

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**Modelo Adaptativo Fuzzy Genético,
Preditivo e Multiobjetivo para Despacho de
AGVs.**

VINICIUS FERNANDES CARIDÁ

ORIENTADOR: PROF. DR. ORIDES MORANDIN JUNIOR

São Carlos - SP

Maio/2011

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**Modelo Adaptativo Fuzzy Genético,
Preditivo e Multiobjetivo para Despacho de
AGVs.**

VINICIUS FERNANDES CARIDÁ

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

Orientador: Prof. Dr. Orides Morandin Junior

São Carlos - SP

Maio/2011

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

C277ma

Caridá, Vinicius Fernandes.

Modelo adaptativo fuzzy genético, preditivo e multiobjetivo para despacho de AGVs / Vinicius Fernandes Caridá. -- São Carlos : UFSCar, 2011.

123 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2011.

1. Inteligência artificial. 2. Despacho de AGVs. 3. Fuzzy logic. 4. Algoritmos genéticos. 5. Sistemas flexíveis de manufatura. 6. Representação de conhecimento. I. Título.

CDD: 006.3 (20^a)

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

“Modelo Adaptativo Fuzzy Genético, Preditivo e Multiobjetivo para Despacho de AGVs”

VINICIUS FERNANDES CARIDÁ

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


Membros da Banca:



Prof. Dr. Orides Morandin Júnior
(Orientador - DC/UFSCar)



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



Prof. Dr. Alexandre Cesar Rodrigues da Silva
(UNESP/Ilha Solteira)

São Carlos
Junho/2011

*Dedico esta conquista à minha mãe Rosimeire Fernandes e aos meus quatro irmãos,
Nathalia, Vitor Hugo, Luan e Amanda.*

AGRADECIMENTOS

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 namorada, por todo seu carinho e atenção.

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

Ao meu orientador, professor Orides, pela confiança e auxílio, dando suporte necessário para realização desse trabalho.

Aos meus grandes amigos peruanos, pela amizade sincera e pela enorme partilha de cultura.

Aos meus antepassados, pois se não fosse pela perda de suas terras, hoje a universidade não existiria.

Ao professor Mauro Biagiz, que foi um dos incentivadores para que eu ingressasse no mestrado.

A professora Heloisa por toda sua simpatia e ajuda nas correções.

Ao meu amigo Carlos Tuma pela grande ajuda na implementação.

Aos meus amigos do laboratório de pesquisa, Marcos, Flávio, Mayra, Wesley, Ageu por toda ajuda, companheirismo e momentos de descontração.

A todos os professores do Departamento de Computação.

A CAPES pela concessão da bolsa de estudos.

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

"Seja você quem for, seja qual for a posição social que você tenha na vida, a mais alta ou a mais baixa, tenha sempre como meta muita força, muita determinação e sempre faça tudo com muito amor e com muita fé em Deus, que um dia você chega lá. De alguma maneira, você chega lá".

Ayrton Senna

RESUMO

Nos últimos anos, mais empresas aplicam técnicas de automação industrial com o objetivo de aumentar sua eficiência. Cada vez mais Veículos Auto Guiados (AGVs) são usados para tarefa de transporte em indústrias e armazéns. O gerenciamento desses AGVs é a chave para um sistema de transporte eficiente. Um dos principais problemas encontrados no gerenciamento dos AGVs é a decisão de despacho. Esse trabalho propõe um método de despacho de AGV com capacidade de otimizar o desempenho de Sistemas Flexíveis de Manufatura (FMS) em tempo real. Na tomada de decisão do despacho podem ser escolhidos dois objetivos para otimização: *makespan* ou *tardiness*. Quando o objetivo é otimizar o *makespan* são avaliadas as variáveis: distância, número de nós, *buffer* de entrada e *buffer* de saída das estações de trabalho. Quando o objetivo é otimizar o *tardiness* são avaliadas as variáveis: distância, número de nós, encadeamento de tarefas, data de entrega do produto. Em ambos os casos também são considerados os tempos dos AGVs e os tempos de processamento das estações de trabalho, para que seja possível adiantar algumas tomadas de decisões. Para realizar os testes foram usados os *softwares* MatLab, CPNtools e Automod, os quais permitem que simulações do funcionamento do método de despacho de AGVs antes que o método seja implementado em fábricas. Com base nos testes realizados, analisou-se as melhoras no *makespan* e *tardiness* em comparação com outros trabalhos, como também a flexibilidade do método no qual é possível a troca de objetivos em tempo de operação.

Palavras-chave: Despacho de AGV. Veículo Auto Guiado. Lógica *Fuzzy*. Algoritmo Genético. Controle de AGV. Sistemas Flexíveis de Manufatura.

ABSTRACT

In recent years, most companies apply techniques of industrial automation with the goal of increasing its efficiency. Increasingly Automated Guided Vehicles (AGVs) are used to transport work in factories and warehouses. The management of these AGV is the key to an efficient transport system. One of the main problems encountered in the management of AGV is the dispatching decision. This work proposes a method of dispatching of AGV with the ability to optimize the performance of flexible manufacturing systems (FMS) in real time. In the dispatching deciding can be chosen to optimize two objectives: makespan or tardiness. When the goal is to optimize the makespan are evaluated variables: distance, number of nodes, the input buffer and output buffer of the workstation. When the goal is to optimize the tardiness are evaluated variables: distance, number of nodes, task chaining, date of delivery. In both cases are also considered the time of the AGVs and the processing times of the workstations, so you can anticipate some decision-making. To perform the tests we used the softwares: Matlab, CPNtools and Automod, which allow simulations of how the method works dispatching AGVs before the method is implemented in factories. Based on the tests, we analyzed the improvement in makespan and tardiness compared to other studies, but also the flexibility of the method in which it is possible to exchange goals in operating time.

Keywords: AGV dispatching. Automated Guided Vehicle. Fuzzy Logic. Genetic Algorithm. AGV control. Flexible Manufacturing Systems.

LISTA DE FIGURAS

Figura 1 - Inserção do projeto	18
Figura 2 - As características de volume e variedade das tecnologias de manufatura, adaptado de (SLACK et al., 1997).....	22
Figura 3 - Exemplo de um FMS, adaptado de Dycus (1998).....	24
Figura 4 - Exemplos de AGVs (Retirado dos sites dos fabricantes JBT Corporation e www.automatedguidedvehicle.org).....	25
Figura 5 - Três tipos de veículos Auto Guiados: (a) Trem sem condutor, (b) AGV transportador de <i>pallets</i> e (c) transportador de Carga Unitária, adaptado de (GROOVER, 2003).....	26
Figura 6 - Representação geral de uma função de pertinência triangular	32
Figura 7 - Representação geral de uma função de pertinência trapezoidal.....	33
Figura 8 - Representação geral de uma função de pertinência Gaussiana	34
Figura 9 - Gráfico representando dois conjuntos <i>fuzzy</i>	35
Figura 10 - Gráfico resultante da intersecção, considerando os conjuntos da Figura 9.....	35
Figura 11 - Gráfico resultante da união, considerando os conjuntos da Figura 9.	36
Figura 12 - Gráfico resultante do complemento, considerando o conjunto F da Figura 9.....	36
Figura 13 - Representação do sistema fuzzy Mamdani.....	43
Figura 14 - Exemplo de Base de Regras criada em Matlab	43
Figura 15 - Exemplo de base de dados criada em Matlab	44
Figura 16 - Exemplo de <i>Fuzzyficação</i>	44
Figura 17 - Exemplo de <i>defuzzificação</i> centroide criada em Matlab.....	45
Figura 18 - Tela inicial do toolbox <i>fuzzy</i>	46
Figura 19 - Tela de configurações da base de dados	46
Figura 20 - Tela de configurações da base de regras	47
Figura 21 - Tela dos valores resultantes	47
Figura 22 - Algoritmo Genético típico (CARVALHO; BRAGA; LUDERMIR, 2003).....	49
Figura 23 - Exemplo de cromossomo / população.	50
Figura 24 - Método de seleção por roleta.....	51
Figura 25 - Método de seleção por torneio.....	52
Figura 26 - Exemplo de cruzamento	53
Figura 27 - Exemplo de mutação	54
Figura 28 - Exemplo de uma rede de Petri lugar/transição	57

Figura 29 - Exemplos de Redes de Petri (MURATA, 1989)	57
Figura 30 - Exemplo da base de dados de uma variável criada em Matlab.....	79
Figura 31 - Exemplo de uma base de regras criada em Matlab	80
Figura 32 - Exemplo de cromossomo	80
Figura 33 - Exemplo de otimização da base de regras apresentada na Figura 31.....	82
Figura 34 - Layout do cenário 1, adaptado de (BENINCASA; MORANDIN JR; KATO, 2003)	85
Figura 35 - Layout do cenário 2, adaptado de (REDDY; RAO, 2006).....	87
Figura 36 - Layout do cenário 3, adaptado de (REDDY; RAO, 2006).....	89
Figura 37 - Base de dados do objetivo <i>makespan</i>	92
Figura 38 - Base de dados do objetivo <i>makespan</i> , variável prioridade.....	92
Figura 39 - Base de dados do objetivo <i>makespan</i> , variável <i>buffer</i> de saída.....	93
Figura 40 - Base de dados do objetivo <i>makespan</i> , variável <i>buffer</i> de entrada	93
Figura 41 - Base de dados do objetivo <i>makespan</i> , variável distância	94
Figura 42 - Base de dados do objetivo <i>makespan</i> , variável número de nós.....	94
Figura 43 - Base de dados do objetivo <i>tardiness</i>	95
Figura 44 - Base de dados do objetivo <i>tardiness</i> , variável prioridade.....	96
Figura 45 - Base de dados do objetivo <i>tardiness</i> , variável data de entrega.....	96
Figura 46 - Base de dados do objetivo <i>tardiness</i> , variável encadeamento de tarefas	97
Figura 47 - Base de dados do objetivo <i>tardiness</i> , variável distância	97
Figura 48 - Base de dados do objetivo <i>tardiness</i> , variável número de nós.....	98
Figura 49 - Exemplo de funcionamento do cruzamento	101
Figura 50 - Exemplo de funcionamento da mutação	101
Figura 51 - Modelo em redes de Petri do cenário 1	103
Figura 52 - Modelo em redes de Petri do cenário 2	104
Figura 53 - Modelo em redes de Petri do cenário 3	104
Figura 54 - Amostra de gráfico para alocação da produção	108
Figura 55 - Simulação do objetivo <i>tardiness</i>	108
Figura 56 - Comparação entre produções do objetivo <i>makespan</i> e do objetivo <i>tardiness</i> ..	109

LISTA DE TABELAS

Tabela 1 - Elementos da Rede de Petri (MURATA, 1989).	56
Tabela 2 - Distâncias do cenário 1	85
Tabela 3 - Número de nós do cenário 1	86
Tabela 4 - Roteiros de produção do cenário 1	86
Tabela 5 - Tempo de processamento das estações de trabalho do cenário 1	86
Tabela 6 - Distâncias do cenário 2.....	87
Tabela 7 - Número de nós do cenário 2.....	87
Tabela 8 - Roteiros de produção do cenário 2	88
Tabela 9 - Tempo de processamento das estações de trabalho do cenário 2.....	88
Tabela 10 - Distâncias do cenário 3.....	89
Tabela 11 - Número de nós do cenário 3.....	90
Tabela 12 - Roteiros de produção do cenário 3	90
Tabela 13 - Tempo de processamento das estações de trabalho do cenário 3.....	91
Tabela 14 - Amostra da base de regras do objetivo <i>makespan</i>	99
Tabela 15 - Atribuição de valores reais.....	100
Tabela 16 – Amostra dos cromossomos do AG.....	100
Tabela 17 - Amostra da base de regras otimizada do objetivo <i>makespan</i>	102
Tabela 18 - Amostra da tabela de recursos do cenário 01	105
Tabela 19 - Resultados do objetivo <i>makespan</i>	107
Tabela 20 - Resultados do objetivo <i>makespan</i>	107

