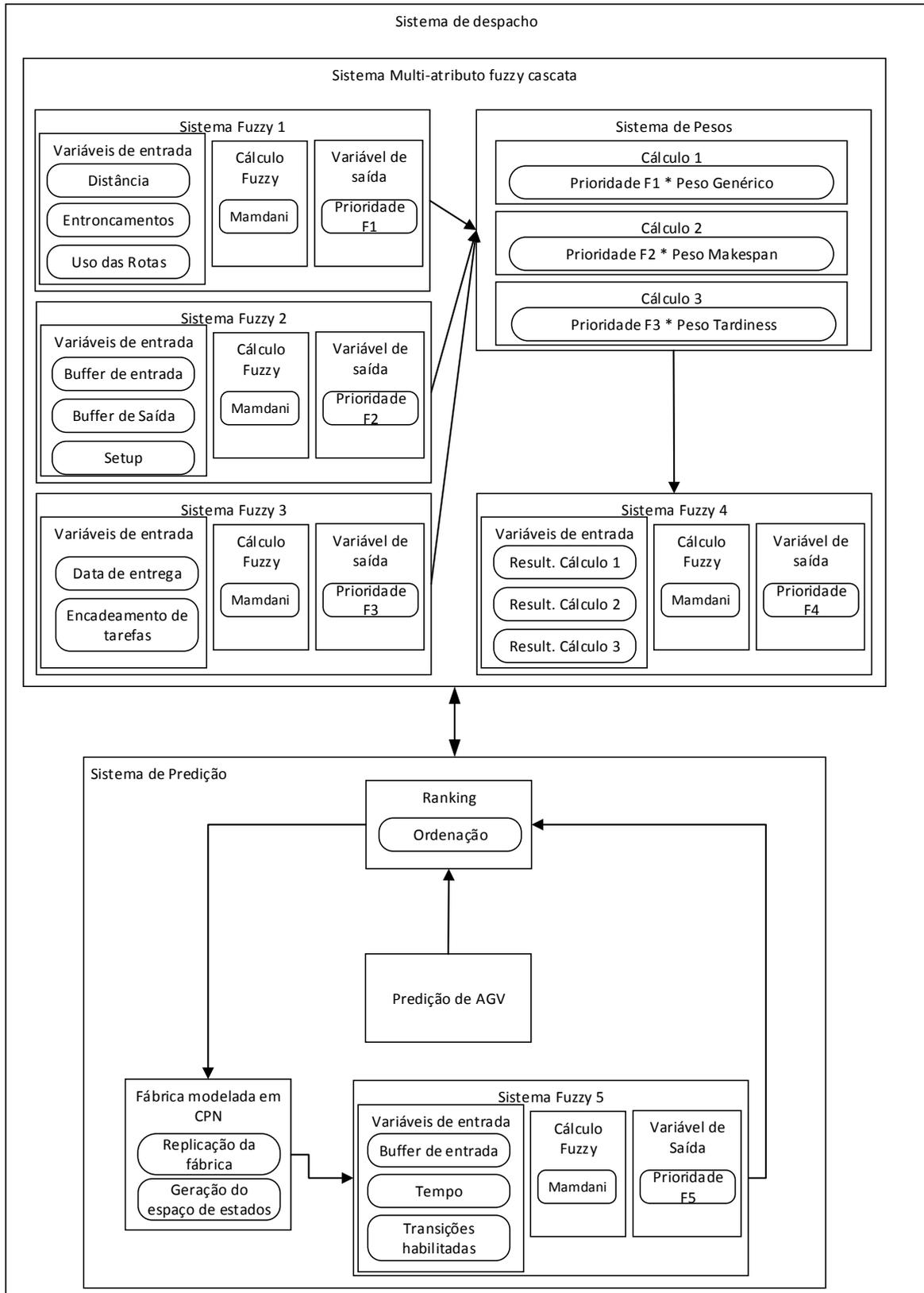


Figura 20 – Esquema da proposta multiatributo fuzzy em conjunto com previsão.

3.3 Considerações finais

A hipótese desta tese de doutorado é a de que incluir outros recursos da fábrica como, o uso das rotas do chão de fábrica e o tempo de *setup* de máquina, por meio de um sistema *fuzzy* em cascata permita alcançar um valor de *makespan* ou *tardiness* menor se comparado com propostas que não consideram os recursos em questão. Conjuntamente, tem-se a hipótese de que uma estrutura baseada em espaço de estados que permita a extração de informações de estados futuros do sistema de produção fabril permita alcançar um valor de *makespan* ou *tardiness* menor se comparado com propostas que não avaliam estados futuros.

Neste capítulo, foi apresentada a proposta de um sistema *fuzzy* em cascata juntamente com a estrutura baseada em espaço de estados aplicados no despacho de AGVs, especificando-se os parâmetros, equações e algoritmos pertinentes.

Pretende-se que, ao final da leitura deste capítulo, o leitor tenha compreendido como esta proposta está definida.

No próximo capítulo é apresentada a forma de implementação da proposta. Alguns aspectos importantes, como por exemplo, *softwares* usados no desenvolvimento, como são realizadas todas as trocas de informações entre os subsistemas, como é replicado o estado atual da fábrica no modelo em redes de Petri, etc., serão explicados e exemplificados.

DESENVOLVIMENTO

4.1 Considerações iniciais

Neste capítulo é apresentado o desenvolvimento da proposta, que consiste na implementação dos sistemas apresentados na proposta. Além disso, não vinculado diretamente à proposta, outros sistemas foram desenvolvidos a fim de fazer com que a proposta se tornasse viável computacionalmente. Conjuntamente, foram definidos dois arranjos físicos de fábrica, um fictício, pautado em publicações, e um inspirado em uma fábrica real, para servir de cenários de ensaio, aplicando o sistema de despacho.

4.2 Desenvolvimento do sistema de despacho de AGVs e dos modelos de fábrica

Para a realização dos testes da proposta, um sistema de despacho de AGV e dois layouts de fábrica foram implementados. Na implementação desses sistemas, foi usada uma fração do Processo Unificado (PU) de desenvolvimento de *software*. Este modelo de desenvolvimento de *software* é iterativo e adaptativo, desta forma consegue produzir um sistema de grande porte como se fossem vários pequenos sistemas, o que diminui o risco do projeto (HIRAMA, 2012).

Na Figura 21 é rerepresentado o esquemático da proposta desta tese, em que se tem o sistema de despacho dividido em dois subsistemas: sistema multiatributo *fuzzy* cascata e sistema de predição. O sistema de despacho realiza o controle da fábrica por meio das decisões de despacho dos AGVs.

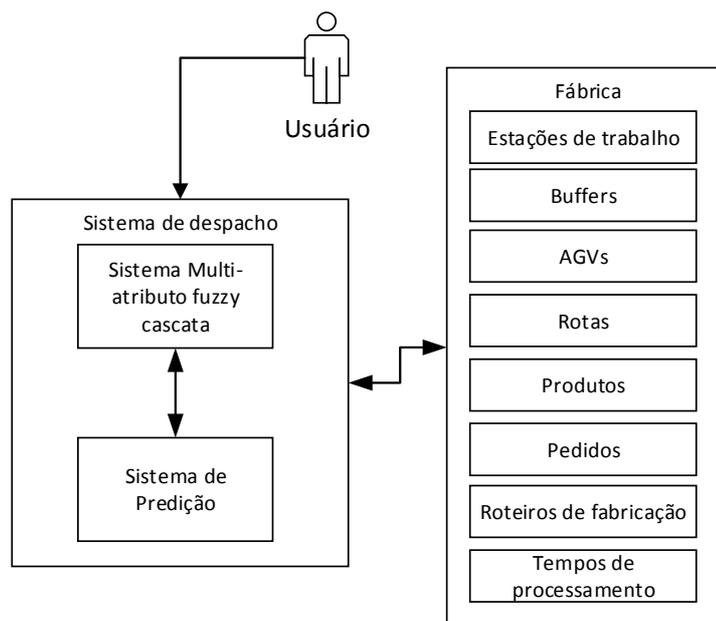


Figura 21 – Proposta da tese conectada ao ambiente de manufatura e com o usuário em um nível maior de detalhes.

Na Figura 22 são rerepresentados os subsistemas em detalhes. Os detalhes que são apresentados na figura são em relação às características de “o que” a proposta da tese realiza.

Para que a proposta seja traduzida em um *software* se faz necessária a implementação de todos os módulos da proposta bem como módulos adicionais de comunicação e cenários de fábrica que serão controlados pelo sistema de despacho. Na Figura 23 são apresentados detalhes com relação às características de “como” a proposta funciona, isto é, o que precisa ser implementado para que a proposta seja viável computacionalmente.

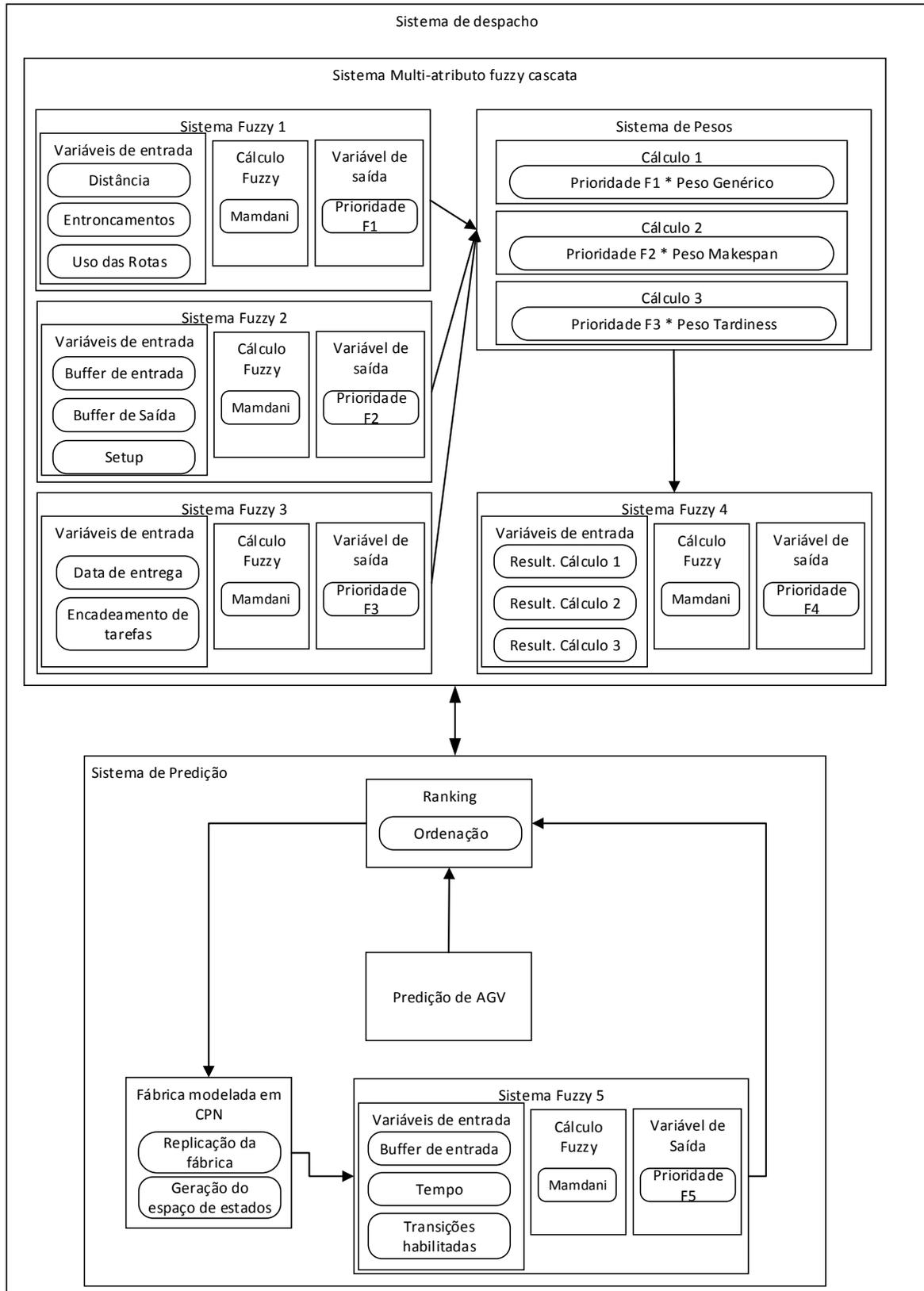


Figura 22 – Esquema da proposta multiatributo fuzzy em conjunto com previsão

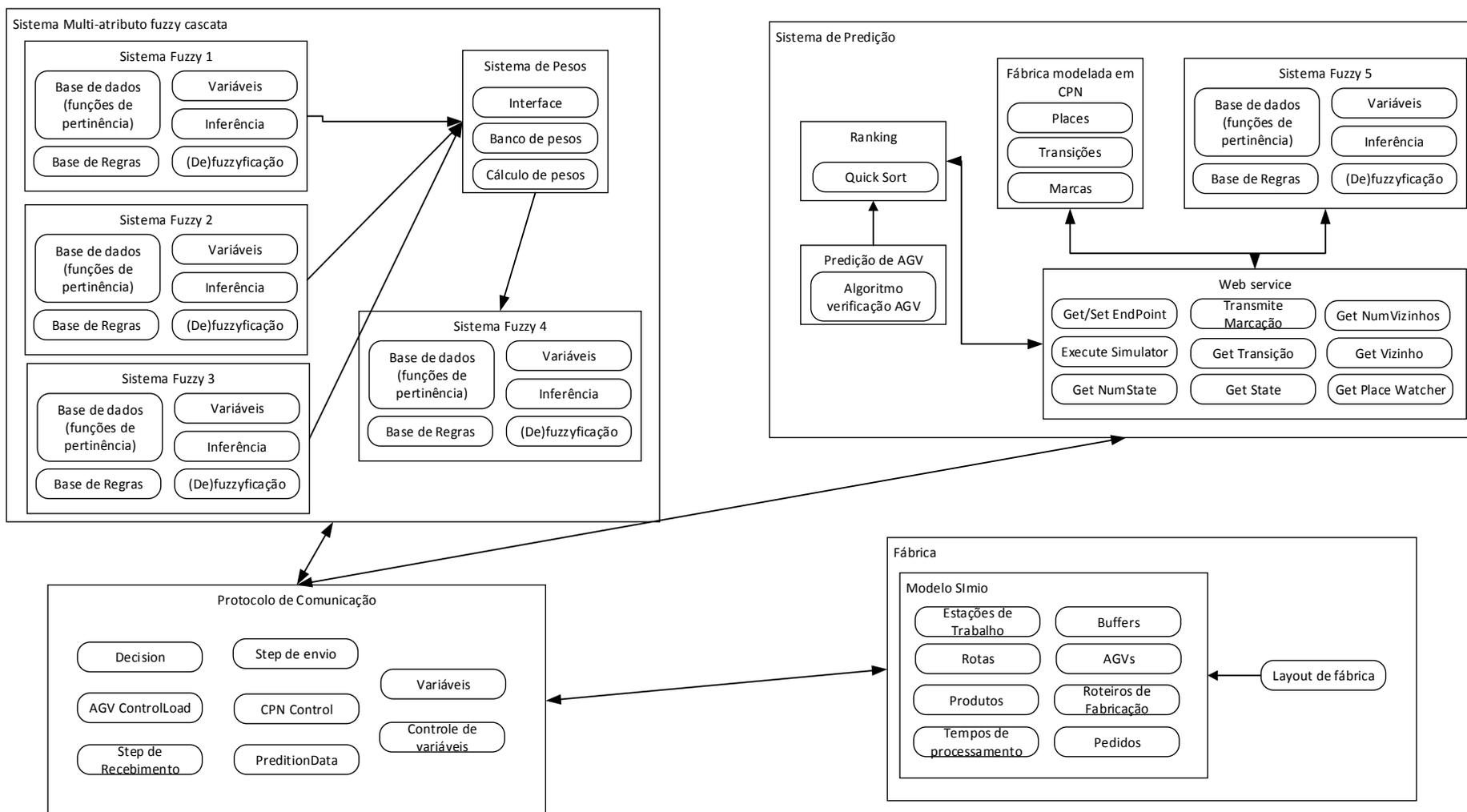


Figura 23 – Esquema dos módulos e suas comunicações para funcionamento e validação da proposta.

As próximas subseções detalham o desenvolvimento e funcionamento de cada módulo apresentado na Figura 23.

4.2.1 Desenvolvimento e características dos cenários de Fábrica

Devido ao fato de não se dispor de fábricas automatizadas disponíveis para implantação e testes do sistema de despacho de AGVs proposto, foram criados dois arranjos físicos de fábrica. Um fictício, pautado nas publicações Benincasa; Morandin; Kato (2003); Morandin et al. (2006); Morandin et al. (2011b) e de menor porte, e outro baseado em um layout de uma indústria moveleira em operação.

Na Figura 24 é apresentado o módulo fábrica, que é um dos módulos apresentados na Figura 23.

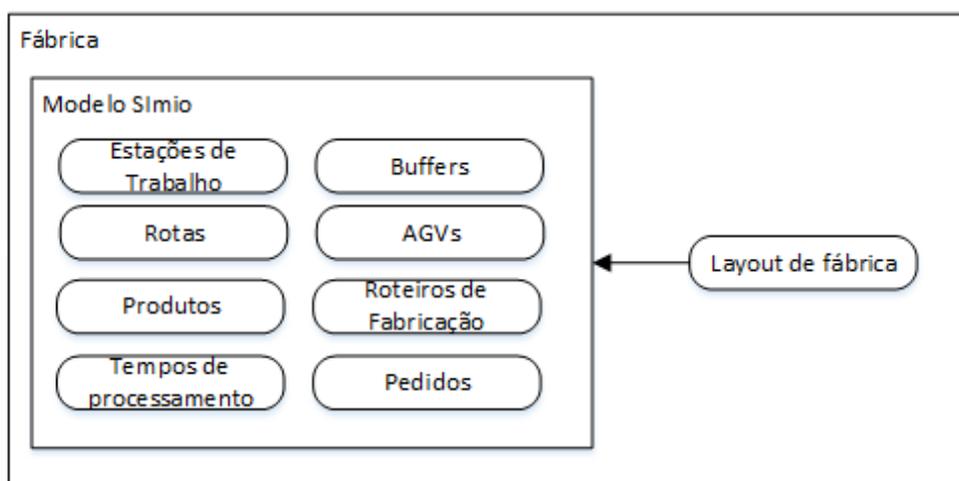


Figura 24 – Módulo fábrica da Figura 23.

O arranjo físico fictício é composto por 6 máquinas, uma estação de carga e uma estação descarga. O arranjo físico da fábrica é apresentado na Figura 25.

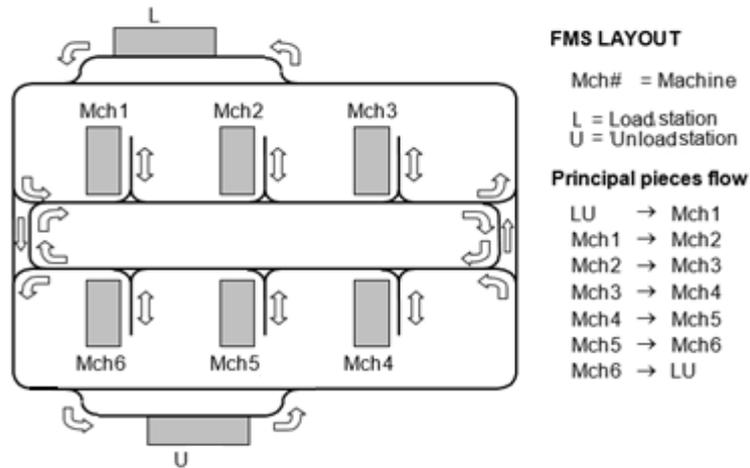
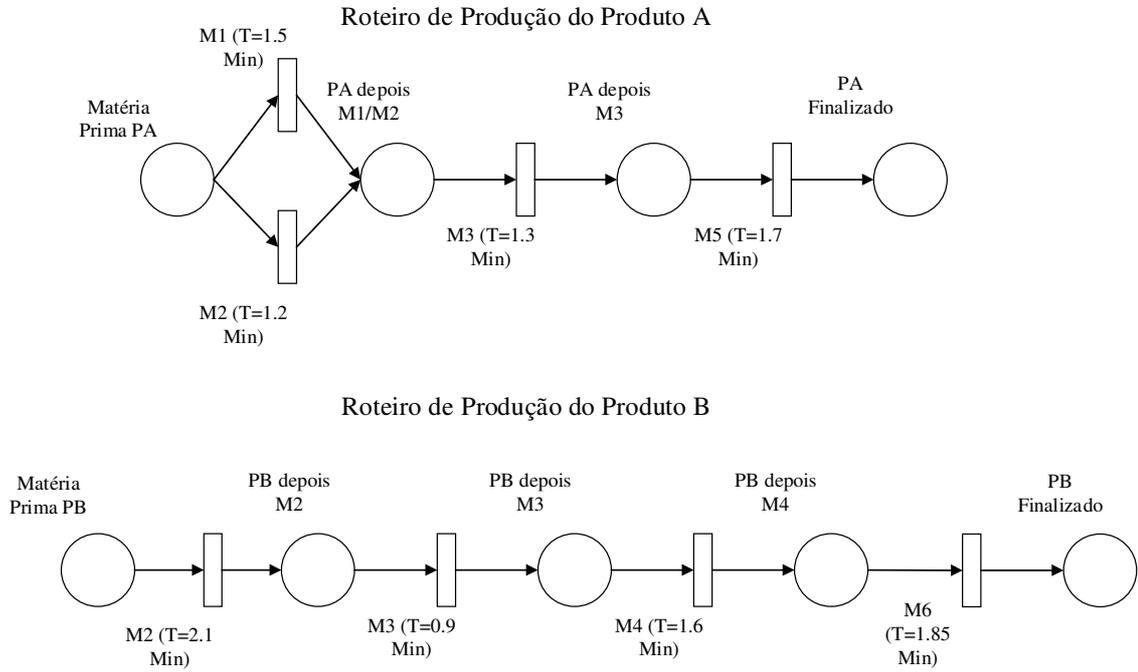


Figura 25 – Arranjo físico da fábrica fictícia (que será chamada de fábrica 1)

Nesta fábrica são produzidos 5 diferentes produtos, sendo que alguns produtos apresentam roteiros alternativos de produção, conforme apresentado na Figura 26.



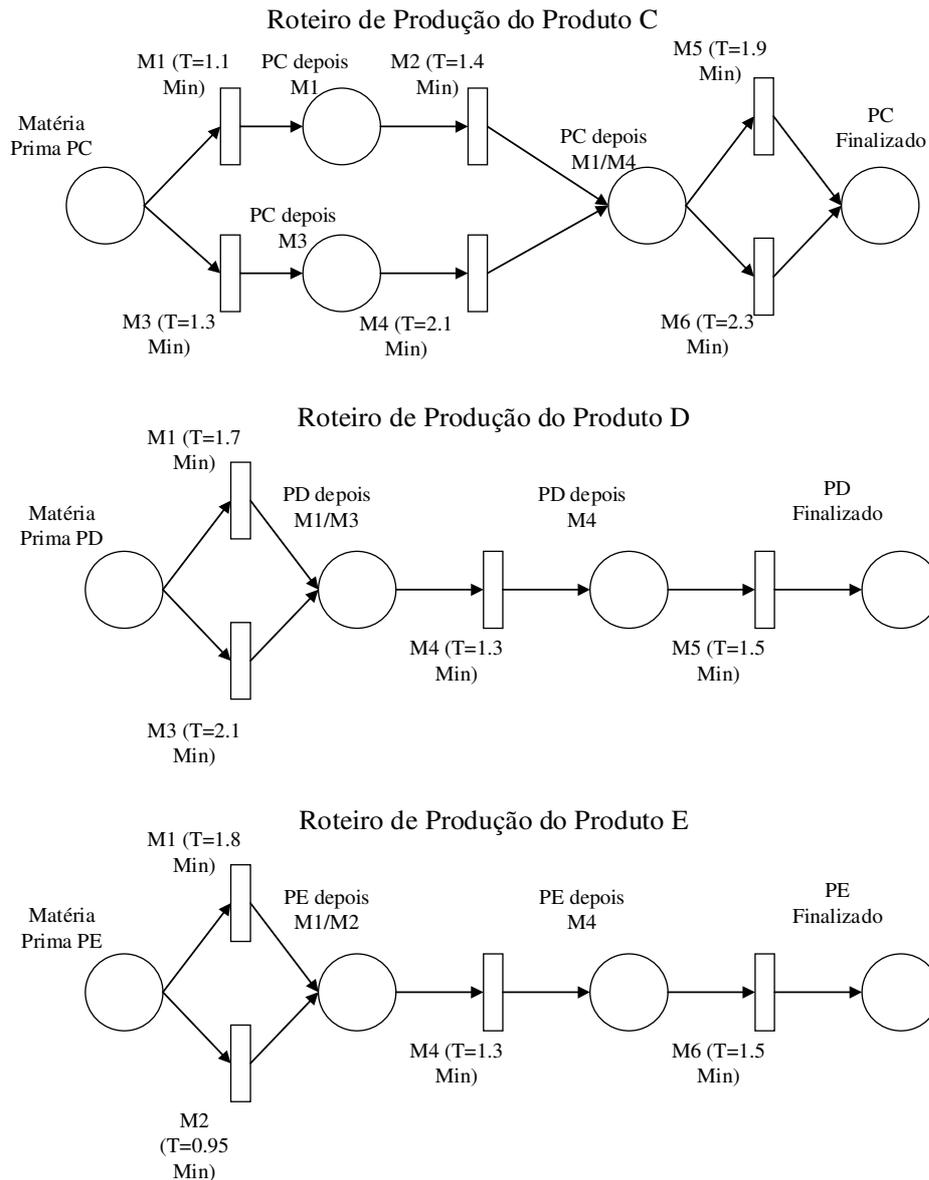


Figura 26 – Roteiros de produção da fábrica 1

Para produzir o produto A, se tem duas opções sendo a primeira passando pelas máquinas 1, 3 e 5; e a segunda pelas máquinas 2, 3 e 5. Para o produto B, o roteiro é: 2, 3, 4 e 6. Para produzir o produto C, as possíveis sequências de máquinas são: 1, 2, e 5; 1, 2 e 6; 3, 4 e 5; 3, 4 e 6. Para o produto D se tem 2 possíveis roteiros: 1, 4 e 5 ou 3, 4 e 5. Por último para o produto E, os roteiros são: 1, 4 e 6 ou 2, 4 e 6.

Para cada produto há uma quantidade de tempo que o mesmo é processado em cada máquina. Esta quantidade de tempo para cada máquina por produto também está representada na Figura 26.

Devido ao fato do laboratório Tear ter contato direto com empresas do ramo moveleiro, propôs-se a criação de um arranjo físico que caracteriza uma indústria desse segmento. No entanto, este arranjo físico pode ser estendido sem muitas modificações a outros setores industriais, uma vez consideradas as muitas semelhanças com outros setores, tais como a indústria metal-mecânica.

Ademais, de acordo com Ferreira et al. (2009), a indústria moveleira pode ser considerada uma das mais antigas do mundo. Os avanços proporcionados pela industrialização permitiram a padronização e os ganhos de economia de escala, de maneira que os móveis deixaram de ser produtos artesanais para se tornarem produtos manufaturados. Sendo assim, a indústria de móveis é classificada como uma indústria tradicional, com tecnologia de produção consolidada e amplamente difundida.

Ferreira et al. (2009) ainda destacam que, no Brasil, esse setor correspondia, em 2005, a 6,1% do total de indústrias, sendo predominada por empresas de pequeno porte. Em 2006, o comércio mundial de móveis movimentou cerca de US\$ 270,5 bilhões.

Ainda segundo Ferreira et al. (2009), o setor moveleiro foi mais eficaz que a média das indústrias para a redução dos custos operacionais, mas teve um desempenho pior na busca pela elevação da produtividade do trabalho e na agregação de valor ao produto. Destaca-se que, na maior parte dos pólos, as empresas ainda não foram capazes de, em conjunto, enriquecer a localidade com novas capacitações técnicas, gerenciais e produtivas que permitissem inovações e que agregassem maior valor ao móvel brasileiro, além de ganhos de escala e de flexibilização produtiva.

Definiu-se, portanto, o arranjo físico prioritariamente do tipo funcional, ou *job-shop*, devido principalmente ao volume e à variedade de peças do setor moveleiro. Cita-se uma grande variedade de empresas dispostas funcionalmente, porém que não chegam a ser consideradas como sistemas flexíveis de manufatura, principalmente por não terem o seu processo produtivo totalmente automatizado, porém o arranjo físico construído será considerado FMS.

Em geral, uma fábrica moveleira é composta por diversos setores que correspondem aos seus principais processos produtivos, sem haver, no entanto, uma regra universal para todas as indústrias do setor. O arranjo físico é composto de quatro setores, ou células, de produção principais, cada um deles responsáveis por uma grande fase do setor de produção moveleiro. São definidos alguns processos de produção para cada um dos setores, conforme descrito na

Tabela 3. Para cada um dos processos de produção, foi alocada uma quantidade definida de máquinas que representa um esboço do arranjo físico. Os setores são: S1 – Desdobro; S2 – Usinagem; S3 – Pré-acabamento; S4 – Acabamento final.

Tabela 3 – Setores e Processos.

Setores	Processos		Setores	Processos	
S1	P1	Cortar Comprimento	S3	P13	Lixação
	P2	Desengrossar		P14	Pintura
	P3	Plainar 4 faces		P15	Secagem
	P4	Sarrafeiar	S4	P16	Lixação Fina
	P5	Respigar		P17	Montagem
P6	Fresamento de canal	P18		Embalagem	
P7	Torneamento				
P8	Torneamento				
S2	P9	Furação Múltipla			
	P10	Furação Simples			
	P11	Fresamento Superior			
	P12	Fresamento Inferior			

Entre os setores de produção da fábrica são usados *buffers* intermediários, o que flexibiliza os fluxos de produção e de AGVs, balanceando o volume e a variedade de peças dentro da mesma.

Três famílias de produtos podem ser produzidas na fábrica, sendo elas: cadeira, mesa e cama. Para cada um dos produtos são definidas peças componentes e a quantidade necessária de cada peça para a produção de uma unidade dos mesmos, conforme apresentado na Tabela 4.

Tabela 4 – Produtos e Peças.

Cadeira		Mesa		Cama	
Quant.	Peças	Quant.	Peças	Quant.	Peças
2	Pés da frente	4	Pés	2	Pés da frente
2	Pés de trás	4	Travessas	2	Pés de Trás
1	Encosto	4	Travas	1	Cabeceira
1	Assento	1	Tampo	1	Peseira
4	Travessas			2	Barra da cama
4	Travas			4	Travessa-estrado
				10	Ripas-estrado

Ainda, estabeleceu-se a existência de cinco tipos de cada produto, diferenciados pela composição de diferentes peças, como pode ser visto na Tabela 5, bem como cinco possibilidades de cores, totalizando vinte e cinco modelos possíveis para cada um dos produtos.

Tabela 5 – Modelos de Produtos.

Cadeira 1	Cadeira 2	Cadeira 3	Cadeira 4	Cadeira 5
Pés da frente				
Pés de trás				
Encosto 1	Encosto 2	Encosto 3	Encosto 1	Encosto 2
Assento 1	Assento 2	Assento 1	Assento 2	Assento 1
Travessas	Travessas	Travessas	Travessas	Travessas
Travas	Travas	Travas	Travas	Travas

Mesa 1	Mesa 2	Mesa 3	Mesa 4	Mesa 5
Pés	Pés	Pés	Pés	Pés
Travessas 1	Travessas 2	Travessas 3	Travessas 4	Travessas 5
Travas	Travas	Travas	Travas	Travas
Tampo 1	Tampo 2	Tampo 3	Tampo 4	Tampo 5

Cama 1	Cama 2	Cama 3	Cama 4	Cama 5
Pés da frente				
Pés de trás				
Cabeceira 1	Cabeceira 2	Cabeceira 3	Cabeceira 4	Cabeceira 5
Peseira 1	Peseira 2	Peseira 3	Peseira 4	Peseira 5
Barra da cama				
Travessa-estrado	Travessa-estrado	Travessa-estrado	Travessa-estrado	Travessa-estrado
Ripas-estrado	Ripas-estrado	Ripas-estrado	Ripas-estrado	Ripas-estrado

Na Figura 27 pode ser visto o esboço do arranjo físico, no qual, *IN* e *OUT* correspondem respectivamente aos *buffers* de entrada e saída da fábrica e EIs correspondem aos *buffers* intermediários. No total, dezoito processos produtivos são definidos, P1-P18 – conforme nomeação apresentada na Tabela 3, entre os quatro setores da fábrica e um total de setenta máquinas, dispostas entre os processos. Na Tabela 6 são exibidos os tipos de máquinas que compõem cada processo, além das peças que podem ser produzidas em cada tipo de máquina, enquanto na Tabela 7 é exposta uma parte dos roteiros das peças/produtos.

Tabela 6 – Processos, máquinas e peças.