LISTA DE ABREVIATURAS E SIGLAS

FMS	<i>Sistema Flexível de Manufatura (do inglês: Flexible Manufacturing System)</i>
AGV	<i>Veículo Auto Guiado (do inglês: Automated Guided Vehicle)</i>
RGV	<i>Veículo Guiado por Trilho (do inglês: Rail Guided Vehicle)</i>
SF	<i>Sistema Fuzzy</i>
SFBR	<i>Sistema Fuzzy Baseado em Regras</i>
BC	<i>Base de Conhecimento</i>
MI	<i>Mecanismo de Inferência</i>
BD	<i>Base de Dados</i>
BR	<i>Base de Regras</i>
SCF	<i>Sistema de Classificação Fuzzy</i>
AG	<i>Algoritmo Genético</i>
STM	<i>Sistema de Transporte de Materiais</i>
PMI	<i>Instituto de Gerência de Projetos (do inglês: Project Management Institute)</i>
QC	<i>Guindaste de Cais (do inglês: Quay Cranes)</i>
CGA	<i>Algoritmo Genético Canônico (do inglês: Canonical Algorithm Genetic)</i>
RAF	<i>Raciocínio para Antecipar o Futuro</i>
AD-FMS	<i>Autonomia Descentralizada em Sistema Flexível de Manufatura (do inglês: Autonomous Decentralized in Flexible Manufacturing Systems)</i>
PMBOK	<i>Conjunto de conhecimento do Gerenciamento de Projetos (do inglês: Project Management Body of Knowledge)</i>
TAD	<i>Tabela de Alocação Dinâmica</i>
SFG	<i>Sistemas Fuzzy Genéticos</i>
PN	<i>Redes de Petri (do inglês: Petri Nets)</i>
RMS	<i>Sistema de Manufatura Reconfigurável</i>

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO	15
1.1 Contextualização	15
1.2 Justificativa e Motivação	16
1.3 Objetivo	18
1.4 Delimitações de escopo	18
1.5 Gerenciamento do projeto.....	19
1.6 Revisão Sistemática	19
1.7 Estrutura do Trabalho	20
CAPÍTULO 2 - AMBIENTE DE MANUFATURA.....	21
2.1 Sistemas Flexíveis de Manufatura (FMS)	21
2.1.1 Componentes do FMS	24
2.2 Veículos Auto Guiados (AGVs).....	25
2.2.1 Tipos de AGVs.....	26
2.2.2 Aplicações de AGV	27
2.2.3 Vantagens dos AGV	27
CAPÍTULO 3 - CONCEITOS FUNDAMENTAIS	29
3.1 Sistemas <i>Fuzzy</i> (Nebulosos)	30
3.1.1 Breve Histórico	30
3.1.2 Conjuntos <i>Fuzzy</i>	31
3.1.3 Funções de pertinência.....	31
3.1.4 Operações básicas sobre conjuntos <i>Fuzzy</i>	34
3.1.4.1 Outros operadores.....	36
3.1.5 Operações Generalizadas sobre conjuntos <i>Fuzzy</i>	37
3.1.6 Computação usando regras <i>fuzzy</i>	38
3.1.7 Variáveis Linguísticas	38

3.1.8 Regras <i>Fuzzy</i>	39
3.1.9 Inferência <i>Fuzzy</i>	41
3.1.10 Sistemas <i>Fuzzy</i>	42
3.1.11 Sistemas fuzzy no Matlab	45
3.2 Algoritmos Genéticos.....	48
3.2.1 Estrutura Clássica de um AG.....	48
3.2.2 Esquemas de Codificação Empregados nos AG.....	49
3.2.3 Seleção.....	50
3.2.4 Elitismo.....	52
3.2.5 Operadores Genéticos.....	53
3.2.6 Controle dos Parâmetros Genéticos	54
3.2.7 Sistemas <i>fuzzy</i> genéticos.....	55
3.3 Redes de Petri.....	55
3.3.1 Definições de Rede de Petri	56
3.3.2 Propriedades em Redes de Petri	58
3.3.3 Métodos de Análises.....	60
3.3.4 Extensões de Redes de Petri.....	60
3.3.5 Considerações finais.....	60
CAPÍTULO 4 - TRABALHOS RELACIONADOS A AGV	62
4.1 Problemas gerais relacionados aos AGVs	63
4.2 Uso de regras de despacho para atribuição de tarefas dos AGVs	67
CAPÍTULO 5 - PROPOSTA	75
CAPÍTULO 6 - APLICAÇÃO DA ABORDAGEM E ANÁLISE DE RESULTADOS.....	83
CAPÍTULO 7 - CONCLUSÃO	111
7.1 Trabalhos futuros.....	113
REFERÊNCIAS BIBLIOGRÁFICAS.....	114

Capítulo 1

INTRODUÇÃO

1.1 Contextualização

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 a função de atingir vantagens competitivas, na medida em que são conciliados para produzir de acordo com as expectativas dos clientes.

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), no qual sua principal função é mover peças e outros materiais de um local para outro, de maneira segura, eficiente e com precisão.

Segundo Groover (2003) e Stecke (1986) a maioria dos FMSs emprega os veículos auto guiados (AGVs) como transportadores. Os AGVs são veículos auto guiados utilizados 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.

Programar corretamente as atividades dos AGVs e as cargas transportadas é um ponto chave para a eficiência e segurança do sistema de transporte. Há necessidade de se tomar várias decisões referentes à programação dos AGVs, tais como controle de trânsito e regras de despacho dos AGVs. A operação do sistema de transporte deve ser eficiente para justificar o alto investimento envolvido.

1.2 Justificativa e Motivação

A busca pela competitividade no meio industrial tornou-se uma 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 utilizadas no intuito de atingir esse propósito (AGUIRRE, 2007).

A “robotização” de centros de fabricação, distribuição e armazenagem de produtos é uma das práticas que vem ocorrendo no Brasil para minimizar custos de produção, de logística e de distribuição. Visto que essas práticas já ocorrem em países desenvolvidos com sucesso, pesquisas nessa área devem ser consideradas como estratégicas para garantir uma maior inserção do Brasil no cenário de comércio internacional.

A movimentação interna de materiais 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. Um sistema de transporte utilizando AGVs pode trazer inúmeros benefícios. Primeiro, o sistema de transporte integrado com outros setores da empresa pode utilizar informações advindas desses setores assim ponderando múltiplas variáveis de diversos setores para tomada de decisão, contribuindo para o objetivo final da empresa. Os AGVs também podem ser utilizados em ambientes hostis para a atividade humana. Mesmo em ambientes favoráveis para a atividade humana, os AGVs podem minimizar outros custos empresariais. Anualmente são registrados milhares de acidentes de trabalho envolvendo empilhadeiras. A visão mais comum sobre o acidente de trabalho é aquela ligada às pessoas, mas não devemos nos esquecer que os acidentes causam inúmeros prejuízos materiais. Quando acontece um acidente, máquinas e equipamentos podem ser quebrados ou danificados paralisando linhas de produção, atrasando entregas e causando perdas financeiras que às vezes levam tempo para voltar à normalidade.

Para empresas em que o investimento para compra de AGVs é inviável, pode ser implantado o método de despacho proposto. Uma vez que o mesmo pode ser embarcado em equipamentos como *palms* e / ou *smarts phones*, esses dispositivos são distribuídos para os operadores das empilhadeiras, assim todas as decisões de atendimento são enviadas pelos dispositivos para que os operadores as cumpram.

Independente do tipo de sistema de transporte, automático (AGVs) ou manual (empilhadeiras), a decisão de atendimento precisa ser em tempo real, o que dificulta na ponderação de múltiplas variáveis, muitas vezes obrigando a decisão se basear em apenas uma variável, por exemplo, atenda o que solicitou primeiro, atenda quem estiver mais perto, etc.

Um método de despacho de AGVs que pondere não somente variáveis do chão de fábrica, mas também variáveis ligadas as estratégias de produção e estratégias de negócios da empresa e que seja robusto o suficiente para ter múltiplos objetivos, seguindo o objetivo atual da empresa e operar em *layouts* pequenos ou de grande porte em tempo real, pode oferecer grandes vantagens no cenário industrial, onde existe grande competitividade e as exigências são cada vez maiores, buscando-se constantemente melhorar a qualidade, diminuir custos, atender os prazos, entre outros.

Seguindo essa temática, bem como apoiado na revisão bibliográfica e sendo direcionado pelas linhas de pesquisas do Laboratório de Pesquisa e Inovação em Tecnologias e Estratégias de Automação (Tear), este trabalho busca dar continuidade aos trabalhos desenvolvidos dentro desse tema de pesquisa melhorando o desempenho do despacho dos veículos auto guiados.

A Figura 1 dá uma visão da inserção do trabalho nas áreas de pesquisa e nos temas do Laboratório Tear. A junção entre os problemas que são abordados com as técnicas e métodos gera a linha de pesquisa que é aplicada nas áreas descritas, que seguem a terminologia e amarrações lógicas definidas pelas agências de fomento.

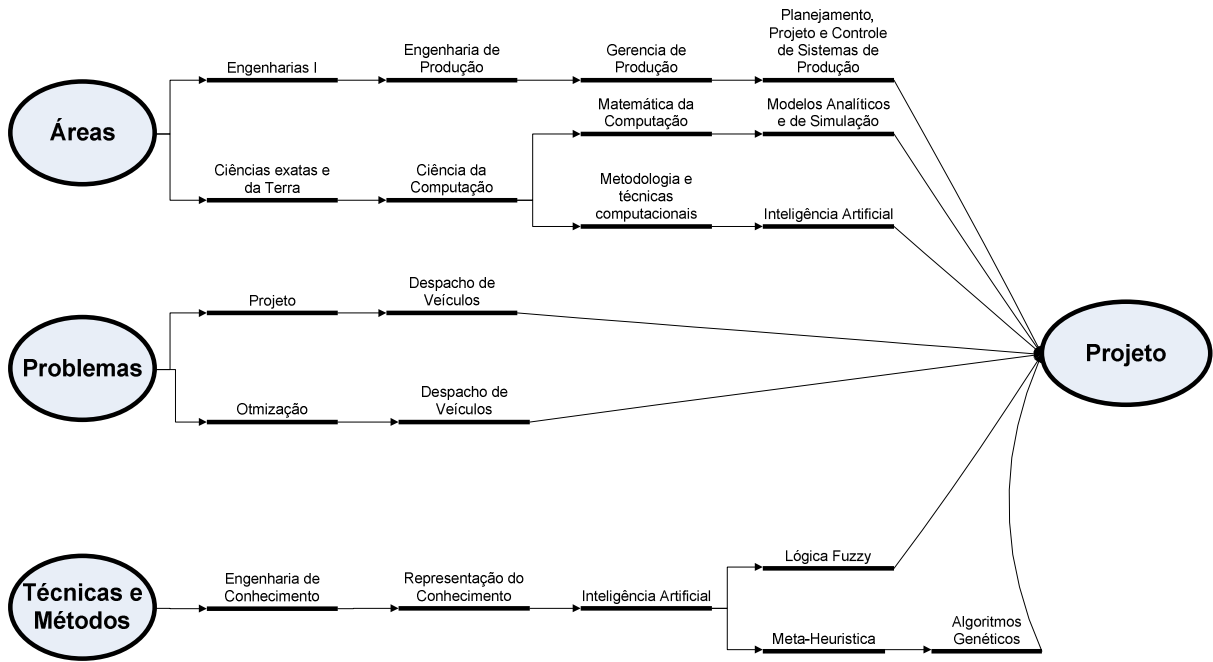


Figura 1 - Inserção do projeto

1.3 Objetivo

Nesse contexto, nesse trabalho desenvolveu-se um modelo de despacho de veículos aplicado em Sistemas Flexíveis de Manufatura. O modelo avalia múltiplas variáveis, sendo elas de vários setores da empresa. O modelo incorpora mais de um objetivo final e que esses podem mudar em função da conjuntura da produção.

Este modelo trata o despacho em tempo real visando à otimização do tempo total de produção (*makespan*) e do atraso de produção (*tardiness*).

1.4 Delimitações de escopo

Embora o trabalho tenha se apoiado em múltiplas variáveis, e também se utilizando de modelagens e simulações para aproximar-se ao máximo de processos produtivos reais, ainda sim algumas variantes deixaram de ser consideradas, como:

- Tempo de *setup* das estações de trabalho;

- Troca de turno dos operadores;
- Manutenção e abastecimento dos veículos;
- Entre outros.

Foram desconsiderados alguns eventos inesperados que possam acontecer, como por exemplo, quebra de máquinas. No objetivo *tardiness* um evento inesperado pode impactar bastante nos prazos. Porém como será descrito no texto, os tempos sem produção podem ser usados para manutenção preventiva, o que pode diminuir a quantidade de eventos inesperados desse tipo.

1.5 Gerenciamento do projeto

Um projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo. O gerenciamento de projetos é a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos (PMBOK, 2008).

O Guia do Conhecimento em Gerenciamento de Projetos (PMBOK) é um trabalho reconhecido para o gerenciamento de projetos que descreve normas, métodos, processos e práticas estabelecidas. O conhecimento contido nesse guia evoluiu a partir das boas práticas reconhecidas de profissionais de gerenciamento de projetos que contribuíram para o seu desenvolvimento (PMBOK, 2008).

Uma das práticas abordadas no laboratório Tear é considerar que todo trabalho de iniciação científica, mestrado ou doutorado é um projeto. A fim de garantir principalmente um controle de escopo, qualidade e prazo foram adotadas as boas práticas descritas no PMBOK para gerenciamento desse projeto de mestrado.

1.6 Revisão Sistemática

De acordo com Kitchenham (2004) e Pai et al. (2004), o objetivo central da revisão sistemática é sintetizar a evidência, identificando, avaliando e interpretando todas pesquisas

disponíveis em relação a uma questão específica. Visa também produzir uma síntese que é completa e imparcial, seguindo um processo bem definido e aberto.

A fim de realizar uma revisão da literatura de forma organizada, relacionada diretamente com o tema, construindo um entendimento dos conceitos teóricos e das terminologias utilizadas, como também sintetizar os resultados e facilitar a elaboração de lista de fontes que foram consultadas, foram usadas no trabalho algumas diretrizes da revisão sistemática.

Para auxiliar na documentação da revisão, foi usado o software start, versão 1.01, que foi desenvolvido pelo Laboratório de Pesquisa em Engenharia de Software (LaPES) do departamento de computação da Universidade Federal de São Carlos (UFSCar) [<http://lapes.dc.ufscar.br/ferramentas/start>].

1.7 Estrutura do Trabalho

No intuito de trazer os principais conceitos envolvidos na proposta desse trabalho, no capítulo 2 discute-se o ambiente ao qual a abordagem se aplica, e no capítulo 3 é apresentado uma visão geral sobre as principais técnicas que serão utilizadas.

A revisão de trabalhos correlatos é apresentada em seguida (capítulo 4), buscando incluir algumas considerações de como o problema de despacho de veículos têm sido tratado, quais as técnicas que são utilizadas na tentativa de resolução do problema e quais as principais questões discutidas nessas aplicações. Inicialmente são apresentados estudos compreendendo diversas áreas pertinentes ao sistema de transporte. Em seguida o trabalho é focado nas regras de despacho, que é o ponto principal da proposta desse trabalho.

A partir das considerações imprescindíveis dos capítulos anteriores, no capítulo 5 é apresentada a proposta de trabalho visando descrever a principal contribuição desse trabalho, cabendo ao capítulo posterior, capítulo 6, o detalhamento do método desenvolvido, discernindo suas principais especificidades e definições que condizem com a perspectiva de resolução dada ao problema considerado.

Para a avaliação da abordagem, são apresentadas ainda no capítulo 6 as definições do cenário avaliado, sendo os resultados apresentados e discutidos no mesmo capítulo.

No capítulo 7 são apresentadas as conclusões e as propostas para trabalhos futuros.

Capítulo 2

AMBIENTE DE MANUFATURA

Neste capítulo são apresentados conceitos fundamentais sobre o ambiente em que o projeto é aplicado.

O capítulo é iniciado ressaltando a importância dos sistemas produtivos, e entre os vários tipos encontrados em atividade estão os Sistemas Flexíveis de Manufatura (FMS). Assim são apresentadas as principais características dos FMS e as três categorias principais de componentes dos FMS. Dentre os componentes de um FMS estão os veículos auto guiados, que é o foco do projeto, sobre o qual serão apresentados seus principais tipos, aplicações e vantagens.

Esses conceitos são de fundamental importância para a compreensão do funcionamento e da complexidade da proposta do projeto.

2.1 Sistemas Flexíveis de Manufatura (FMS)

Excelência em sistemas de manufatura tem sido reconhecido como um dos principais fatores por trás do sucesso de empresas industriais ou de produção. As novas tecnologias de processos de fabricação desempenham um papel significativo neste processo. Atingir o potencial dessas inovações tecnológicas de produção, no entanto, necessita de uma ampla gama de gestão, engenharia e questões relativas ao sistema (RAOUF; BEN-DAYA, 1995).

Tradicionalmente, a implantação de sistemas de produção tem dois objetivos conflitantes: flexibilidade e produtividade. Flexibilidade refere-se a produzir uma série de produtos distintos em um ambiente *jobshop*¹ no qual existe oportunidade de produção variável. Produtividade, por outro lado, refere-se a produção de alta velocidade que é semelhante a uma linha de montagem. Estudos têm demonstrado que a produtividade em um ambiente *jobshop* é baixo quando comparado ao *flowshop*² (RAOUF; BEN-DAYA, 1995); (HUTCHINSON, 1984).

A flexibilidade tem uma influência positiva no aumento da lucratividade e da participação no mercado. Vale ainda lembrar que a flexibilidade gera impactos em outros fatores de desempenho como custo, velocidade e confiabilidade de entrega (AGUIRRE, 2007).

A relação geral entre flexibilidade e volume de produção é representada na Figura 2. Esta figura define que sistemas flexíveis de manufatura são adequados para produção com média variedade de peças e médio volume de produção.

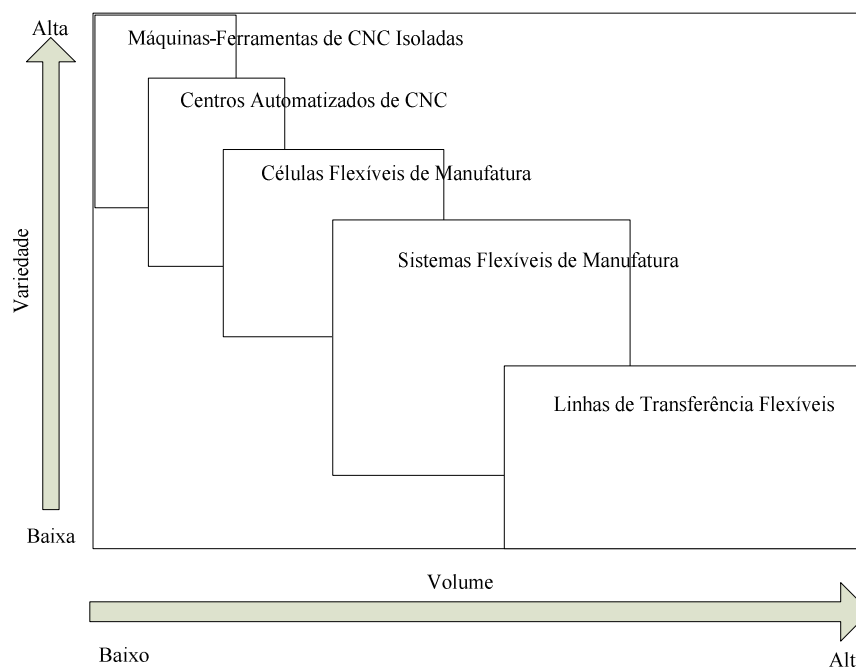


Figura 2 - As características de volume e variedade das tecnologias de manufatura, adaptado de (SLACK et al., 1997)

¹ Sistema de produção *jobshop*: os itens fabricados não têm o mesmo roteiro de fabricação, alta personalização, volumes mais baixos (FERNANDES; FILHO, 2010).

² Sistema de produção *flowshop*: produção em linha, todos os itens seguem a mesma sequência de operações, ou com pequenas variações, volume alto (FERNANDES; FILHO, 2010).

Apesar de todo o interesse em sistemas flexíveis de manufatura, não existe um padrão sobre a definição de Sistemas Flexíveis de Manufatura (FMS). A principal característica que distingue o FMS de sistemas de produção tradicionais é a "flexibilidade" (ZHOU, 1998).

Um sistema flexível de manufatura (FMS) pode ser caracterizado como um agrupamento de equipamentos, considerados como estações de processamento, interligados por um sistema automatizado de carga e descarga e de movimentação de peças e ferramentas e um sistema de armazenamento. Todo sistema é controlado por computador (MORANDIN, 1999); (RAOUF; BEN-DAYA, 1995). No esquema de classificação para sistemas de manufatura, um FMS deve possuir múltiplas estações automatizadas e envolver vários conceitos e tecnologias como: máquinas CNC, controle computadorizado, transporte de material e armazenamento automatizado (GROOVER, 2003).

Após revisão bibliográfica Gupta e Goyal (1989), Sethi e Sethi (1990) enumeraram vários tipos de flexibilidade e são sugeridas formas para medi-las e avaliá-la. Tempelmeier e Kuhn (1993) apresentaram os tipos essenciais de flexibilidade encontrados na literatura. A seguir, são citados somente os tipos que são pertinentes a esse trabalho:

- **Flexibilidade de roteamento:** é a habilidade do sistema de manufatura em produzir uma peça por rotas alternativas através do sistema;
- **Flexibilidade na manipulação e transporte de material:** é a habilidade do sistema automático de manipulação e transporte de material em mover diferentes peças eficientemente para posições apropriadas e conduzi-las pelo sistema;
- **Flexibilidade de mercado:** é a habilidade do sistema, em se adaptar as mudanças e flutuações do mercado. Ela permite a organização responder às mudanças sem comprometer seus negócios de novas oportunidades de negócios.

Na Figura 3 é ilustrado um Sistema de Flexível de Manufatura e seus componentes.

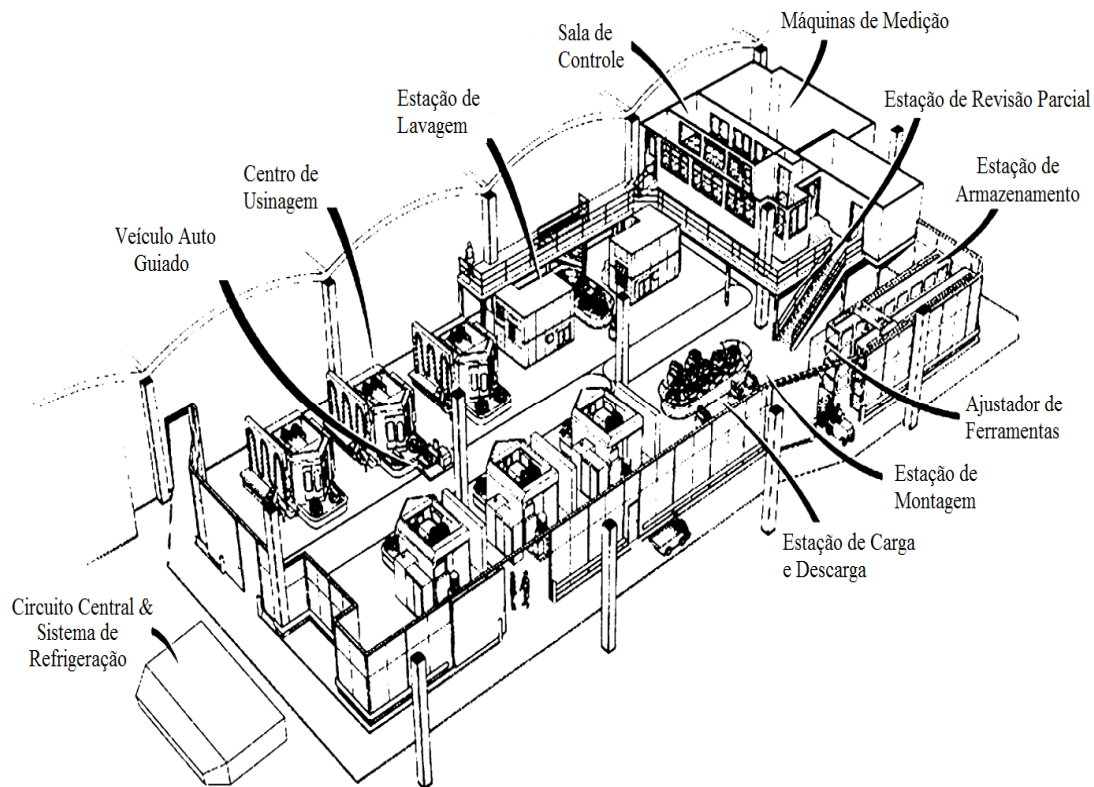


Figura 3 - Exemplo de um FMS, adaptado de Dycus (1998)

2.1.1 Componentes do FMS

Segundo Morandin (1999), os componentes de um FMS consistem de três categorias principais:

- Estações de Processamento, do inglês “*Workstations*”: Embora existam vários tipos de estações de processamento, dependendo do processo de manufatura, as principais são: estações de máquinas, estações de carga e descarga, estações de montagem, estações de controle da qualidade, limpeza, refrigeração, etc;
- Sistemas de manipulação, movimentação e armazenagem de materiais: os principais são: Braço robô, veículo guiado por trilho (RGV) e veículo auto guiado (AGV);
- Sistemas de Controle por Computador: Esse sistema é usado para coordenar as atividades das estações de processamento e o sistema de manipulação, movimentação e armazenagem de materiais.