	Processos	Máquina	Peças
P1	Cortar Comprimento	Esquadrejadeira	Todas as peças
P2	Desengrossar	Desengrossadeira	Todos os pés / travessas / travas / tampos / cabeceiras / peseiras / barras da cama
P3	Plainar 4 faces	Plaina de 4 faces	Todos os pés / travessas / travas / tampos / cabeceiras / peseiras / barras da cama
P4	Sarrafeiar	Tupia	Todos os pés / travessas / travas / tampos / cabeceiras / peseiras / barras da cama
P5	Respigar	Respigadeira	Travessas / travas / barras da cama / todos os pés
P6	Fresamento de canal	Fresadora horizontal	Todos os pés / cabeceira / peseiras
P7	Torneamento	Torno Copiador	Cabeceiras / pés de cadeira
P8	Torneamento	Torno CNC	Cabeceiras / pés de cadeiras
P9	Furação Múltipla	Furadeira Múltipla	Cabeceiras / encostos
P10	Furação Simples	Furadeira Vertical	Peseiras / barras de cama
P11	Fresamento Superior	Fresadora interpolada	Cabeceiras / encostos
P12	Fresamento Inferior	Fresadora pneumática	Cabeceiras / encostos
P13	Lixação	Lixadeira vertical	Todas as peças
P14	Pintura	Cabine de pintura	Todas as peças
P15	Secagem	Estufa	Todas as peças
P16	Lixação Fina	Lixadeira pneumática	Todas as peças
P17	Montagem	Estação de montagem	Todas as peças
P18	Embalagem	Estação de embalagem	Todas as peças

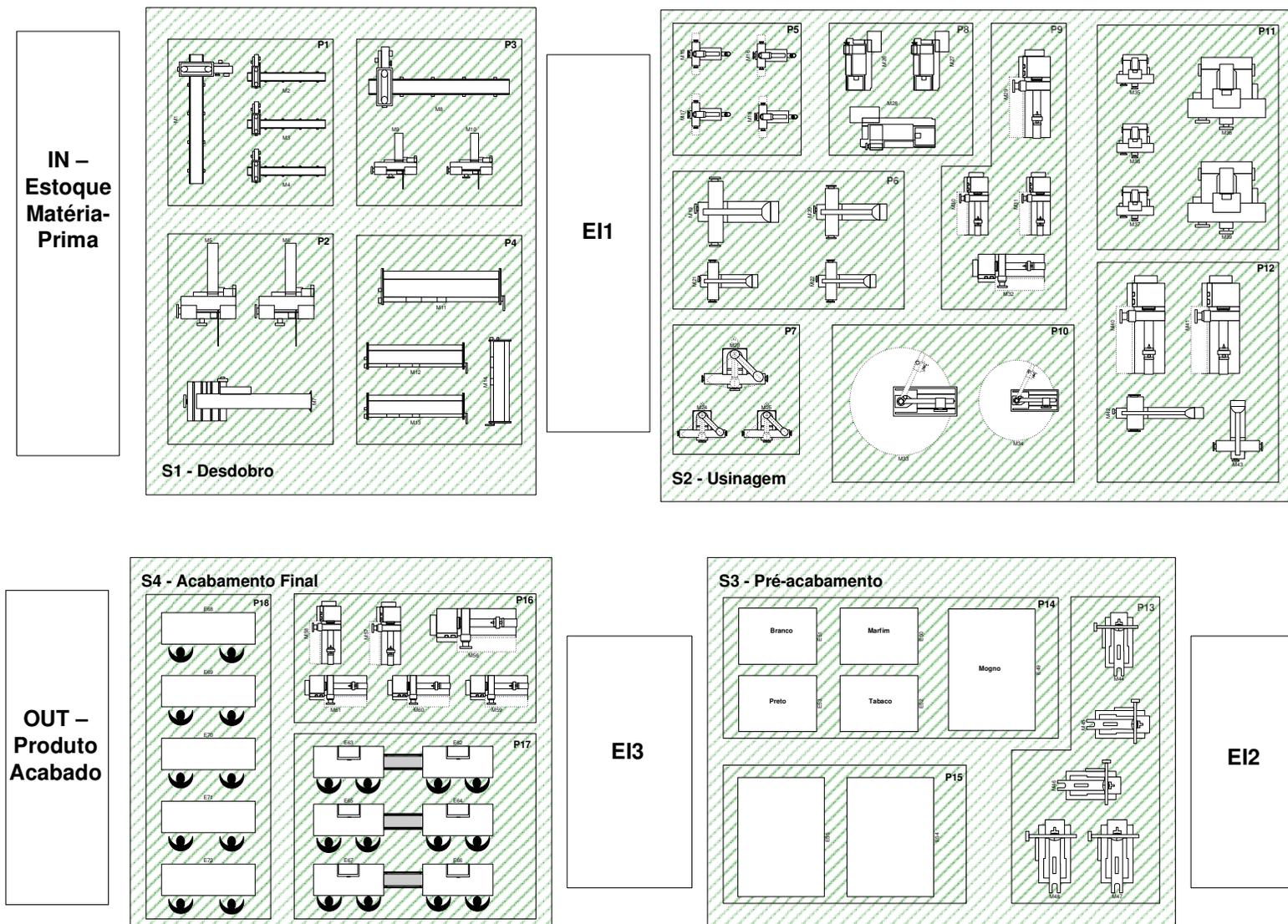


Figura 27 – Arranjo físico da fábrica inspirada em uma indústria moveleira em operação (que será chamada de fábrica 2)

Tabela 7 – Roteiros dos Produtos (Processos e Máquinas) da fábrica 2

Produto	Peças	Roteiro (processos necessários)								
COMUNS	CADEIRA	Pés da frente - cadeira	P1 (M3 M4)	P2 (M5 M6)	P3 (M9 M10)	P4 (M13 M14)	EI1	P5 (M15 M16 M17 M18)	P6 (M21 M22)	...
		Pés de trás - cadeira	P1 (M3 M4)	P2 (M5 M6)	P3 (M9 M10)	P4 (M13 M14)	EI1	P5 (M15 M16 M17 M18)	P6 (M21 M22)	...
		Travessas - cadeira	P1 (M3 M4)	P2 (M5 M6)	P3 (M9 M10)	P4 (M13 M14)	EI1	P5 (M15 M16 M17 M18)	EI2	...
		Travas - cadeira	P1 (M3 M4)	P2 (M5 M6)	P3 (M9 M10)	P4 (M13 M14)	EI1	P5 (M15 M16 M17 M18)	EI2	...
	MESA	Pés - mesa	P1 (M3 M4)	P2 (M5 M6)	P3 (M9 M10)	P4 (M13 M14)	EI1	P5 (M15 M16 M17 M18)	P6 (M21 M22)	...
		Travas - mesa	P1 (M3 M4)	P2 (M5 M6)	P3 (M9 M10)	P4 (M13 M14)	EI1	P5 (M15 M16 M17 M18)	EI2	...
	CAMA	Pés da frente - cama	P1 (M3 M4)	P2 (M5 M6)	P3 (M9 M10)	P4 (M13 M14)	EI1	P5 (M15 M16 M17 M18)	P6 (M21 M22)	...
		Pés de trás - cama	P1 (M3 M4)	P2 (M5 M6)	P3 (M9 M10)	P4 (M13 M14)	EI1	P5 (M15 M16 M17 M18)	P6 (M21 M22)	...
		Barra da cama	P1 (M3 M4)	P2 (M5 M6)	P3 (M9 M10)	P4 (M13 M14)	EI1	P5 (M15 M16 M17 M18)	P10 (M34)	...
		Travessa-estrado	P1 (M3 M4)	P2 (M5 M6)	P3 (M9 M10)	P4 (M13 M14)	EI1	P5 (M15 M16 M17 M18)	EI2	...
		Ripas-estrado	P1 (M3 M4)	P2 (M5 M6)	P3 (M9 M10)	P4 (M13 M14)	EI1	P5 (M15 M16 M17 M18)	EI2	...
	DIFERENCIAIS	CADEIRA	Encosto 1	P1 (M1 M2)	EI1	P9 (M30)	P11 (M35)	P12 (M42 M43)	EI2	P13 (M47)
Encosto 2			P1 (M1 M2)	EI1	P9 (M31)	P11 (M36)	P12 (M43)	EI2	P13 (M47)	...
Encosto 3			P1 (M1 M2)	EI1	P9 (M32)	P11 (M37)	P12 (M43)	EI2	P13 (M47)	...
Assento 1			P1 (M2)	EI2	P13 (M47)	P14 (E49 E50 E51 E52 E53)	P15 (E54)	EI3	P16 (M59 M60 M61)	...
Assento 2			P1 (M2)	EI2	P13 (M47)	P14 (E49 E50 E51 E52 E53)	P15 (E54)	EI3	P16 (M59 M60 M61)	...
MESA		Travessas 1 - mesa	P1 (M3 M4)	P2 (M5 M6)	P3 (M9 M10)	P4 (M14)	EI1	P5 (M15 M16 M17 M18)	EI2	...
		Travessas 2 - mesa	P1 (M3 M4)	P2 (M5 M6)	P3 (M9 M10)	P4 (M14)	EI1	P5 (M15 M16 M17 M18)	EI2	...
		Travessas 3 - mesa	P1 (M3 M4)	P2 (M5 M6)	P3 (M9 M10)	P4 (M14)	EI1	P5 (M15 M16 M17 M18)	EI2	...

Para efeito da criação do arranjo físico não foram consideradas as dimensões da fábrica, o mesmo ocorreu para as dimensões das máquinas. As máquinas apresentadas na Figura 27 possuem caráter ilustrativo, não representando as respectivas máquinas de cada processo.

4.2.2 Sistema multiatributo *fuzzy* cascata

Nesta subseção é apresentado o desenvolvimento da simulação dos cenários de fábrica no software Simio. Também é apresentado o desenvolvimento e forma de funcionamento dos sistemas *fuzzy*. Uma vez que os sistemas *fuzzy* necessitam dos dados da fábrica para realizar os cálculos, a criação de *steps* durante o desenvolvimento da simulação é um fator crucial para permitir a comunicação da fábrica com o sistema multiatributo *fuzzy* cascata. Na Figura 28 são apresentados os módulos que serão descritos nessa subseção e que fazem parte da Figura 23.

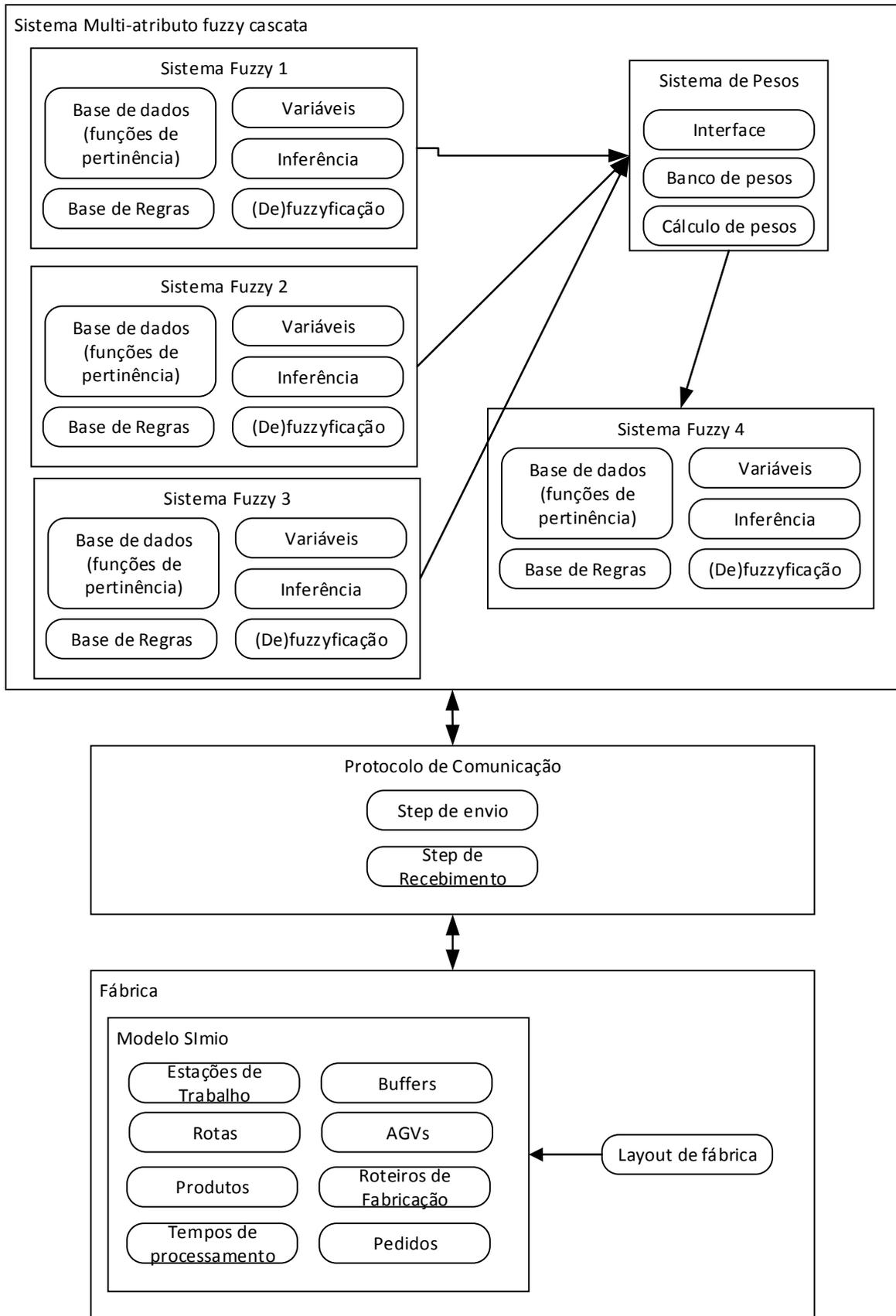


Figura 28 – Módulos da Figura 23 que serão apresentados em detalhes nas subseções a seguir.

4.2.2.1 Implementação dos cenários de fábrica no *software* Simio

O uso dos *softwares* de simulação possibilita avaliar os impactos de mudanças nos processos produtivos, testar diferentes configurações na alocação de recursos limitados, determinar parâmetros das operações, validar novos projetos, entre outras aplicações. Porém, existe uma grande diversidade de *softwares* de simulação de eventos discretos disponíveis.

Assim uma das primeiras etapas desenvolvidas foi um estudo visando comparar oito *softwares* distintos (AutoMod, Arena, Simul8, Gazebo, Stage, Simio, FlexSim e ProModel) em aspectos considerados importantes para a aplicação dos mesmos na modelagem da indústria: visualização da simulação, custos, comunicação, construção do modelo, otimização do modelo, apresentação dos resultados. O *software* que apresentou melhores resultados e que, portanto, foi escolhido para uso neste trabalho, bem como para o laboratório Tear, foi o Simio. Neste trabalho a versão usada foi a 7.119.12241.

Nas Figura 29 e Figura 30 é ilustrada a fábrica 1 modelada no Simio. Nas Figura 31 e Figura 32 é ilustrada a fábrica 2 modelada no Simio. Por meio das imagens é possível verificar a disposição dos recursos e sentido das rotas.

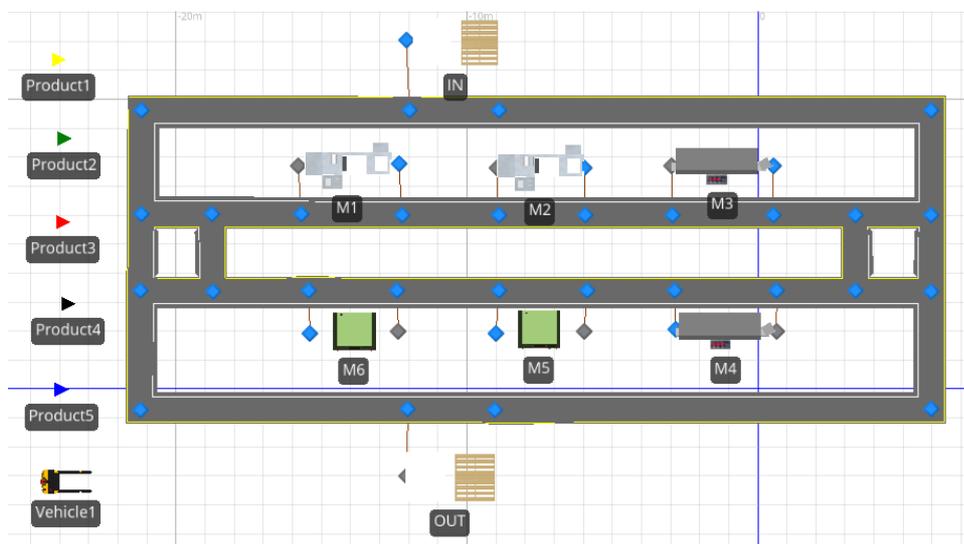


Figura 29 – Visualização em 2D da fábrica 1 modelada no *software* SIMIO.

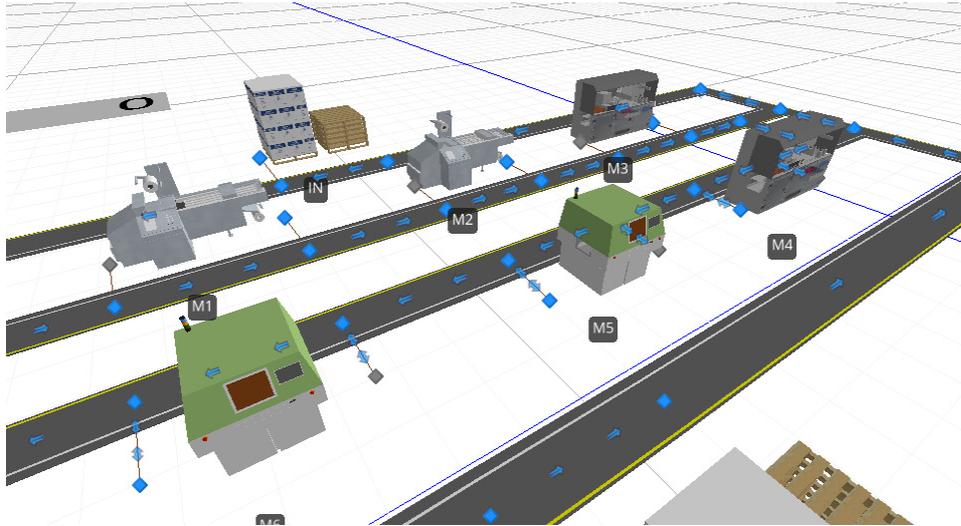


Figura 30 – Visualização em 3D da fábrica 1 modelada no software SIMIO.

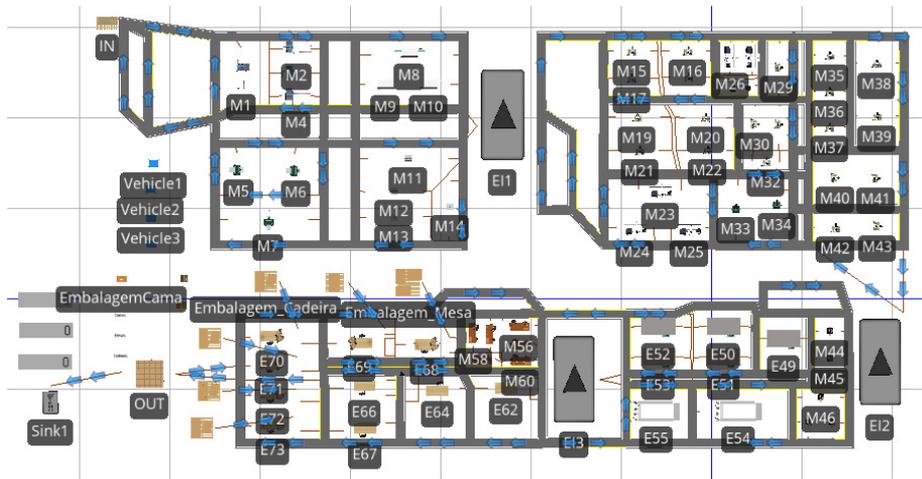


Figura 31 – Visualização em 2D da fábrica 2 modelada no software SIMIO.

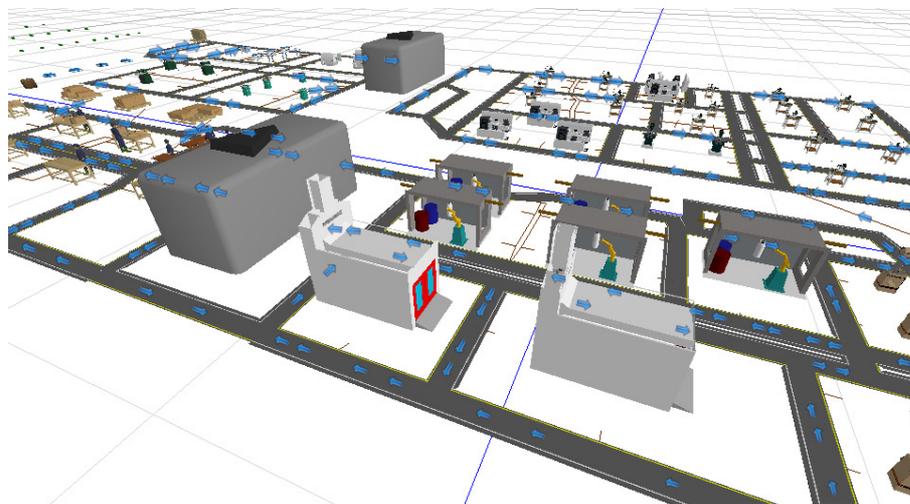


Figura 32 – Visualização em 3D da fábrica 2 modelada no software SIMIO.

Após a criação da parte física da simulação, são necessárias as configurações da parte lógica. Primeiramente são necessários os cadastros dos itens que serão produzidos na fábrica, como exemplificado na Figura 33.

Posteriormente são cadastrados os roteiros de produção de cada item bem como os tempos de processamento de cada etapa de produção e os tempos de setup. Essa configuração é ilustrada na Figura 34.

Subsequentemente é necessário realizar configurações em todos os recursos da fábrica. Como no exemplo apresentado na Figura 35, para a estação de trabalho M8 são configurados os tempos de setup e os tempos de processamento, que são vinculadas as tabelas de dados anteriormente criadas, a capacidade dos *buffers* de entrada e de saída.

Por fim são criados os processos para auxiliar o funcionamento lógico da fábrica. É possível modificar o comportamento de um objeto por meio de "*processes*" (processos). Os processos oferecem uma gama de possibilidades permitindo estender e expandir a capacidade de modelagem nativa do SIMIO. Os processos são criados por meio de "*steps*" (passos). Existem passos padrões do SIMIO e passos que podem ser desenvolvidos no *Microsoft visual studio*, na linguagem C#, usando a API disponibilizada pelo SIMIO. Na Figura 36 são apresentados alguns processos da modelagem da fábrica real.

Por meio do desenvolvimento de passos é que se torna possível a comunicação do Simulação com *softwares* externos, possibilitando assim recuperar as informações necessárias da fábrica para realizar os cálculos *fuzzy* e a recuperação do estado atual da fábrica e replicá-lo no modelo em redes de Petri. Mais detalhes da comunicação estão presentes na seção 4.2.4, intitulada protocolo de comunicação.

The screenshot shows the SIMIO software interface for configuring data for manufactured items. The main window displays a table of manufacturing orders for the item 'pes_da_frente_cadeira'. The table has the following columns: Entity, Step, Node List, SetupTime (Minutes), and ProcessTime (Minutes). The data is as follows:

Entity	Step	Node List	SetupTime (Minutes)	ProcessTime (Minutes)
pes_da_frente_...	1	List_M3M4	5.0	0.5
pes_da_frente_...	2	List_M5M6	2.0	0.5
pes_da_frente_...	3	List_M9M10	10.0	0.5
pes_da_frente_...	4	List_M13M14	5.0	0.5
pes_da_frente_...	5	List_EI1	0.0	0.0
pes_da_frente_...	6	List_M15M16M17M18	2.0	0.5
pes_da_frente_...	7	List_M21M22	2.0	0.3
pes_da_frente_...	8	List_M24M25	2.0	0.2
pes_da_frente_...	9	List_M26M27	2.0	0.5
pes_da_frente_...	10	List_EI2	0.0	0.0
pes_da_frente_...	11	List_M45M46	5.0	0.5
pes_da_frente_...	12	List_E49E50E51E5...	15.0	0.2
pes_da_frente_...	13	List_E55	0.0	120.0

The right sidebar shows the 'Browse: FMS: ManufacturedItems' tree with the following structure:

- Navigation: FMS
 - FMS v1 75 buffer10
 - ModelEntity
 - FMS
 - Montadora_Mesa
 - Montadora_Cadeira

The 'Properties: ManufacturedItems (Table)' panel shows the following options:

- Show Commonly Used Properties Only
- Advanced Options
 - General
 - Name: ManufacturedIt
 - Description:

Figura 33 – Configuração de dados do SIMIO, cadastro dos itens que a fábrica produz.

The screenshot displays a software interface with a central table and a right-hand sidebar. The table is bound to an Excel file and contains manufacturing data. The sidebar includes a navigation tree and a properties panel for the selected table.

	Entity	Step	Node List	SetupTime (Minutes)	ProcessTime (Minutes)
1	pes_da_frente_...	1	List_M3M4	5.0	0.5
2	pes_da_frente_...	2	List_M5M6	2.0	0.5
3	pes_da_frente_...	3	List_M9M10	10.0	0.5
4	pes_da_frente_...	4	List_M13M14	5.0	0.5
5	pes_da_frente_...	5	List_EI1	0.0	0.0
6	pes_da_frente_...	6	List_M15M16M17M18	2.0	0.5
7	pes_da_frente_...	7	List_M21M22	2.0	0.3
8	pes_da_frente_...	8	List_M24M25	2.0	0.2
9	pes_da_frente_...	9	List_M26M27	2.0	0.5
10	pes_da_frente_...	10	List_EI2	0.0	0.0
11	pes_da_frente_...	11	List_M45M46	5.0	0.5
12	pes_da_frente_...	12	List_E49E50E51E5...	15.0	0.2
13	pes_da_frente_...	13	List_E55	0.0	120.0
14	pes_da_frente_...	14	List_EI3	0.0	0.0
15	pes_da_frente_...	15	List_M57M58	1.0	0.5
16	pes_da_frente_...	16	List_E64E65_PesF...	1.0	5.0
17	pes_de_tras_ca...	1	List_M3M4	5.0	0.75
18	pes_de_tras_ca...	2	List_M5M6	2.0	0.75
19	pes_de_tras_ca...	3	List_M9M10	10.0	0.75
20	pes_de_tras_ca...	4	List_M13M14	5.0	0.75

The right-hand sidebar contains the following elements:

- Browse: FMS : PartRoutings**
- Navigation: FMS
- FMS v1.75 buffer 10
- ModelEntity
- FMS
- Montadora_Mesa
- Montadora_Cadeira
- Properties: PartRoutings (Table)
- Show Commonly Used Properties Only
- Advanced Options
- General

Name	PartRoutings
Description	
- Advanced Options

Figura 34 – Configuração de dados do SIMIO, cadastro do roteiro de produção dos itens a serem produzidos e tempos de processamento e setup.

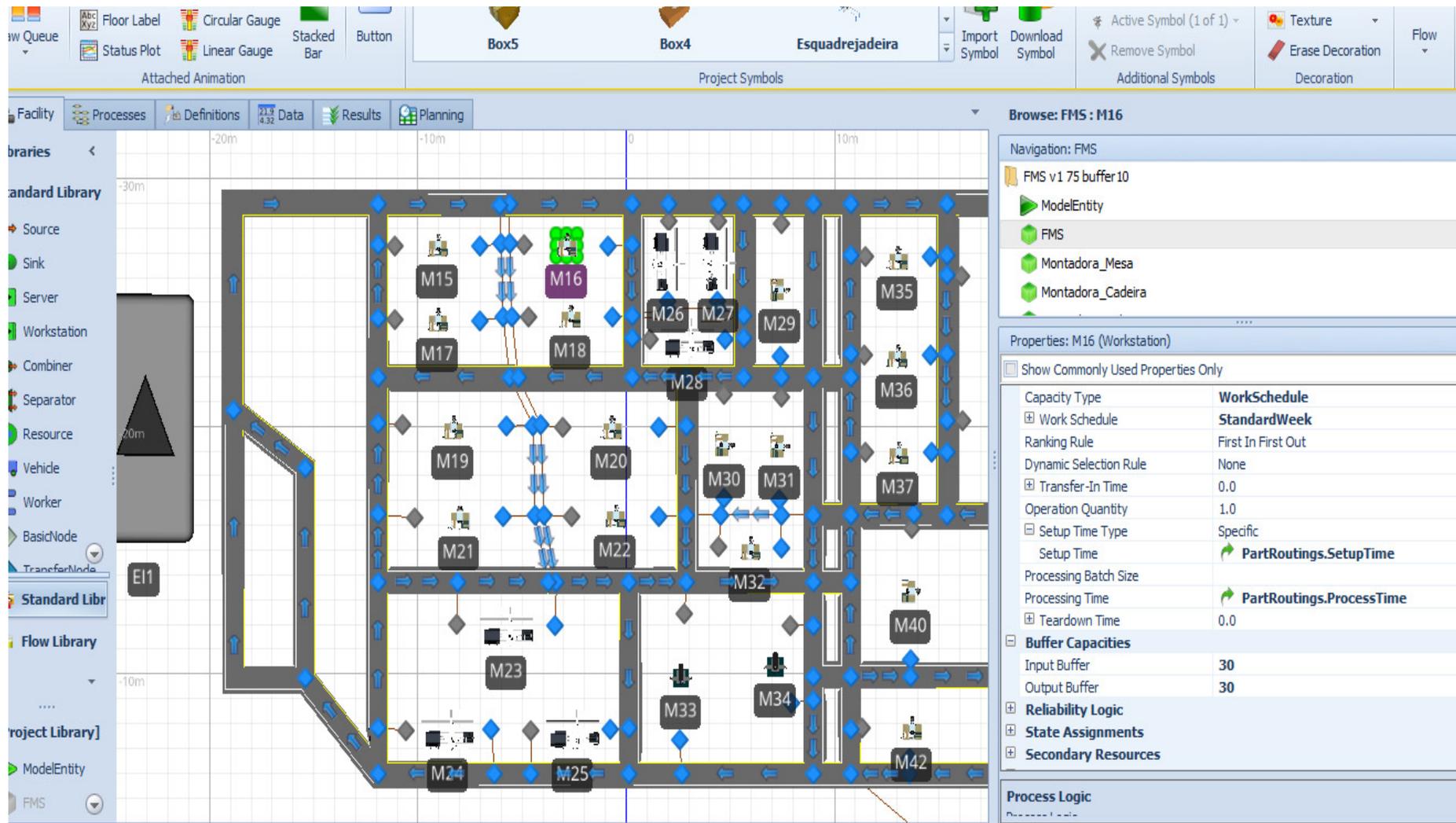


Figura 35 – Configuração dos recursos da fábrica.

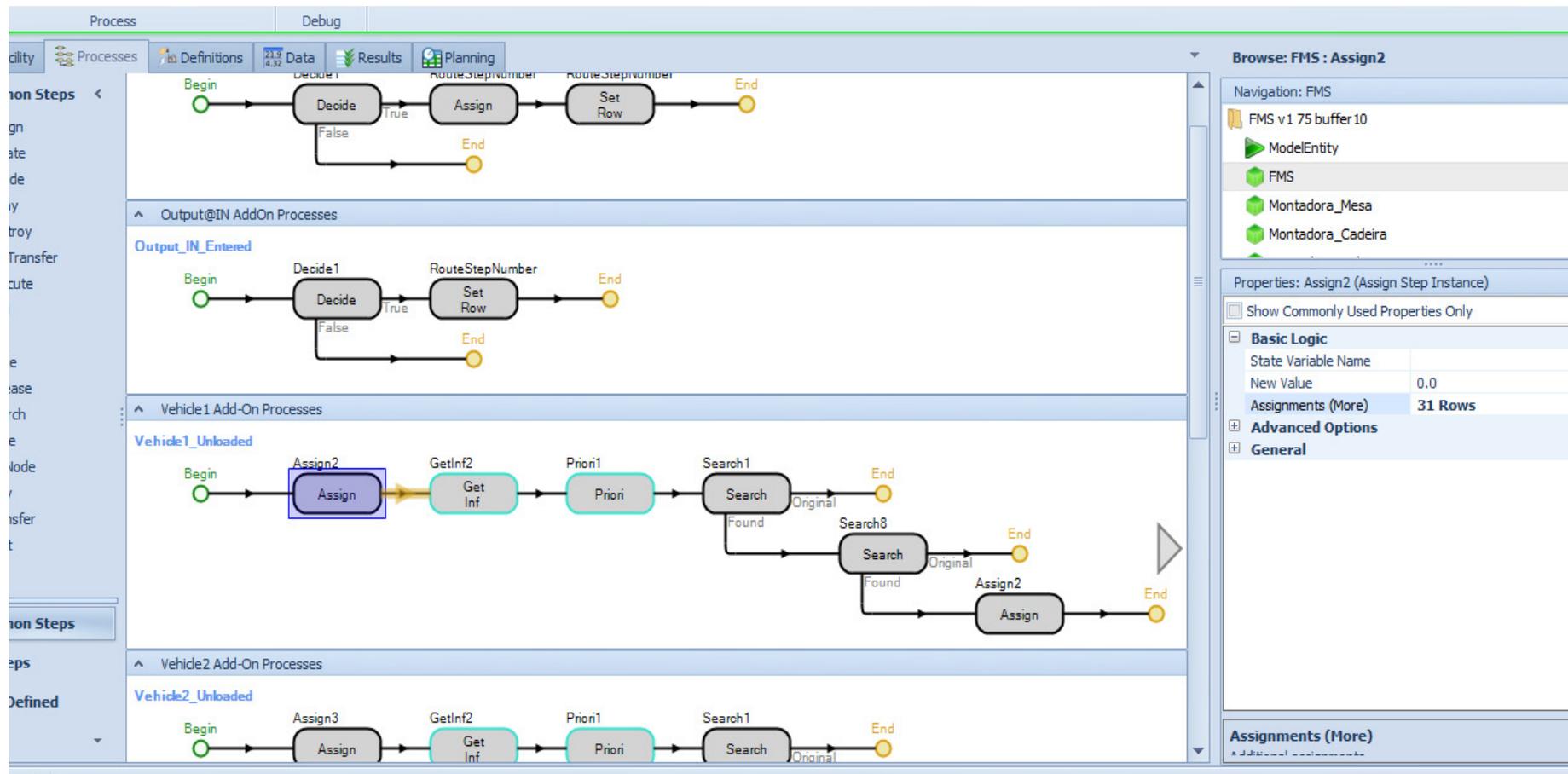


Figura 36 – Criação de processos e steps.

4.2.2.2 Implementação dos sistemas *fuzzy*

Para implementação dos sistemas *fuzzy* que compõem o sistema multiatributo *fuzzy* em cascata foi usado o toolbox *fuzzy* do *software* MatLab versão R2012a.

Primeiramente cria-se a estrutura do sistema *fuzzy*, inserindo as variáveis de entrada, de saída, métodos de inferência, agregação, conversão escalar/*fuzzy* e conversão *fuzzy*/escalar. Na Figura 37 é apresentada a estrutura geral do sistema *fuzzy* 1, em que se tem três variáveis de entrada (distância, entroncamentos e uso de rotas), e uma de saída (prioridade), como descrito anteriormente no capítulo de proposta.

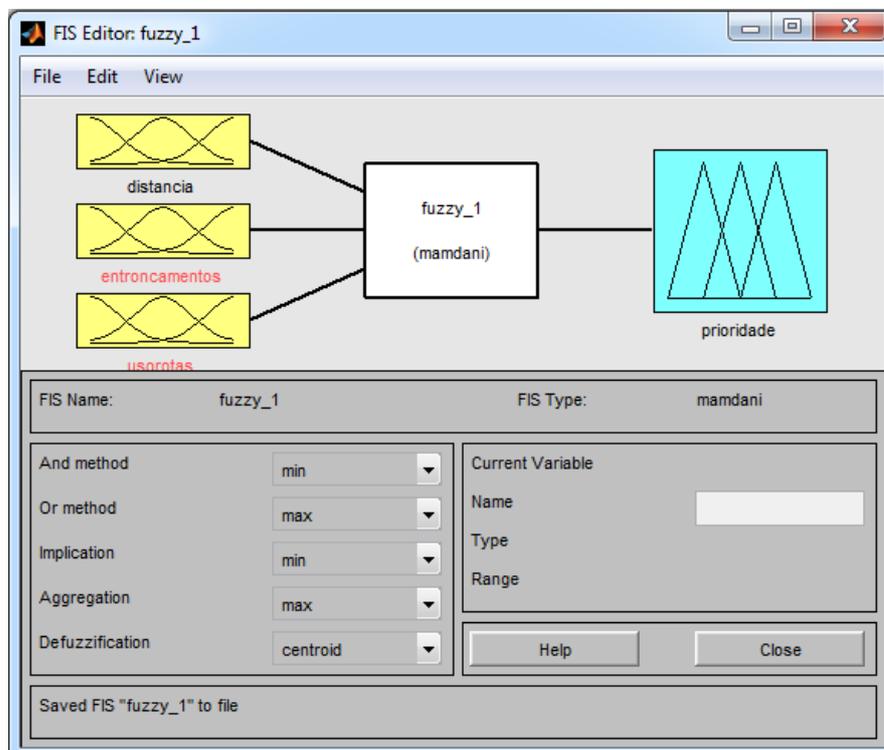


Figura 37 – Estrutura geral do sistema *fuzzy* 1.