2.2 Veículos Auto Guiados (AGVs)

Em uma fábrica moderna, a flexibilidade e a reconfiguração dinâmica estão rapidamente se tornando uma norma. Ao mesmo tempo, há uma crescente percepção de que a manipulação de materiais é um componente chave no cumprimento das metas de flexibilidade, fabricação, dinamismo e agilidade. Capacidade de manipulação avançada de materiais é essencial, pois sem esta capacidade de fornecer o material necessário para a estação de trabalho adequada na hora certa e na quantidade certa toda a instalação irá travar, tornando-se menos eficientes e produzindo menos lucros e/ou operando com custos mais elevados (JOSHI; SMITH, 1994).

Uma tecnologia de movimentação de material que aborda essas preocupações é o Veículo Auto Guiado (AGV). Um AGV é um móvel utilizado no transporte automático de materiais em ambientes de manufatura, projetado para receber e executar instruções, percorrer um caminho ou trajetória, receber e distribuir materiais. Os veículos geralmente seguem uma trilha que pode assumir muitas formas e que geralmente podem ser facilmente re-configuradas de acordo com a planta da fábrica. As instruções para um AGV indicam para onde o veículo deve se locomover, como deve chegar ao destino e o que fazer quando chegar ao mesmo (JOSHI; SMITH, 1994).



Figura 4 - Exemplos de AGVs (Retirado dos sites dos fabricantes JBT Corporation e www.automatedguidedvehicle.org)

Os AGVs podem satisfazer as exigências da nova abordagem de produção desde que o sistema de trilha seja flexível e os comandos para os AGVs reflitam condições de chão de fábrica. Embora esta tecnologia tenha sido utilizada com sucesso, há ainda espaço para melhorias. Em sistemas flexíveis de manufatura, é necessário fazer pleno uso da

flexibilidade mecânica fornecida por um sistema de AGV. Tratar condições não previstas como um sistema que possa raciocinar e chegar a uma solução, seria uma importante adição à tecnologia, produtividade e eficiência dos sistemas de fabricação (JOSHI; SMITH, 1994).

2.2.1 Tipos de AGVs

Veículos Auto Guiados podem ser divididos em três categorias: (1) trens sem condutor, (2) transportadores de *pallets* e (3) transportadores de carga única, ilustrado na Figura 5 (GROOVER, 2003).

O trem sem condutor consiste em um veículo de tração que puxa um ou mais carros de reboque, como ilustrado na Figura 5(a). Foi o primeiro tipo de AGV a ser introduzido e é amplamente usado até hoje. Uma aplicação comum é mover cargas pesadas em longas distâncias em armazém ou fábricas, com ou sem coleta intermediária e descarrega-os em pontos ao longo do percurso.

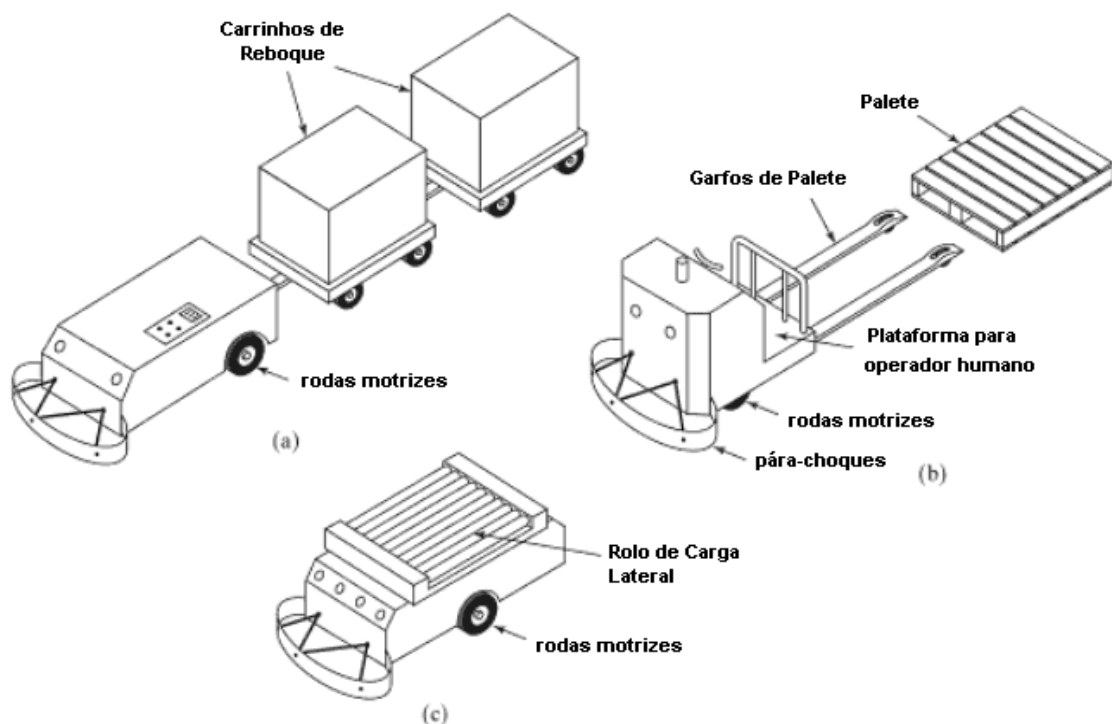


Figura 5 - Três tipos de veículos Auto Guiados: (a) Trem sem condutor, (b) AGV transportador de *pallets* e (c) transportador de Carga Unitária, adaptado de (GROOVER, 2003).

AGV transportador de *pallets*, Figura 5(b), são usados para mover cargas ao longo de rotas pré-determinadas. Em sua aplicação típica, o veículo é manobrado por um trabalhador humano que guia o AGV e usa seus garfos para elevar a carga ligeiramente e programa seu destino. A capacidade de um AGV transportador de *pallets* varia até milhares de quilos, e alguns AGVs são capazes de manipular dois *pallets*. Este veículo pode atingir significativa versatilidade no movimento dos seus garfos para alcançar as cargas em prateleiras (GROOVER, 2003).

Transportadores de carga unitária são usados para movimentar cargas de uma estação para outra. Eles são muitas vezes equipados para carregamento e descarregamento automático de *pallets* por meio de rolos, plataformas de elevação mecanizadas, ou outros dispositivos construídos dentro do veículo. O AGV de carga unitária é ilustrado na Figura 5(c). O AGV de carga unitária por ser subdividido em AGVs de carga leve e AGVs de linha de montagem. O AGV de carga leve é um veículo relativamente pequeno, normalmente com capacidade de 250 kg ou menos. Os AGVs de carga leve são projetados para levar pequenas cargas (peças únicas, cestas de pequeno porte, lote de peças) através das plantas de tamanho limitado envolvido em manufatura. O AGV de linha de montagem foi concebido para transportar uma parte parcialmente concluída através de uma seqüência de estações de trabalho para construir o produto (GROOVER, 2003).

2.2.2 Aplicações de AGV

Sistemas de veículos auto guiados são usados em um crescente número e com variedade de aplicação. As aplicações tendem a correlacionar com os tipos de veículos descritos anteriormente. As principais aplicações de AGV em produção e logística são: operações de trens sem condutor, armazenamento e distribuição, linha de montagem e sistemas flexível de manufatura (GROOVER, 2003).

2.2.3 Vantagens dos AGV

Segundo os autores, Kulwiec (1984); Hammond (1986); Sule (1988); Goetz Jr e Egbelu (1990); King e Wilson (1991); Morandin (1999), comparando os AGVs com outros equipamentos de transporte de material, o mesmo pode apresentar várias vantagens como:

- Controle de Transporte de Material em Tempo-Real;
- Flexibilidade;

- Melhor Utilização do espaço do “chão de fábrica”;
- Adaptabilidade à Automação;
- Integração entre células de produção;
- Adaptabilidade para instalações existentes.

Em FMS, os sistemas de AGV podem proporcionar a flexibilidade, versatilidade e adaptabilidade requerida pelo ambiente. Eles podem ser re-configurados para adaptar-se a possíveis mudanças no volume de produção, variedade de produtos, rotas de produtos, etc (SCHNEIDER, 1987); (OZDEN, 1988).

Capítulo 3

CONCEITOS FUNDAMENTAIS

Neste capítulo apresenta-se conceitos básicos das técnicas que são usadas no desenvolvimento do projeto de pesquisa.

No item 3.1 são apresentados conceitos básicos da lógica *fuzzy*, que tem por objetivo modelar de modo aproximado o raciocínio humano, visando desenvolver sistemas computacionais capazes de tomar decisões racionais em um ambiente de incerteza e imprecisão. Será usada para composição das variáveis pertinentes, criando uma base de dados e uma base de regras para tomar as decisões de despacho.

Em seguida no item 3.2 são apresentados conceitos básicos dos algoritmos genéticos (AGs). Os AGs são voltados para problemas de otimização, em geral, são modelos de otimização capazes de fornecer um conjunto de soluções satisfatórias em aplicações complexas consideradas difíceis. Os AGs serão usados para fazer a otimização da base de regras do sistema *fuzzy*.

Por fim, no item 3.3 apresenta-se a rede de Petri, que tem a característica de modelar situações que ocorrem concorrência de recursos, e será usada como ferramenta para modelagem e verificação do modelo.

3.1 Sistemas *Fuzzy* (Nebulosos)

Lidar com fatores como ambiguidade, incerteza e informações vagas na resolução de problemas são características do pensamento humano, que usa o conhecimento adquirido e experiências para encontrar melhores resoluções (KLIR; YUAN, 1995); (MUKAIDONO, 2001).

Em problemas de difícil solução, em que se faz necessário o auxílio matemático/computacional, modelar tais fatores é extremamente difícil. A lógica clássica não trabalha com ambiguidades, pois se baseia nos conceitos de verdadeiro ou falso (MUKAIDONO, 2001).

A precisão computacional também é limitada, deixando sempre uma margem, por mínima que seja para a incerteza (KLIR; YUAN, 1995). Para lidar com isso de forma matemática, foi desenvolvida por Zadeh a Teoria dos Conjuntos *Fuzzy* (Nebulosos), a qual permitia serem tratados níveis de incerteza e ambiguidade.

Colocada inicialmente como uma extensão da lógica clássica, ela se baseia em que uma proposição lógica não é necessariamente verdadeira ou falsa, mas possui graus de verdade. Os valores que definem o grau de verdade de uma proposição podem não ser exatos ou quantitativos, mas de alguma forma definidos pelo contexto social, linguístico, ou uma base de referência experimental. A definição e a modelagem destes termos e valores constituem a base da teoria e do estudo dos sistemas *fuzzy*.

3.1.1 Breve Histórico

Na década de 60, Lofti A. Zadeh, professor da Universidade da Califórnia em Berkeley, desenvolveu uma variação da teoria clássica dos conjuntos e extensão da lógica de valores múltiplos de Lukasiewicz e publicou em 1965, um artigo com conceitos sobre conjuntos *fuzzy* (ZADEH, 1965). No Japão, pesquisadores esforçaram-se para estender os fundamentos da lógica *fuzzy*, introduzindo conceitos novos e desenvolvendo outras abordagens da teoria (ORTEGA, 2001).

O engenheiro britânico Ebrahim Mamdani foi o primeiro a utilizar o conceito de conjuntos *fuzzy* no projeto de um sistema de controle prático. Em 1974, Mamdani desenvolveu um controle automático de uma máquina a vapor baseado no conhecimento de um operador humano especialista (MCNEILL; THRO, 1994).

Desde a publicação dos conceitos da lógica *fuzzy* em 1965, houve muitas outras aplicações. Nos anos 80, surgiram as primeiras aplicações industriais e a partir dos anos 90, houve aumento em várias outras aplicações, destacando-se, por exemplo, os controladores *fuzzy* de refinarias, processos biológicos e químicos, tratamento de água e sistema de operação automática de trens (ORTEGA, 2001).

3.1.2 Conjuntos *Fuzzy*

Na teoria clássica dos conjuntos, um elemento pertence ou não a um dado conjunto. Isto pode ser representado por uma função característica, que associa o valor 1 (um) ao elemento de um conjunto universo se ele for membro do conjunto ou 0 (zero) se ele não for (NICOLETTI; CAMARGO, 2004).

Porém, existem situações em que os conjuntos clássicos não são flexíveis o suficiente para uma descrição de elementos, pois a teoria dos conjuntos clássicos força, a transição de pertinência absoluta para a não pertinência absoluta.

Os conjuntos *fuzzy* são usados para modelar informação imprecisa, permitindo que cada elemento pertença parcialmente a um determinado conjunto, os quais podem ser abordados como uma generalização da noção clássica de conjuntos. 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 num 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).

3.1.3 Funções de pertinência

A base da teoria de conjuntos *fuzzy* é o conceito de função de pertinência. Os formatos mais comuns são: triangular, trapezoidal e Gaussiana (KLIR; YUAN, 1995). Além desses formatos tradicionais existe uma forma de conjunto unitário (*singleton*).