Os valores do conjunto universo de cada variável são definidos pelos valores presentes na fábrica em que o sistema *fuzzy* irá atuar. Na fábrica 2 usada nesse trabalho, por exemplo, tem como menor distância o valor zero, isto é, o veículo está estacionado exatamente no local onde se encontra a tarefa, e tem como maior distância que pode ser percorrida o valor de cento e vinte metros. Portanto os valores do conjunto universo devem ser ajustados de acordo com a fábrica. Cada variável tem seu conjunto universo igualmente dividido em três funções de pertinência triangulares. Na Figura 38 é apresentada a configuração da viável distância para atuar na fábrica 2 usada nesta tese.

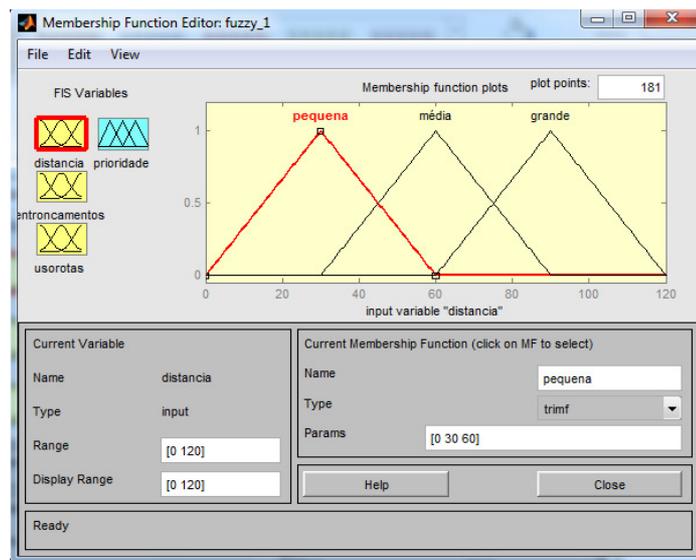


Figura 38 – Variável distância do sistema fuzzy 1.

Após a configuração de todas as variáveis se faz necessário a criação da base de regras fuzzy. Para criação das bases de regras, foram consideradas todas as possíveis combinações das variáveis de entrada e a saída foi atribuída de acordo com o conhecimento do especialista. No sistema fuzzy 1, por exemplo, são três variáveis de entrada e cada uma dividida em três funções de pertinência, chegando a um total de 27 regras. Na Figura 39 é apresentada a base de regras do sistema fuzzy 1.

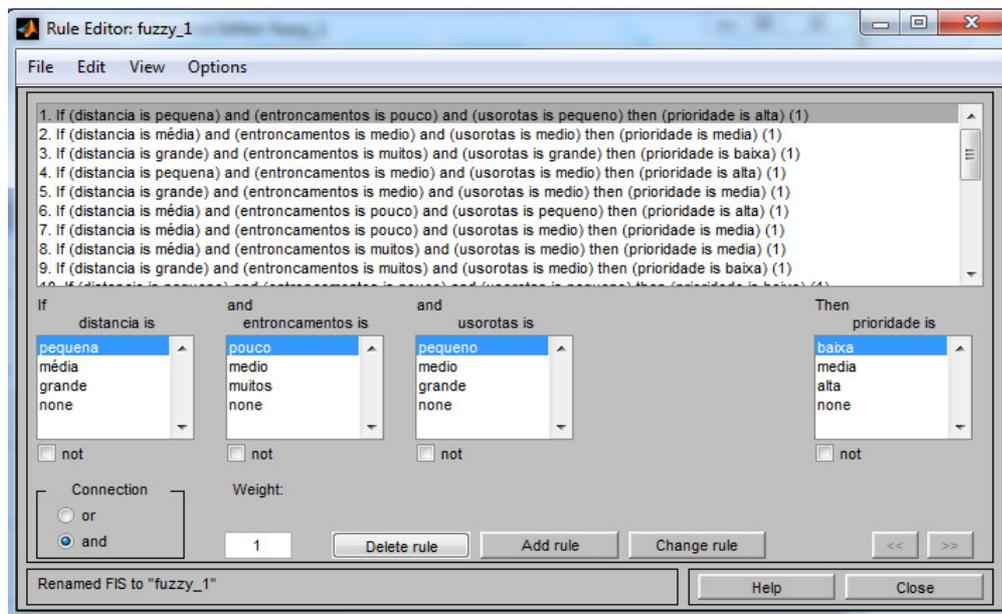


Figura 39 – Base de regras do sistema fuzzy 1.

Os sistemas *fuzzy* são construídos separadamente, porém por se tratar de um sistema *fuzzy* em cascata, se faz necessária troca de informações entre eles. A maneira como essa troca de informações funciona é detalhada na seção 4.2.4, intitulada protocolo de comunicação.

4.2.3 Sistema de Predição

Nessa seção é apresentado o desenvolvimento e funcionamento dos módulos que compõem o sistema de predição. Na Figura 40 são apresentados esses módulos que são uma parte do sistema de despacho de AGVs apresentado na Figura 23.

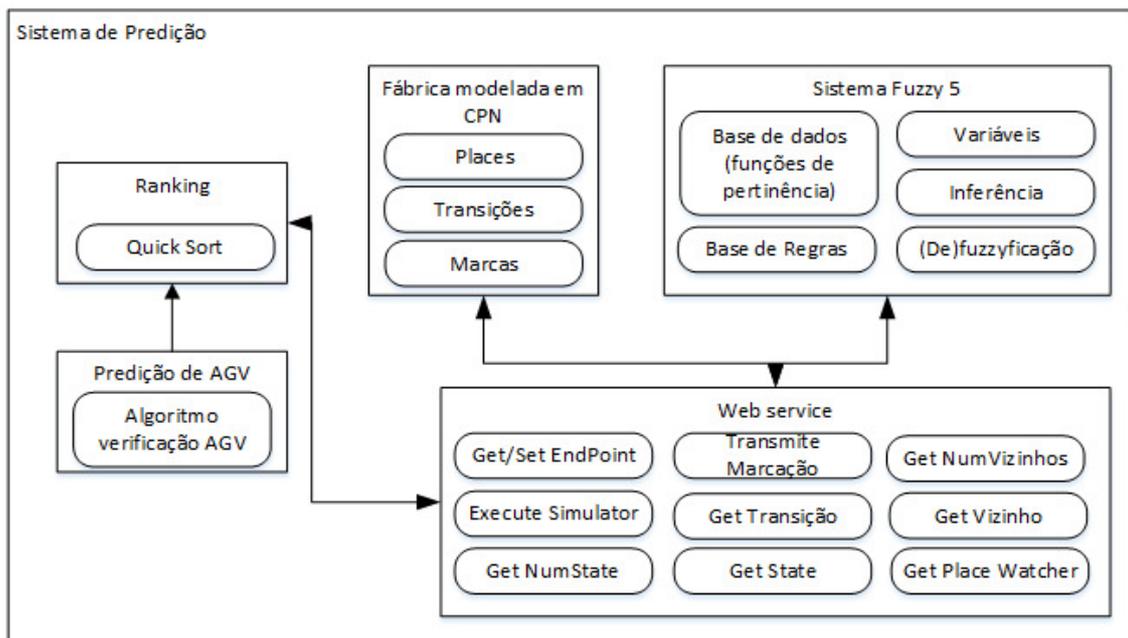


Figura 40 –Módulo predição da Figura 23.

4.2.3.1 Implementação dos roteiros de produção em redes de Petri

Devido ao fato de que a proposta do sistema de predição é baseada na geração e exploração de espaços de estados, se faz necessário a modelagem dos roteiros de produção da fábrica em redes de Petri. De mesmo modo, se faz necessário o uso de uma ferramenta computacional que, além de interface gráfica para criação e manipulação da rede, forneça funcionalidades como a de simulação e geração do espaço de estados da rede. Neste trabalho é usado o *software* CPN Tools versão 3.4.0.

A modelagem é realizada considerando os aspectos físicos da fábrica em que se representa as estações de trabalho presentes no sistema, as áreas de *buffers* de entrada e saída, AGVs, estoques e, principalmente, os roteiros dos produtos. A modelagem segue os roteiros de produção de cada produto. Os *places* representam as estações de trabalhos, estoque de matéria prima, estoque de produtos acabados, *buffers* de entrada, saída e intermediários.

Devido ao fato da fábrica 1 ser de menor porte, todos os roteiros foram modelados em uma única aba do *software*. Na Figura 41 é apresentada a aba com os roteiros dos cinco produtos da fábrica 1. Na Figura 42 é apresentado uma parte da modelagem, que engloba o início do roteiro dos produtos 1, 2 e 3, sendo possível a visualização dos detalhes.

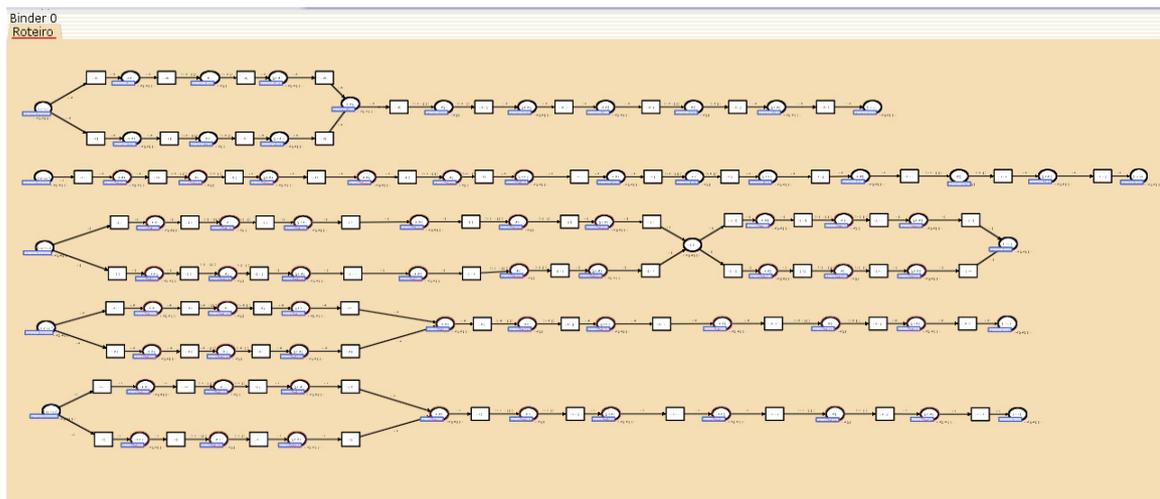


Figura 41 – Modelo em redes de Petri dos roteiros de fabricação da fábrica 1 modelada no software CPN Tools.

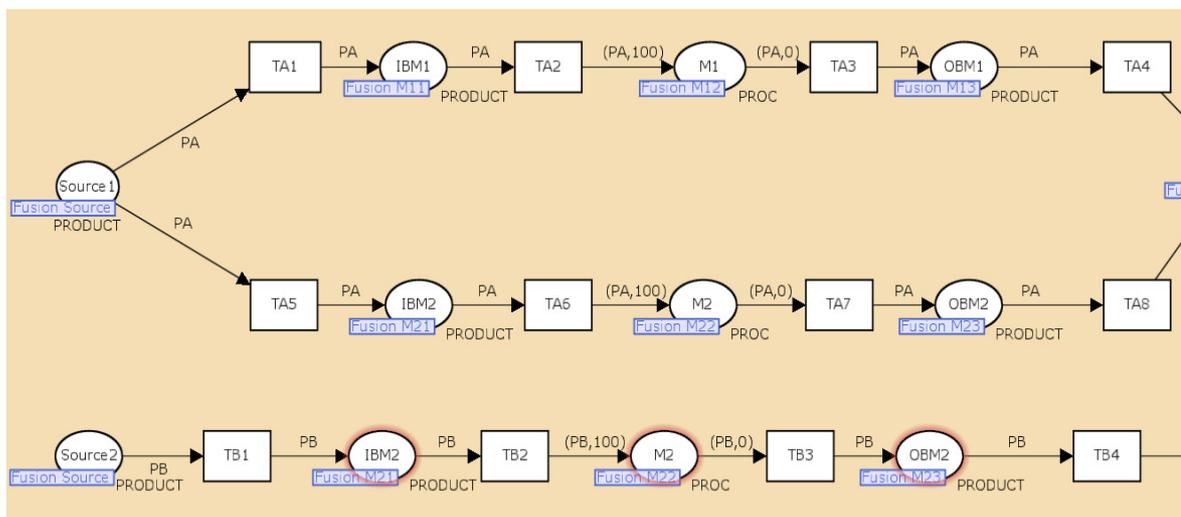


Figura 42 – Parte do modelo em redes de Petri dos roteiros de fabricação da fábrica 1 modelada no software CPN Tools.

Devido ao fato da fábrica real ser de maior porte, foi necessário modelar cada roteiro em uma aba diferente *software*. Na Figura 43 é apresentada a aba com uma parte do roteiro do produto cadeira. Na Figura 44 é apresentado uma parte da modelagem do roteiro do produto cadeira possibilitando a visualização de detalhes de construção.

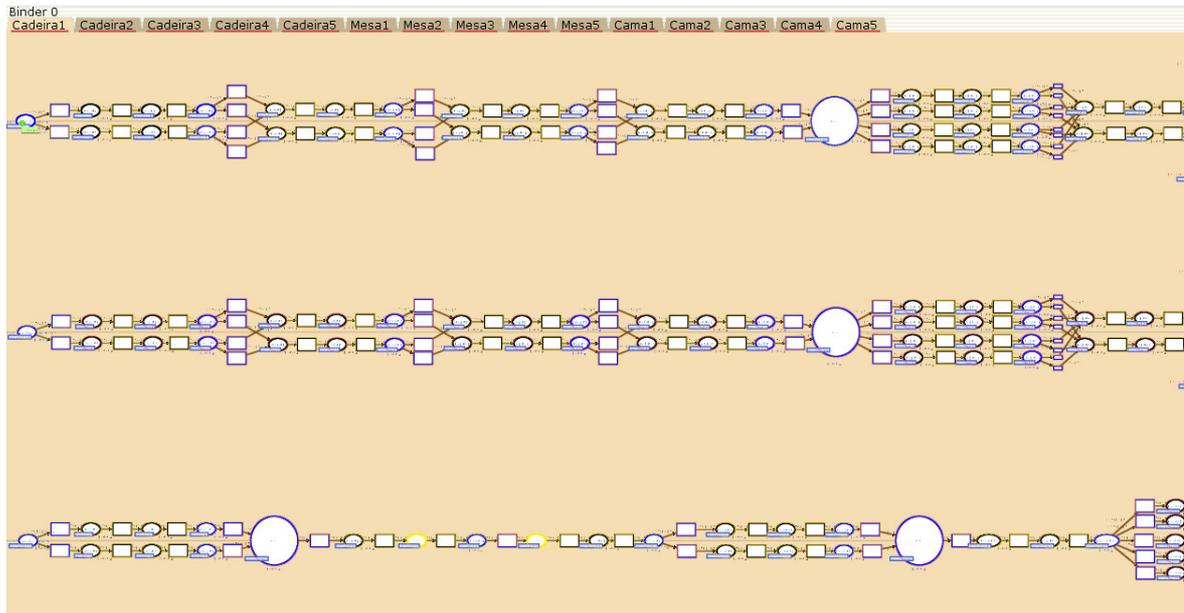


Figura 43 – Modelo em redes de Petri dos roteiros de fabricação da fábrica 2 modelada no *software* CPN Tools.

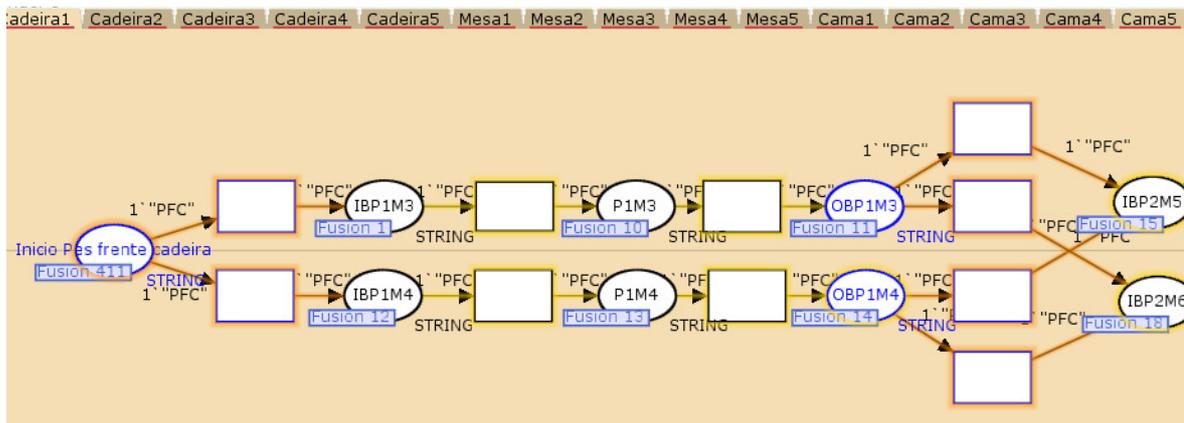


Figura 44 – Parte do modelo em redes de Petri dos roteiros de fabricação da fábrica 2 modelada no *software* CPN Tools.

4.2.3.2 Implementação do módulo *Webservice*

Com o objetivo de manter todo o sistema com maior independência de plataforma de *software*, foi construído um *webservice* baseado em classes que estenderam a biblioteca *AccessCPN* para que estas classes possam ser acessadas por sistemas de diferentes plataformas por meio de protocolos de internet. Para a construção do *webservice* foi usada a ferramenta Eclipse versão Juno Service Release 1.

O projeto do *webservice* é responsável por estabelecer comunicação entre o *software* protocolo de comunicação e o modelo simulado no *software* CPNtools. Para tal, foram realizadas as seguintes tarefas:

- Criação do projeto no Eclipse que estenda a biblioteca *AccessCPN* para abrir interface para os métodos:
 - *HighLevelSimulator.getMarking(PlaceNode)*;
 - *HighLevelSimulator.setMarking(PlaceNode, String)*;
 - *HighLevelSimulator.execute()*.
- No projeto criado no Eclipse foi desenvolvido um *webservice* SOAP baseado na classe que contém os métodos estendidos da biblioteca *AccessCPN*.

O Projeto do *webservice* é composto por três classes que estendem a biblioteca *AccessCPN*:

Simulator: responsável por referenciar a instância local do simulador do CPNTools. Uma vez que a classe *Simulator* referencia o simulador é possível que outras classes se comuniquem com o simulador, sendo possível enviar e receber sinais do simulador.

PNWatcher: Esta classe tem a responsabilidade de fazer o reconhecimento da rede de Petri que está sendo simulada pelo CPNTools e localizar todos os *places* que compõe a rede.

PlaceWatcher: responsável por realizar as operações efetivas de alteração de marca de *place*, leitura de marca de *place*, e disparo das transições habilitadas na rede de Petri que está em simulação.

As relações entre esses componentes do sistema encontram-se ilustradas no diagrama de comunicação da Figura 45.

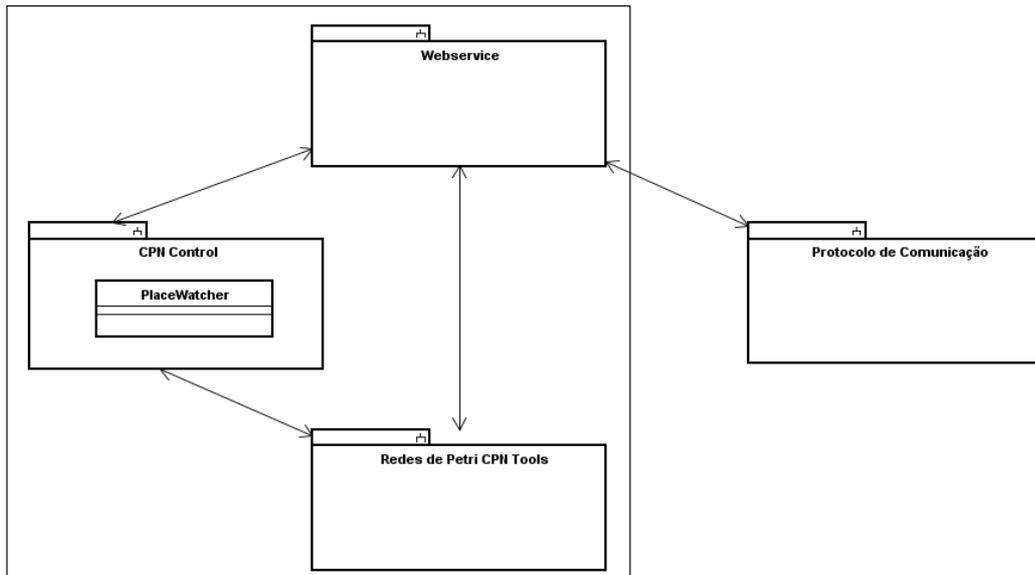


Figura 45 – Diagrama de comunicação entre os elementos do sistema. As setas em ambos os sentidos indicam possibilidade tanto de envio como de obtenção de informações.

No diagrama da Figura 45, o *software* chamado de CPN Control foi construído em Java e faz uso da biblioteca Access/CPN, disponibilizada pelos desenvolvedores do CPN Tools para fornecer mecanismos para interação do mesmo com *softwares* externos. Tanto o uso da linguagem Java como da IDE Eclipse é recomendado no uso da Access/CPN.

Todos os serviços de interação do CPN Tools com o meio externo podem ser acionados diretamente pelo CPN Control e visualizados na tela do console, por exemplo; no entanto, o *webservice* foi introduzido como uma interface entre o CPN Tools e o *software* Protocolo de Comunicação (SPC). Uma vez configurado, o *webservice* usando os métodos de uma das classes do CPN Control (a classe *PlaceWatcher*). A utilidade do CPN Control é a de inicializar o *webservice* e mantê-lo durante a execução do sistema. Dessa forma, o *webservice* abstrai o funcionamento da interação com o CPN Tools ao SPC, e o uso se dá apenas pelo acionamento dos serviços por meio do SPC.

O serviço responsável por obter o número de estados gerados recebe o argumento que informa o tipo de busca a ser efetuada, no caso deste trabalho, a busca direcionada com poda que é explicada na seção 4.2.3.3. Conjuntamente é adicionado o serviço que retorna o nome das transições habilitadas do estado e as marcações de seus respectivos *places* de entrada. Esse serviço complementa informações obtidas durante a geração e as mantém à disposição do SPC.

Os serviços disponíveis do *webservice* podem ser melhor compreendidos com a modelagem UML do diagrama de casos de uso do sistema, ilustrado na Figura 46.

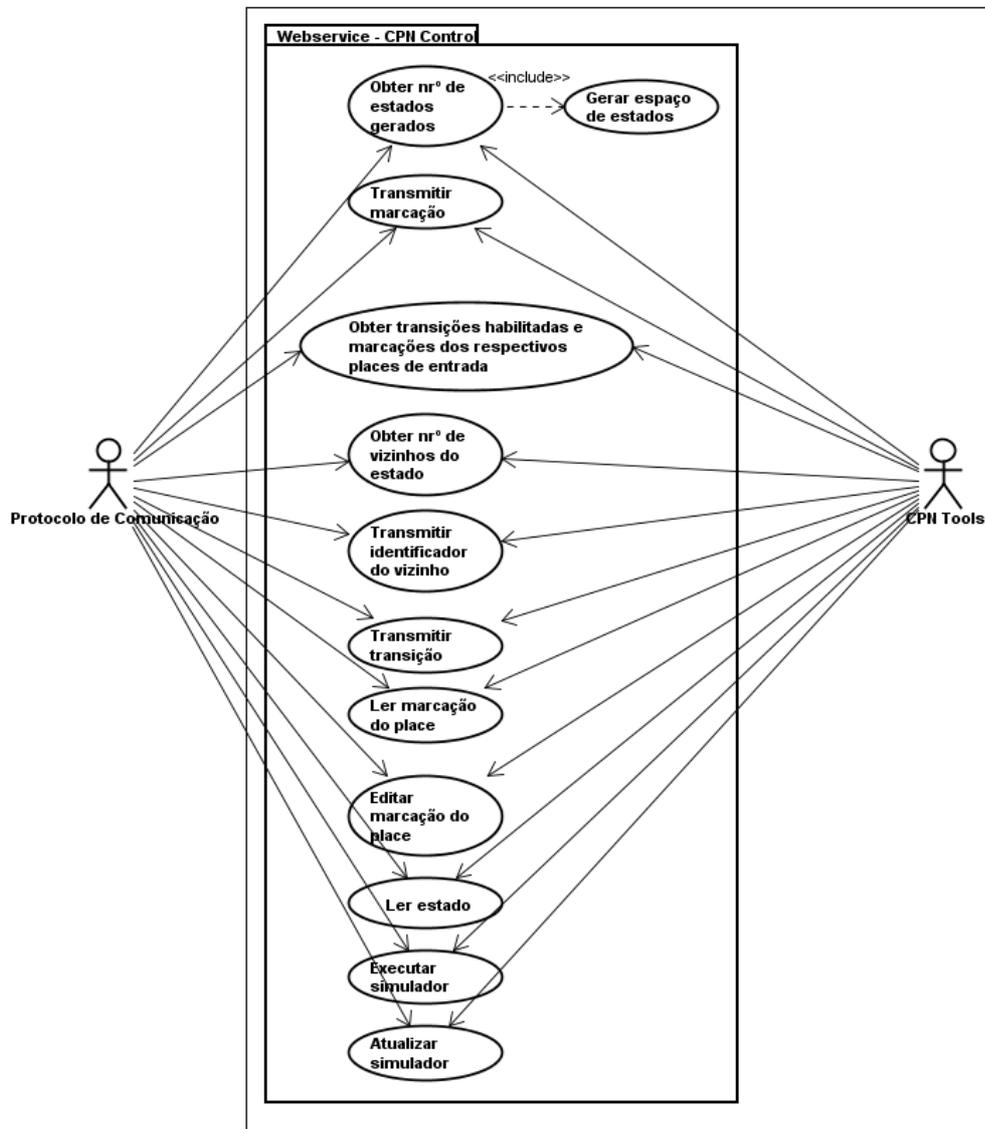


Figura 46 – Diagrama de casos de uso do sistema.

No diagrama da Figura 46, cada caso de uso corresponde a um serviço do *webservice*. Dentre as funcionalidades básicas estão a leitura e edição da marcação de um determinado *place* da rede de Petri. As funcionalidades de atualização do simulador, ou seja, atualização da tela do CPN Tools frente a uma mudança, e a de execução do simulador, ou seja, de ativação de uma transição da rede de Petri que esteja habilitada, são funcionalidades secundárias e são acionadas principalmente por outros serviços. As demais funcionalidades estão relacionadas à extração do espaço de estados da rede de Petri.

Olhando mais de perto o projeto CPN Control e descendo o nível de abstração, pode-se distinguir onde foi usada a biblioteca Access/CPN e os métodos da classe *PlaceWatcher*, que foram transformados em serviços do *webservice*, contidos no diagrama de classes da Figura 47.

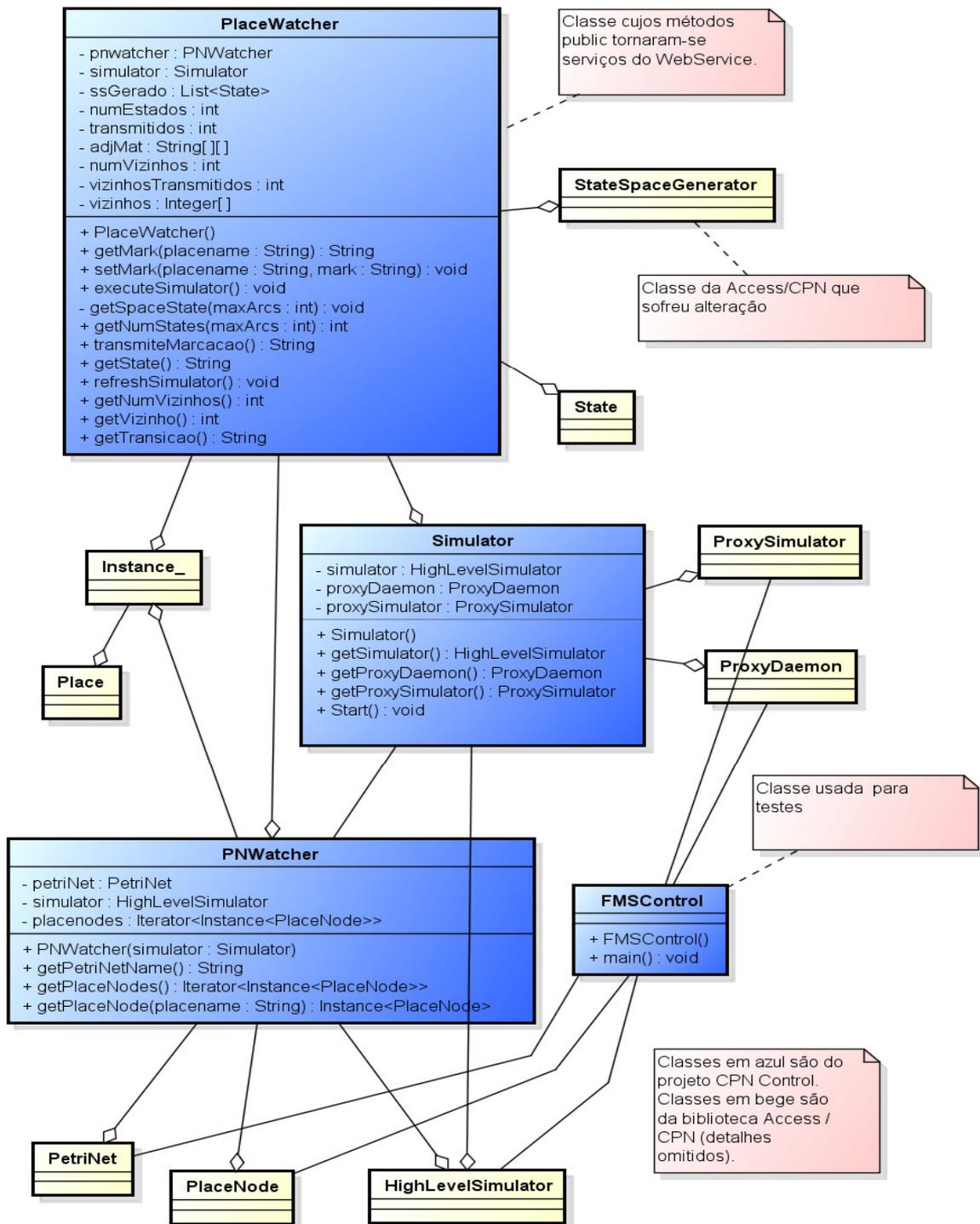


Figura 47 – Diagrama de classes do Projeto CPN Control. As classes em azul são do projeto CPN Control. Classes em bege são originais da biblioteca Access/CPN, e seus detalhes foram omitidos.

Sendo assim, a sequência de ações e acionamento de serviços necessária para se transmitir o espaço de estados pelo *webservice* foi planejada para que determinados serviços sejam acionados em sequência, como sugerido no seguinte algoritmo apresentado na Tabela 8.

Tabela 8 – Algoritmo para acionamento do serviço

```

numStates = getNumStates( int maxArcs )
  Para 1 .. numStates Faça
    transmiteMarcacao()
    getTransicoesHabilitadas()
    numVizinhos = getNumVizinhos()
    Para 1 .. numVizinhos Faça
      getVizinho()
      getTransicao()

```

Se o algoritmo for seguido serão transmitidos, para cada estado e de maneira sequencial:

- A marcação correspondente ao estado
- O nome das transições habilitadas e marcação dos respectivos *places* de entrada
- O número de vizinhos (descendentes diretos ou adjacentes) do estado
- Para cada um dos vizinhos:
 - O número (a partir de zero) identificador desse vizinho
 - A transição correspondente à mudança de estado para o estado vizinho

No diagrama de classes da Figura 47, a classe *PlaceWatcher* é a mais importante, sendo aquela cujos métodos públicos foram transformados em serviços do *webservice*. As demais classes são agregadas à classe *PlaceWatcher* para melhor modularizar a sua implementação.

A classe *StateSpaceGenerator* é proveniente de um projeto demo da Access/CPN com o intuito de extrair o espaço de estados completo da rede de Petri e exibí-lo na janela do console da IDE Eclipse, e teve que ser devidamente adaptada para limitar o tamanho do espaço de estados gerado, gerar o espaço de estados a partir do estado atual da rede e não necessariamente o estado inicial, possibilitar múltiplas gerações de espaços de estados intercaladas com outros serviços e também armazenar e transmitir o espaço de estados via *webservice*.

Subsequentemente são explicados de maneira objetiva cada serviço do *webservice*.

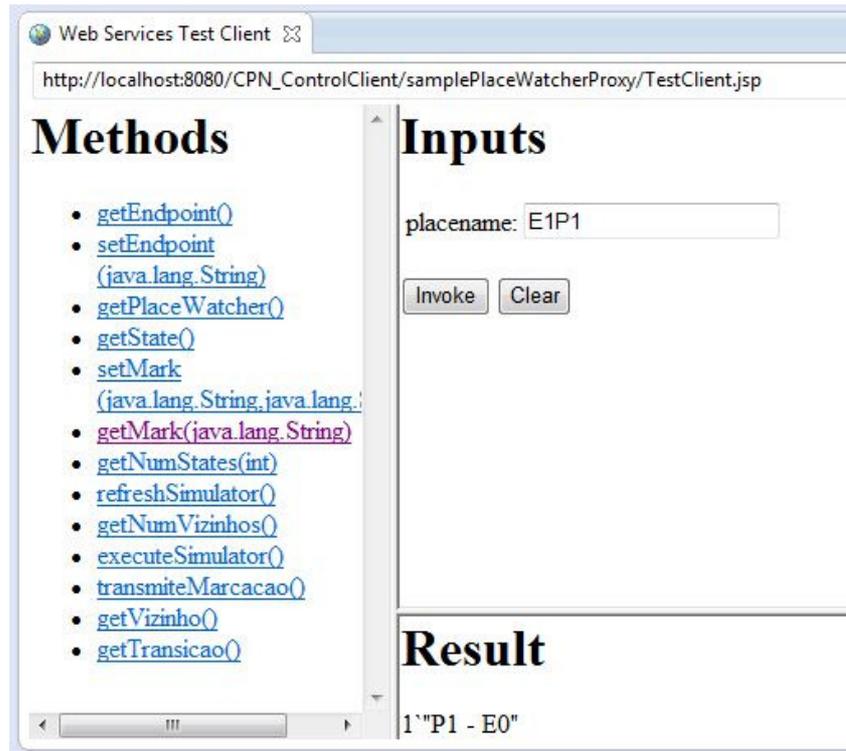


Figura 48 – Serviço `getMark`. O place de nome E1P1 possui como marcação “1”P1-E0” Isso significa que o place E1P1 possui 1 token, do tipo string, com o valor “P1-E0”.

- *String Mark* `getMark(String placeName)`: fornecido o nome do *place* (exemplos: “p1”, P2”, “place 1”) retorna a marcação atual. Um exemplo de chamada a esse serviço encontra-se na Figura 48.
- *void* `setMark(String placeName, String newMark)`: fornecido o nome do *place* e a nova marcação, substitui a marcação atual do *place* pela marcação fornecida.
- *String* `getState()`: retorna o estado atual da rede na forma de uma String.
- *int numStates* `getNumStates(int maxArcs)`: gera o espaço de estados da rede de Petri a partir do estado em que ela se encontra, explorando no máximo *maxArcs* transições. Retorna o número de estados distintos encontrados, incluindo o estado de partida. Um exemplo de chamada a esse serviço encontra-se na Figura 49.

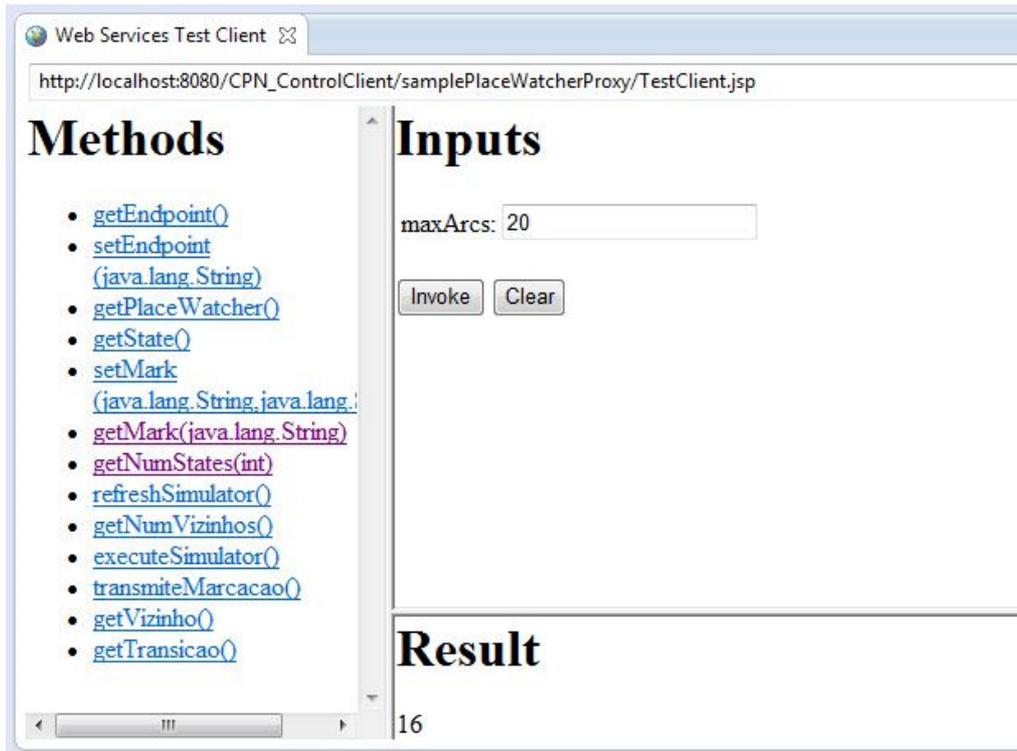


Figura 49 – Serviço `getNumStates`. Nesse exemplo foram explorados todos os arcos possíveis (15), uma vez que o limite especificado foi 20, e foram encontrados 16 estados distintos. Isso significa que a partir da rede na configuração atual podem-se disparar diferentes combinações de transições e serão obtidas até 15 outras configurações distintas (estados) para a rede.

- *String transmiteMarcacao()*: transmite até *numStates* estados distintos encontrados, um de cada vez, a cada acionamento do método transmite estado. Cada estado é representado por uma *String*, por sua vez dividida em *Strings* separadas por quebra de linha, cada uma contendo a marcação de um dos *places* da rede, mas apenas para aqueles em que a marcação do *place* não for vazia. Quando todos os estados forem transmitidos será retornada a string “Todos os estados já foram transmitidos.” Um exemplo de chamada a esse serviço encontra-se na Figura 50.

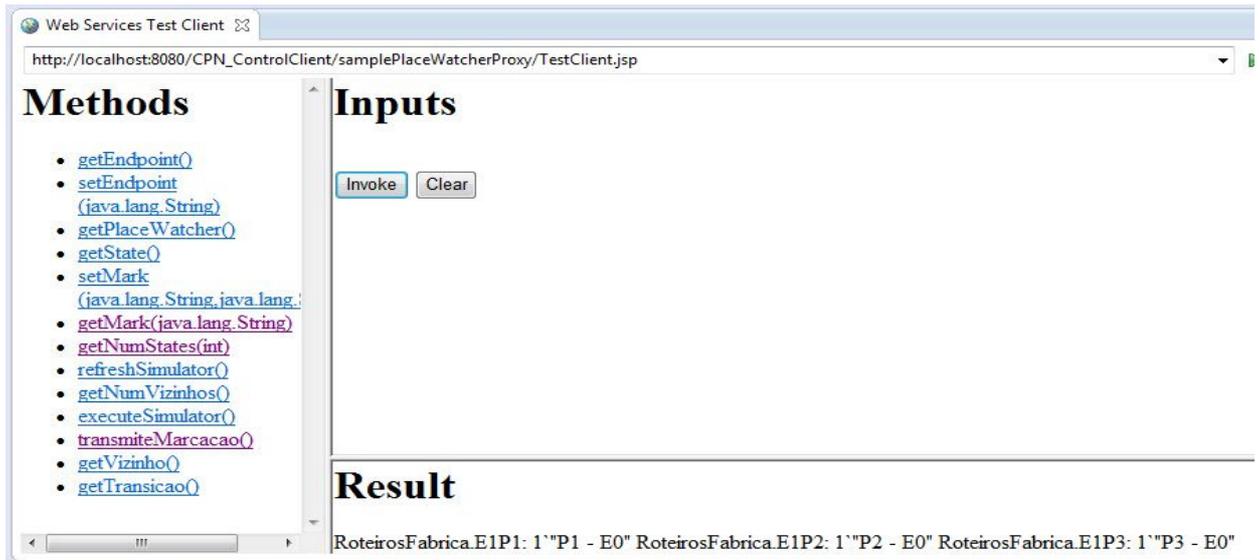


Figura 50 – Serviço `transmiteMarcacao` – transmissão de um dos estados. Nesse exemplo, o estado transmitido foi o estado atual da fábrica. Note que a marcação de cada *place* possui a seguinte notação: nome da página da rede (`RoteirosFabrica`), seguida de um ponto e o nome do *place* (por exemplo, `E1P1`), seguido de dois pontos e a marcação propriamente dita (por exemplo: “1”`P1-E0`”). Observe que os *places* que possuem marcação vazia são omitidos.

- `int numVizinhos getNumVizinhos()`: retorna o número de vizinhos do estado sendo transmitido. Retorna -1 caso o serviço tenha sido requisitado de maneira incorreta (quando não existir espaço de estados gerado ou quando todos os estados já tiverem sido transmitidos). Um exemplo de chamada a esse serviço encontra-se na Figura 51.

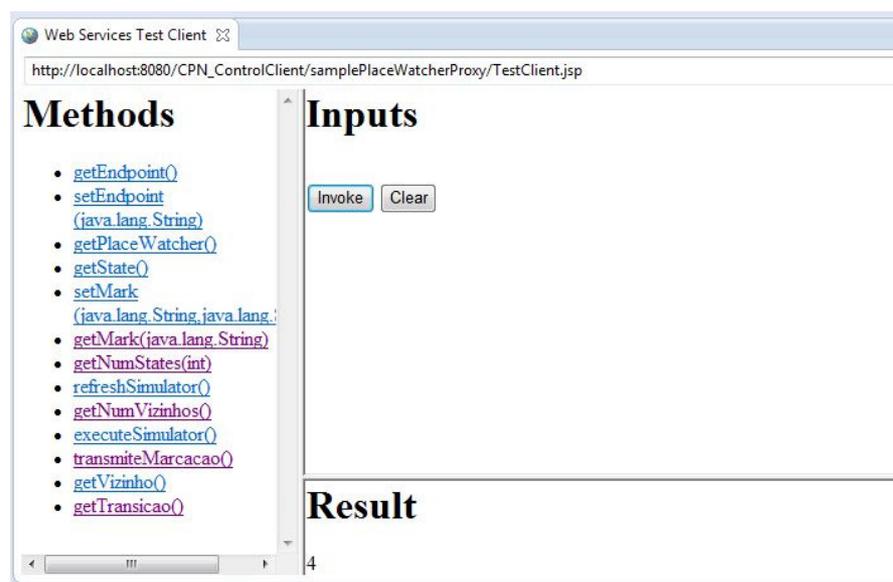


Figura 51 – Exemplo serviço `getNumVizinhos`. A partir da configuração da atual existem 4 outras configurações distintas imediatamente posteriores, uma vez que existem 4 transições habilitadas na rede.

- *int getVizinho()*: retorna o identificador (inteiro não-negativo) de um dos vizinhos do estado em transmissão (o estado atual da rede sempre possui identificador zero). Retorna -1 se for utilizado de maneira incorreta (todos os vizinhos já tiverem sido transmitidos).
- *String getTransicao()*: retorna informações relacionadas à transição que ocasionaria a mudança de estado do estado sendo transmitido para o vizinho sendo transmitido (cujo identificador foi obtido por *getVizinho* imediatamente antes). Essas informações estão contidas em uma *String* com o seguinte formato: *nomeDaPáginaDaRede.nomeDaTransição{atribuiçãoDeVariáveisDeArco }*. Um exemplo de chamada a esse serviço encontra-se na Figura 52.

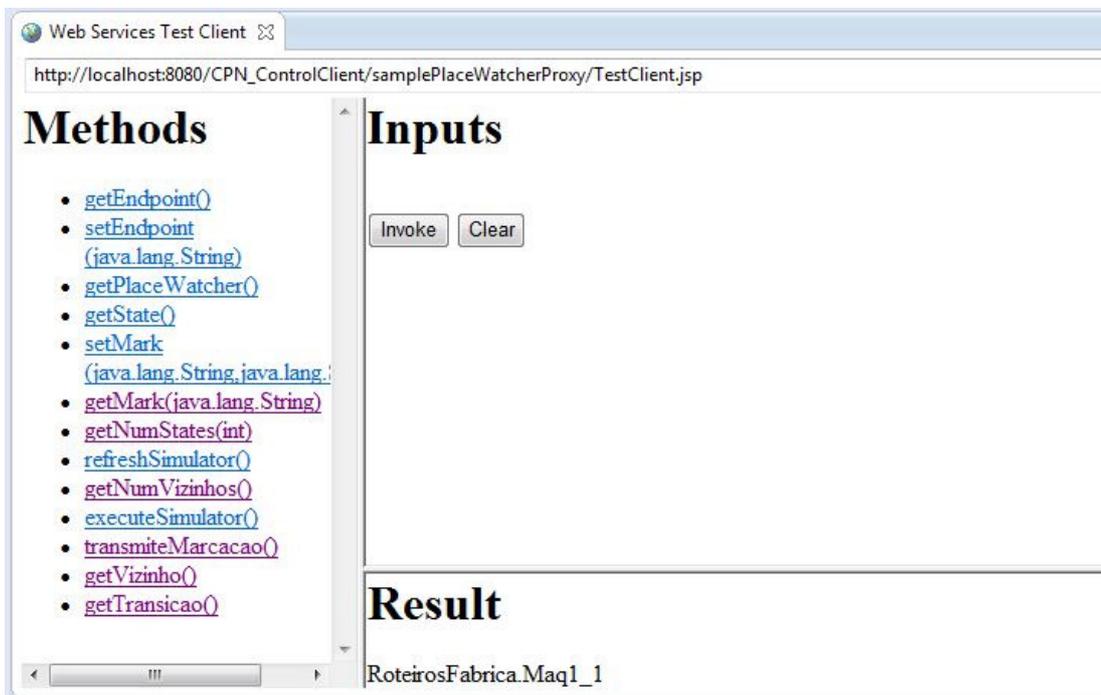


Figura 52 – Serviço *getTransicao*. Nesse exemplo, para que a rede mudasse da configuração no estado de identificador zero (estado atual) para a configuração no estado 1, seria necessária a ativação da transição *Maq1_1*.

- *getTransicoesHabilitadas()*: que retorna todas as transições habilitadas e as marcações de seus respectivos *places* de entrada. Um exemplo de execução se encontra na Figura 53.

Os demais serviços descritos abaixo são secundários, e são usados de maneira indireta por outros serviços de maior importância:

- *void executeSimulator()*: ativa aleatoriamente alguma transição que esteja habilitada na rede de Petri.
- *void refreshSimulator()*: atualiza a tela do CPN Tools para que ela represente alguma atualização efetuada na rede (mudança de estado). É acionado automaticamente pelo *setMark*, *executeSimulator* e *getNumStates*.

4.2.3.3 Algoritmo de busca direcionado com poda

A fim de lidar com a explosão combinatória ao se gerar um espaço de estados completo, implementou-se um algoritmo direcionado com poda.

A busca direcionada com poda tem o intuito de gerar parte do espaço de estados da rede de Petri, no entanto, ao contrário da geração por meio da busca em largura ou em níveis completos, isso é feito a partir do disparo de determinada transição habilitada de interesse, com exploração apenas de estados derivados do disparo dessa transição e de transições subsequentes contidas nesses estados. Dessa forma, o algoritmo de busca implementado é basicamente uma busca em largura com condições adicionais analisadas no momento da expansão de um estado, as quais restringem a geração de estados àqueles obtidos por meio de disparos de transições de interesse.

Neste trabalho, definiu-se transição de interesse como sendo aquela que passou a estar habilitada, mas no estado anterior da busca não estava (isto é, representam possibilidades de exploração que surgiram a partir do disparo de outra transição de interesse), ou transições cujas marcações dos *places* de entrada, ou seja, responsáveis por determinar se dada transição estaria habilitada ou não, sofreu algum tipo de alteração, em termos de quantidade e tipo das marcas (subsequentes alterações nos *places* de entrada dessa transição poderiam ocasionar em sua eventual habilitação). Inicialmente, a transição de interesse é dada como entrada no algoritmo, e representa determinado elemento do modelo em redes de Petri que pode ser acionado e as implicações que se deseja investigar. O funcionamento básico do algoritmo é apresentado na Tabela 9.

Tabela 9 – Pseudo código do algoritmo

<p>Entradas:</p> <ul style="list-style-type: none"> - Rede de Petri em um dado estado - Transição de interesse, que deve estar habilitada - O número máximo de arcos a serem explorados (valor deverá ser estimado com base na estrutura da rede e na quantidade de níveis máxima a ser considerada) <ol style="list-style-type: none"> 1. Armazenar estado inicial da rede e marcá-lo como visitado 2. Obter lista de todas as transições habilitadas e as marcações de seus respectivos <i>places</i> de entrada e armazená-los 3. Disparar a transição de interesse 4. Adicionar na fila cada estado novo obtido a partir do passo 3, armazená-lo e marcá-lo como visitado, criando uma aresta do estado inicial para o estado novo 5. Enquanto a fila não estiver vazia <ol style="list-style-type: none"> 5.1. Retirar primeiro estado da fila (estado "u") 5.2. Obter lista de todas as transições habilitadas e as marcações de seus respectivos <i>places</i> de entrada e armazená-los 5.3. Selecionar apenas as transições de interesse do estado "u", que são: <ol style="list-style-type: none"> 5.3.1. Transições que não estavam no estado pai de "u", mas que agora estão no estado "u" 5.3.2. Transições que estão em ambos os estados e cuja marcação dos <i>places</i> de entrada sejam distintas 5.4. Disparar cada uma das transições selecionadas, obtendo um ou mais estados "v": <ol style="list-style-type: none"> 5.4.1. Se o estado "v" não tiver sido visitado, então armazenar e inserir "v" na fila, criando uma aresta do estado "u" para o estado "v"
--

Como consequência do algoritmo acima, o espaço de estados gerado representará apenas ações desencadeadas pelo disparo de uma dada transição, o que resulta em um grafo parcial do espaço de estados menos denso.

Além disso, uma vantagem bastante importante é que o grafo selecionará informações úteis relacionadas apenas a uma dada transição. Considerando que a modelagem em redes de Petri representa um ambiente de manufatura, essa transição pode estar associada com uma máquina e o grafo, por sua vez, com ações futuras desencadeadas pela passagem de um produto nessa máquina, por exemplo.

Vale ressaltar que, mesmo para uma rede de tamanho modesto, o espaço de estados obtido é relativamente grande e complexo (porém ainda pôde ser representado de maneira

gráfica). A geração e visualização do espaço de estados em todos os níveis para a rede que representa a fábrica real seria impraticável.

Portanto, ao aplicar o algoritmo de busca direcionada com poda se tem uma drástica redução no número de estados do espaço de estados sem perda de informação, uma vez que a intenção é investigar o que se decorre ao disparar a transição “X” e as diversas possibilidades subsequentes, ou equivalentemente e de acordo com o significado da modelagem, analisar as consequências de se usar a máquina “Y” para processar o produto “P” da etapa 0 para etapa 1, ao invés de produzir algum outro produto em determinada máquina antes. Também podem ser analisados aspectos como conflito no uso de máquinas, paralelismo entre tarefas e comparação de benefícios entre os caminhos, por meio da escolha de diferentes transições de interesse.

4.2.4 Implementação do *software* Protocolo de comunicação

Nessa seção é apresentado o desenvolvimento e funcionamento dos módulos que compõe o *software* de comunicação, chamado de protocolo de comunicação. Na Figura 54 são apresentados esses módulos que são uma parte do sistema de despacho de AGVs apresentado na Figura 23.

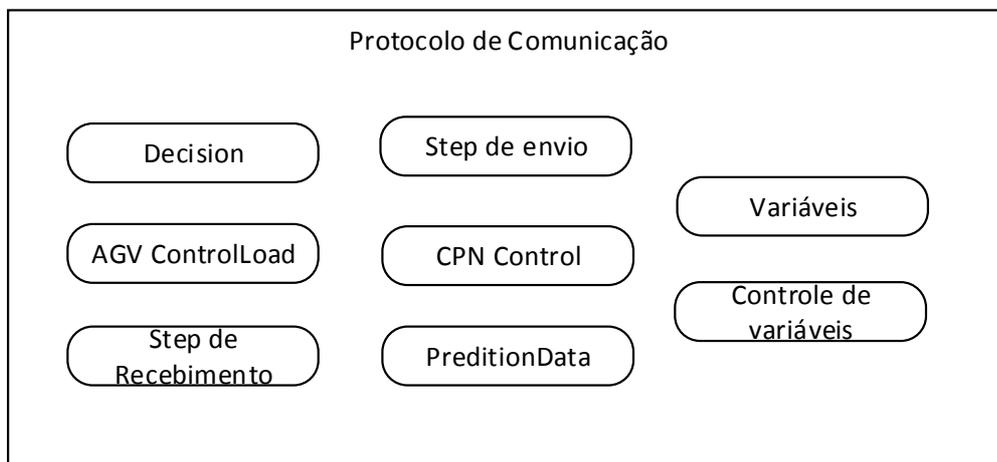


Figura 54 –Módulo protocolo de comunicação da Figura 23.

O *software* protocolo de comunicação é responsável pelo controle de todo fluxo de informações entre as simulações, sistemas *fuzzy* e fábrica. O sistema foi desenvolvido no *software* Microsoft Visual Studio 2010 e escrito na linguagem *c#*. O sistema possui vários módulos, cada qual com funções específicas.

Módulo GetInfo (step de recebimento): responsável pela comunicação com o *software* Simio. Este módulo se torna um *step user defined* e é usado em alguns processos da simulação. Sua função é a de capturar as informações da fábrica como, distâncias, quantidade de peças nos *buffers* de entrada e saída, quais estações de trabalho estão em processamento, quais estações de trabalho estão paradas, etc, enfim, todas as informações necessárias para replicar o estado atual no modelo em redes de Petri e também informações que servirão como variáveis de entrada nos sistemas *fuzzy*.

Módulo Priori (step de envio): responsável pela comunicação com o *software* Simio. Este módulo se torna um *step user defined* e é usado em alguns processos da simulação. Sua função é a de atribuir as prioridades calculadas nas respectivas tarefas que estão aguardando por transporte na fábrica.

Na Figura 55 é apresentada a tela do *software* Simio em que se tem um processo usando os *steps* “GetInf” e “Priori”.

Na Tabela 10 são apresentados trechos de código que são base nos módulos GetInf e Priori. Primeiramente é necessário criar um campo de opção lógica para ser preenchido no *software* Simio, em seguida é criada uma variável que receberá o valor lido na fábrica, por exemplo, a variável *distancia1* recebe a distância da posição atual do veículo até a estação de trabalho 1 sendo vinculada ao comando `Vehicle.CurrentNetwork.Distance(Vehicle.CurrentNode , Output@M1`. Por fim é realizada a vinculação entre a variável e o campo de opção lógica, fazendo com que as informações lidas da fábrica fiquem disponíveis para uso nos outros módulos.

Módulo ControlInterface: responsável pela comunicação com o modelo em rede de Petri. Este projeto faz uso direto dos serviços do *webservice*. Por meio do *webservice*, o projeto *ControlInterface* consegue resgatar as marcas nos *places* da rede de Petri, alterar as marcas da rede, executar as transições habilitadas na rede e gerar espaço de estados a partir do estado vigente da rede.

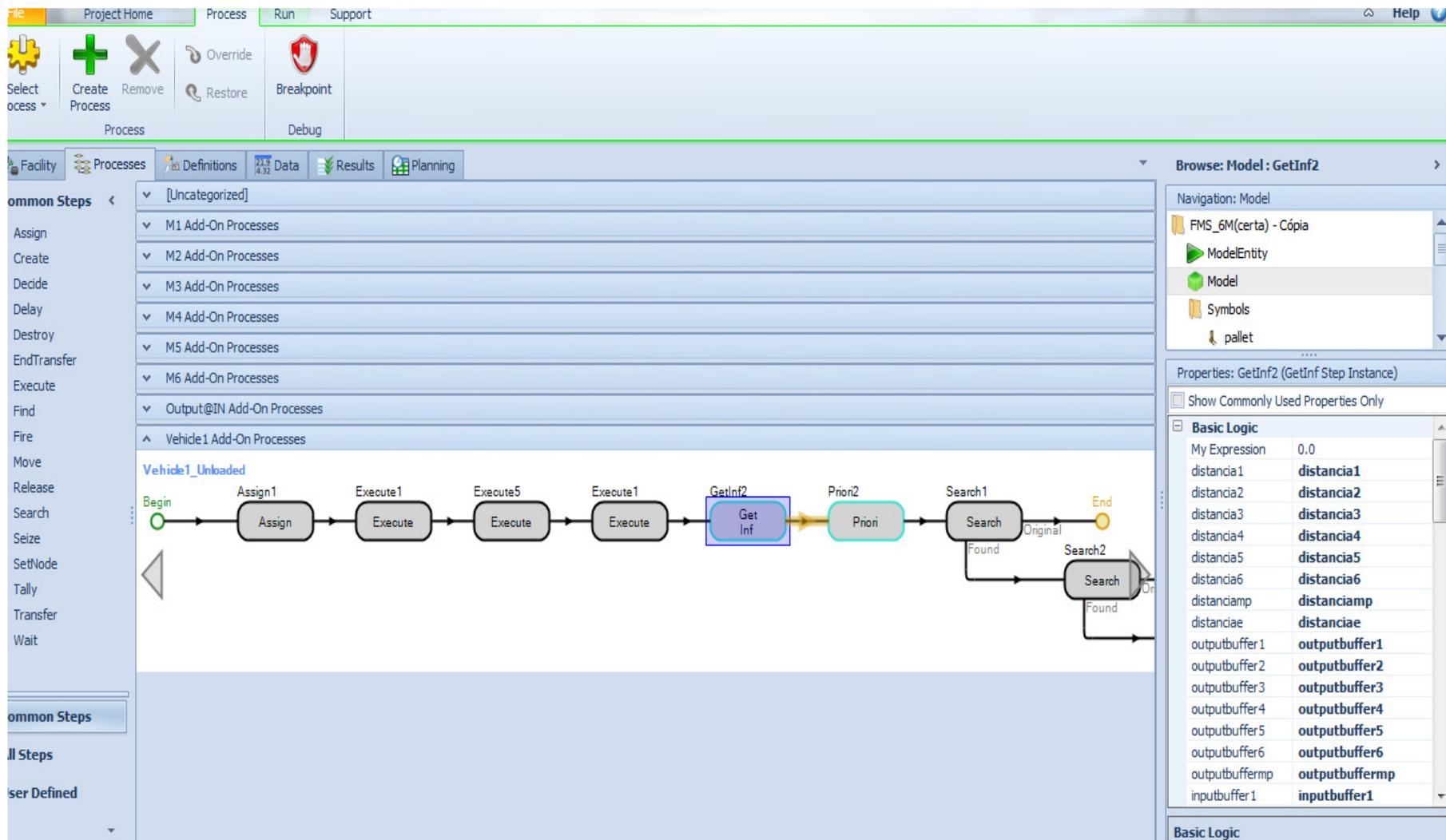


Figura 55 – Tela do software Simio apresentado o processo com os steps

Tabela 10 – Trechos do código do módulo GetInfo

```

pd = schema.AddStateProperty("distancia1"); // Cria um
campo de opção de lógica com o nome descrito

static double distanciam1;

IPropertyReader distancia1 =
_properties.GetProperty("distancia1") as IPropertyReader;
// leitura ou vinculação de uma propriedade do step
    IStateProperty getDistancia1 =
(IStateProperty)distancia1;
    IState distancia1obtain =
getDistancia1.GetState(context);
    distanciam1 = distancia1obtain.StateValue;

```

Na Tabela 11 são apresentados os trechos de código que são base no módulo *ControlInterface*. Primeiramente, converte-se o tempo de trabalho da estação de trabalho para *string*. Subsequentemente são concatenados os valores da marca, o tipo da peça que está sendo processada e o tempo de processamento. Inicia-se a comunicação com o *webservice*, realiza uma nova concatenação de valores e então as informações são enviadas para o modelo em redes de Petri. Por fim, é solicitado a criação de um novo espaço de estados.

Tabela 11 – trechos de código do módulo ControlInterface

```

string s_mM1work_time = Convert.ToString(iM1work_time); //
converte valor do para string

String mark_mM1working = String.Concat(s_um, p_mM1working,
s_mM1work_time, ""); // concatena strings

//COMUNICAÇÃO WEBSERVICE
    AccessCPNService.PlaceWatcherClient client =
new AccessCPNService.PlaceWatcherClient(bdg,epa); // inicio
da comunicação com webservice
    AccessCPNService.PlaceWatcherClient
placewatcherclient = new
AccessCPNService.PlaceWatcherClient(bdg, epa);

String mark_OBM1 = String.Concat(mark_mP1outputbuffer1,
juncao, mark_mP2outputbuffer1, juncao,
mark_mP3outputbuffer1, juncao, mark_mP4outputbuffer1,
juncao, mark_mP5outputbuffer1); // concatena todas as
strings do mesmo place, senão um sobrepe o outro
    client.setMark("OBM1", mark_OBM1); // Envio da
marca P1 para outputbuffer M1

StateSpace ss = new StateSpace(placewatcherclient);

```

Módulo *Decision*: contém o código responsável por receber dados das situações de decisão do sistema e enviar estes dados ao sistema *fuzzy* para que sejam processados. O sistema de inferência *fuzzy* devolve para o projeto *Decision* os valores de prioridade calculadas e o Projeto *Decision* então repassa esta informação ou para outro sistema *fuzzy*, caso não seja a prioridade final, ou para o módulo *Priori*.

Para que se possam usar os sistemas *fuzzy* criados no *software* Matlab no módulo *decision* é necessário criar um script para cada sistema *fuzzy* no próprio Matlab como o apresentado na Tabela 12.

Tabela 12 – Script criado no Matlab

```
function out=fisparameter(in1, in2, in3)
    fismat = readfis('fuzzy_1');

    out = evalfis([in1 in2 in3],fismat);
```

No código são especificadas três variáveis como parâmetro da função (*in1*, *in2*, *in3*) e em seguida o sistema *fuzzy* é importado pela função "readfis". Os parâmetros são passados para a função *evalfis* que retorna a saída calculada pelo sistema *fuzzy*.

Na sequência é usada a ferramenta "deploytool", em que são incluídos os arquivos do sistema *fuzzy* e do script. Na Figura 56 são apresentadas as telas de configuração do "deploytool".

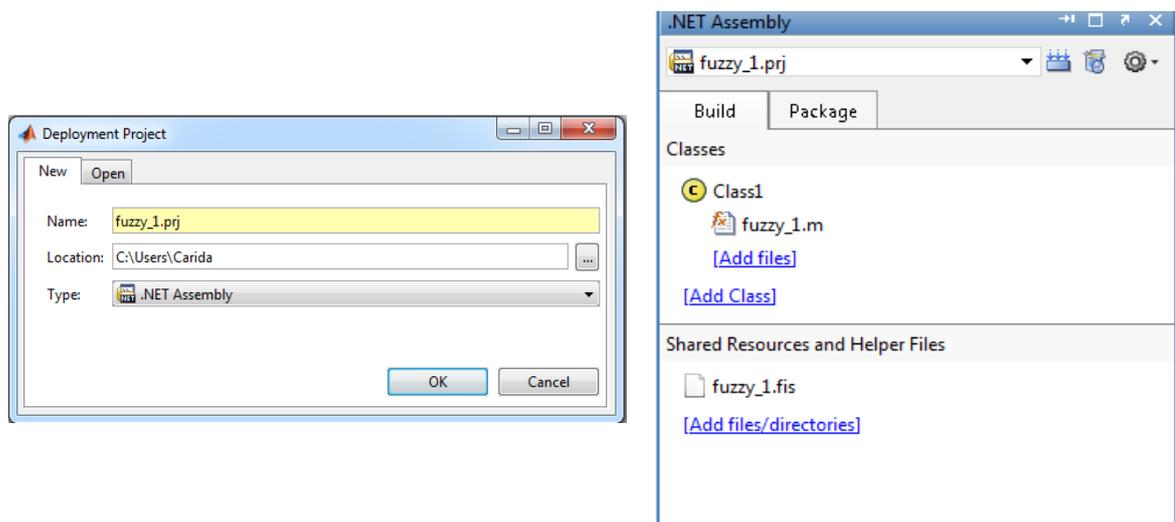


Figura 56 – Telas de configuração do *deploytool*

A ferramenta “*deploytool*” gera um arquivo .dll que é importado para o módulo *decision*, como exemplificado na Figura 57.

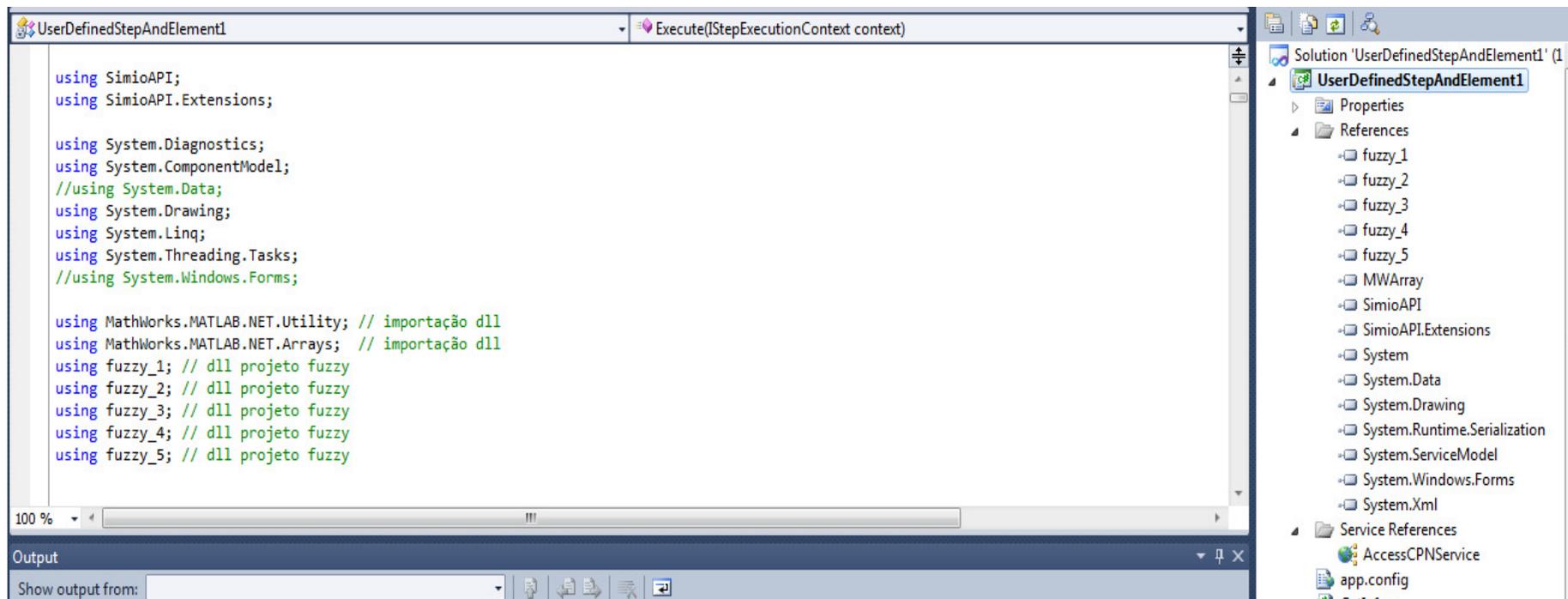


Figura 57 – Importação dos arquivos .dll para o módulo *decision*.

Na Tabela 13 são apresentados trechos do código para o uso dos sistemas *fuzzy*. Primeiramente é criada a classe do tipo *fuzzy* e uma variável do tipo *MWArray* (tipo nativo do Matlab). Na sequência é invocada a função que envia os valores das variáveis para o sistema *fuzzy*. São necessárias conversões do tipo do valor de retorno para que o mesmo possa ser usado pelos outros módulos.

Tabela 13 – Trechos do código para uso dos sistemas *fuzzy*.

```
Class1 fuzzy = new Class1(); // classe criada no matlab
MWArray saida1p1 = 0; // declaração da variável
legivel pelo matlab
saida1p1 = fuzzy.fuzzy_2(inputbufferm1,
outputbufferm1, setupm1); // função para enviar e receber
os dados do fuzzy
double[,] saida2p1 =
(double[,])(MWNumericArray)saida1p1).ToArray(MWArrayCompon
ent.Real); // converter o valor que vem do matlab para
array de double
saida3m1 = (double)saida2p1[0, 0]; // converter
de array de double para double
```

4.3 Considerações finais

Com o intuito de possibilitar a verificação das hipóteses desta tese foram desenvolvidos *softwares* que transpõem a proposta e *softwares* que se tornam necessário para validar a proposta.