Função triangular: A função triangular apresenta como:

$$\mu_{\text{triangular}}(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b \\ \frac{c-x}{c-b}, & b < x \leq c \\ 0, & x > c \end{cases} \quad (3.1)$$

Alternativamente, pode-se reescrever a formulação 3.1 como:

$$\mu_{\text{triangular}}(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (3.2)$$

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 6.

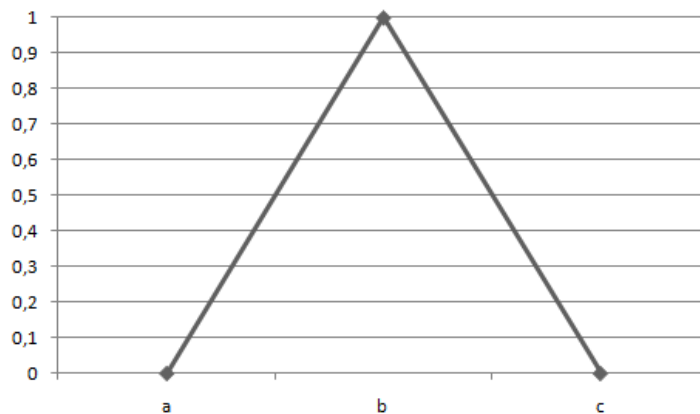


Figura 6 - Representação geral de uma função de pertinência triangular

Função trapezoidal : A função trapezoidal é especificada por um conjunto de quatro parâmetros: $\{a, b, c, d\}$, conforme a formulação 3.3:

$$\mu_{trapezoidal}(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b \\ 1, & b < x \leq c \\ \frac{d-x}{d-c}, & c < x \leq d \\ 0, & d \leq x \end{cases} \quad (3.3)$$

Alternativamente, pode-se reescrever a formulação 3.3 como:

$$\mu_{trapezoidal}(x; a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{d-x}{d-c}\right), 0\right) \quad (3.4)$$

Graficamente, pode-se interpretar o conjunto $\{a, b, c, d\}$ como as coordenadas dos quatro vértices do trapézio formado pela função de pertinência. Para o caso especial em que $b = c$ a função trapezoidal se reduz a uma função triangular. Para o caso especial $a = b$ e $c = d$ tem-se um conjunto clássico, como ilustrado Figura 7.

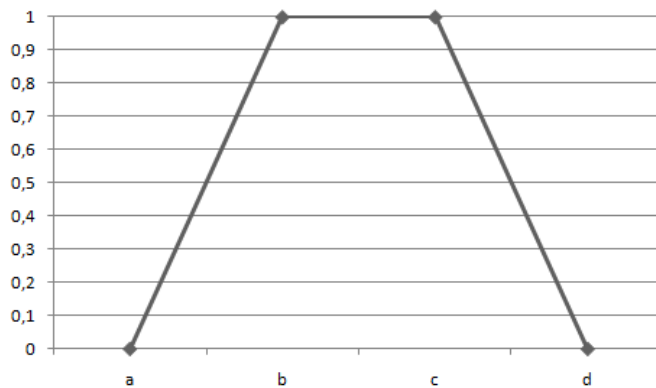


Figura 7 - Representação geral de uma função de pertinência trapezoidal

Função Gaussiana: Uma função de pertinência Gaussiana é especificada por dois parâmetros, a média (ω) e a abertura (σ), conforme definida a seguir:

$$\mu_{Gaussiana}(x;\omega,\sigma) = e^{-\left(\frac{x-\omega}{2\sigma}\right)^2} \quad (3.5)$$

A função Gaussiana é simétrica ao redor de ω e sua abertura é especificada pelo parâmetro σ . Podemos também afirmar que o suporte de $Gaussiana(x,\omega,\sigma)$ é infinito. Na Figura 8 é apresentado um exemplo de representação da função Gaussiana.

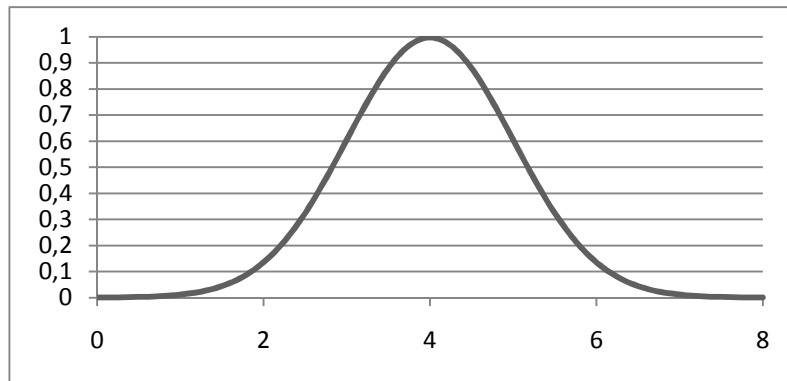


Figura 8 - Representação geral de uma função de pertinência Gaussiana

A especificação de um formato para uma função de pertinência nem sempre é óbvia, podendo inclusive não estar ao alcance do conhecimento de um especialista para a aplicação desejada. No entanto existem sistemas *fuzzy* cujos parâmetros das funções de pertinência são completamente definidos pelo especialista. Nestes casos, a escolha de funções triangulares e trapezoidais é mais comum porque 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. Entretanto, em trabalhos mais recentes existe a tendência ao projeto automático de sistemas *fuzzy* nos quais os parâmetros das funções de pertinência são ajustados no sentido de aperfeiçoar alguma função objetivo. Em geral essa otimização é realizada a partir de um conjunto significativo de dados de treinamento, ou seja, um conjunto de dados de treinamento que descreva adequadamente o comportamento do sistema.

3.1.4 Operações básicas sobre conjuntos *Fuzzy*

Assim como os conjuntos matemáticos convencionais, os conjuntos fuzzy possuem algumas características que os definem. Dentre as propriedades básicas pode-se listar: comutatividade, associatividade, idem potência e distributividade.

Além das propriedades comuns aos conjuntos, há também as operações entre conjuntos. A seguir serão apresentadas três operações básicas: intersecção, união e complemento (KLIR; YUAN, 1995).

Na Figura 9 são apresentados dois conjuntos *fuzzy* que serão usados para geração de exemplos das operações entre conjuntos.

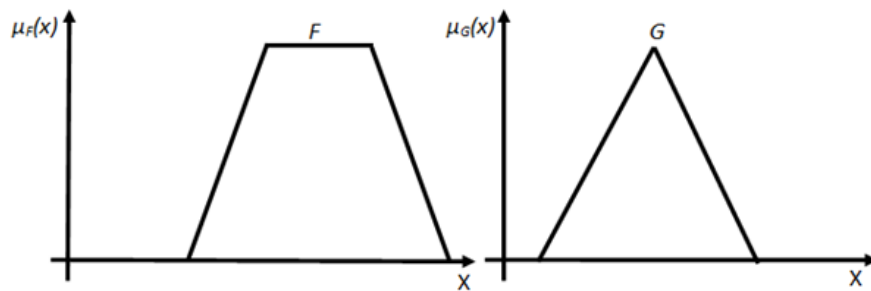


Figura 9 - Gráfico representando dois conjuntos *fuzzy*.

- Intersecção Padrão de Conjuntos *Fuzzy*: Sendo F e G dois conjuntos *Fuzzy* em U , a intersecção de F e G é o conjunto *Fuzzy* E , como ilustrado na Figura 10 e denotado como $E = F \cap G$, sendo para todo $x \in U$:

$$\mu_{F \cap G} = \min(\mu_F(x), \mu_G(x)) \quad (3.6)$$

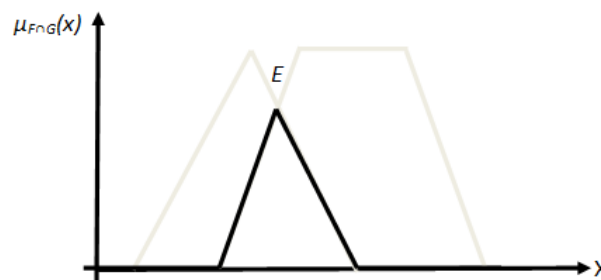


Figura 10 - Gráfico resultante da intersecção, considerando os conjuntos da Figura 9.

- União Padrão de Conjuntos *Fuzzy*: Sendo F e G dois conjuntos *Fuzzy* em U , a união de F e G é um conjunto *Fuzzy* E , como ilustrado na Figura 11 e denotado como $E = F \cup G$, sendo para cada $x \in U$:

$$\mu_{F \cup G} = \max(\mu_F(x), \mu_G(x)) \tag{3.7}$$

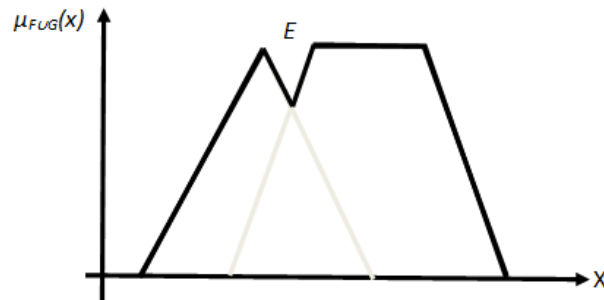


Figura 11 - Gráfico resultante da união, considerando os conjuntos da Figura 9.

- Complemento Padrão de um Conjunto *Fuzzy*: Sendo F um conjunto *Fuzzy* em U . Enquanto $\mu_F(x)$ representa o grau com que x pertence a F , o complemento denotado por $\mu_{\sim F}$, pode representar não só o grau com que x pertence a $\sim F$, senão também, o grau com que x não pertence a F . Um exemplo de complemento padrão é ilustrado na Figura 12.

$$\mu_{\sim F} = 1 - \mu_F(x) \tag{3.8}$$

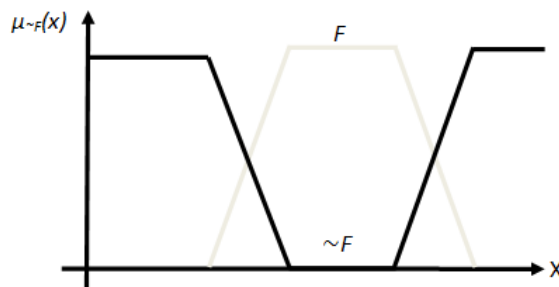


Figura 12 - Gráfico resultante do complemento, considerando o conjunto F da Figura 9

3.1.4.1 Outros operadores

Além dos operadores apresentados nesse capítulo, existem alguns operadores de agregação específicos como os operadores compensatórios propostos por Zimmermann e

Zysno (1980), operadores de soma simétrica de Dubois (1981), operador de média generalizada de Dyckhoff e Pedrycz (1984) e o operador de média ponderada ordenada, proposto por Yager (1988).

Além da agregação, algumas operações com conjuntos *fuzzy* podem resultar em medidas de distância, igualdade, possibilidade, necessidade e compatibilidade (DUBOIS, 1981); (KLEMENT; MESIAR; PAP, 2001); (PEDRYCZ; GOMIDE, 1998).

3.1.5 Operações Generalizadas sobre conjuntos Fuzzy

As operações generalizadas são as operações entre conjuntos Fuzzy (complemento, interseção e união) que assumem formas diferentes das operações padrão. No caso da interseção e união, são operações que utilizam outros operadores em substituição ao mínimo e máximo. Esses operadores pertencem a categorias denominadas genericamente por normas triangulares, as quais garantem que propriedades de operações entre conjuntos serão satisfeitas. Formalmente essas operações podem ser definidas como (PEDRYCZ; GOMIDE, 1998):

- Normas triangulares (t-normas): Utilizada para a intersecção de conjuntos Fuzzy. Uma t-norma t é uma operação binária $t: [0,1]^2 \rightarrow [0,1]$, que satisfaz os seguintes axiomas, para toda $a, b, c \in [0,1]$.

$$t(a, 1) = a \quad (\text{condição limite})$$

$$b \leq c \text{ implica que } t(a, b) \leq t(a, c) \quad (\text{monotonicidade})$$

$$t(a, b) = t(b, a) \quad (\text{comutatividade})$$

$$t(a, t(b, c)) = t(t(a, b), c) \quad (\text{associatividade})$$

Exemplos de t-normas:

- Mínimo: $x \mathop{t} y = \min(x, y)$
- Produto algébrico: $x \mathop{t} y = xy$
- Diferença limitada: $x \mathop{t} y = \max(0, x + y - 1)$
- Co-normas triangulares (s-normas): Utilizada para a união de conjuntos Fuzzy. Uma s-norma s é uma operação binária $s: [0,1]^2 \rightarrow [0,1]$, que satisfaz os seguintes axiomas, para toda $a, b, c \in [0,1]$.

$$s(a, 0) = a \quad (\text{condição limite})$$

$b \leq c$ implica que $s(a, b) \leq s(a, c)$ (monotonicidade)

$s(a, b) = s(b, a)$ (comutatividade)

$s(a, s(b, c)) = s(s(a, b), c)$ (associatividade)

Exemplos de s-normas:

- Máximo: $x \text{ s } y = \max(x, y)$
- Soma algébrica: $x \text{ s } y = x + y - xy$
- Soma limitada: $x \text{ s } y = \min(1, x + y)$

3.1.6 Computação usando regras *fuzzy*

As regras *fuzzy* são ferramentas eficientes na modelagem de sentenças em linguagem natural (JANG; SUN; MIZUTANI, 1997). Este tipo de modelagem utiliza o conceito de variáveis linguísticas para gerar regras *fuzzy* que são mapeadas através 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.