Neste capítulo foram apresentadas explicações, modelos, trechos de códigos e *screenshots* de telas dos *softwares* desenvolvidos. Também foram apresentadas as ferramentas usadas para a implementação do projeto e os diagramas de caso de uso e de classes de domínio.

Pretende-se que, ao final da leitura deste capítulo, o leitor tenha compreendido como a proposta está implementada e seu funcionamento.

No próximo capítulo são apresentados os testes do sistema desenvolvido e os resultados obtidos a partir desses mesmos testes.

Capítulo 5

VALIDAÇÃO

5.1 Considerações iniciais

As hipóteses levantadas nesta tese são:

- Ao se incluir recursos da fábrica como, o uso das rotas do chão de fábrica e o tempo de *setup* de máquina, além das variáveis habituais por meio de um sistema *fuzzy* em cascata alcançar-se-á um valor de *makespan* ou *tardiness* mais correto e que na média seja menor se comparado com propostas que não consideram os recursos em questão.
- Ao se usar informações de estados futuros do sistema de produção fabril por meio de uma estrutura baseada em espaço de estados alcançar-se-á um valor de *makespan* ou *tardiness* menor se comparado com propostas que não avaliam estados futuros.
- Ao se incluir outros recursos da fábrica por meio de um sistema *fuzzy* em cascata e usar informações de estados futuros do sistema de produção fabril por meio de uma estrutura baseada em espaço de estados, alcançar-se-á um valor de *makespan* ou *tardiness* mais correto e menor se comparado com os valores obtidos por meio das hipóteses separadas.

Este trabalho se pauta pelo método hipotético-dedutivo tendo em vista que permite corroborar uma hipótese dentro de um determinado contexto e contra determinados trabalhos alternativos. O resultado está sujeito ao momento em que ocorre, podendo no futuro, quando posto em comparação a outros concorrentes e/ou outros contextos, obter resultado diferente.

A validação é realizada pela comparação de seis métodos de despacho de AGVs:

- FIFO;
- Menor distância;
- *Fuzzy* proposto em (BENINCASA; MORANDIN; KATO, 2003) (*Fuzzy1*);
- Multiatributo *fuzzy* cascata proposto nesta tese (*Fuzzy 2*);
- *Fuzzy1* acrescido do sistema de predição proposto nesta tese;
- *Fuzzy2* acrescido do sistema de predição. Proposta completa da tese.

Para que a comparação tenha maior abrangência, são analisados 30 diferentes tipos de demanda com diferentes prazos de entrega (chamados nesta tese de pedidos) no cenário de fábrica 1 e 15 diferentes pedidos no cenário de fábrica 2. Para cada pedido de fábrica também é variada a quantidade de veículos disponíveis para transporte.

Neste capítulo, serão descritos o contexto da validação, o planejamento dos experimentos, seus resultados e as análises da execução dos experimentos planejados.

5.2 Verificação e Validação do Software

A verificação e validação do *software* tem como finalidade avaliar se o *software* desenvolvido está isento de erros de implementação, dessa forma aumentando a confiabilidade nos resultados obtidos na validação da proposta. Portanto testes de unidade e testes de sistema foram executados a fim de garantir que o *software* corresponda às especificações para o qual ele foi concebido.

Detalhes sobre a aplicação dos testes bem como os resultados obtidos estão contemplados no relatório técnico, que complementa esta tese e encontra-se disponível para consulta no repositório do laboratório Tear (<http://www.tear.dc.ufscar.br/>) ou pode ser solicitado ao autor dessa tese.

5.3 Validação da Proposta

Nesta validação, busca-se saber se o sistema multiatributo *fuzzy* cascata e a estrutura baseada em espaço de estados propostos, buscando o menor *makespan* e *tardiness*, é melhor do que os outros métodos de despacho de AGVs. A validação se dará pela comparação dos resultados obtidos de forma empírica e a análise foi realizada usando ferramentas de estatística.

A validação é realizada pela comparação de seis métodos de despacho aplicados na solução do problema de atribuição de tarefas aos veículos em *FMS* em dois cenários.

Cenários são os conjuntos de dados de entrada para cada problema, contendo os roteiros de todas as tarefas, indicando máquina e tempo de cada operação. Para que os experimentos tivessem maior abrangência, dois cenários de fábrica descritos na seção 4.2.1 foram utilizados.

5.3.1 Planejamento dos Experimentos

A comparação entre os trabalhos se dará aos pares, cenário a cenário. A comparação de apenas uma amostra de cada execução em cada cenário não é suficiente, pois este único valor pode não refletir a verdade.

A execução de um número maior de vezes, obtendo uma média e desvio padrão de cada população de resultados, permite uma comparação com melhor qualidade. Porém, se a diferença das médias e desvio padrão entre os trabalhos comparados não for grande o suficiente, não se pode garantir a constância dos resultados.

A diferença absoluta é a diferença entre os valores *makespan* ou *tardiness* alcançados pelo uso de um determinado método em comparação e os valores de *makespan* ou *tardiness* alcançados pelo método de despacho proposto.

Para que se obtenha um número de execuções que garanta a confiança nos resultados além da diferença absoluta, devem-se usar técnicas de estatística para calcular o número de amostras necessário para cada dupla de trabalhos comparados.

Entre as várias técnicas estatísticas disponíveis, escolheu-se o teste de hipóteses *t-student* por ser adequado ao problema (MONTGOMERY, 2005). O teste de hipóteses *t-student* fazendo inferência na diferença dos resultados de duas populações independentes com variâncias desconhecidas (ver Apêndice A, seção 7.3) possui os seguintes requisitos:

- Distribuição normal;

- Populações independentes.

Considera-se a suposição de normalidade por conta do Teorema de Limite Central, que afirma que toda distribuição tende a normalidade na medida em que aumenta (MONTGOMERY, 2005) e que, usualmente, se costuma considerar população acima de 30 indivíduos como populações grandes (MONTGOMERY; RUNGER, 2009)

Consideram-se as populações como sendo independentes por serem geradas aleatoriamente (MONTGOMERY, 2005).

O teste necessita também da definição de dois parâmetros:

- O nível de significância, que é a margem assumida para o erro Tipo I; e
- O poder de teste, que é a margem assumida para o erro Tipo II.

Definiu-se o nível de confiança em 3%. O poder de teste calculou-se em 97%, a ser garantido pelos tamanhos das populações amostrais a serem geradas na fase de cálculo de tamanho amostral.

Após gerarem-se as populações com o tamanho adequado e com tamanho mínimo para garantir a exigência de normalidade da distribuição (Teorema do Limite Central), aplica-se o teste de hipóteses *t-student*. A aplicação deste método está explicada no Apêndice A.

Com base nos resultados obtidos pela aplicação dos testes de hipóteses, pode-se concluir sobre a corroboração da hipótese e finaliza-se a validação.

5.3.2 Análise da hipótese

Para cada par de trabalhos comparados, usou-se o teste de hipótese *t-student* fazendo inferência na diferença para duas amostras independentes e com variâncias desconhecidas.

Este teste verificou se a hipótese de que o trabalho proposto nesta tese é melhor do que os métodos com os quais foi comparado. Para tal quatro hipóteses nulas foram cunhadas:

- O *makespan* da população amostral da proposta é menor ou igual o *makespan* da população amostral do outro método na comparação;
- O *makespan* da população amostral da proposta é igual o *makespan* da população amostral do outro método na comparação.
- O *tardiness* da população amostral da proposta é menor ou igual o *tardiness* da população amostral do outro método na comparação;

- O *tardiness* da população amostral da proposta é igual o *tardiness* da população amostral do outro método na comparação.

Ao aceitarmos a primeira e refutarmos a segunda, corroboramos que o *makespan* da população amostral da proposta é menor do que o *makespan* da população amostral do outro trabalho na comparação. Ao aceitarmos a terceira e refutarmos a quarta, corroboramos que o *tardiness* da população amostral da proposta é menor do que o *tardiness* da população amostral do outro trabalho na comparação. Esta afirmação encontra-se comprovada no Apêndice A.

5.3.3 Execução dos Experimentos

Como descrito anteriormente, a validação da proposta se dá em dois cenários de fábrica. Um fictício com seis estações de trabalho (fábrica 1) e cinco tipos de produtos e um baseado em uma fábrica moveleira simulada (fábrica 2), com setenta e três estações de trabalho e quinze tipos de produtos.

No sentido de iniciar as simulações foram criados 30 pedidos de produção para fábrica 1 e 15 pedidos de produção para fábrica 2. Para fábrica 1 os pedidos foram criados fixando o volume em 1875 produtos, assim tendo em média uma semana de produção, porém a quantidade de cada produto é gerada de maneira aleatória. Na Tabela 14 é apresentado o pedido 1 para a fábrica 1, os demais pedidos são apresentados no relatório técnico.

Tabela 14 – Pedido 1 para fábrica 1

OrderID	Products	Quantity	ReleaseDate	DueDate
Order1.1	Product1	35	03/10/2015 08:00	04/10/2015 19:00
Order2.1	Product1	30	03/10/2015 08:00	05/10/2015 00:10
Order3.1	Product1	25	03/10/2015 08:00	05/10/2015 01:00
Order4.1	Product1	30	03/10/2015 08:00	04/10/2015 18:00
Order5.1	Product1	40	03/10/2015 08:00	04/10/2015 19:00
Order1.2	Product2	110	03/10/2015 08:00	05/10/2015 02:50
Order2.2	Product2	165	03/10/2015 08:00	05/10/2015 05:00
Order3.2	Product2	80	03/10/2015 08:00	04/10/2015 19:20
Order4.2	Product2	75	03/10/2015 08:00	04/10/2015 17:50
Order5.2	Product2	135	03/10/2015 08:00	04/10/2015 22:00
Order1.3	Product3	140	03/10/2015 08:00	05/10/2015 01:50
Order2.3	Product3	105	03/10/2015 08:00	05/10/2015 04:00
Order3.3	Product3	110	03/10/2015 08:00	05/10/2015 03:20
Order4.3	Product3	70	03/10/2015 08:00	04/10/2015 21:00

Order5.3	Product3	80	03/10/2015 08:00	05/10/2015 00:50
Order1.4	Product4	105	03/10/2015 08:00	05/10/2015 04:50
Order2.4	Product4	135	03/10/2015 08:00	05/10/2015 01:30
Order3.4	Product4	95	03/10/2015 08:00	05/10/2015 02:50
Order4.4	Product4	55	03/10/2015 08:00	04/10/2015 23:00
Order5.4	Product4	145	03/10/2015 08:00	05/10/2015 05:40
Order1.5	Product5	20	03/10/2015 08:00	04/10/2015 17:00
Order2.5	Product5	15	03/10/2015 08:00	04/10/2015 16:20
Order3.5	Product5	20	03/10/2015 08:00	04/10/2015 14:00
Order4.5	Product5	25	03/10/2015 08:00	04/10/2015 15:20
Order5.5	Product5	30	03/10/2015 08:00	04/10/2015 17:00

Para fábrica 2 os pedidos foram criados fixando o volume em 2275 peças, que equivalem no final a 150 produtos. Os pedidos para a fábrica 2 seguem padrões de pedidos reais recebido pela fábrica, por exemplo, o pedido de uma mesa é acompanhado do pedido de quatro cadeiras. Na Tabela 15 é apresentado o pedido 1 para a fábrica 2, os demais pedidos são apresentados no relatório técnico.

Tabela 15 – Pedido 1 para fábrica 2

OrderID	PartType	Quantity	ReleaseDate	DueDate
Order1.1	encosto1	24	03/10/2015 08:00	08/11/2015 14:30
Order1.2	encosto2	8	03/10/2015 08:00	08/11/2015 14:30
Order1.3	encosto3	0	03/10/2015 08:00	08/11/2015 14:30
Order1.4	assento1	24	03/10/2015 08:00	08/11/2015 14:30
Order1.5	assento2	8	03/10/2015 08:00	08/11/2015 14:30
Order1.6	pes_da_frente_cadeira	64	03/10/2015 08:00	08/11/2015 14:30
Order1.7	pes_de_tras_cadeira	64	03/10/2015 08:00	08/11/2015 14:30
Order1.8	travessas_cadeira	128	03/10/2015 08:00	08/11/2015 14:30
Order1.9	travas_cadeira	128	03/10/2015 08:00	08/11/2015 14:30
Order2.1	encosto1	24	03/10/2015 08:00	05/11/2015 15:20
Order2.2	encosto2	8	03/10/2015 08:00	05/11/2015 15:20
Order2.3	encosto3	0	03/10/2015 08:00	05/11/2015 15:20
Order2.4	assento1	24	03/10/2015 08:00	05/11/2015 15:20
Order2.5	assento2	8	03/10/2015 08:00	05/11/2015 15:20
Order2.6	pes_da_frente_cadeira	64	03/10/2015 08:00	05/11/2015 15:20
Order2.7	pes_de_tras_cadeira	64	03/10/2015 08:00	05/11/2015 15:20
Order2.8	travessas_cadeira	128	03/10/2015 08:00	05/11/2015 15:20
Order2.9	travas_cadeira	128	03/10/2015 08:00	05/11/2015 15:20
Order3.1	encosto1	28	03/10/2015 08:00	07/11/2015 23:40
Order3.2	encosto2	8	03/10/2015 08:00	07/11/2015 23:40
Order3.3	encosto3	0	03/10/2015 08:00	07/11/2015 23:40
Order3.4	assento1	28	03/10/2015 08:00	07/11/2015 23:40

Order3.5	assento2	8	03/10/2015 08:00	07/11/2015 23:40
Order3.6	pes_da_frente_cadeira	72	03/10/2015 08:00	07/11/2015 23:40
Order3.7	pes_de_tras_cadeira	72	03/10/2015 08:00	07/11/2015 23:40
Order3.8	travessas_cadeira	144	03/10/2015 08:00	07/11/2015 23:40
Order3.9	travas_cadeira	144	03/10/2015 08:00	07/11/2015 23:40
Order4.1	tampo1	5	03/10/2015 08:00	07/11/2015 12:50
Order4.2	tampo2	1	03/10/2015 08:00	07/11/2015 12:50
Order4.3	tampo3	0	03/10/2015 08:00	07/11/2015 12:50
Order4.4	tampo4	1	03/10/2015 08:00	07/11/2015 12:50
Order4.5	tampo5	1	03/10/2015 08:00	07/11/2015 12:50
Order4.6	travessas1_mesa	20	03/10/2015 08:00	07/11/2015 12:50
Order4.7	travessas2_mesa	4	03/10/2015 08:00	07/11/2015 12:50
Order4.8	travessas3_mesa	0	03/10/2015 08:00	07/11/2015 12:50
Order4.9	travessas4_mesa	4	03/10/2015 08:00	07/11/2015 12:50
Order4.10	travessas5_mesa	4	03/10/2015 08:00	07/11/2015 12:50
Order4.11	pes_mesa	32	03/10/2015 08:00	07/11/2015 12:50
Order4.12	travas_mesa	32	03/10/2015 08:00	07/11/2015 12:50
Order5.1	tampo1	5	03/10/2015 08:00	08/11/2015 13:20
Order5.2	tampo2	1	03/10/2015 08:00	08/11/2015 13:20
Order5.3	tampo3	0	03/10/2015 08:00	08/11/2015 13:20
Order5.4	tampo4	1	03/10/2015 08:00	08/11/2015 13:20
Order5.5	tampo5	1	03/10/2015 08:00	08/11/2015 13:20
Order5.6	travessas1_mesa	20	03/10/2015 08:00	08/11/2015 13:20
Order5.7	travessas2_mesa	4	03/10/2015 08:00	08/11/2015 13:20
Order5.8	travessas3_mesa	0	03/10/2015 08:00	08/11/2015 13:20
Order5.9	travessas4_mesa	4	03/10/2015 08:00	08/11/2015 13:20
Order5.10	travessas5_mesa	4	03/10/2015 08:00	08/11/2015 13:20
Order5.11	pes_mesa	32	03/10/2015 08:00	08/11/2015 13:20
Order5.12	travas_mesa	32	03/10/2015 08:00	08/11/2015 13:20
Order6.1	tampo1	7	03/10/2015 08:00	06/11/2015 16:10
Order6.2	tampo2	2	03/10/2015 08:00	06/11/2015 16:10
Order6.3	tampo3	0	03/10/2015 08:00	06/11/2015 16:10
Order6.4	tampo4	0	03/10/2015 08:00	06/11/2015 16:10
Order6.5	tampo5	0	03/10/2015 08:00	06/11/2015 16:10
Order6.6	travessas1_mesa	28	03/10/2015 08:00	06/11/2015 16:10
Order6.7	travessas2_mesa	8	03/10/2015 08:00	06/11/2015 16:10
Order6.8	travessas3_mesa	0	03/10/2015 08:00	06/11/2015 16:10
Order6.9	travessas4_mesa	0	03/10/2015 08:00	06/11/2015 16:10
Order6.10	travessas5_mesa	0	03/10/2015 08:00	06/11/2015 16:10
Order6.11	pes_mesa	36	03/10/2015 08:00	06/11/2015 16:10
Order6.12	travas_mesa	36	03/10/2015 08:00	06/11/2015 16:10
Order7.1	placa_cabeceira1	2	03/10/2015 08:00	08/11/2015 13:50
Order7.2	placa_peseira1	2	03/10/2015 08:00	08/11/2015 13:50
Order7.3	placa_cabeceira2	2	03/10/2015 08:00	08/11/2015 13:50
Order7.4	placa_peseira2	2	03/10/2015 08:00	08/11/2015 13:50

Order7.5	placa_cabeceira3	2	03/10/2015 08:00	08/11/2015 13:50
Order7.6	placa_peseira3	2	03/10/2015 08:00	08/11/2015 13:50
Order7.7	placa_cabeceira4	1	03/10/2015 08:00	08/11/2015 13:50
Order7.8	placa_peseira4	1	03/10/2015 08:00	08/11/2015 13:50
Order7.9	placa_cabeceira5	1	03/10/2015 08:00	08/11/2015 13:50
Order7.10	placa_peseira5	1	03/10/2015 08:00	08/11/2015 13:50
Order7.11	pes_da_frente_cama	16	03/10/2015 08:00	08/11/2015 13:50
Order7.12	pes_de_tras_cama	16	03/10/2015 08:00	08/11/2015 13:50
Order7.13	barra_da_cama	16	03/10/2015 08:00	08/11/2015 13:50
Order7.14	travessa_estrado	32	03/10/2015 08:00	08/11/2015 13:50
Order7.15	ripas_estrado	80	03/10/2015 08:00	08/11/2015 13:50
Order8.1	placa_cabeceira1	2	03/10/2015 08:00	07/11/2015 22:50
Order8.2	placa_peseira1	2	03/10/2015 08:00	07/11/2015 22:50
Order8.3	placa_cabeceira2	2	03/10/2015 08:00	07/11/2015 22:50
Order8.4	placa_peseira2	2	03/10/2015 08:00	07/11/2015 22:50
Order8.5	placa_cabeceira3	2	03/10/2015 08:00	07/11/2015 22:50
Order8.6	placa_peseira3	2	03/10/2015 08:00	07/11/2015 22:50
Order8.7	placa_cabeceira4	1	03/10/2015 08:00	07/11/2015 22:50
Order8.8	placa_peseira4	1	03/10/2015 08:00	07/11/2015 22:50
Order8.9	placa_cabeceira5	1	03/10/2015 08:00	07/11/2015 22:50
Order8.10	placa_peseira5	1	03/10/2015 08:00	07/11/2015 22:50
Order8.11	pes_da_frente_cama	16	03/10/2015 08:00	07/11/2015 22:50
Order8.12	pes_de_tras_cama	16	03/10/2015 08:00	07/11/2015 22:50
Order8.13	barra_da_cama	16	03/10/2015 08:00	07/11/2015 22:50
Order8.14	travessa_estrado	32	03/10/2015 08:00	07/11/2015 22:50
Order8.15	ripas_estrado	80	03/10/2015 08:00	07/11/2015 22:50
Order9.1	placa_cabeceira1	1	03/10/2015 08:00	06/11/2015 19:20
Order9.2	placa_peseira1	1	03/10/2015 08:00	06/11/2015 19:20
Order9.3	placa_cabeceira2	1	03/10/2015 08:00	06/11/2015 19:20
Order9.4	placa_peseira2	1	03/10/2015 08:00	06/11/2015 19:20
Order9.5	placa_cabeceira3	1	03/10/2015 08:00	06/11/2015 19:20
Order9.6	placa_peseira3	1	03/10/2015 08:00	06/11/2015 19:20
Order9.7	placa_cabeceira4	3	03/10/2015 08:00	06/11/2015 19:20
Order9.8	placa_peseira4	3	03/10/2015 08:00	06/11/2015 19:20
Order9.9	placa_cabeceira5	3	03/10/2015 08:00	06/11/2015 19:20
Order9.10	placa_peseira5	3	03/10/2015 08:00	06/11/2015 19:20
Order9.11	pes_da_frente_cama	18	03/10/2015 08:00	06/11/2015 19:20
Order9.12	pes_de_tras_cama	18	03/10/2015 08:00	06/11/2015 19:20
Order9.13	barra_da_cama	18	03/10/2015 08:00	06/11/2015 19:20
Order9.14	travessa_estrado	36	03/10/2015 08:00	06/11/2015 19:20
Order9.15	ripas_estrado	90	03/10/2015 08:00	06/11/2015 19:20

Todos os pedidos da fábrica 1 foram simulados variando a quantidade de AGVs em 1, 2, 3, 4, 5, 7, 10, 15 e 20 AGVs. Cada variação é simulada com os métodos de despacho: FIFO; Menor distância; *Fuzzy* proposto em (BENINCASA; MORANDIN; KATO, 2003) (que será chamado de *Fuzzy1*); Multiatributo *fuzzy* cascata proposto nesta tese (que será chamado de *Fuzzy2*); *Fuzzy1* acrescido do sistema de predição proposto nesta tese; *Fuzzy2* acrescido do sistema de predição, proposta completa da tese.

O valor de *makespan* é equivalente ao tempo de conclusão da última tarefa do ambiente de manufatura. O *makespan* mínimo geralmente implica em uma boa utilização das estações de trabalho (PINEDO, 2008), (NETO, 2010). O valor de *makespan* pode ser definido como $\max\{C_i\}$ ou C_{max} quando o tempo inicial for zero, ou definido pela equação a baixo.

$$C_{max} - T_0 \quad (2)$$

O valor de *tardiness* é equivalente ao tempo de atraso para conclusão de uma determinada tarefa em relação a sua data de entrega. O valor de *tardiness* é calculado pela seguinte equação.

$$T_i = \max\{C_i - d_i, 0\} \quad (3)$$

Onde:

- C_i equivale a data de conclusão da tarefa i
- d_i equivale a data de entrega da tarefa i
- T_0 é o tempo de início da produção

Para avaliação de desempenho é usado o valor de *tardiness* total, que equivale a soma do *tardiness* de todas as tarefas.

$$\sum T_i \quad (4)$$

Devido ao fato de que os testes foram iniciados no dia 03 de outubro de 2015, em todas as simulações é mantida essa data de início. Na Tabela 16 são apresentados os resultados de

makespan para do pedido 1 da fábrica 1, os resultados dos outros 29 pedidos encontram-se no relatório técnico. A normalização é calculada para realização dos testes estatísticos e segue a seguinte inequação:

$$M_i / ((\sum M) / 6) \quad (5)$$

Onde:

- M equivale ao valor do *makespan*
- M_i equivale ao valor do *makespan* encontrado na execução de um determinado método de despacho de AGVs

Na Tabela 17 são apresentados os valores médios dos trinta pedidos. Na Tabela 18 são apresentados os ganhos percentuais na comparação por pares.

Tabela 16 – Resultados de makespan do pedido 1

TESTE NA FÁBRICA 1 - PEDIDO 01									
	1 AGV			2 AGVs			3 AGVs		
	Finalizado em:	Tempo	Normal.	Finalizado em:	Tempo	Normal.	Finalizado em:	Tempo	Normal.
Fifo	5/10/15 11:08 PM	63:08:24	1,1507	5/10/15 8:24 AM	48:24:00	1,0842	5/10/15 7:46 AM	47:46:56	1,0847
Menor Dist.	5/10/15 8:38 PM	60:38:49	1,1053	5/10/15 8:16 AM	48:16:28	1,0814	5/10/15 7:46 AM	47:46:56	1,0847
Fuzzy1	5/10/15 1:27 PM	53:27:16	0,9742	5/10/15 7:56 AM	47:56:20	1,0739	5/10/15 5:16 AM	45:16:34	1,0278
Fuzzy1+Predição	5/10/15 1:00 PM	53:00:47	0,9662	5/10/15 7:55 AM	47:55:09	1,0734	5/10/15 4:18 AM	44:18:56	1,0060
Fuzzy2	5/10/15 10:52 AM	50:52:06	0,9271	4/10/15 10:15 PM	38:15:33	0,8570	4/10/15 11:35 PM	39:35:27	0,8988
Fuzzy2+Predição	5/10/15 8:05 AM	48:05:58	0,8766	4/10/15 9:03 PM	37:03:19	0,8301	4/10/15 11:33 PM	39:33:28	0,8980

	4 AGVss			5 AGVs			7 AGVs		
	Finalizado em:	Tempo	Normal.	Finalizado em:	Tempo	Normal.	Finalizado em:	Tempo	Normal.
Fifo	5/10/15 7:43 AM	47:43:47	1,0911	5/10/15 7:41 AM	47:41:12	1,0901	5/10/15 7:38 AM	47:38:53	1,0804
Menor Dist.	5/10/15 7:43 AM	47:43:27	1,0909	5/10/15 7:41 AM	47:41:12	1,0901	5/10/15 7:38 AM	47:38:53	1,0804
Fuzzy1	5/10/15 5:15 AM	45:15:12	1,0344	5/10/15 5:14 AM	45:14:18	1,0341	5/10/15 5:09 AM	45:09:25	1,0239
Fuzzy1+Predição	5/10/15 3:40 AM	43:40:27	0,9983	5/10/15 3:14 AM	43:14:05	0,9883	5/10/15 2:51 AM	42:51:05	0,9717
Fuzzy2	4/10/15 11:03 PM	39:03:34	0,8929	4/10/15 11:20 PM	39:20:18	0,8993	5/10/15 12:40 AM	40:40:00	0,9221
Fuzzy2+Predição	4/10/15 11:02 PM	39:02:19	0,8924	4/10/15 11:17 PM	39:17:09	0,8981	5/10/15 12:38 AM	40:38:12	0,9214

	10 AGVs			15 AGVs			20 AGVs		
	Finalizado em:	Tempo	Normal.	Finalizado em:	Tempo	Normal.	Finalizado em:	Tempo	Normal.
Fifo	5/10/15 7:37 AM	47:37:09	1,0760	5/10/15 7:36 AM	47:36:35	1,0750	5/10/15 7:36 AM	47:36:22	1,0750
Menor Dist.	5/10/15 7:37 AM	47:37:09	1,0760	5/10/15 7:36 AM	47:36:35	1,0750	5/10/15 7:36 AM	47:36:22	1,0750
Fuzzy1	5/10/15 5:08 AM	45:08:15	1,0199	5/10/15 5:06 AM	45:06:53	1,0186	5/10/15 5:06 AM	45:06:31	1,0186
Fuzzy1+Predição	5/10/15 2:33 AM	42:33:47	0,9617	5/10/15 2:30 AM	42:30:43	0,9599	5/10/15 2:31 AM	42:31:15	0,9601
Fuzzy2	5/10/15 1:16 AM	41:16:06	0,9325	5/10/15 1:26 AM	41:26:39	0,9358	5/10/15 1:26 AM	41:26:21	0,9357
Fuzzy2+Predição	5/10/15 1:20 AM	41:20:05	0,9340	5/10/15 1:26 AM	41:26:39	0,9358	5/10/15 1:26 AM	41:26:21	0,9357

Tabela 17 – Resultados de *makespan* médios dos testes realizados na fábrica 1

MÉDIA GERAL									
	1 AGV			2 AGVs			3 AGVs		
	SOMA	MÉDIA	Média Normal.	SOMA	MÉDIA	Média Normal.	SOMA	MÉDIA	Média Normal.
Fifo	23/3/00 10:07 PM	67:08:14	1,1739	7/3/00 3:31 AM	53:43:04	1,1111	6/3/00 3:26 AM	52:54:53	1,1118
Menor Dist.	17/3/00 4:40 AM	61:45:21	1,0805	6/3/00 5:26 AM	52:58:54	1,0964	6/3/00 12:44 AM	52:49:30	1,1101
Fuzzy1	11/3/00 11:27 AM	57:10:56	0,9998	4/3/00 1:09 PM	51:38:19	1,0681	1/3/00 11:04 PM	49:34:09	1,0399
Fuzzy1+Predição	10/3/00 12:03 AM	56:00:07	0,9791	29/2/00 10:02 AM	48:20:05	0,9984	25/2/00 11:25 PM	45:34:50	0,9551
Fuzzy2	5/3/00 11:05 PM	52:46:11	0,9224	21/2/00 8:40 PM	42:17:21	0,8731	22/2/00 8:38 AM	42:41:17	0,8936
Fuzzy2+Predição	29/2/00 9:58 AM	48:19:57	0,8443	20/2/00 3:27 PM	41:18:54	0,8528	22/2/00 2:32 AM	42:29:05	0,8894

	4 AGVs			5 AGVs			7 AGVs		
	SOMA	MÉDIA	Média Normal.	SOMA	MÉDIA	Média Normal.	SOMA	MÉDIA	Média Normal.
Fifo	5/3/00 11:50 PM	52:47:40	1,1148	5/3/00 9:56 PM	52:43:54	1,1163	5/3/00 8:11 PM	52:40:22	1,1188
Menor Dist.	5/3/00 10:58 PM	52:45:57	1,1143	5/3/00 9:56 PM	52:43:52	1,1163	5/3/00 7:31 PM	52:39:03	1,1185
Fuzzy1	1/3/00 12:23 PM	49:12:48	1,0377	1/3/00 5:40 AM	48:59:20	1,0355	29/2/00 6:52 PM	48:37:46	1,0312
Fuzzy1+Predição	24/2/00 10:12 PM	44:44:25	0,9421	24/2/00 8:34 AM	44:17:08	0,9345	23/2/00 3:18 PM	43:42:37	0,9254
Fuzzy2	22/2/00 7:47 AM	42:39:35	0,8971	22/2/00 8:40 AM	42:41:22	0,8998	22/2/00 10:51 AM	42:45:43	0,9046
Fuzzy2+Predição	22/2/00 3:22 AM	42:30:45	0,8940	22/2/00 5:32 AM	42:35:04	0,8976	22/2/00 6:36 AM	42:37:13	0,9016

	10 AGVs			15 AGVs			20 AGVs		
	SOMA	MÉDIA	Média Normal.	SOMA	MÉDIA	Média Normal.	SOMA	MÉDIA	Média Normal.
Fifo	5/3/00 7:56 PM	52:39:53	1,1200	5/3/00 7:43 PM	52:39:28	1,1203	5/3/00 7:37 PM	52:39:14	1,1207
Menor Dist.	5/3/00 7:56 PM	52:39:53	1,1200	5/3/00 7:43 PM	52:39:28	1,1203	5/3/00 7:37 PM	52:39:14	1,1207
Fuzzy1	29/2/00 4:26 PM	48:32:54	1,0305	29/2/00 2:58 PM	48:29:57	1,0299	29/2/00 2:02 PM	48:28:05	1,0296
Fuzzy1+Predição	23/2/00 5:21 AM	43:22:43	0,9194	23/2/00 3:02 AM	43:18:05	0,9182	23/2/00 1:20 AM	43:14:41	0,9171
Fuzzy2	22/2/00 11:14 AM	42:46:30	0,9062	22/2/00 11:11 AM	42:46:22	0,9067	22/2/00 10:53 AM	42:45:47	0,9068
Fuzzy2+Predição	22/2/00 7:56 AM	42:39:53	0,9039	22/2/00 8:17 AM	42:40:35	0,9046	22/2/00 8:37 AM	42:41:16	0,9052

Tabela 18 – Resultados dos ganhos percentuais para *makespan* dos testes realizados na fábrica 1

	1 AGV Ganho	2 AGVs Ganho	3 AGVs Ganho	4 AGVs Ganho	5 AGVs Ganho	7 AGVs Ganho	10 AGVs Ganho	15 AGVs Ganho	20 AGVs Ganho	MÉDIA DE GANHO
<i>Fuzzy2 -> Fifo</i>	21,4002	21,2753	19,3267	19,1966	19,0440	18,8158	18,7785	18,7720	18,7847	19,4882
<i>Fuzzy2 -> Menor Dist.</i>	14,5511	20,1816	19,1897	19,1525	19,0433	18,7818	18,7785	18,7719	18,7847	18,5817
<i>Fuzzy2 -> Fuzzy1</i>	7,7164	18,1059	13,8817	13,3165	12,8590	12,0655	11,8918	11,8073	11,7706	12,6016
<i>Fuzzy1+Pred. ->Fuzzy1</i>	2,0641	6,3983	8,0463	9,0887	9,6006	10,1158	10,6485	10,7175	10,7769	8,6063
<i>Fuzzy2+Pred. ->Fuzzy2</i>	8,4086	2,3032	0,4761	0,3454	0,2457	0,3317	0,2575	0,2257	0,1763	1,4189
<i>Fuzzy2+Pred. -> Fuzzy1+Pred.</i>	13,6947	14,5230	6,7919	4,9798	3,8413	2,4936	1,6455	1,4435	1,2880	5,6335

Na Figura 58 é apresentado um gráfico com os valores médios de *makespan* encontrados com cada método de despacho de AGVs nos testes realizados na fábrica 1.

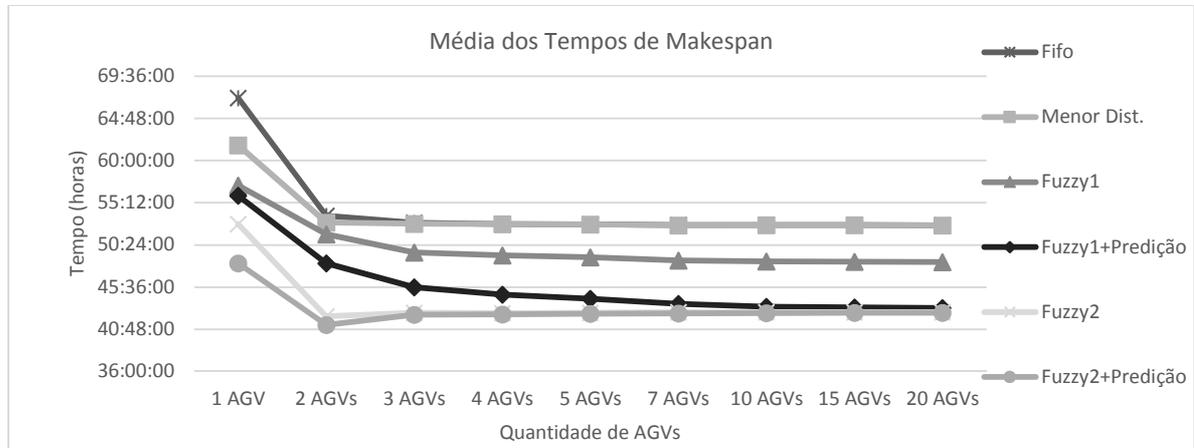


Figura 58 – Valores médios de *makespan* dos testes realizados na fábrica 1

Na Tabela 19 são apresentados os resultados de *tardiness* para do pedido 1 da fábrica 1, os resultados dos outros 29 pedidos encontram-se no relatório técnico. Na Tabela 20 são apresentados os valores médios dos trinta pedidos. Na Tabela 21 são apresentados os ganhos percentuais na comparação por pares.

Tabela 19 – Resultados de *tardiness* do pedido 1

TESTE NA FÁBRICA 1 - PEDIDO 01						
	1 AGV		2 AGVs		3 AGVs	
	Soma Ti	Normal.	Soma Ti	Normal.	Soma Ti	Normal.
Fifo	52:30:08	1,3048	31:34:17	1,2110	29:49:46	1,1904
Menor Dist.	47:55:39	1,1911	31:10:19	1,1957	30:15:42	1,2077
<i>Fuzzy</i>	37:55:48	0,9426	30:28:52	1,1692	26:56:40	1,0753
<i>Fuzzy+Predição</i>	37:18:59	0,9274	30:40:16	1,1765	25:46:17	1,0285
<i>Fuzzy2</i>	34:49:46	0,8656	17:10:17	0,6587	18:31:31	0,7393
<i>Fuzzy2+Predição</i>	30:55:31	0,7685	15:21:11	0,5889	19:00:55	0,7589

	4 AGVss		5 AGVs		7 AGVs	
	Soma Ti	Normal.	Soma Ti	Normal.	Soma Ti	Normal.
Fifo	29:45:25	1,2197	30:20:36	1,2330	29:51:33	1,2018
Menor Dist.	30:36:42	1,2548	30:33:30	1,2417	30:04:26	1,2104
<i>Fuzzy</i>	26:20:23	1,0797	26:42:01	1,0849	26:01:00	1,0471
<i>Fuzzy+Predição</i>	23:59:07	0,9832	23:23:00	0,9502	22:51:29	0,9200
<i>Fuzzy2</i>	18:03:00	0,7399	18:10:37	0,7386	20:00:36	0,8054
<i>Fuzzy2+Predição</i>	17:38:04	0,7228	18:29:53	0,7516	20:15:29	0,8153

	10 AGVs		15 AGVs		20 AGVs	
	Soma Ti	Normal.	Soma Ti	Normal.	Soma Ti	Normal.
Fifo	29:23:24	1,1629	30:26:57	1,2054	30:00:55	1,1897
Menor Dist.	30:27:45	1,2053	29:22:37	1,1629	30:13:47	1,1982
<i>Fuzzy</i>	26:44:56	1,0584	25:57:32	1,0276	25:57:02	1,0286
<i>Fuzzy+Predição</i>	23:07:08	0,9147	22:23:35	0,8865	22:34:07	0,8946
<i>Fuzzy2</i>	20:50:25	0,8246	21:41:39	0,8588	21:13:44	0,8414
<i>Fuzzy2+Predição</i>	21:05:01	0,8342	21:41:39	0,8588	21:22:53	0,8475

Tabela 20 – Resultados médios de *tardiness* dos testes realizados na fábrica 1

MÉDIA GERAL						
	1 AGV		2 AGVs		3 AGVs	
	MÉDIA (Soma Ti)	Média Normal.	MÉDIA (Soma Ti)	Média Normal.	MÉDIA (Soma Ti)	Média Normal.
Fifo	53:21:35	1,3580	34:38:40	1,2878	33:30:26	1,2956
Menor Dist.	45:51:44	1,1666	33:29:06	1,2450	33:22:13	1,2893
<i>Fuzzy</i>	39:20:52	0,9975	31:38:14	1,1740	28:42:15	1,0965
<i>Fuzzy+Predição</i>	37:42:12	0,9547	27:04:29	1,0008	23:12:30	0,8831
<i>Fuzzy2</i>	33:19:32	0,8438	18:36:51	0,6746	19:09:10	0,7240
<i>Fuzzy2+Predição</i>	27:00:31	0,6795	17:17:25	0,6177	18:50:16	0,7115

	4 AGVss		5 AGVs		7 AGVs	
	MÉDIA (Soma Ti)	Média Normal.	MÉDIA (Soma Ti)	Média Normal.	MÉDIA (Soma Ti)	Média Normal.
Fifo	33:17:13	1,3018	33:15:57	1,3073	33:07:31	1,3151
Menor Dist.	33:15:14	1,3000	33:22:16	1,3112	33:05:38	1,3154
<i>Fuzzy</i>	28:23:16	1,0942	27:58:28	1,0827	27:34:36	1,0777
<i>Fuzzy+Predição</i>	22:02:31	0,8470	21:29:14	0,8285	20:41:15	0,8045
<i>Fuzzy2</i>	19:08:16	0,7324	19:08:27	0,7359	19:17:36	0,7481
<i>Fuzzy2+Predição</i>	18:56:59	0,7245	19:06:12	0,7344	19:03:45	0,7392

	10 AGVs		15 AGVs		20 AGVs	
	MÉDIA (Soma Ti)	Média Normal.	MÉDIA (Soma Ti)	Média Normal.	MÉDIA (Soma Ti)	Média Normal.
Fifo	33:08:35	1,3220	33:12:21	1,3274	33:12:10	1,3265
Menor Dist.	33:00:36	1,3164	33:07:31	1,3242	33:14:55	1,3284
<i>Fuzzy</i>	27:23:08	1,0736	27:12:26	1,0692	27:11:51	1,0673
<i>Fuzzy+Predição</i>	20:15:16	0,7902	20:01:40	0,7829	19:56:56	0,7775
<i>Fuzzy2</i>	19:19:29	0,7519	19:16:44	0,7505	19:19:16	0,7517
<i>Fuzzy2+Predição</i>	19:11:00	0,7458	19:08:54	0,7457	19:14:59	0,7485

Tabela 21 – Resultados dos ganhos percentuais para *tardiness* dos testes realizados na fábrica 1

	1 AGV Ganho	2 AGVs Ganho	3 AGVs Ganho	4 AGVs Ganho	5 AGVs Ganho	7 AGVs Ganho	10 AGVs Ganho	15 AGVs Ganho	20 AGVs Ganho	MÉDIA DE GANHO
<i>Fuzzy 2 -> Fifo</i>	37,6807	45,7677	42,7341	42,6194	41,9314	41,7137	41,1335	41,7035	41,7114	41,8884
<i>Fuzzy 2 -> Menor Dist.</i>	27,4705	44,3204	42,6140	42,5631	42,3211	41,8227	41,0826	41,6004	41,6259	40,6023
<i>Fuzzy 2 -> Fuzzy</i>	16,0004	41,1012	33,4388	32,3455	31,4191	30,0356	29,3156	28,9778	29,0762	30,1900
<i>Fuzzy1+Pred. ->Fuzzy1</i>	4,7645	14,7565	19,1624	21,9171	23,7512	24,8115	26,4593	26,4435	26,8708	20,9930
<i>Fuzzy2+Pred. ->Fuzzy2</i>	18,4361	7,8030	1,1280	0,8962	0,8543	1,1165	1,2539	0,8404	0,4947	3,6470
<i>Fuzzy2+Pred. -> Fuzzy1+Pred.</i>	28,0590	36,2967	18,5894	14,1320	10,8249	7,9870	5,0893	4,2569	3,4955	14,3034

Na Figura 59 é apresentado um gráfico com os valores médios de *tardiness* encontrados com cada método de despacho de AGVs nos testes realizados na fábrica 1.

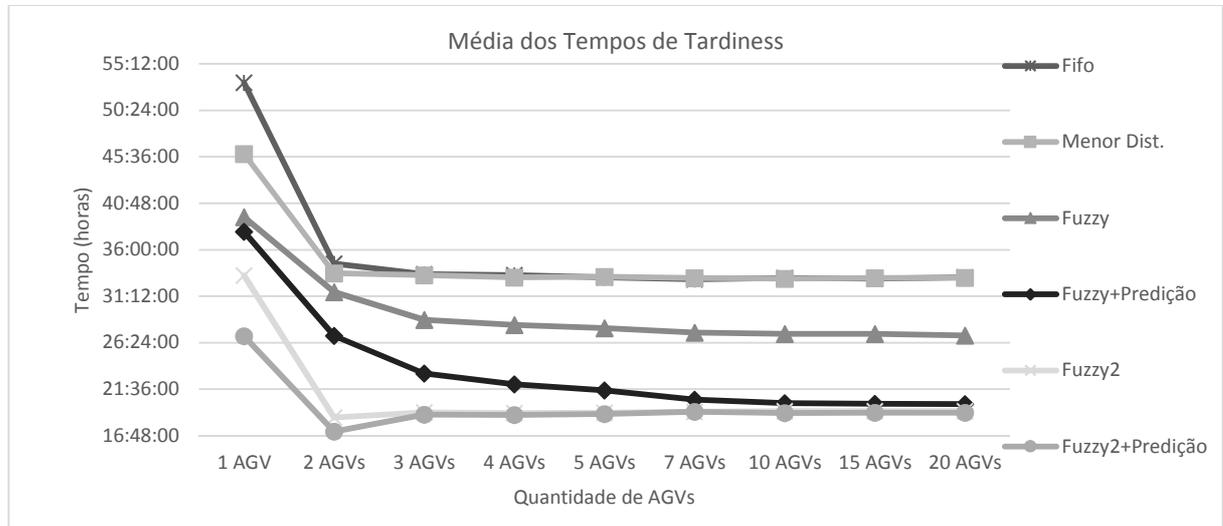


Figura 59 – Valores médios de *tardiness* dos testes realizados na fábrica 1

Na Tabela 22 são apresentados os resultados de *makespan* para do pedido 1 da fábrica 2, os resultados dos outros 14 pedidos encontram-se no relatório técnico. Na Tabela 23 são apresentados os valores médios dos quinze pedidos. Na Tabela 24 são apresentados os ganhos percentuais na comparação por pares.

Tabela 22 – Resultados de *makespan* do pedido 1 dos testes realizados na fábrica 2

TESTE NA FÁBRICA 2 - PEDIDO 01									
	4 AGVs			8 AGVs			12 AGVs		
	Finalizado em:	Tempo	Normal.	Finalizado em:	Tempo	Normal.	Finalizado em:	Tempo	Normal.
Fifo	21/1/16 12:36 PM	2644:36:18	1,1543	28/11/15 8:42 PM	1356:42:51	1,0781	21/11/15 5:58 PM	1185:58:56	1,1036
Menor Dist.	16/1/16 10:25 AM	2522:25:14	1,1010	28/11/15 2:07 PM	1350:07:22	1,0728	21/11/15 2:01 PM	1182:01:17	1,0999
Fuzzy1	9/1/16 11:02 AM	2355:02:58	1,0280	27/11/15 2:25 PM	1326:25:47	1,0540	18/11/15 11:03 AM	1107:03:18	1,0302
Fuzzy1+Predição	4/1/16 2:36 AM	2226:36:55	0,9719	25/11/15 12:41 PM	1276:41:58	1,0145	14/11/15 7:26 PM	1019:26:31	0,9486
Fuzzy2	27/12/15 11:23 AM	2043:23:14	0,8919	19/11/15 2:37 PM	1134:37:57	0,9016	13/11/15 11:11 AM	987:11:08	0,9186
Fuzzy2+Predição	23/12/15 5:55 PM	1953:55:27	0,8529	18/11/15 10:17 AM	1106:17:56	0,8791	12/11/15 2:11 PM	966:11:58	0,8991

	16 AGVs			20 AGVs		
	Finalizado em:	Tempo	Normal.	Finalizado em:	Tempo	Normal.
Fifo	20/11/15 10:18 AM	1154:18:40	1,1481	15/11/15 12:32 PM	1036:32:44	1,1298
Menor Dist.	19/11/15 1:01 PM	1133:01:34	1,1270	12/11/15 3:53 PM	967:53:27	1,0550
Fuzzy1	13/11/15 10:38 AM	986:38:18	0,9814	11/11/15 10:34 AM	938:34:11	1,0230
Fuzzy1+Predição	11/11/15 4:33 PM	944:33:42	0,9395	8/11/15 10:36 AM	866:36:27	0,9446
Fuzzy2	11/11/15 12:44 PM	940:44:38	0,9357	7/11/15 6:34 PM	850:34:47	0,9271
Fuzzy2+Predição	8/11/15 4:59 PM	872:59:00	0,8683	7/11/15 12:25 PM	844:25:28	0,9204

	28 AGVs			40 AGVs		
	Finalizado em:	Tempo	Normal.	Finalizado em:	Tempo	Normal.
Fifo	7/11/15 3:22 PM	847:22:32	1,0277	7/11/15 10:18 AM	842:18:32	1,0377
Menor Dist.	6/11/15 1:35 PM	821:35:23	0,9964	6/11/15 10:36 AM	818:36:15	1,0084
Fuzzy1	6/11/15 12:31 PM	820:31:22	0,9951	6/11/15 10:36 AM	818:36:15	1,0084
Fuzzy1+Predição	6/11/15 12:28 PM	820:28:49	0,9951	5/11/15 5:34 PM	801:34:37	0,9875
Fuzzy2	6/11/15 10:58 AM	818:58:41	0,9932	5/11/15 10:49 AM	794:49:11	0,9791
Fuzzy2+Predição	6/11/15 10:24 AM	818:24:55	0,9925	5/11/15 10:33 AM	794:33:31	0,9788

Tabela 23 – Resultados de *makespan* médios dos testes realizados na fábrica 2

MÉDIA GERAL									
	4 AGVs			8 AGVs			12 AGVs		
	SOMA	MÉDIA	Média Normal.	SOMA	MÉDIA	Média Normal.	SOMA	MÉDIA	Média Normal.
Fifo	8/7/04 9:31 PM	2643:02:06	1,1519	7/5/02 10:46 AM	1373:31:07	1,0929	12/1/02 4:56 PM	1189:55:47	1,1060
Menor Dist.	16/4/04 1:52 AM	2508:55:30	1,0934	27/4/02 12:34 AM	1356:50:17	1,0797	7/1/02 5:03 PM	1181:56:13	1,0986
Fuzzy1	9/1/04 12:59 PM	2352:52:00	1,0254	7/4/02 12:42 PM	1325:38:51	1,0549	23/11/01 1:45 AM	1108:55:02	1,0307
Fuzzy1+Predição	10/11/03 10:45 AM	2256:43:01	0,9835	5/3/02 2:12 PM	1272:56:52	1,0130	5/10/01 10:12 AM	1031:04:49	0,9583
Fuzzy2	29/6/03 1:04 AM	2041:40:17	0,8898	2/12/01 10:15 AM	1123:53:02	0,8943	2/9/01 1:37 AM	977:42:29	0,9087
Fuzzy2+Predição	11/5/03 5:58 PM	1964:23:55	0,8561	9/11/01 12:55 PM	1087:15:43	0,8651	25/8/01 4:15 PM	965:53:04	0,8978

	16 AGVs			20 AGVs		
	SOMA	MÉDIA	Média Normal.	SOMA	MÉDIA	Média Normal.
Fifo	21/12/01 12:06 PM	1154:24:24	1,1468	8/10/01 5:18 AM	1035:33:13	1,1279
Menor Dist.	7/12/01 8:26 AM	1131:45:46	1,1244	30/8/01 6:29 AM	973:13:59	1,0599
Fuzzy1	7/9/01 6:07 PM	986:48:32	0,9803	7/8/01 5:23 AM	936:21:34	1,0197
Fuzzy1+Predição	13/8/01 9:54 PM	947:03:36	0,9408	24/6/01 8:36 AM	866:10:26	0,9434
Fuzzy2	10/8/01 2:10 PM	941:44:41	0,9356	15/6/01 6:13 PM	852:24:54	0,9284
Fuzzy2+Predição	1/7/01 3:13 PM	877:48:53	0,8720	11/6/01 10:23 AM	845:29:35	0,9209

	28 AGVs			40 AGVs		
	SOMA	MÉDIA	Média Normal.	SOMA	MÉDIA	Média Normal.
Fifo	12/6/01 6:38 PM	847:38:32	1,0271	9/6/01 11:47 AM	842:23:10	1,0368
Menor Dist.	29/5/01 1:47 AM	824:07:11	0,9986	25/5/01 8:20 PM	818:57:22	1,0080
Fuzzy1	27/5/01 9:56 AM	821:27:48	0,9954	25/5/01 7:49 PM	818:55:19	1,0079
Fuzzy1+Predição	26/5/01 10:15 PM	820:41:03	0,9945	15/5/01 5:57 PM	802:47:51	0,9881
Fuzzy2	25/5/01 11:03 PM	819:08:15	0,9926	12/5/01 3:57 AM	797:03:51	0,9810
Fuzzy2+Predição	25/5/01 1:32 PM	818:30:10	0,9918	10/5/01 5:02 PM	794:44:09	0,9782

Tabela 24 – Resultados dos ganhos percentuais para *makespan* dos testes realizados na fábrica 2

	4 AGVs Ganho	8 AGVs Ganho	12 AGVs Ganho	16 AGVs Ganho	20 AGVs Ganho	28 AGVs Ganho	40 AGVs Ganho	MÉDIA DE GANHO
<i>Fuzzy2 -> Fifo</i>	22,75276279	18,17484146	17,83481189	18,42175559	17,6850816	3,362824504	5,380207965	11,5124762
<i>Fuzzy2 -> Menor Dist.</i>	18,62365577	17,16890101	17,27916698	16,78956029	12,41409099	0,604539573	2,673152642	9,505896361
<i>Fuzzy2 -> Fuzzy1</i>	13,22621168	15,22001163	11,83218177	4,566646978	8,964977025	0,283123935	2,669107424	6,306917826
<i>Fuzzy1+Pred. ->Fuzzy1</i>	4,086489645	3,975401818	7,019183178	4,028000846	7,495572539	0,094853315	1,969004554	3,185389544
<i>Fuzzy2+Pred. ->Fuzzy2</i>	3,784777538	3,258515342	1,209322975	6,788455981	0,812030551	0,077502472	0,292092787	1,802521961
<i>Fuzzy2+Pred. -> Fuzzy1+Pred.</i>	12,95325011	14,58707353	6,323059391	7,311608964	2,387601053	0,265805789	1,004171571	4,981396711

Na Figura 60 é apresentado um gráfico com os valores médios de *makespan* encontrados com cada método de despacho de AGVs nos testes realizados na fábrica 2.

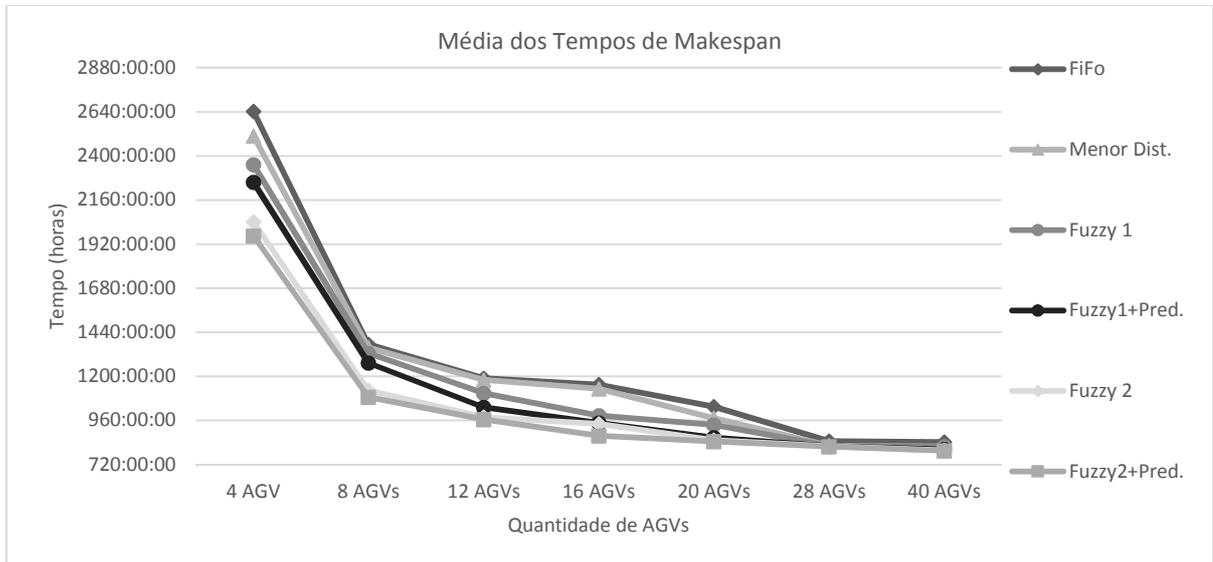


Figura 60 – Valores médios de *makespan* dos testes realizados na fábrica 2

Na Tabela 25 são apresentados os resultados de *tardiness* para do pedido 1 da fábrica 2, os resultados dos outros 14 pedidos encontram-se no relatório técnico. Na Tabela 26 são apresentados os valores médios dos quinze pedidos. Na Tabela 27 são apresentados os ganhos percentuais na comparação por pares.

Tabela 25 – Resultados de *tardiness* do pedido 1 dos testes realizados na fábrica 2

TESTE NA FÁBRICA 2 - PEDIDO 01								
	4 AGVs		8 AGVs		12 AGVs		16 AGVs	
	Soma Ti	Normal.	Soma Ti	Normal.	Soma Ti	Normal.	Soma Ti	Normal.
Fifo	4539:22:06	1,2325	1371:09:25	1,2028	958:53:21	1,4142	887:26:40	1,7297
Menor Dist.	4256:01:32	1,1556	1376:58:25	1,2079	937:35:09	1,3828	834:13:55	1,6260
<i>Fuzzy</i>	3858:10:33	1,0475	1317:44:27	1,1560	753:55:05	1,1119	468:15:45	0,9127
<i>Fuzzy+Predição</i>	3568:12:17	0,9688	1183:52:05	1,0385	541:28:02	0,7986	355:48:34	0,6935
<i>Fuzzy2</i>	3047:55:55	0,8276	834:53:42	0,7324	465:52:25	0,6871	352:06:44	0,6863
<i>Fuzzy2+Predição</i>	2828:44:51	0,7680	755:08:07	0,6624	410:29:26	0,6054	180:26:33	0,3517

	20 AGVs		28 AGVs		40 AGVs	
	Soma Ti	Normal.	Soma Ti	Normal.	Soma Ti	Normal.
Fifo	590:39:30	2,0148	118:11:02	1,8919	105:43:12	3,0695
Menor Dist.	421:23:37	1,4374	54:45:03	0,8765	47:35:56	1,3820
<i>Fuzzy</i>	341:07:45	1,1636	52:45:42	0,8446	47:47:30	1,3876
<i>Fuzzy+Predição</i>	168:11:07	0,5737	52:52:02	0,8463	5:32:30	0,1609
<i>Fuzzy2</i>	127:05:28	0,4335	49:06:42	0,7862	0:00:00	0,0000
<i>Fuzzy2+Predição</i>	110:28:24	0,3768	47:07:57	0,7545	0:00:00	0,0000

Tabela 26 – Resultados médios de *tardiness* dos testes realizados na fábrica 2.

MÉDIA GERAL								
	4 AGVs		8 AGVs		12 AGVs		16 AGVs	
	MÉDIA (Soma Ti)	Média Normal.						
Fifo	4531:01:45	1,2337	1392:13:05	1,2642	935:17:37	1,4337	847:04:39	1,7636
Menor Dist.	4196:53:06	1,1428	1345:58:34	1,2232	917:00:42	1,4054	794:26:31	1,6529
<i>Fuzzy</i>	3817:09:26	1,0393	1271:20:26	1,1552	735:47:48	1,1260	434:00:58	0,8972
<i>Fuzzy+Predição</i>	3589:43:17	0,9774	1144:05:36	1,0393	543:22:54	0,8276	335:32:43	0,6892
<i>Fuzzy2</i>	3048:05:30	0,8299	772:45:20	0,7010	411:21:31	0,6266	321:58:18	0,6627
<i>Fuzzy2+Predição</i>	2853:18:29	0,7769	682:11:48	0,6171	381:24:40	0,5807	164:09:04	0,3343

	20 AGVs		28 AGVs		40 AGVs	
	MÉDIA (Soma Ti)	Média Normal.	MÉDIA (Soma Ti)	Média Normal.	MÉDIA (Soma Ti)	Média Normal.
Fifo	526:40:37	2,0888	84:57:15	2,5015	78:54:14	2,2231
Menor Dist.	400:33:46	1,5673	46:41:28	0,7273	36:06:46	1,0170
<i>Fuzzy</i>	308:07:28	1,1972	40:45:44	0,6340	35:58:09	1,0129
<i>Fuzzy+Predição</i>	135:20:42	0,4976	39:20:44	0,6119	5:07:52	0,1436
<i>Fuzzy2</i>	101:03:38	0,3555	36:47:42	0,5724	0:07:45	0,0034
<i>Fuzzy2+Predição</i>	84:37:16	0,2937	35:33:21	0,5530	0:00:00	0,0000

Tabela 27 – Resultados dos ganhos percentuais para *tardiness* dos testes realizados na fábrica 2

	4 AGVs Ganho	8 AGVs Ganho	12 AGVs Ganho	16 AGVs Ganho	20 AGVs Ganho	28 AGVs Ganho	40 AGVs Ganho	MÉDIA DE GANHO
<i>Fuzzy 2 -> Fifo</i>	32,8702	44,4700	56,0713	62,0038	80,6344	56,4679	99,8343	48,0391
<i>Fuzzy 2 -> Menor Dist.</i>	27,6479	42,8256	55,1872	59,1877	74,6019	21,0065	99,6381	42,2328
<i>Fuzzy 2 -> Fuzzy</i>	20,2765	39,2054	43,9927	25,7884	67,1739	9,5678	99,6367	33,9602
<i>Fuzzy1+Pred. ->Fuzzy1</i>	6,5160	10,2158	25,8968	23,0725	56,3470	3,0868	85,7479	23,4314
<i>Fuzzy2+Pred. ->Fuzzy2</i>	6,3789	11,8396	7,2557	49,1468	16,7108	3,1488	100,0000	21,6089
<i>Fuzzy2+Pred. -> Fuzzy1+Pred.</i>	20,1596	40,3049	29,9037	50,9421	37,3683	9,6256	100,0000	32,0338

Na Figura 61 é apresentado um gráfico com os valores médios de *tardiness* encontrados com cada método de despacho de AGVs nos testes realizados na fábrica 2.

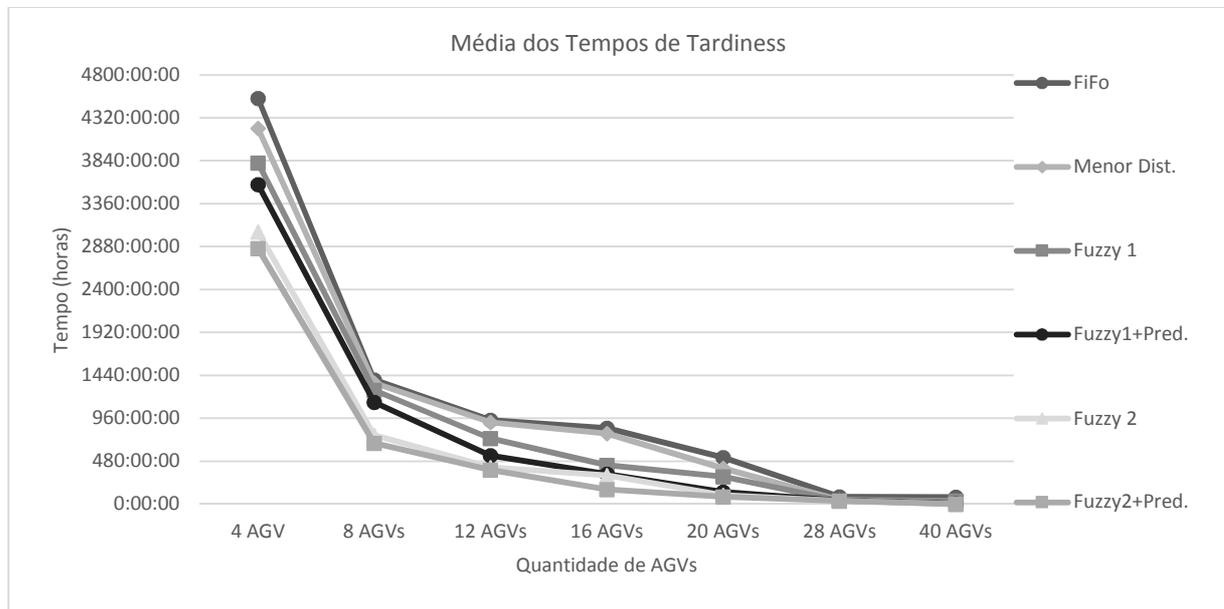


Figura 61 – Valores médios de *tardiness* dos testes realizados na fábrica 2.

Com os resultados gerados, inicia-se a verificação de tamanho amostral necessário para que se possa inferir com uma alta confiança. Esse cálculo é feito para cada par de métodos de despacho comparados.

Como parâmetros para este cálculo são usados: nível de significância de 3% (confiança de 97%) e poder de teste de no mínimo 97%. Os significados de tais parâmetros e as fórmulas que os equacionam estão no Apêndice A.

Para cada par de trabalhos comparados, usou-se o teste de hipótese *t-student* fazendo inferência na diferença das médias para duas amostras independentes e com variâncias desconhecidas.

Este teste verificou se a hipótese de que o trabalho proposto nesta tese (*Fuzzy2+Predição*) é melhor do que os trabalhos com os quais foi comparado (*FIFO*, *Menor Distância*, *Fuzzy1*, *Fuzzy1+Predição*, *Fuzzy2*).

A hipótese a ser testada é se os valores de *makespan* e os valores de *tardiness* obtidos pelo *Fuzzy 2+Predição* é menor do que os valores de *makespan* e *tardiness* dos trabalhos com quem está sendo comparado. Porém o suplemento usado para o cálculo do teste de hipóteses só permite as hipóteses nulas na forma: igual, maior igual e menor igual.

Portanto quatro hipóteses nulas foram cunhadas:

- O *makespan* da população amostral da proposta é menor ou igual o *makespan* da população amostral do outro método na comparação;
- O *makespan* da população amostral da proposta é igual o *makespan* da população amostral do outro método na comparação.
- O *tardiness* da população amostral da proposta é menor ou igual o *tardiness* da população amostral do outro método na comparação;
- O *tardiness* da população amostral da proposta é igual o *tardiness* da população amostral do outro método na comparação.

A hipótese nula é rejeitada sempre que o p-valor for menor que 0,03.

Ao aceitarmos a primeira e refutarmos a segunda, corroboramos que o *makespan* da população amostral da proposta é menor do que o *makespan* da população amostral do outro trabalho na comparação. Ao aceitarmos a terceira e refutarmos a quarta, corroboramos que o *tardiness* da população amostral da proposta é menor do que o *tardiness* da população amostral do outro trabalho na comparação.

Para realização dos testes de hipóteses é usado o complemento do Excel chamado *action* versão 3.0. Na Figura 62 é apresentada a tela de configuração para a realização do teste de verificação das amostras independentes e na Tabela 28 é apresentado o resultado do teste em que foi verificado se os resultados de *makespan* na fábrica 1 usando a abordagem *fuzzy2*+predição é maior que os resultados usando a abordagem FIFO.

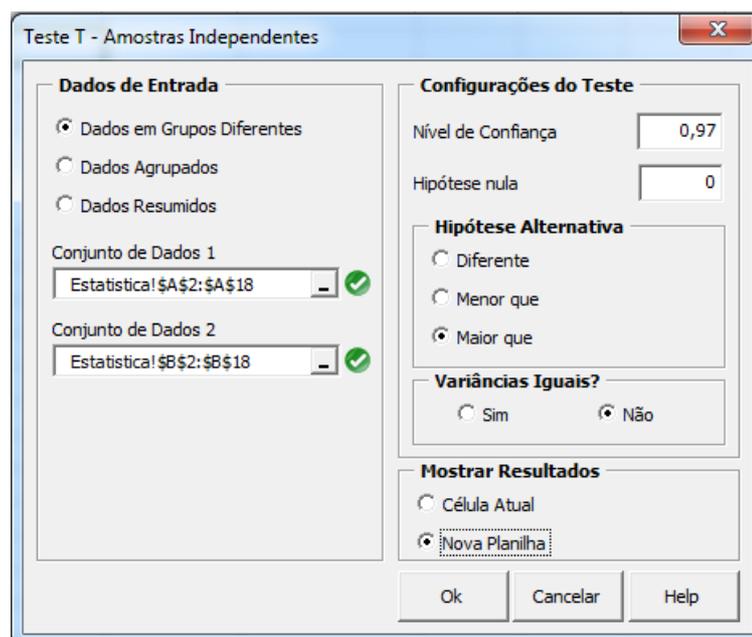


Figura 62 – Realização do Teste T, verificação de amostras independentes

Tabela 28 – Resultados da Análise de Teste T *Fuzzy2*+Predição > FIFO para *makespan* na fábrica 1

Resultados	
Estatística T	-53,86065
Graus de Liberdade	535,0117
P-valor	1
Média da Amostra 1	0,8881502
Média da Amostra 2	1,123087
Desvio Padrão da Amostra 1	0,04875047
Desvio Padrão da Amostra 2	0,0525408
Tamanho da Amostra 1	270
Tamanho da Amostra 2	270
Hipótese Alternativa Maior que	0
Nível de Confiança	0,97
Limite Inferior	-0,2431579
Limite Superior	Inf

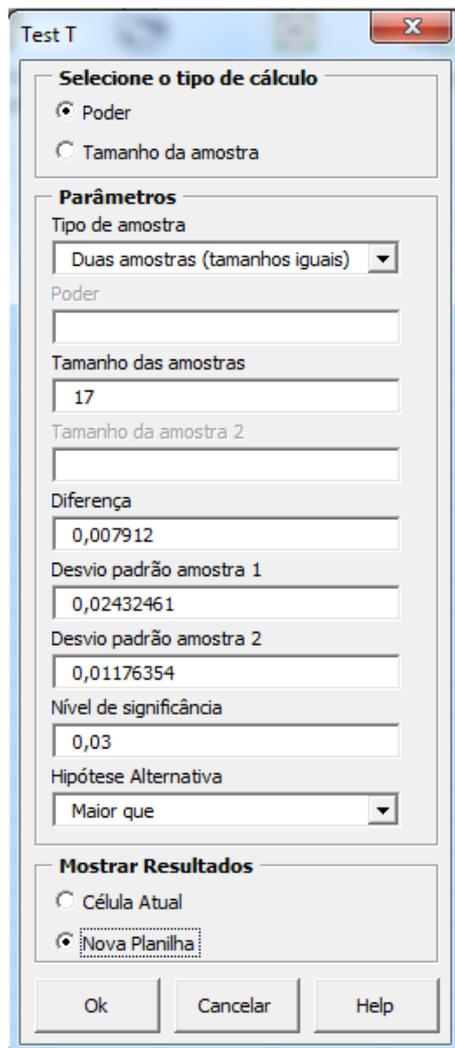


Figura 63 – Realização do Teste T, verificação do poder do teste pelo tamanho da amostra

Na Figura 63 é apresentada a tela de configuração para verificação do poder do teste de verificação das amostras independentes e na Tabela 29 é apresentado o resultado da verificação do teste apresentado na Tabela 28.