3.1.7 Variáveis Linguísticas

Variáveis linguísticas são variáveis cujos valores são palavras ou sentenças em linguagem natural em vez de números (ZIMMERMANN, 1991). Elas são definidas sobre um determinado domínio, o qual é granularizado em termos linguísticos definidos por conjuntos *fuzzy*. O processo de granularização de um domínio de uma variável em conjuntos *fuzzy* define a chamada partição *fuzzy*.

Conforme apontado por Zadeh, técnicas convencionais para análise de sistemas são essencialmente inadequadas para o tratamento de sistemas baseados no conhecimento humano, cujo comportamento é influenciado pela percepção, julgamento e emoções. Essa é uma manifestação do princípio da incompatibilidade enunciado por Zadeh: "Com o aumento da complexidade do sistema, nossa habilidade de realizar indicações precisas e significativas sobre seu funcionamento diminui até que é alcançado um nível em que precisão e significado se tornam características quase mutuamente exclusivas" (ZADEH, 1973). Isso motivou Zadeh a propor o conceito de variáveis linguísticas como uma

alternativa na modelagem do pensamento humano em que a informação é processada através de conjuntos *fuzzy*.

Não existe uma metodologia consistente para a determinação da partição *fuzzy* ideal. Em geral essa tarefa é realizada manualmente através da intervenção de um especialista ou utilizando um método de particionamento automático a partir de dados de treinamento. Na ausência de um especialista ou de abordagens de ajuste automático, partições utilizando de 5 a 7 termos linguísticos uniformemente distribuídos são utilizadas com frequência (PEDRYCZ; GOMIDE, 1998).

3.1.8 Regras *Fuzzy*

O uso de regras como mecanismo formal para representação do conhecimento remonta às primeiras décadas da criação da área de Inteligência Artificial quando as regras de produção foram propostas como um par condição/ação que define uma porção de conhecimento para solução de um problema (LUGER, 2004).

As regras são uma das técnicas mais antigas para representação de conhecimento e permanecem sendo largamente utilizadas. Sua popularidade vem do fato de possibilitar a expressão clara de diretivas e estratégias, captar o conhecimento experimental de especialistas humanos e possuir um formato linguístico de fácil compreensão. As regras são uma maneira conveniente e popular para expressar o conhecimento, propiciando transparência e compreensibilidade ao sistema. As regras *fuzzy* possuem o seguinte formato:

SE *antecedente* ENTÃO *consequente*

Este formato tem a finalidade de estabelecer relações entre as variáveis que aparecem no antecedente, também chamado de condição ou premissa, e as que aparecem no consequente, também chamado de conclusão ou ação.

Por exemplo, a regra: SE a *velocidade* é *alta* ENTÃO o *risco_de_acidente* é *grande*, estabelece uma relação entre a variável *velocidade* e *risco_de_acidente* usando os termos *alto* e *grande*, que são inerentemente imprecisos.

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.

Por exemplo, se X for a variável *velocidade* e A o termo *alta*, a proposição X é A representa formalmente a parcela de informação do exemplo anterior que diz “a velocidade é alta”.

As proposições *fuzzy* também podem ser compostas, isto é, construídas a partir de proposições simples pelos operadores de conjunção ou disjunção, resultando em proposições com os formatos.

$$X1 \text{ é } A \text{ E } X2 \text{ é } B \text{ OU } X1 \text{ é } C \text{ OU } X2 \text{ é } D$$

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:

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

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.

Em uma regra *fuzzy*, tanto o antecedente quanto o conseqüente podem ser formados por proposições compostas, o que dá origem, por exemplo, a regras da forma:

$$\text{SE } X1 \text{ é } A1 \text{ E } X2 \text{ é } A2 \text{ E } X3 \text{ é } A3 \text{ ENTÃO } Y1 \text{ é } B1 \text{ OU } Y2 \text{ é } B2$$

Onde $A1$, $A2$, $A3$ e $B1$, $B2$ são conjuntos *fuzzy* nos universos $X1$, $X2$, $X3$, $Y1$, $Y2$, respectivamente, e $X1$, $X2$, $X3$ e $Y1$, $Y2$ são variáveis sobre os domínios $X1$, $X2$, $X3$, $Y1$, $Y2$, respectivamente. Assim, a parte antecedente de uma regra é sempre uma proposição *fuzzy* simples ou composta. A parte conseqüente também pode ser uma proposição *fuzzy* ou uma função, dependendo do modelo de regra utilizado.

3.1.9 Inferência *Fuzzy*

O verbo inferir significa evidenciar, deduzir ou obter uma consequência lógica. Sendo assim, a inferência *fuzzy* busca tirar conclusões a partir de um conjunto de regras ou mesmo de uma única regra. Logo, a computação com regras *fuzzy* envolve a aplicação de um método de raciocínio, o qual está baseado em regras de inferência. A regra de inferência composicional é a regra de inferência básica para os processos de raciocínio nos sistemas *fuzzy*. A regra de inferência composicional, também chamada de regra da composição, será apresentada para o caso simples, isto é, quando a regra tem apenas uma variável no antecedente e uma no consequente.

O processo de inferência é aplicado a partir de uma regra, a qual estabelece um conhecimento sobre algum domínio, e de um fato, o qual estipula uma informação específica sobre a situação da qual se deseja obter alguma conclusão.

Seja A um conjunto *fuzzy* sobre X , e B um conjunto *fuzzy* sobre Y . A regra SE X é A então Y é B induz uma relação *fuzzy* $R(x, y)$ sobre X e Y . A definição da relação R consiste na definição da semântica da regra. Essa relação é definida pela função $R(x, y) = f(A(x), B(y))$ sendo que f pode ser de uma das três classes de funções: conjunção *fuzzy*, disjunção *fuzzy* e implicação *fuzzy* (detalhes em (PEDRYCZ; GOMIDE, 1998)).

Para usar essa regra em uma inferência, é necessário conhecer algum fato que estabeleça um valor específico para a variável do antecedente X . Dado o fato X é A' , pode-se inferir a conclusão Y é B' , sendo que B' é definido pela equação 3.9 (PEDRYCZ; GOMIDE, 1998), para todo $y \in Y$, onde \sup denota o operador de supremo e t uma t-norma.

$$B'(y) = \sup_{x \in X} [A'(x)tR(x, y)] \quad (3.9)$$

A aplicação da regra de inferência composicional pode ser facilmente estendida para regras com múltiplas variáveis, e também para situações em que o conhecimento do problema é descrito por um conjunto de regras. O processo geral de computação usando regras *fuzzy* é baseado na regra de inferência composicional.

Alternativamente a este, pode ser aplicado o chamado método de inferência escalonada, que envolve um número pequeno de cálculos numéricos com relação ao método geral. Esses dois processos gerais de computação com regras estão descritos em (PEDRYCZ; GOMIDE, 1998).

3.1.10 Sistemas *Fuzzy*

Sistemas *fuzzy* utilizam a lógica *fuzzy* em seu processo de raciocínio e as variáveis do problema são representadas por valores linguísticos definidos por conjuntos *fuzzy* (KLIR; YUAN, 1995).

O tipo específico de Sistema *Fuzzy* (SF) de interesse neste trabalho são os Sistemas *Fuzzy* Baseados em Regras (SFBR), o qual é composto por dois componentes principais: a Base de Conhecimento (BC) e o Mecanismo de Inferência (MI).

A Base de Conhecimento é composta pela Base de Dados (BD), a qual contém as definições dos conjuntos *fuzzy* relacionados aos termos linguísticos usados nas regras *fuzzy* e pela Base de Regras (BR), que armazena o conjunto de regras que modelam um determinado problema.

O Mecanismo de Inferência é responsável pelo processamento das regras, o qual é realizado por algum método de raciocínio. Este consiste da aplicação de um procedimento de inferência para derivar conclusões a partir das regras e de fatos conhecidos, como os procedimentos apresentados na seção anterior.

O tipo de consequente usado nas regras distingue dois tipos de sistemas *fuzzy*: Mamdani (1977), Takagi e Sugeno (1985). O primeiro possui proposições *fuzzy* no antecedente e no consequente da regra. Já os sistemas do tipo Takagi-Sugeno possuem proposições *fuzzy* no antecedente da regra e o consequente da regra é uma função f aplicada sobre os valores de entrada.

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. No entanto, os sistemas do tipo Takagi-Sugeno já possuem uma saída numérica exata como resultado da aplicação de uma função sobre os valores de entrada, não sendo necessária, portanto, a conversão de *fuzzy* para escalar.

O sistema Mamdani, que é o tipo que será usado no trabalho, é composto por alguns mecanismos principais apresentados na Figura 13.

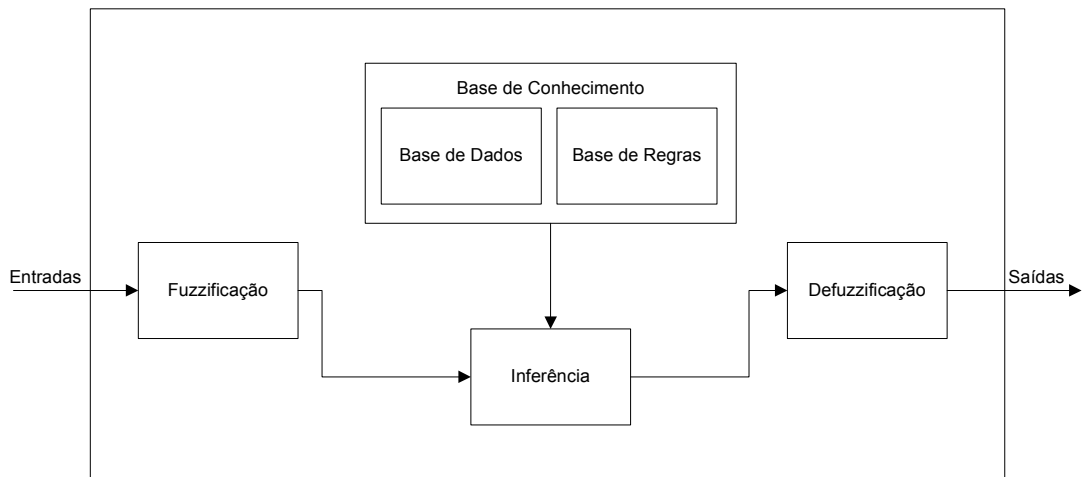


Figura 13 - Representação do sistema fuzzy Mamdani

Base de Regras: contém um conjunto de regras/proposições fuzzy onde as variáveis antecedentes/consequentes são variáveis linguísticas. Na Figura 14 é apresentada uma base de regras criada no software Matlab.

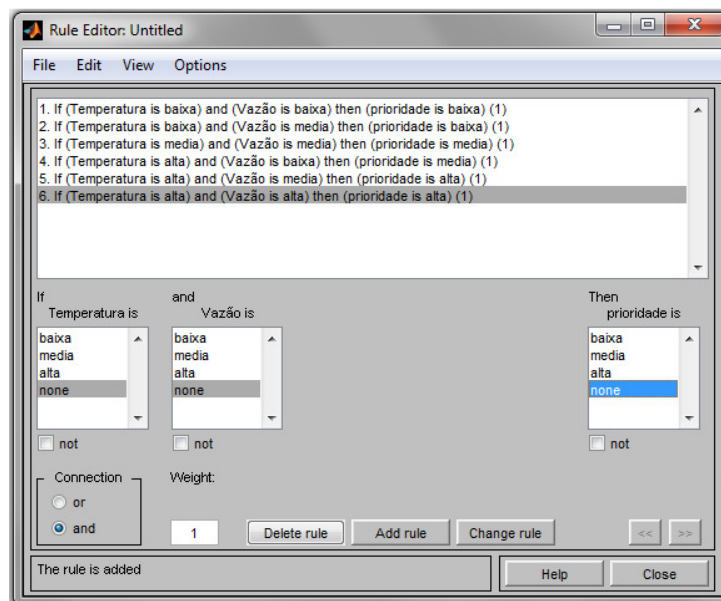


Figura 14 - Exemplo de Base de Regras criada em Matlab

Base de Dados: Define as funções de pertinência do conjunto *fuzzy* nas regras *fuzzy*. Na Figura 15 é ilustrada uma base de dados de uma variável de entrada chamada *temperatura*.