Tabela 29 – Resultados da Análise de Teste T *Fuzzy2*+Predição > FIFO para *makespan* na fábrica 1

Teste T - Duas amostras (tamanhos iguais)	
Poder	1
Tamanho da amostra	270
Diferença	0,2349368
Desvio padrão 1	0,04875047
Desvio padrão 2	0,0525408
Nível de significância	0,03
Hipótese Alternativa	Maior que

Nas Tabela 30 e Tabela 31 são apresentados todos os resultados das análises de teste-T realizados na fábrica 1 e fábrica 2 para *makespan* e *tardiness*.

Tabela 30 – Todos os resultados da Análise de Teste T para *makespan* e *tardiness* na fábrica 1

Resultados de <i>Makespan</i> na fábrica 1						
Método de Despacho	Verificação	Método de Despacho	P-Valor	Poder	Tam.Amostra	Nível de Confiança
Fuzzy 2+Predição	>	FIFO	1	1	270	0,97
Fuzzy 2+Predição	≠	FIFO	3,26E-218	1	270	0,97
Fuzzy 2+Predição	>	Menor Dist.	1	1	270	0,97
Fuzzy 2+Predição	≠	Menor Dist.	1,63E-210	1	270	0,97
Fuzzy 2+Predição	>	Fuzzy 1	1	1	270	0,97
Fuzzy 2+Predição	≠	Fuzzy 1	6,13E-156	1	270	0,97
Fuzzy 2+Predição	>	Fuzzy 1+Predição	1	1	270	0,97
Fuzzy 2+Predição	≠	Fuzzy 1+Predição	1,28E-35	1	270	0,97
Fuzzy 2+Predição	>	Fuzzy 2	0,999376	0,912886	270	0,97
Fuzzy 2+Predição	≠	Fuzzy 2	0,001249	0,857167	270	0,97

Resultados de <i>Tardiness</i> na fábrica 1						
Método de Despacho	Verificação	Método de Despacho	P-Valor	Poder	Tam.Amostra	Nível de Confiança
Fuzzy 2+Predição	>	FIFO	1	1	270	0,97
Fuzzy 2+Predição	≠	FIFO	2,00E-209	1	270	0,97
Fuzzy 2+Predição	>	Menor Dist.	1	1	270	0,97
Fuzzy 2+Predição	≠	Menor Dist.	1,04E-194	1	270	0,97
Fuzzy 2+Predição	>	Fuzzy 1	1	1	270	0,97
Fuzzy 2+Predição	≠	Fuzzy 1	5,42E-156	1	270	0,97
Fuzzy 2+Predição	>	Fuzzy 1+Predição	1	1	270	0,97
Fuzzy 2+Predição	≠	Fuzzy 1+Predição	3,82E-32	1	270	0,97
Fuzzy 2+Predição	>	Fuzzy 2	0,998352	0,856835	270	0,97
Fuzzy 2+Predição	≠	Fuzzy 2	0,004189	0,757881	270	0,97

Tabela 31 – Todos os resultados da Análise de Teste T para *makespan* e *tardiness* na fábrica 2

Resultados de <i>Makespan</i> na fábrica 2						
Método de Despacho	Verificação	Método de Despacho	P-Valor	Poder	Tam.Amostra	Nível de Confiança
Fuzzy 2+Predição	>	FIFO	1	1	105	0,97
Fuzzy 2+Predição	≠	FIFO	1,85E-70	1	105	0,97
Fuzzy 2+Predição	>	Menor Dist.	1	1	105	0,97
Fuzzy 2+Predição	≠	Menor Dist.	5,06E-59	1	105	0,97
Fuzzy 2+Predição	>	Fuzzy 1	1	1	105	0,97
Fuzzy 2+Predição	≠	Fuzzy 1	7,23E-41	1	105	0,97
Fuzzy 2+Predição	>	Fuzzy 1+Predição	1	1	105	0,97
Fuzzy 2+Predição	≠	Fuzzy 1+Predição	3,27E-21	1	105	0,97
Fuzzy 2+Predição	>	Fuzzy 2	0,999571	0,932080	105	0,97
Fuzzy 2+Predição	≠	Fuzzy 2	0,000859	0,884423	105	0,97

Resultados de <i>Tardiness</i> na fábrica 2						
Método de Despacho	Verificação	Método de Despacho	P-Valor	Poder	Tam.Amostra	Nível de Confiança
Fuzzy 2+Predição	>	FIFO	1	1	105	0,97
Fuzzy 2+Predição	≠	FIFO	3,84E-25	1	105	0,97
Fuzzy 2+Predição	>	Menor Dist.	1	1	105	0,97
Fuzzy 2+Predição	≠	Menor Dist.	1,27E-35	1	105	0,97
Fuzzy 2+Predição	>	Fuzzy 1	1	1	105	0,97
Fuzzy 2+Predição	≠	Fuzzy 1	1,03E-28	1	105	0,97
Fuzzy 2+Predição	>	Fuzzy 1+Predição	1	0,999814	105	0,97
Fuzzy 2+Predição	≠	Fuzzy 1+Predição	1,12E-07	0,999518	105	0,97
Fuzzy 2+Predição	>	Fuzzy 2	0,981802	0,585644	105	0,97
Fuzzy 2+Predição	≠	Fuzzy 2	0,035819	0,472377	105	0,97

Analisando-se os resultados estatísticos verifica-se que:

- Quando a hipótese nula é de que o valor de *makespan* alcançado pelo *Fuzzy2+Predição* é menor ou igual ao valor de *makespan* de qualquer um dos métodos em comparação, o p-valor é igual a um ou aproxima-se de um, caracterizando a aceitação desta hipótese nula.
- Quando a hipótese nula é de que o valor de *makespan* alcançado pelo *Fuzzy2+Predição* é igual ao valor de *makespan* de qualquer um dos métodos em comparação, o p-valor é bem menor que 0,03, caracterizando a não aceitação desta hipótese nula; e
- Quando a hipótese nula é de que o valor de *tardiness* alcançado pelo *Fuzzy2+Predição* é menor ou igual ao valor de *tardiness* de qualquer um dos métodos em comparação, o p-valor é igual a um ou aproxima-se de um, caracterizando a aceitação desta hipótese nula; e
- Quando a hipótese nula é de que o valor de *tardiness* alcançado pelo *Fuzzy2+Predição* é igual ao valor de *tardiness* de qualquer um dos métodos em comparação, o p-valor é bem menor que 0,03, caracterizando a não aceitação desta hipótese nula.

Portanto, estatisticamente, com confiança de 97%, usando populações adequadas (pois possuem distribuição normal e são independentes), o método de despacho de AGVs desta tese (*Fuzzy2+Predição*) é corroborado como sendo melhor que os outros métodos nos dois cenários e contexto dados.

5.4 Considerações finais

Neste capítulo foram apresentados os testes realizados para verificação das funcionalidades do *software* desenvolvido. Também foi apresentado o contexto da validação da proposta, o planejamento e a execução dos experimentos com a análise estatística dos dados obtidos.

Ao final, pode-se afirmar que a proposta desta tese está corroborada, pois não se conseguiu falsear as hipóteses de que esta seja melhor do que os métodos usados na comparação, no contexto definido.

Capítulo 6

CONCLUSÃO

6.1 Síntese do contexto da proposta e dos objetivos

No âmbito da manufatura, é comum pesquisas por métodos e técnicas para a melhoria da qualidade e aumento da produção. Um problema encontrado é que estas técnicas e métodos devem ser adaptados ao ambiente da manufatura que geralmente é caótico, não linear e complexo. Por conta deste ambiente, os sistemas de manufatura se tornam imprevisíveis e seu controle uma tarefa difícil. A dificuldade no controle também está associada às constantes perturbações que ocorrem no ambiente, tanto de conotação externa (por exemplo: demandas do mercado) quanto interna (por exemplo: falta ou falha nos recursos compartilhados).

Dentro do ambiente de manufatura, os FMS possuem um sistema de transporte que é responsável por realizar a integração da produção, transportando os produtos de um ponto a outro da fábrica; este fluxo de produção, segundo Groover (2003), tem uma grande influência no custo total de produção.

Existem várias formas de controlar o sistema de transporte, porém uma forma eficiente e usada comumente na prática são as regras de despacho. O principal problema do despacho é definir tarefas para os AGVs atender em um determinado momento e que ao atender essas tarefas minimiza-se o critério de otimização buscado.

A partir desse ambiente, foi definida a proposta de um sistema *fuzzy* cascata multiatributo que permite avaliar vários recursos da fábrica acrescida de uma estrutura baseada em espaço de estados que permite extrair informações futuras da fábrica. Portanto, em posse dessas informações, a proposta realiza o despacho de AGVs de maneira eficiente, permitindo alcançar um valor de *makespan* ou *tardiness* menor se comparado com propostas da literatura.

6.2 Contribuições e delimitações

Esta proposta de tese de doutorado foi desenvolvida considerando as delimitações listadas na Seção 1.5 e validada usando dois cenários de fábrica de diferentes dimensões, buscando a minimização de *makespan* e *tardiness*.

Foi feita uma comparação do método proposto nesta tese, nomeado *Fuzzy2+Predição*, com cinco abordagens (FIFO, Menor Distância, *Fuzzy1*, *Fuzzy1+Predição*, *Fuzzy2*) e os resultados corroboraram a hipótese de que se considerando múltiplos atributos e informações futuras da fábrica obtém resultados melhores aos obtidos por outras abordagens usadas atualmente. Na Tabela 32 são apresentados os ganhos médios finais dos dois cenários de fábrica para os critérios de *makespan* e *tardiness*.

Tabela 32 – Resumo dos resultados finais

	MÉDIA DE GANHO MAKESPAN (%)	MÉDIA DE GANHO TARDINESS (%)
<i>Fuzzy2 -> Fifo</i>	15,50	44,96
<i>Fuzzy2 -> Menor Dist.</i>	14,04	41,42
<i>Fuzzy2 -> Fuzzy1</i>	9,45	32,08
<i>Fuzzy1+Pred. ->Fuzzy1</i>	5,90	22,21
<i>Fuzzy2+Pred. ->Fuzzy2</i>	1,61	12,63
<i>Fuzzy2+Pred. -> Fuzzy1+Pred.</i>	5,31	23,17

As amostras foram produzidas em número suficiente para garantir atendimento a Teoria do Limite Central (> 30 elementos permite considerar como distribuição normal) e ao poder de teste desejado (97%). As amostras são comparadas usando teste de hipótese de *t-student* por inferência nas médias de *makespan* e *tardiness* com 97% de confiança. Feitas as análises, pôde-se afirmar que a proposta conseguiu, com 97% de confiança e no mínimo 97% de poder de teste, ser melhor aos trabalhos comparados, dentro dos contextos dados e das delimitações impostas.

Como estes contextos e delimitações aproximam este trabalho da realidade, pôde-se afirmar que a principal contribuição desta tese de doutorado está na melhoria do *makespan* e

tardiness encontrados para problemas de despacho de AGVs por meio da criação de um novo método de despacho de AGVs.

Como contribuições secundárias deste trabalho, pode-se listar:

- Por meio do sistema criado é possível simular vários métodos de despacho de AGVs e avaliar a quantidade ideal de AGV para um determinado cenário de fábrica.
- Os modelos criados podem ser simulados e obter outros dados importantes para gestores de produção e que não são o foco da proposta como, por exemplo: taxa de uso de cada veículo, taxa de uso de cada estação de trabalho, ocupação média dos *buffers*, *flowtime*, *throughput* e *work-in-progress* (WIP).

6.3 Trabalhos futuros

Durante a elaboração desta tese, tanto na fase de definição da proposta quanto nas fases de desenvolvimento e validação, foram identificados vários pontos a serem estendidos, acrescentados e ou melhorados e que são listados como possíveis trabalhos futuros:

- Devido ao ganho que se obteve ao incluir mais variáveis no sistema de despacho, um estudo que poderia derivar deste é a avaliação de outras variáveis a serem incluídas no sistema de despacho e, além disso, por meio de exaustivas simulações identificar o grau de interferência de cada variável nos valores de *makespan* e *tardiness*.
- Otimização das bases de conhecimento e de regras dos sistemas *fuzzy* usando algoritmos genéticos ou redes neurais.
- Uso de técnicas de mineração de dados para que por meio dos históricos de produção e decisões da fábrica se ajuste os pesos do sistema *fuzzy*.
- Criar novos modelos de fábricas para aumentar a base de dados e melhorar os resultados de testes.
- Avaliar o tempo em se leva para realizar a verificação completa do espaço de estados gerados com o intuito de balancear um tempo de resposta hábil há uma maior profundidade de busca.
- Avaliar outros métodos de busca para percorrer e analisar o espaço de estados a fim de se conseguir aumentar a eficiência da busca.

REFERÊNCIAS

AGUIRRE, L. A. **Enciclopédia de Automática**. São Paulo: Blucher, 2007.

ANBUUDAYASANKAR, S. P.; GANESH, K.; MOHAPATRA, S. **Models for Practical Routing Problems in Logistics: Design and Practices**. Switzerland: Springer International 2014.

ANDERSON, D. R. **Estatística Aplicada à Economia**. São Paulo: Pioneira Thomson Learning, 2002.

ARENALES, M.; ARMENTATO, V.; MORABITO, R.; YANASSE, H. **Pesquisa Operacional**. Rio de Janeiro: Campus-Elsevier, 2007.

AZADEH, A.;SHOJA, B. M.;MOGHADDAM, M.;ASADZADEH, S. M.;AKBARI, A. A neural network meta-model for identification of optimal combination of priority dispatching rules and makespan in a deterministic job shop scheduling problem. **The International Journal of Advanced Manufacturing Technology**, v. 67, n. 5-8, p. 1549-1561, 2013.

BANDARA, D.; MAYORGA, M. E.; MCLAY, L. A. Priority dispatching strategies for EMS systems. **Journal Operational Research Society**, v. 65, n. 4, p. 572-587, 2014.

BARTHOLDI III, J. J.; PLATZMAN, L. K. Decentralized control of automated guided vehicles on a simple loop. **IIE Transactions (Institute of Industrial Engineers)**, v. 21, n. 1, p. 76-81, 1989.

BENINCASA, A. X.; MORANDIN, O. J.; KATO, E. R. R. Reactive fuzzy dispatching rule for automated guided vehicles. IEEE International Conference on Systems, Man and Cybernetics, 2003. Washington, DC. USA. p.4375-4380.

BRISKORN, D.;HARTMANN, S.;HAASIS, H.-D.;KOPFER, H.;SCHÖNBERGER, J. Simulating Dispatching Strategies for Automated Container Terminals. In: (Ed.). **Operations Research Proceedings**: Springer Berlin Heidelberg, v.5, 2006. p.97-102.

CARIDÁ, V. F.; MORANDIN, O.; CERIONI, P. R.; TUMA, C. C. M. Prediction Model Using Coverability Tree from a Modeling in Petri Nets Applied in AGVs Dispatching. Brazilian Conference on Intelligent Systems (BRACIS) 2014. p.140-145.

CARIDÁ, V. F.; MORANDIN, O.; TUMA, C. C. M. Approaches of fuzzy systems applied to an AGV dispatching system in a FMS. **The International Journal of Advanced Manufacturing Technology**, v. 79, n. 1, p. 615-625, 2015.

CHANG, X.; DONG, M.; YANG, D. Multi-objective real-time dispatching for integrated delivery in a Fab using GA based simulation optimization. **Journal of Manufacturing Systems**, n. 2, 2013.

CHEN, B.; MATIS, T. I. A flexible dispatching rule for minimizing tardiness in job shop scheduling. **International Journal of Production Economics**, v. 141, n. 1, p. 360-365, 2013.

CHEN, T. An effective dispatching rule for bi-objective job scheduling in a wafer fabrication factory—considering the average cycle time and the maximum lateness. **The International Journal of Advanced Manufacturing Technology**, v. 67, n. 5-8, p. 1281-1295, 2013.

CHEN, T.; RAJENDRAN, C.; WU, C.-W. Advanced dispatching rules for large-scale manufacturing systems. **The International Journal of Advanced Manufacturing Technology**, v. 67, n. 1-4, p. 1-3, 2013.

CHIBA, R.; ARAI, T.; OTA, J. Integrated design for automated guided vehicle systems using cooperative co-evolution. **Advanced Robotics**, v. 24, n. 1-2, p. 25-45, 2010.

CHOE, R.; KIM, J.; RYU, K. R. Online preference learning for adaptive dispatching of AGVs in an automated container terminal. **Applied Soft Computing**, v. 38, p. 647-660, 2016.

CHOE, R.; PARK, T.; RYU, K. R. **Dispatching AGVs with noisy estimation of crane operation time**. *ACM Symposium on Applied Computing*. 2: 1288-1293 p. 2010.

CORDÓN, O.; HERRERA, F.; MAGDALENA, L.; VILLAR, P. A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base. **Information Sciences**, v. 136, n. 1-4, p. 85-107, 2001.

DAHARI, M.; LIU, Y. A Review of Auto-Guided-Vehicles Routing Algorithms. **Advanced Materials Research**, v. 479-481, p. 443-456, 2012.

DANG, T. T. Combination of dispatching rules and prediction for solving multi-objective scheduling problems. **International Journal of Production Research**, v. 51, n. 17, p. 5180-5194, 2013.

DUINKERKEN, M. B.; OTTJES, J. A.; LODEWIJKS, G. Comparison of routing strategies for AGV systems using simulation. Winter Simulation Conference, 2006. Monterey, United states. p.1523-1530.

- EGBELU, P. J. Pull Versus Push Strategy for Automated Guided Vehicle Load Movement in a Batch Manufacturing System. **Journal of Manufacturing Systems**, v. 6, n. 3, p. 209-221, 1987.
- EGBELU, P. J.; TANCHOCO, J. M. A. Characterization of Automatic Guided Vehicle Dispatching Rules. **International Journal of Production Research**, v. 22, n. 3, p. 359-374, 1984.
- EL-BOURI, A. A cooperative dispatching approach for minimizing mean tardiness in a dynamic flowshop. **Computers & Operations Research**, v. 39, n. 7, p. 1305-1314, 2012.
- EVERS, J. J. M.; KOPPERS, S. A. J. Automated guided vehicle traffic control at a container terminal. **Transportation Research Part A: Policy and Practice**, v. 30, n. 1, p. 21-34, 1996.
- FANTI, M. P.; MANGINI, A. M.; PEDRONCELLI, G.; UKOVICH, W. Decentralized deadlock-free control for AGV systems. American Control Conference (ACC), 2015. p.2414-2419.
- FERREIRA, M. J. B.; CUNHA, S. F. E.; ARAÚJO, R. D. D.; MELLO, C. H.; BOEIRA, J. **Relatório de Acompanhamento Setorial Madeira e Móveis – Indústria Moveleira UNICAMP e Agência Brasileira de Desenvolvimento Industrial** p.15. 2009
- GHANNADPOUR, S. F.; NOORI, S.; TAVAKKOLI-MOGHADDAM, R.; GHOSEIRI, K. A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application. **Applied Soft Computing**, v. 14, Part C, p. 504-527, 2014.
- GONZALEZ-R, P. L.; FRAMINAN, J. M.; RUIZ-USANO, R. A multi-objective comparison of dispatching rules in a drum–buffer–rope production control system. **International Journal of Computer Integrated Manufacturing**, v. 23, n. 2, p. 155-167, 2010.
- GROOVER, M. P. **Automation, Production Systems, and Computer Integrated Manufacturing**. New Jersey: Prentice-Hall, 2003.
- GUAN, X.; DAI, X. Deadlock-free multi-attribute dispatching method for AGV systems. **International Journal of Advanced Manufacturing Technology**, v. 45, n. 5-6, p. 603-615, 2009.
- HAN, M.-H.; MCGINNIS, L. F. Control material handling transporter in automated manufacturing. **IIE Transactions (Institute of Industrial Engineers)**, v. 21, n. 2, p. 184-190, 1989.

HIDEHIKO, Y. Addition and Deletion of Agent Memories to Realize Autonomous Decentralized Manufacturing Systems. *New Frontiers in Applied Artificial Intelligence*, 2008. Wroclaw, Poland. p.738-748.

HIRAMA, K. **Engenharia de Software: Qualidade e Produtividade com Tecnologia**. Rio de Janeiro - RJ: Elsevier, 2012. ISBN 9788535248821.

HO, Y.-C.; LIU, H.-C.; YIH, Y. A multiple-attribute method for concurrently solving the pickup-dispatching problem and the load-selection problem of multiple-load AGVs. **Journal of Manufacturing Systems**, v. 31, n. 3, p. 288-300, 2012.

JANG, J. S. R.; SUN, C. T.; MIZUTANI, E. Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence. **Upper Saddle River, NJ, USA: Prentice-Hall, Inc.**, 1997.

JEONG, B. H.; RANDHAWA, S. U. A multi-attribute dispatching rule for automated guided vehicle systems. **International Journal of Production Research**, v. 39, n. 13, p. 2817-2832, 2001.

KALINOVICIC, L.; PETROVIC, T.; BOGDAN, S.; BOBANAC, V. Modified Banker's algorithm for scheduling in multi-AGV systems. *IEEE Conference on Automation Science and Engineering*, 2011. p.351-356.

KATO, E. R. R.; MORANDIN, O.; FONSECA, M. A. S. Ant colony optimization algorithm for reactive production scheduling problem in the job shop system. *IEEE International Conference on Systems, Man and Cybernetics*, 2009. p.2199-2204.

KATO, E. R. R.; MORANDIN, O. J.; SGAVIOLI, M. A conflict solution manufacturing system modeling using Fuzzy Coloured Petri net. *IEEE International Conference on Systems Man and Cybernetics*, 2010. p.3983-3988.

KHANMOHAMMADI, S.; MIRNIA, M. K.; REZVANI, K.; BADAMCHIZADEH, M. A. Multi AGV hybrid path planning using fuzzy inference systems. 2010. Singapore, Singapore. *IEEE Computer Society*. p.789-792.

KIM, C. W.; TANCHOCO, J. M. A.; KOO, P. H. AGV dispatching based on workload balancing. **International Journal of Production Research**, v. 37, n. 17, p. 4053-4066, 1999.

KIM, Y.-H.; YARLAGADDA, P. Control of Automated Guided Vehicles Based on Multi-Attribute Dispatching Rule. **Applied Mechanics and Materials**, v. 278-280, p. 1432-1435, 2013.

- KIZIL, M.; OZBAYRAK, M.; PAPADOPOULOU, T. C. Evaluation of dispatching rules for cellular manufacturing. **International Journal of Advanced Manufacturing Technology**, v. 28, n. 9, p. 985-992, 2006.
- KLEI, C. M.; KIM, J. AGV dispatching. **International Journal of Production Research**, v. 34, n. 1, p. 95-110, 1996.
- KLIR, G. J.; YUAN, B. **Fuzzy sets and fuzzy logic: Theory and applications**. Prentice Hall, 1995.
- KULWIEC, R. Trends in Automatic Guided Vehicle Systems. **Plant Engineering (Barrington, Illinois)**, v. 38, n. 23, p. 66-73, 1984.
- LAKATOS, E. M.; MARCONI, M. A. **Metodologia Científica**. São Paulo: Atlas, 2008. 312
- LE-ANH, T.; DE KOSTER, M. B. M. A review of design and control of automated guided vehicle systems. **European Journal of Operational Research**, v. 171, n. 1, p. 1-23, 2006.
- LEITÃO, P. A holonic disturbance management architecture for flexible manufacturing systems. **International Journal of Production Research**, v. 49, n. 5, p. 1269-1284, 2011.
- LIN, L.; SHIN, S. W.; GEN, M.; HWANG, H. Network model and effective evolutionary approach for AGV dispatching in manufacturing system. **Journal of Intelligent Manufacturing**, v. 17, n. 4, p. 465-477, 2006.
- LONG LV, Y.; ZHANG, G.; ZHANG, J.; DONG, Y. J. Integrated scheduling of the job and AGV for flexible manufacturing system. **Applied Mechanics and Materials**, v. 80-81, p. 1335-1339, 2011.
- LU, M.-S.; HSIEH, T. C. A predictive approach for dispatching of tasks in a pull-type robotic cell. **Expert Systems with Applications**, v. 38, n. 7, p. 8825-8834, 2011.
- LUGER, G. Artificial intelligence: Structures and strategies for complex problem solving. **5 ed. Addison Wesley Longman**, p. 928, 2004.
- MAMDANI, E. H.; ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. **International Journal of Man-Machine Studies**, v. 7, n. 1, p. 1-13, 1975/01/01 1975.
- MANIYA, K. D.; BHATT, M. G. A multi-attribute selection of automated guided vehicle using the AHP/M-GRA technique. **International Journal of Production Research**, v. 49, n. 20, p. 6107-6124, 2011.

- MARTÍNEZ-BARBERÁ, H.; HERRERO-PÉREZ, D. Autonomous navigation of an automated guided vehicle in industrial environments. **Robotics and Computer-Integrated Manufacturing**, v. 26, n. 4, p. 296-311, 2010.
- MARTINS, P. G.; ALT, P. R. C. **Administração de Materiais e Recursos Patrimoniais**. São Paulo: Saraiva, 2009.
- MARVIZADEH, S. Z.; CHOUBINEH, F. F. Entropy-based dispatching for automatic guided vehicles. **International Journal of Production Research**, v. 52, n. 11, p. 3303-3316, 2014.
- MCCLAVE, J. T. **Estatística para Administração e Economia**. São Paulo: Prentice Hall, 2009.
- MCHANEY, R. Modelling battery constraints in discrete event automated guided vehicle simulations. **International Journal of Production Research**, v. 33, n. 11, p. 3023-3040, 1995.
- MIYAMOTO, T.; INOUE, K. Local and random searches for dispatch and conflict-free routing problem of capacitated AGV systems. **Computers & Industrial Engineering**, v. 91, p. 1-9, 2016.
- MONTGOMERY, D. C. **Design and Analysis of Experiments**. New York: John Wiley: 2005.
- MONTGOMERY, D. C.; RUNGER, G. C. **Estatística aplicada e probabilidade para engenheiros**. Rio de Janeiro: LTC: 2009.
- MORANDIN, O.; CASTRO, P. A. D.; KATO, E. R. R.; CAMARGO, H. A. A Genetic Fuzzy System for Defining a Reactive Dispatching Rule for AGVs. IEEE International Conference on Systems, Man and Cybernetics, 2006. p.56-61.
- MORANDIN, O.; KATO, E. R. R.; MAGGIO, E. G. R.; SANCHES, D. S.; DERIZ, A. C. A Heuristic based on Petri Nets modeling for FMS Scheduling problem of makespan minimization. 33rd Annual Conference of the IEEE Industrial Electronics Society, 2007. p.2683-2688.
- MORANDIN, O.; KATO, E. R. R.; SANCHES, D. S.; MUNIZ, B. D. Modeling strategy by adaptive genetic algorithm for production reactive scheduling with simultaneous use of machines and AGVs. IEEE International Conference on Systems, Man and Cybernetics, 2009. p.704-709.
- MORANDIN, O.; KATO, E. R. R.; TUMA, C. C. M. A strategy of production scheduling with the fitness function of genetic algorithm using Timed Petri net and considering AGV and the

input buffer. 36th Annual Conference on IEEE Industrial Electronics Society, 2010. p.1311-1316.

MORANDIN, O.;SANCHES, D. S.;DERIZ, A. C.;KATO, E. R. R.;TSUNAKI, R. H. An Adaptive Genetic Algorithm based approach for production reactive scheduling of manufacturing systems. 34th Annual Conference of IEEE Industrial Electronics, 2008. p.1461-1466.

MORANDIN, O. J.; CARIDA, V. F.; KATO, E. R. R.; FONSECA, M. A. S. A hierarchical fuzzy rule-based building model applied to a AGV dispatching system in an FMS. IEEE International Conference on Robotics and Automation, 2011a. p.3764-3769.

MORANDIN, O. J.; CARIDA, V. F.; KATO, E. R. R.; TUMA, C. C. M. Adaptive genetic fuzzy, predictive and multiobjective approach for AGVs dispatching. 37th Annual Conference on IEEE Industrial Electronics Society, 2011b. p.2317-2322.

MORANDIN, O. J.; CASTRO, P. A. D.; KATO, E. R. R.; CAMARGO, H. A. A genetic fuzzy system for defining a reactive dispatching rule for AGVs. IEEE International Conference on Systems, Man and Cybernetics, 2007. Taipei, Taiwan. p.56-61.

MORANDIN, O. J.;TUMA, C. C. M.;CARIDA, V. F.;PEDRINO, E. C.;TSUNAKI, R. H. PSO in 2D-space to solve reactive scheduling problems in FMS to reduce the makespan. IEEE International Symposium on Industrial Electronics, 2013. p.1-6.

MORETTIN, P. A.; BUSSAB, W. O. **Estatística Básica**. São Paulo: Saraiva, 2002.

MUKAIDONO, M. **Fuzzy Logic for beginners**. Singapore: World Scientific, 2001.

MURATA, T. Petri nets: Properties, analysis and applications. **Proceedings of the IEEE**, v. 77, n. 4, p. 541-580, 1989.

NAIQI, W.; MENGCHU, Z. Deadlock Resolution in Automated Manufacturing Systems With Robots. **IEEE Transactions on Automation Science and Engineering**, v. 4, n. 3, p. 474-480, 2007.

NASO, D.; TURCHIANO, B. Multicriteria meta-heuristics for AGV dispatching control based on computational intelligence. **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics**, v. 35, n. 2, p. 208-226, 2005.

NETO, R. F. T. **Proposta de Solução de Problemas de Scheduling Considerando Possibilidade de Terceirização Usando a Técnica de Otimização por Colônia de Formigas**.

2010. 148 (Tese de Doutorado). Departamento de Engenharia de Produção, Universidade Federal de São Carlos

NGUYEN, V. D.; KIM, K. H. A dispatching method for automated lifting vehicles in automated port container terminals. **Computers & Industrial Engineering**, v. 56, n. 3, p. 1002-1020, 2009.

NICOLETTI, M. D. C.; CAMARGO, H. A. **Fundamentos da Teoria de Conjuntos Fuzzy**. São Carlos: Editora UFSCar, 2004. 65

NISHI, T.; MAENO, R. Petri Net Decomposition Approach to Optimization of Route Planning Problems for AGV Systems. **IEEE Transactions on Automation Science and Engineering**, v. 7, n. 3, p. 523-537, 2010.

NISHI, T.; TANAKA, Y. Petri Net Decomposition Approach for Dispatching and Conflict-Free Routing of Bidirectional Automated Guided Vehicle Systems. **IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans**, v. 42, n. 5, p. 1230-1243, 2012.

OLIVEIRA, M. M.;GALDAMES, J. P. M.;VIVALDINI, K. T.;MAGALHAES, D. V.;BECKER, M. Battery state estimation for applications in intelligent warehouses. IEEE International Conference on Robotics and Automation, 2011. p.5511-5516.

PEDRYCZ, W.; GOMIDE, F. Introduction to Fuzzy Sets: Analysis and Design. **MIT Press**, 1998.

PETERS, B. A.; SMITH, J. S.; VENKATESH, S. A control classification of automated guided vehicle system. **International Journal of Industrial Engineering**, v. 3, p. 29-39, 1996.

PINEDO, M. L. **Scheduling: Theory, Algorithms, and Systems**. New York, NY: 2008.

POPPER, K. R. **A Lógica da Pesquisa Científica**. São Paulo: Cultrix, 2007.

PRADHAN, S. K.; KUMAR, A.; SINHA, A. N. Some analysis of automated guided vehicle. International Mechanical Engineering Congress, 2014. Tiruchirappalli, India. p.2225-2228.

PRESSMAN, R. S. **Engenharia de Software**. São Paulo-SP: MacGraw-Hill, 2010.

QIU, L.; HSU, W. J.; HUANG, S. Y.; WANG, H. Scheduling and routing algorithms for AGVs: A survey. **International Journal of Production Research**, v. 40, n. 3, p. 745-760, 2002.

- REIS, F. B.; CARIDÁ, V. F.; MORANDIN, O. J.; CASTRO, R. L.; TUMA, C. C. M. A Collaborative Fuzzy CPN System for Conflict Solution of Flexible Manufacturing System. 39th Annual Conference of the IEEE Industrial Electronics Society, 2013. p.0-0.
- SANTOS, J.; COSTA, P.; ROCHA, L.; VIVALDINI, K.; MOREIRA, A. P.; VEIGA, G. Validation of a Time Based Routing Algorithm Using a Realistic Automatic Warehouse Scenario. In: REIS, P. L.; MOREIRA, P. A., *et al* (Ed.). **Second Iberian Robotics Conference: Advances in Robotics**. Cham: Springer International Publishing, v.2, 2016. p.81-92. ISBN 978-3-319-27149-1.
- SARIN, S. C.; VARADARAJAN, A.; WANG, L. A survey of dispatching rules for operational control in wafer fabrication. **Production Planning & Control**, v. 22, n. 1, p. 4-24, 2011.
- SEIFERT, R. W.; MORITO, S. Cooperative dispatching - exploiting the flexibility of an FMS by means of incremental optimization. **European Journal of Operational Research**, v. 129, n. 1, p. 116-133, 2001.
- SHAHZAD, A.; MEBARKI, N. Learning Dispatching Rules for Scheduling: A Synergistic View Comprising Decision Trees, Tabu Search and Simulation. **Computers**, v. 5, n. 1, p. 3, 2016.
- SILVA, E. D.; MENEZES, E. M. **Metodologia da Pesquisa e Elaboração de Dissertação**. Florianópolis: UFSC, 2005. 138
- SINGH, N.; SARNGADHARAN, P. V.; PAL, P. AGV scheduling for automated material distribution: a case study. **Journal of Intelligent Manufacturing**, v. 22, n. 2, p. 219-228, 2011.
- SLACK, N.; CHAMBERS, S.; HARLAND, C.; HARRISON, A.; JOHNSTON, R. **Administração da Produção**. São Paulo: Atlas, 1997.
- SMOLIC-ROCAK, N.; BOGDAN, S.; KOVACIC, Z.; PETROVIC, T. Time windows based dynamic routing in multi-AGV systems. **IEEE Transactions on Automation Science and Engineering**, v. 7, n. 1, p. 151-155, 2010.
- SOYLU, M.; ÖZDEMIREL, N. E.; KAYALIGIL, S. A self-organizing neural network approach for the single AGV routing problem. **European Journal of Operational Research**, v. 121, n. 1, p. 124-137, 2000.
- STECKE, K. E. Hierarchical Approach to Solving Machine Grouping and Loading Problems of Flexible Manufacturing Systems. **European Journal of Operational Research**, v. 24, n. 3, p. 369-378, 1986.

SU, T.-S. Optimal vehicle-dispatch decisions for cement silos using a fuzzy multi-objective approach. **International Journal of Production Research**, v. 52, n. 4, p. 947-966, 2014.

TAGHABONI-DUTTA, F.; TANCHOCO, J. M. A. Comparison of dynamic routing techniques for automated guided vehicle system. **International Journal of Production Research**, v. 33, n. 10, p. 2653-2669, 1995.