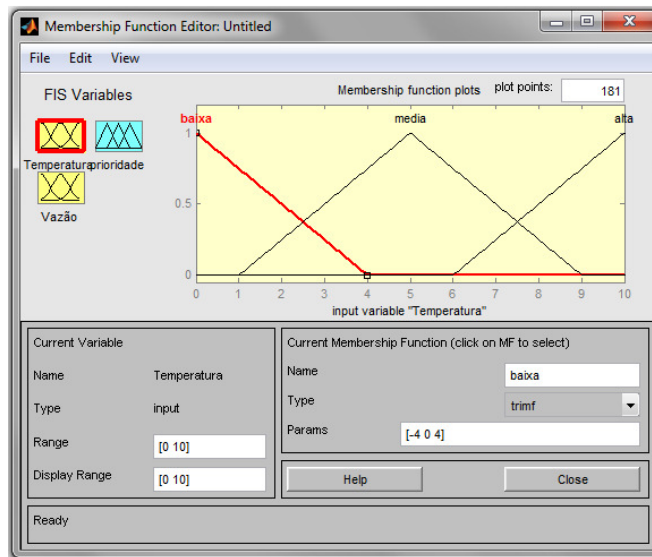


Figura 15 - Exemplo de base de dados criada em Matlab

Inferência: realiza operações de inferência, para obter, a partir da avaliação dos níveis de compatibilidade das entradas com as condições impostas pela base de regras, uma ação a ser realizada pelo sistema.

Fuzzyficação (Conversão de escalar para *fuzzy*): utilizando as funções de pertinência pré-estabelecidas, mapeia cada variável de entrada do sistema em graus de pertinência de algum conjunto *fuzzy* que representa a variável em questão. Na Figura 16 é apresentado um exemplo do funcionamento da *fuzzyficação*.

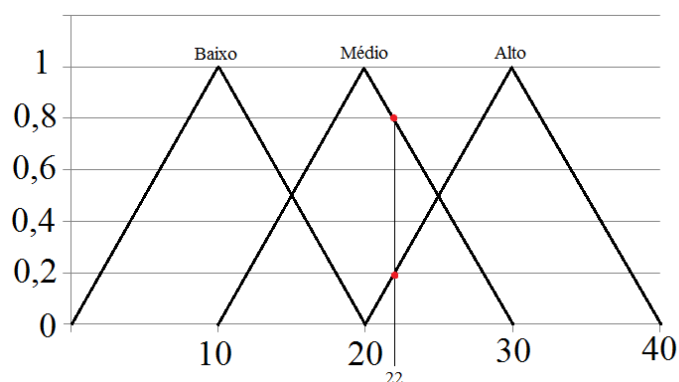


Figura 16 - Exemplo de *Fuzzyficação*

Defuzzyficação (Conversão de *fuzzy* para escalar): transforma os resultados *fuzzy* da inferência em valores de saída. A *defuzzyficação* centroide, que será usada nesse trabalho,

é o valor no universo que corresponde ao centro de gravidade do conjunto *fuzzy* de saída. Na Figura 17 é apresentado um exemplo da *defuzzificação* centroide.

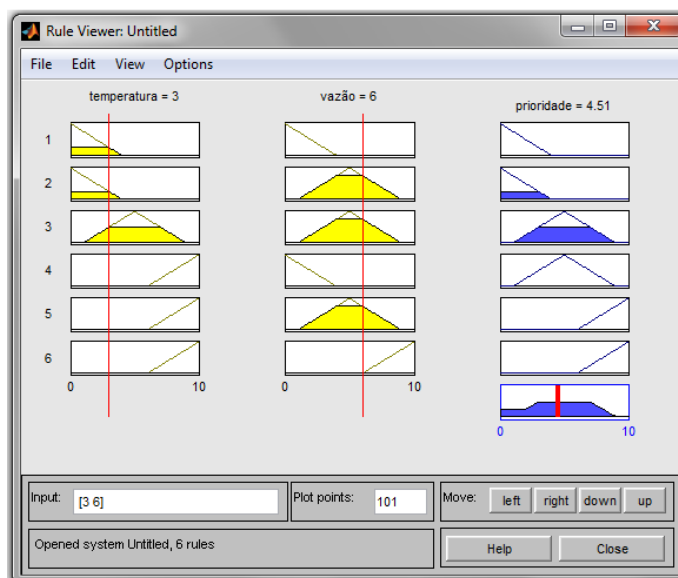


Figura 17 - Exemplo de *defuzzificação* centroide criada em Matlab

3.1.11 Sistemas fuzzy no Matlab

Incorporado ao software Matlab existe um toolbox chamado “*Fuzzy Logical Toolbox*”. Nesse toolbox estão disponibilizados arquivos e funções destinados ao uso da teoria de sistemas *fuzzy*. O Matlab permite a análise gráfica dos resultados e possui um bom ambiente de simulação de sistemas de controle. Outra vantagem de usar o matlab ao invés de refazer toda a programação é que os resultados gerados tem maior confiança, pois toda matemática já está testada e validada.

Como o sistema *fuzzy* proposto nesse trabalho foi implementado usando a toolbox do Matlab, e muitos dos exemplos e resultados são apresentados em *print screen* de telas do software, a seguir é explicado rapidamente algumas das telas principais, para que se possam compreender melhor alguns pontos do trabalho.

Na Figura 18 é apresentada a tela inicial do toolbox fuzzy. Nessa tela são incluídas as variáveis de entrada e de saída, é escolhido o método de inferência e de *defuzzificação* que serão usados.

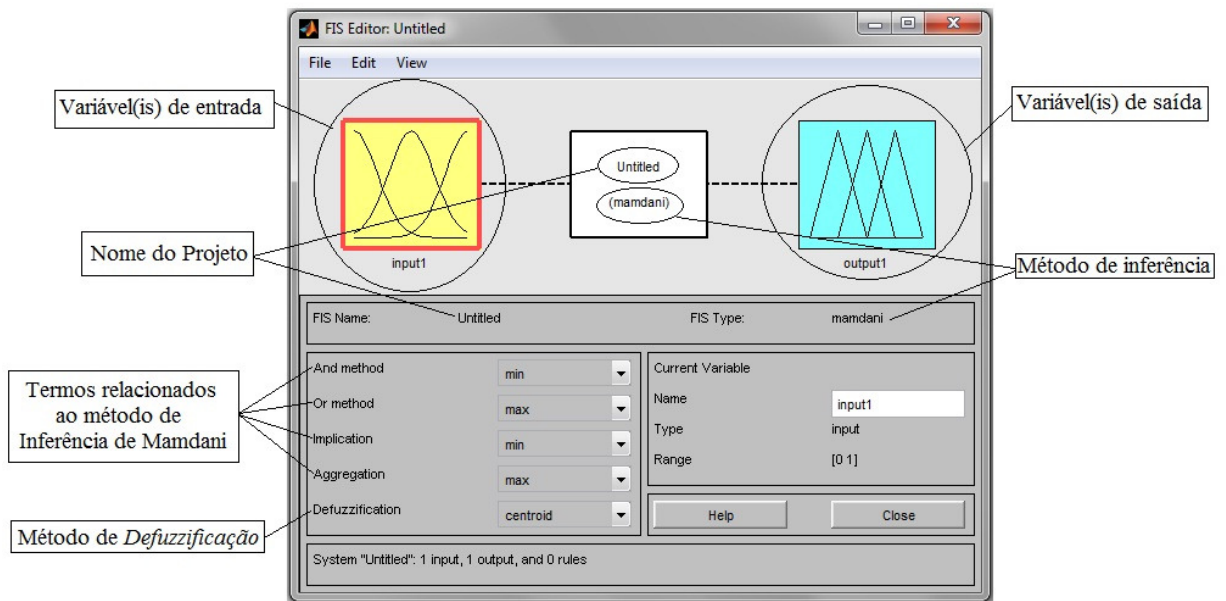


Figura 18 - Tela inicial do toolbox fuzzy

Dando um duplo clique em qualquer variável na tela inicial, o software abrirá uma nova janela, conforme apresentado na Figura 19, que servirá para configuração da base de dados.

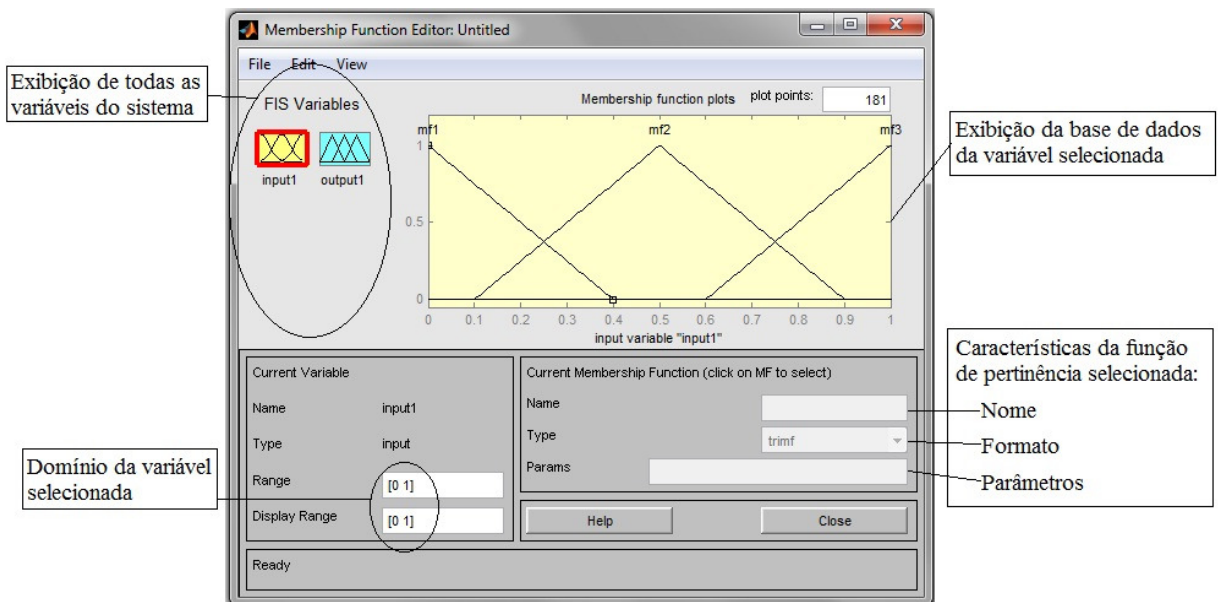


Figura 19 - Tela de configurações da base de dados

Na tela inicial clicando em *Edit/Rules*, o software abrirá uma nova janela, conforme apresentado na Figura 20, que servirá para configuração da base de regras.

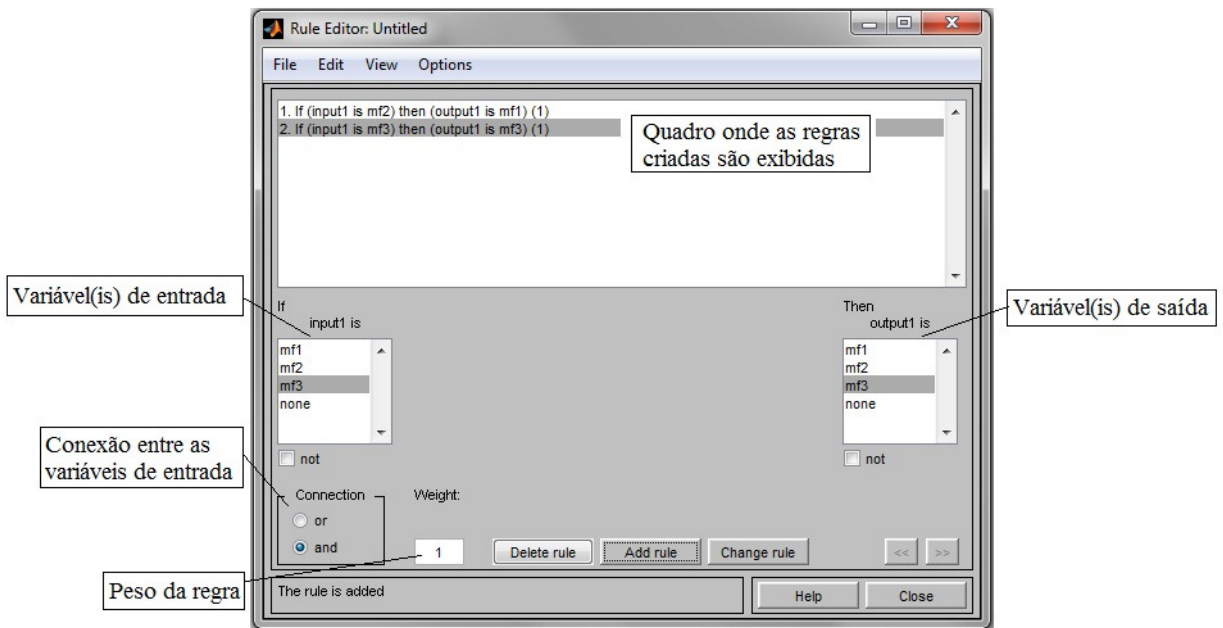


Figura 20 - Tela de configurações da base de regras

Na Figura 21 é apresentada a tela em que são mostrados os valores de entrada, onde esses valores de entrada cortam as funções de pertinência. Também é apresentado a função de saída e o valor *defuzzificado*.