TAGHABONI, F. **Scheduling and control of manufacturing systems with critical material handling**. 1989. Ph. D. Thesis, Purdue University, West Lafayette, Indiana.

TAN, K. K.; TANG, K. Z. Simulation of an evolutionary tuned fuzzy dispatching system for automated guided vehicles. Winter Simulation Conference, 2000. Orlando, FL, USA. p.1339-1343.

TAO, X. Local obstacle avoidance planning of logistics system AGV based vector field. International Conference on Management Science and Industrial Engineering, 2011. p.1256-1259.

TAY, J. C.; HO, N. B. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. **Computers & Industrial Engineering**, v. 54, n. 3, p. 453-473, 2008.

TOMPKINS, J. A.; WHITE, J. A.; BOZER, Y. A.; TANCHOCO, J. M. A. **Facilities planning**. New York: Wiley, 2013. 864

TORRES, I. R.; DESSENS, L. F. R.; FLORES, J. L. M.; BENÍTEZ, E. O. Review of Comprehensive Approaches in Optimizing AGV Systems. Operational Excellence in Logistics and Supply Chains, 2015. Hamburg.

TRIOLA, M. F. **Introdução à Estatística**. Rio de Janeiro: LTC, 2008.

TUMA, C. C. M.; MORANDIN, O. J.; CARIDÁ, V. F. Minimizing the Makespan for the Problem of Reactive Production Scheduling in a FMS with AGVs Using a New Structure of Chromosome in a Hybrid GA with TS. 18th IEEE International Conference on Emerging Technologies & Factory Automation, 2013. p.1-6.

UMAR, U. A.; ARIFFIN, M. K. A.; ISMAIL, N.; TANG, S. H. Hybrid multiobjective genetic algorithms for integrated dynamic scheduling and routing of jobs and automated-guided vehicle (AGV) in flexible manufacturing systems (FMS) environment. **The International Journal of Advanced Manufacturing Technology**, v. 81, n. 9, p. 2123-2141, 2015.

UMASHANKAR, N.; KARTHIK, V. N. Multi-criteria intelligent dispatching control of automated guided vehicles in FMS. IEEE Conference on Cybernetics and Intelligent Systems, 2006.

VIS, I. F. A. Survey of research in the design and control of automated guided vehicle systems. **European Journal of Operational Research**, v. 170, n. 3, p. 677-709, 2006.

VIVALDINI, K.;ROCHA, L. F.;MARTARELLI, N. J.;BECKER, M.;MOREIRA, A. P. Integrated tasks assignment and routing for the estimation of the optimal number of AGVS. **The International Journal of Advanced Manufacturing Technology**, v. 82, n. 1, p. 719-736, 2015.

VIVALDINI, K. C. T.; ROCHA, L. F.; BECKER, M.; MOREIRA, A. P. Comprehensive Review of the Dispatching, Scheduling and Routing of AGVs. In: MOREIRA, P.

WU, X.; LOU, P.; TANG, D. Multi-objective Genetic Algorithm for System Identification and Controller Optimization of Automated Guided Vehicle. **Journal of Networks**, v. 6, 2011.

YAMASHITA, H. Analysis of dispatching rules of AGV systems with multiple vehicles. **IIE Transactions (Institute of Industrial Engineers)**, v. 33, n. 10, p. 889-895, 2001.

YANG, H. B.; YAN, H. S. An adaptive approach to dynamic scheduling in knowledgeable manufacturing cell. **International Journal of Advanced Manufacturing Technology**, v. 42, n. 3-4, p. 312-320, 2009.

YE, R.; HSU, W.-J.; VEE, V.-Y. Distributed routing and simulation of Automated Guided Vehicles. IEEE Region 10 International Conference (TENCON), 2000. Kuala Lumpur, Malaysia. p.315-320.

ZHANG, J.; ZHU, C.; CHAN, C. C. A wireless power charging method for Automated Guided Vehicle. IEEE International Electric Vehicle Conference (IEVC), 2014. p.1-5.

Apêndice A

CONCEITOS TEÓRICOS DA PROPOSTA

Com o intuito de diminuir a densidade de material apresentado na tese alguns conceitos teóricos foram omitidos no capítulo de proposta. Esses conceitos são apresentados nesse apêndice com o objetivo de esclarecer possíveis indagações do leitor.

7.1 Explicações e equações dos sistemas *fuzzy* usados na Tese

A função característica de um conjunto A, chamada de função de pertinência na teoria dos conjuntos *fuzzy*, associa a cada elemento do universo de discurso um valor em um intervalo, geralmente $[0, 1]$, que indica o grau de pertinência do elemento ao conjunto A. O grau de pertinência 0 (zero) significa a exclusão total de um elemento em relação ao conjunto, e o grau de pertinência 1 (um) significa pertinência total do elemento ao conjunto (KLIR; YUAN, 1995), (NICOLETTI; CAMARGO, 2004).

A base da teoria de conjuntos *fuzzy* é o conceito de função de pertinência. Na construção dos sistemas *fuzzy* da proposta todas as variáveis são representadas por funções triangulares, pois a ideia de definição de regiões de pertinência total, média e nula é mais intuitiva que a especificação de valor modal e dispersão associada ao projeto de funções Gaussianas. A função triangular apresenta como:

$$\mu_{triangular}(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x - a}{b - a}, & a < x \leq b \\ \frac{c - x}{c - b}, & b < x \leq c \\ 0, & x > c \end{cases} \quad (6)$$

Alternativamente, pode-se reescrever a equação (6) como:

$$\mu_{triangular}(x; a, b, c) = \max\left(\min\left(\frac{x - a}{b - a}, \frac{c - x}{c - b}\right), 0\right) \quad (7)$$

Onde b é o valor modal de μ e a e c representam os limites inferiores e superiores dos elementos compatíveis com o conjunto *fuzzy* respectivamente. Graficamente, pode-se interpretar o conjunto $\{a, b, c\}$ como as coordenadas dos três vértices do triângulo formado pela função de pertinência, como ilustrado na Figura 64.

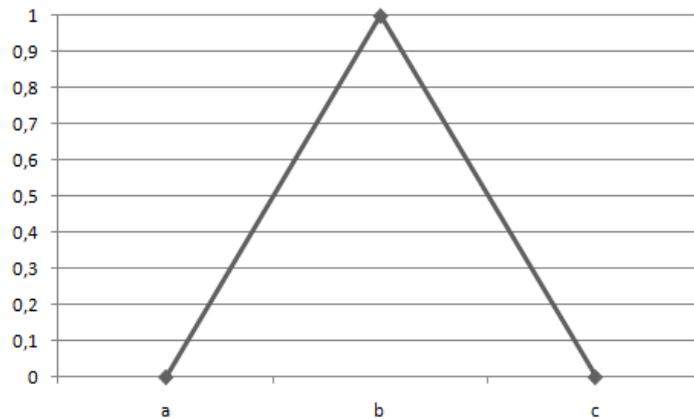


Figura 64 - Representação geral de uma função de pertinência triangular.

A *Fuzzyficação* (conversão de um escalar para *fuzzy*) usa as funções de pertinência pré-estabelecidas, mapeia cada variável de entrada do sistema em graus de pertinência do *fuzzy* que representa a variável em questão. Na Figura 65 é apresentado um exemplo do funcionamento da *fuzzyficação*, no qual o valor escalar recebido é 22.

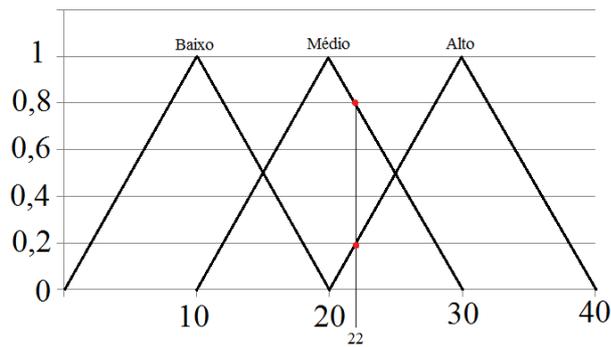


Figura 65 - Exemplo de Fuzzyficação.

Nos quais os valores 0,2 e 0,8 são obtidos por meio da aplicação da equação (7) apresentada anteriormente.

$$\text{Para a função de pertinência Médio: } \frac{c-x}{c-b} = \frac{30-22}{30-20} = 0,8$$

$$\text{Para função de pertinência Alto: } \frac{x-a}{b-a} = \frac{22-20}{30-20} = 0,2$$

As regras *fuzzy* são ferramentas eficientes na modelagem de sentenças em linguagem natural (JANG; SUN; MIZUTANI, 1997). Este tipo de modelagem usa o conceito de variáveis linguísticas para gerar regras *fuzzy* que são mapeadas por meio de relações *fuzzy* permitindo a investigação de diferentes esquemas de raciocínio aproximado. Nestes esquemas, procedimentos de inferência baseados no conceito de regra composicional de inferência são utilizados para derivar conclusões, a partir de uma base de conhecimento contendo regras e fatos conhecidos.

Uma proposição *fuzzy* é a parcela de informação básica que pode aparecer em uma regra e seu formato mais simples é $X \text{ é } A$ onde X é uma variável linguística e A é um termo linguístico que representa um conjunto *fuzzy*, o qual é caracterizado por uma função de pertinência.

As proposições *fuzzy* compostas podem representar conhecimento do tipo: “A temperatura é alta E a umidade é baixa” e “O tempo é nublado OU o trabalho é difícil”, respectivamente. Uma regra *fuzzy* assume o formato de uma proposição condicional, que no caso mais simples fica:

SE $X \text{ é } A$ ENTÃO $Y \text{ é } B$

Onde X e Y são variáveis linguísticas sobre os domínios X e Y, respectivamente, e A e B são termos linguísticos, que rotulam conjuntos *fuzzy* definidos sobre os domínios X e Y, respectivamente.

A proposta usa o método de inferência composicional simplificada com entradas numéricas, usando os operadores max-min para a inferência (conjunção pelo mínimo e agregação pelo máximo). A conjunção é usada nos antecedentes da regra encontrando o mínimo deles e a agregação é usada no consequente das regras encontrando o máximo delas. Para facilitar o entendimento, é apresentado um exemplo simples a baixo.

Pressupõe-se que é requerido ao sistema que calcule a prioridade de produção de um determinado produto de uma fábrica, e este evento dependa da margem de contribuição do produto juntamente com o volume de vendas, e tenha as seguintes regras:

SE V1 é Alta e V2 é Alta ENTÃO U é Alta.

SE V1 é Média e V2 é Alta ENTÃO U é Alta.

Sendo que V1 é margem de contribuição, V2 o volume de vendas e U é a prioridade de produção. A agregação é feita da seguinte forma:

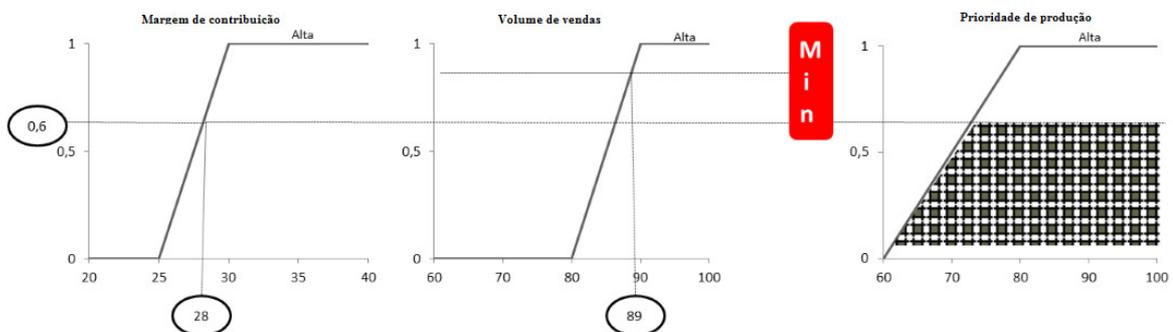


Figura 66 - Agregação da regra 1 utilizando operador de min.

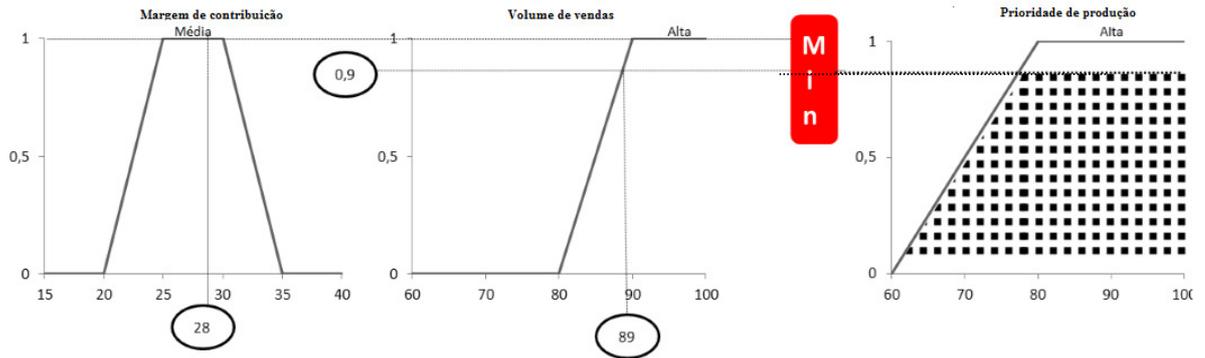


Figura 67 - Agregação da regra 2 utilizando operador de min.

Na composição, usa-se o operador de Max em todas as agregações, como apresentado na Figura 68.

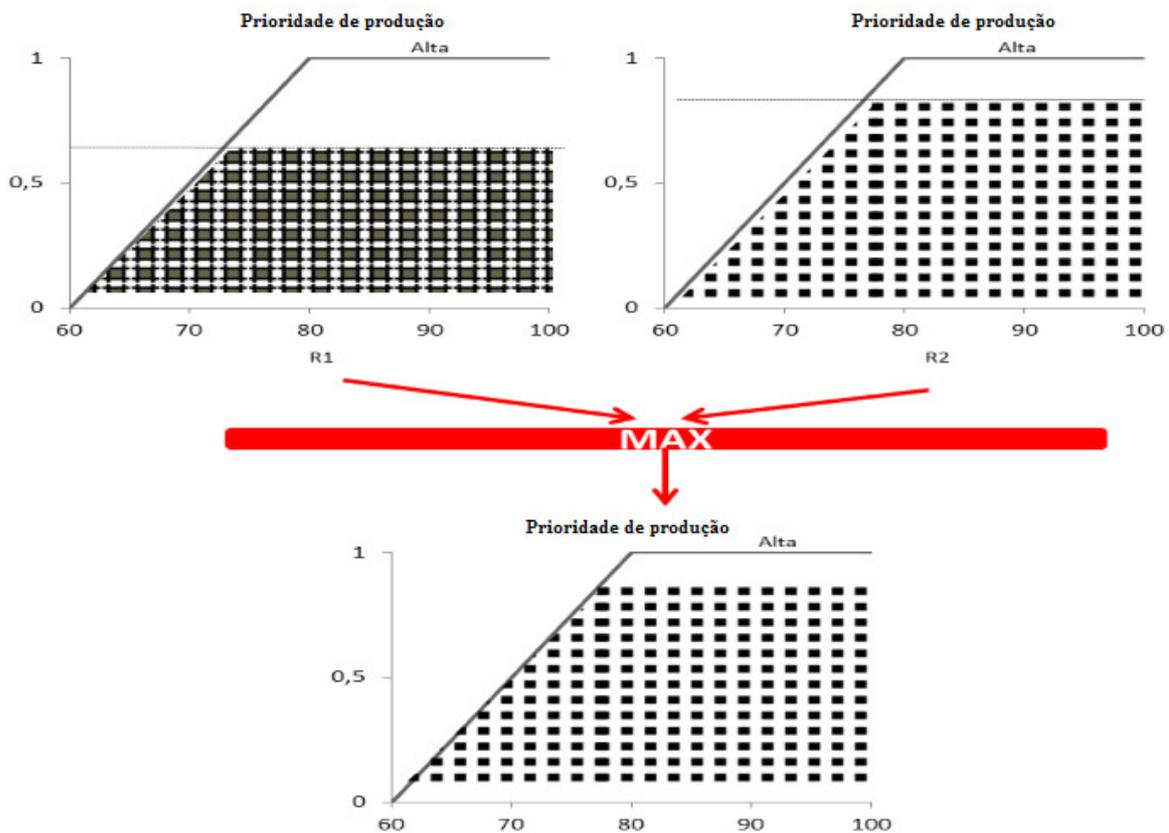


Figura 68 - Composição das regras 1 e 2 utilizando operador máx.

Os sistemas do tipo Mamdani obtêm como resultado do seu processamento de inferência um conjunto *fuzzy* que deve ser convertido de *fuzzy* para escalar para a obtenção de uma saída numérica exata. A *Defuzzyficação* (conversão de *fuzzy* para escalar) centro de área (CoA) é

usada nesse trabalho. Este método calcula o ponto central da saída *fuzzy* para uma determinada área, obtendo assim o valor escalar. Para definir o centro de área, aplica-se a seguinte equação:

$$CoA(B) = \frac{\sum_{i=0}^D B(y_i)y_i}{\sum_{i=0}^D B(y_i)} \tag{8}$$

No qual B é o conjunto base:

$$Y = \{y_1, y_2, y_3, \dots, y_n\}$$

Sendo $B(y_i)$ o grau de pertinência do elemento y_i no conjunto base. Para simplificar, tomemos como exemplo o seguinte conjunto base:

$$Y = \{0, 5, 10, 15, 20, 25, 30, 25\}$$

E a seguinte relação entre os elementos e seus respectivos graus de pertinência obtidos como resposta:

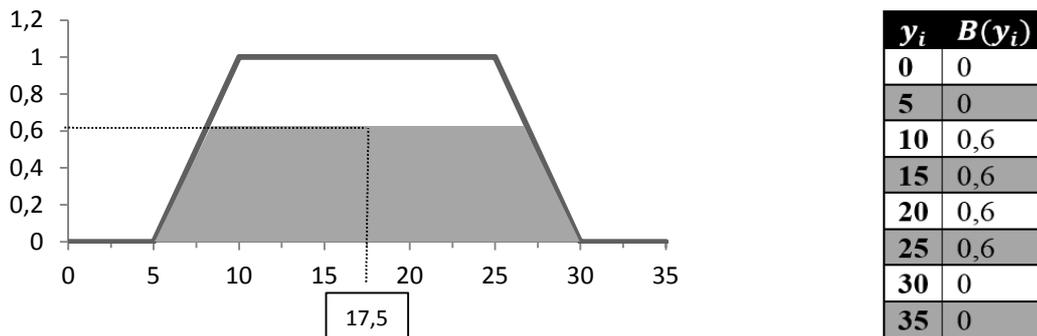


Figura 69 - Método de defuzificação CoA

Para calcular o CoA, basta aplicar os valores na equação (8):

$$CoA(B) = \frac{[(0 * 0) + (0 * 5) + (0,6 * 10) + (0,6 * 15) + (0,6 * 20) + (0,6 * 25) + (0 * 30) + (0 * 35)]}{0 + 0 + 0,6 + 0,6 + 0,6 + 0,6 + 0 + 0}$$

$$CoA(B) = \frac{6 + 9 + 12 + 15}{2,4} = 17,5$$

7.2 Explicações sobre geração e exploração do espaço de estados

Dada uma rede de Petri com marcação inicial, a partir da marcação inicial pode-se obter uma ou mais novas marcações de acordo com as transições habilitadas e ativadas. Para cada nova marcação pode-se obter novas marcações e assim sucessivamente. Este processo gera um espaço de estados cuja raiz é a marcação inicial.

Os nós representam as marcações e os arcos representam as transições ativadas que levam de uma marcação para outra marcação.

Para modelar um problema e traçar estratégias é uma ferramenta poderosa, porém este espaço de estados tende a crescer na medida em que o número de lugares e transições na rede de Petri aumenta, tornando sua visualização impossível a partir de determinado ponto.

O espaço de estados é o espaço de soluções para um problema de despacho modelado com uma Rede de Petri. Pode-se obter uma solução iniciando pelo nó raiz e terminando em algum nó folha.

Um caminho nesse espaço de estados inclui todas as operações que devem ocorrer no ambiente de produção, na exata ordem em que são ativadas, o que lhe garante alta correlação com os critérios de otimização.

Considere a rede de Petri apresentada na Figura 70. Verifica-se que a mesma possui 11 lugares (três dos quais compartilhados ou em disputa) e seis transições.

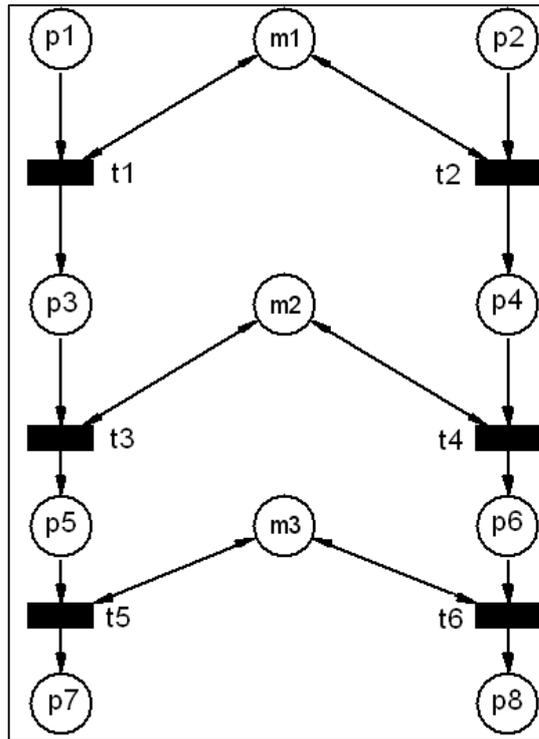


Figura 70 – Exemplo de um ambiente de manufatura modelado em redes de Petri

Nesta rede de Petri pode-se notar que após um número de ativações a rede encerra as ativações. Pode-se imaginar que se esta rede possuísse alguns laços realimentando os lugares p1 e p2, seria uma rede sem fim determinado. Para esta rede, na Figura 71 pode-se ver seu espaço de estados. As marcações foram mapeadas do seguinte modo (p1 p2 p3 p4 p5 p6 p7 p8 m1 m2 m3).

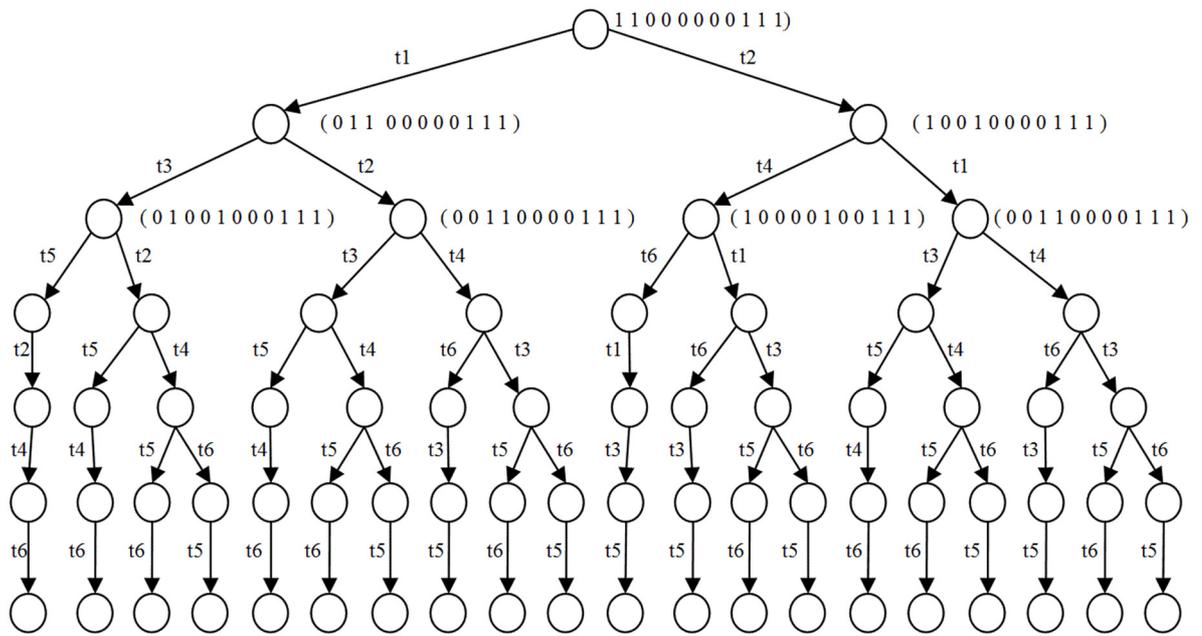


Figura 71 – Espaço de estados total da rede de Petri apresentada na Figura 70

Percebe-se que embora o número de lugares e transições da rede de Petri usada como exemplo seja pequeno, o espaço de estados gerado se torna algo complexo, porém ainda útil na visualização de estratégias de trabalho. Pode-se perceber inclusive que as marcações só estão colocadas nos três primeiros níveis de nós do espaço de estado, posto que ficaria ilegível caso se tentasse legendá-la por completo.

Devida a dificuldade da representação gráfica de uma rede de Petri e seu espaço de estados a medida que esses tendem a crescer o uso da matriz de transição (ou de incidência) e das equações de estado tornam o trabalho mais produtivo.

A equação de estados é representada abaixo:

$$M_k = M_{k-1} + A^T U_k, k = 1, 2, 3, \dots \tag{9}$$

Considere a matriz de transição $A = \{a_{ij} \mid a_{ij} \text{ é um inteiro, com } i=1, 2, \dots, n \text{ e } j=1, 2, \dots, m, \text{ sendo } n \text{ o número de transições e } m \text{ o número de lugares}\}$. Os elementos a_{ij} da matriz são calculados da seguinte forma: $a_{ij} = w(i, j) - w(j, i)$, em que $w(i, j)$ é o número de arcos que saem da transição i e vão para o lugar j e $w(j, i)$ é o número de arcos que saem do lugar j e vão para a transição i .

A matriz coluna U_k indica que a k transição será ativada. Para os objetivos desta tese, sempre é composta por zeros e apenas um único elemento igual a 1 (MURATA, 1989). O uso de mais de um elemento diferente de zero implicaria em ativações ocorrendo concorrentemente, o que conflitaria com o objetivo nesta tese de obter uma sequência inequívoca das ativações.

A matriz coluna M_k é a marcação na k ativação de uma transição.

A matriz coluna M_{k-1} é a marcação anterior a M_k e que dada a ativação indicada na matriz U_k transforma-se em M_k .

Portanto, a equação de estados permite que se calcule a marcação M_k desde que se conheça sua marcação anterior (M_{k-1}), a matriz de incidência da Rede de Petri (A) e a transição a ser ativada (indicada em U_k).

$$\begin{array}{ccccccc}
 M_k & & M_{k-1} & & A^t & & U_k \\
 \begin{array}{|c|} \hline 0 \\ \hline 1 \\ \hline 1 \\ \hline 0 \\ \hline 1 \\ \hline 1 \\ \hline 1 \\ \hline \end{array} & = & \begin{array}{|c|} \hline 1 \\ \hline 1 \\ \hline 0 \\ \hline 1 \\ \hline 1 \\ \hline 1 \\ \hline \end{array} & + & \begin{array}{|cccccc|} \hline -1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & -1 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & -1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & -1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & -1 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & -1 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} & & \begin{array}{|c|} \hline 1 \\ \hline 0 \\ \hline \end{array}
 \end{array}$$

Figura 72 – Equação de estados instanciada para o exemplo dado na Figura 71

7.3 Noções de estatística

Todos os conceitos apresentados neste apêndice vieram das seguintes referências: (MONTGOMERY, 2005); (MONTGOMERY; RUNGER, 2009); (MORETTIN; BUSSAB, 2002); (MCCLAVE, 2009); (TRIOLA, 2008); (ANDERSON, 2002).

O tamanho amostral, testes de normalidade das distribuições e testes de hipóteses possuem várias formas de serem calculados e, usualmente, lança-se mão de ferramentas computacionais.

7.3.1 Testes de Hipóteses *t-student*

Nesta seção, pretende-se explicar os conceitos básicos para o entendimento do uso de *t-student* para teste de hipóteses, fazendo inferência na diferença das médias de duas amostras independentes com distribuições normais e com variâncias desconhecidas.

Fórmulas complexas e explicações mais detalhadas não fazem parte do contexto desta obra. Limitou-se aos conceitos e forma de interpretação de seus resultados, uma vez que todos os métodos comentados possuem implementação em vários programas de distribuição gratuita.

Amostras independentes são duas amostras que não possuem algum tipo de correlação entre elas.

Variância é a raiz quadrada do desvio padrão.

Um teste de hipóteses formula duas hipóteses: a hipótese nula (HN) e a hipótese alternativa (HA). Um teste de hipóteses tenta aceitar ou rejeitar a hipótese nula e, pela sua rejeição, aceitar a hipótese alternativa.

Para garantir qualidade das inferências, devem-se considerar os dois possíveis erros:

- Erro I – Quando se rejeita a hipótese nula e esta é verdadeira. Neste caso, a sua probabilidade de ocorrência chama-se nível de significância (α). Quanto menor esse nível, maior a confiança em não se rejeitar a hipótese nula erroneamente. Este nível, normalmente, é estipulado com valores de 0,05 a 0,001. Confiança do teste é dado por $(1 - \alpha)$;

- Erro II – Quando não se rejeita a hipótese nula e esta é falsa. Neste caso, a sua probabilidade de ocorrer é denotada por β . Chama-se poder do teste ao valor $(1 - \beta)$. β é importante na definição do tamanho amostral.

Para o teste de hipótese usando *t-student* algumas condições devem ser atendidas:

- Definir nível de significância (α), definido nesta tese como 0,03;
- Definir poder do teste $(1-\beta)$, definido nesta tese como 0,97. Caso nesse poder de teste o tamanho amostral extrapole 200, estabelece-se o tamanho amostral como 200 e calcula-se o poder de teste para esse tamanho amostral;
- Garantir que as amostras: sejam grandes (>30); ou que possuem distribuição normal suficientemente boa para que se possa inferir, verificado com o uso dos métodos indicados na próxima seção; ou pela observação do papel de probabilidade;
- Garantir que as amostras são independentes, o que, normalmente, ocorre com amostras geradas aleatoriamente.

A hipótese nula é rejeitada sempre que o valor de p (p -valor) for menor que o nível de significância. Quando a hipótese nula é rejeitada, automaticamente se aceita a hipótese alternativa.

Os pares H_N e H_A normalmente formuladas são:

1 - H_N afirma que a tese possui resultados iguais aos do concorrente e H_A afirma que a tese possui resultados diferentes aos do concorrente;

2 - H_N afirma que a tese possui resultados menores ou iguais aos do concorrente e H_A afirma que a tese possui resultados maiores que os do concorrente;

3 - H_N afirma que a tese possui resultados maiores ou iguais aos do concorrente e H_A afirma que a tese possui resultados menores que os do concorrente;

Para verificar-se se a hipótese possui resultados menores que os do concorrente, faz-se necessário elaborar dois pares de H_N e H_A . O primeiro com H_N afirmando que a hipótese possui valores menores ou iguais aos do concorrente e o segundo com H_N afirmando que a hipótese possui resultados iguais aos do concorrente. A aceitação da primeira H_N e a rejeição da segunda H_N implica em afirmar que a hipótese possui resultados menores que os do concorrente.

Para verificar a veracidade do parágrafo acima segue uma prova usando lógica proposicional.

Sejam as proposições:

p: a hipótese possui valores menores que o concorrente;

q: a hipótese possui valores iguais ao concorrente;

HN1: a hipótese possui valores menores ou iguais ao concorrente (p ou q);

HN2: a hipótese possui valores iguais ao concorrente (q).

Se HN1 não for rejeitada então p ou q = V;

Se HN2 for rejeitada então q = F;

Substituindo q em HN1 pelo seu valor verdade então p ou F = V;

Usando a propriedade de identidade então p = V.

Os passos para se efetuar uma comparação de duas amostras usando testes de hipótese com *t-student* são os seguintes:

- Formular as hipóteses nula (HN) e alternativa (HA);
- Definir o nível de significância (quanto se permite rejeitar HN erroneamente);
- Definir o poder do teste (quanto se permite aceitar a HN erroneamente);
- Gerar duas amostras de tamanho razoável para poder calcular os tamanhos amostrais. Considera-se que estas amostras são independentes, com distribuições normais e variâncias desconhecidas;
- Calcular o tamanho amostral usando como entradas os parâmetros indicados na Seção 8.2.3;
- Gerar os complementos das amostras, se necessário;
- Testar a suposição de normalidade das amostras, conforme indicado na Seção 8.2.2;
- Aplicar o teste de hipóteses;
- Se o p-valor obtido for superior ao grau de significância do teste, então a hipótese é aceita (corroborada), caso o p-valor seja inferior ao grau de significância do teste, então a hipótese é rejeitada.

7.3.2 Método de verificação de normalidade de uma distribuição

Considera-se que uma amostra possui distribuição normal quando:

- Possuir mais de 30 elementos (Teorema do Limite Central);
- For testada por métodos específicos com esse fim (Anderson-Darling, Kolmogorov-Smirnov e Shapiro-Wilk) e for aceita. Sempre se aceita quando p-valor for maior ou igual a 0,03. No processo destes testes pode ocorrer de serem rejeitados, sendo possível então partir para duas estratégias de manipulação de população antes de refazer o cálculo:
 - 1 – elevar toda a população a alguma potência;
 - 2 – aumentar o tamanho das populações, quando possível.

Estes testes normalmente são usados quando o número de elementos da amostra for inferior a 30 ou quando há fortes indícios de que a distribuição não seja normal.

- Pode-se perceber sua normalidade pela sua curva de distribuição de frequência de ocorrências de seus valores, a qual se assemelha a uma curva de Gauss;
- Analisando seu papel de probabilidade da amostra. Papel de probabilidade é um gráfico que plota uma diagonal e os pontos referentes às amostras, passados por uma função. Considera-se distribuição normal quando estes pontos não fogem muito dessa diagonal. Percebe-se que é um teste subjetivo, mas comumente usados.

Estas duas últimas formas são bastante subjetivas e normalmente são usadas mais como forma de ilustrar a escolha.

7.3.3 Método de cálculo de tamanho amostral por *t-student*

Um fator que influencia a qualidade do poder de teste é o tamanho das amostras usado nele. O tamanho amostral é calculado usando-se as médias, desvios padrões e tamanho do efeito das amostras iniciais. Este tamanho indica o tamanho mínimo que a amostra deve possuir para que se obtenha o poder de teste desejado.

Tamanho do efeito é calculado da seguinte forma: divide-se o valor absoluto da diferença das médias das amostras pela média de seus desvios padrões.

$$te = |u_1 - u_2| / ((\sigma_1 + \sigma_2) / 2) \quad (10)$$

Onde te é o tamanho de efeito, u_1 e u_2 são as médias das amostras e σ_1 e σ_2 são os desvios padrões das amostras.

Como o cálculo do tamanho amostral é feito de forma recursiva, a maioria dos pacotes estatísticos gratuitos o possuem implementado.