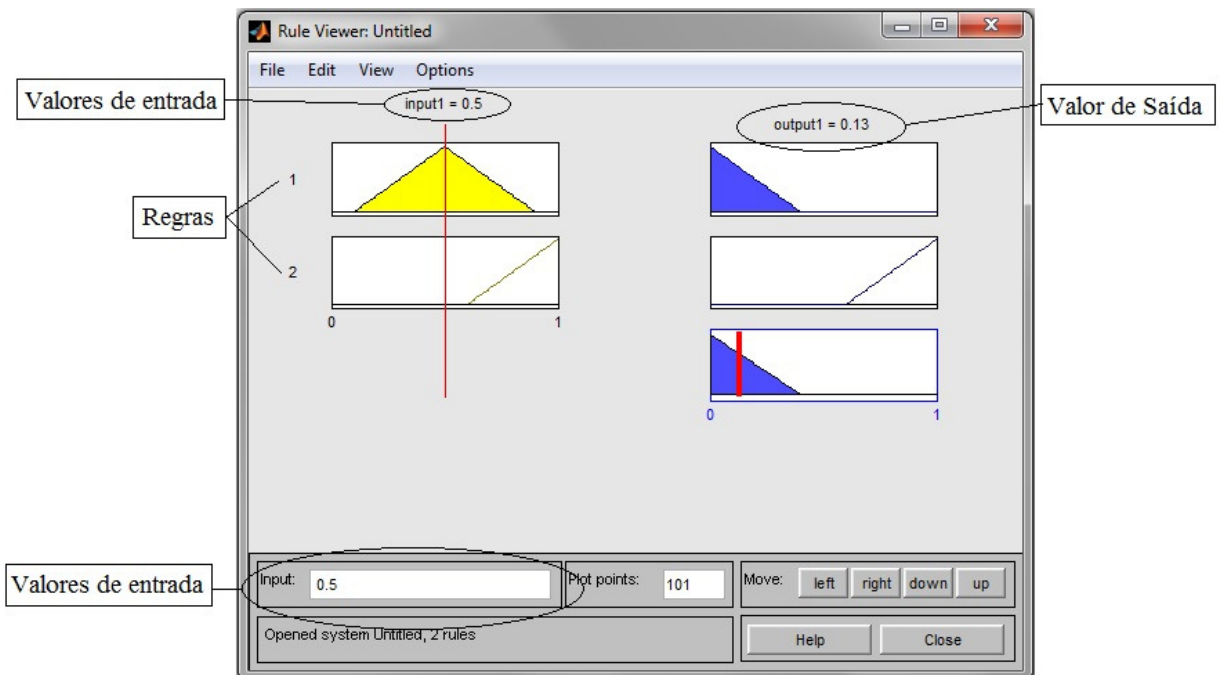


Figura 21 - Tela dos valores resultantes

3.2 Algoritmos Genéticos

Os Algoritmos Genéticos (AGs) são algoritmos de busca e otimização, que utilizam regras baseadas em uma metáfora do processo evolutivo proposto por Charles Darwin, operando sobre um espaço de soluções codificado (GOLDBERG, 1989); (HOLLAND, 1992). Os AGs simulam o mecanismo evolucionário dos sistemas biológicos naturais, onde os indivíduos mais aptos têm maior probabilidade de se reproduzir gerando descendentes.

Os AGs foram criados na década de 60 por John Holland e outros pesquisadores da Universidade de Michigan. Com base nas pesquisas realizadas sobre os AGs, John Holland publicou um livro sobre este assunto, chamado de “*Adaptation in Natural and Artificial Systems*” (HOLLAND, 1975). Após isso, os AGs têm sido analisados por diversos pesquisadores e aplicado para vários tipos de problemas nas mais diversas áreas.

Algoritmos Genéticos partem do pressuposto que, em uma dada população, indivíduos com boas características genéticas têm maiores chances de sobrevivência e de produzirem indivíduos cada vez mais aptos. Como resultado, os indivíduos menos aptos tenderão a desaparecer. Assim, Algoritmos Genéticos favorecem a combinação dos indivíduos mais aptos (CARVALHO; BRAGA; LUDERMIR, 2003).

3.2.1 Estrutura Clássica de um AG

Em um AG clássico, como apresentado por Goldberg (1989), o algoritmo inicia gerando um conjunto de soluções aleatórias codificadas em cadeias de dígitos binários, chamado população. Cada indivíduo da população recebe o nome de cromossomo, representando uma solução candidata para o problema. A avaliação de cada cromossomo determina seu índice de aptidão; indivíduos mais aptos têm maior probabilidade de gerarem filhos. Os cromossomos evoluem através de iterações sucessivas chamadas de gerações.

O processo de seleção e os operadores de cruzamento e mutação são os responsáveis por criar as novas gerações. O processo de seleção visa escolher indivíduos, de acordo com seu índice de aptidão, para combiná-los através do operador de cruzamento, gerando novos indivíduos que mantêm características de seus pais. Posteriormente estes novos indivíduos podem ser modificados pelo operador de mutação. Após várias gerações o AG pode produzir soluções aceitáveis para um problema.

O elemento de ligação entre o AG e o problema a ser resolvido é a função de aptidão. Esta recebe como entrada um cromossomo e retorna um número, ou uma lista de

números, que representam a medida de aptidão do cromossomo com relação ao problema a ser resolvido. O bom desempenho de um AG está condicionado à qualidade da avaliação da aptidão dos indivíduos (OLIVEIRA, 2000).

São apresentadas na Figura 22 as etapas básicas de um algoritmo genético típico.

```
1:  $t = 0$ ;  
2: Gerar População Inicial  $P(0)$ ;  
3: para todo indivíduo  $i$  da população atual  $P(t)$  faça  
4:     Avaliar aptidão do indivíduo  $i$ ;  
5: fim para  
6: enquanto Critério de parada não for satisfeito faça  
7:      $t = t + 1$ ;  
8:     Selecionar população  $P(t)$  a partir de  $P(t - 1)$ ;  
9:     Aplicar operadores de cruzamento sobre  $P(t)$ ;  
10:    Aplicar operadores de mutação sobre  $P(t)$ ;  
11:    Avaliar  $P(t)$ ;  
12: fim enquanto
```

Figura 22 - Algoritmo Genético típico (CARVALHO; BRAGA; LUDERMIR, 2003)

3.2.2 Esquemas de Codificação Empregados nos AG

Os esquemas usados na codificação dos cromossomos variam de problema para problema e de AG para AG. A codificação clássica usada no trabalho de Holland, e até hoje a mais usada, utiliza cadeias de 0 e 1. Com o passar do tempo, outros pesquisadores apresentaram outras formas de codificação.

A codificação clássica, quando utilizada em problemas que possuem variáveis contínuas e cujas soluções requeridas necessitam boa precisão numérica, torna os cromossomos longos. Para cada ponto decimal acrescentado na precisão, é necessário adicionar 3 dígitos (0 ou 1) na cadeia (LACERDA; CARVALHO, 1999).

É demonstrado na Figura 23 um exemplo de população, seus indivíduos e a codificação dos genes de cada indivíduo.

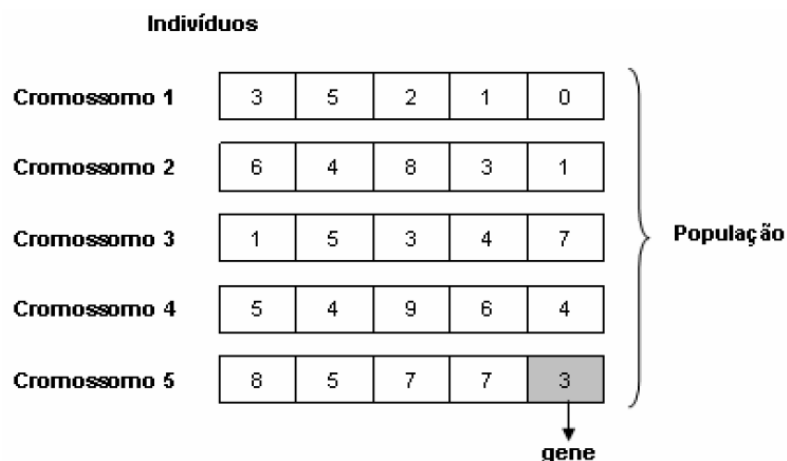


Figura 23 - Exemplo de cromossomo / população.

A consequência imediata do aumento da cadeia, que representa o cromossomo, é o aumento no tempo necessário para calcular o equivalente decimal deste cromossomo. Por este motivo, formas não clássicas de codificação dos cromossomos foram desenvolvidas, gerando codificações adequadas para problemas específicos (HERRERA; LOZANO; MORAGA, 1998).

Uma das formas não clássicas de codificação é a codificação real. Esta forma de codificação consiste em representar variáveis numéricas contínuas através de seu próprio valor real. Um cromossomo pode ser composto por múltiplos genes quando o problema a ser resolvido envolve duas ou mais variáveis. As primeiras aplicações da codificação real foram propostas por (LUCASIUS; KATEMAN, 1989); (DAVIS, 1989). A partir de então a codificação real tornou-se padrão em problemas de otimização numérica com variáveis contínuas.

Castro (1999) afirma que nenhuma forma de codificação funcionaria igualmente bem em todas as situações e que, para cada caso, deve-se fazer uma escolha cuidadosa do tipo de codificação a ser utilizada, pois uma codificação ruim pode não levar ao resultado esperado.

3.2.3 Seleção

A finalidade da seleção é escolher os elementos da população que devem se reproduzir. O Algoritmo Genético começa com uma população inicial de N cromossomos. Quando não existe nenhum conhecimento prévio sobre a região do espaço de busca onde se encontra a solução do problema, os cromossomos são gerados aleatoriamente.

Para que o processo de seleção possa dar prioridade aos indivíduos mais aptos, a cada cromossomo da população é atribuído um valor denominado valor de aptidão, que é calculado pela função de aptidão. Esta função recebe como entrada os valores dos genes do cromossomo e fornece como resultado a sua aptidão. A aptidão pode ser vista como uma nota que mede a qualidade da solução codificada por um indivíduo e é baseada no valor da função-objetivo, que é específica para cada problema.

Após ter associado uma nota ou aptidão a cada indivíduo da população, o processo de seleção escolhe um subconjunto de indivíduos da população atual, gerando uma população intermediária. Os métodos de seleção mais usados são: roleta e torneio, que serão apresentados a seguir.

- **Método da Roleta**

Neste método, os indivíduos de uma geração (ou população) são selecionados para a próxima geração utilizando uma roleta. Cada indivíduo da população é representado na roleta por uma fatia proporcional ao seu índice de aptidão. Assim, indivíduos com aptidão alta ocupam fatias maiores da roleta, enquanto indivíduos de aptidão mais baixa ocupam fatias menores.

Na Figura 24 é apresentada a criação de uma roleta a partir dos valores de aptidão dos indivíduos de uma população. Para a seleção dos indivíduos, a roleta é girada N vezes, onde N é número de indivíduos da população inicial. A cada vez que a roleta parar de girar, o cromossomo selecionado pelo marcador será copiado para a próxima geração. Cromossomos com maior espaço na roleta terão maior chance de serem selecionados. O algoritmo Roleta não funciona com aptidões negativas. Além disso, pode gerar também problemas como convergência prematura (CARVALHO; BRAGA; LUDERMIR, 2003).

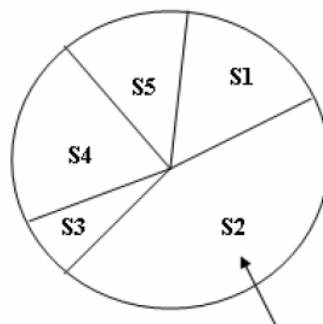


Figura 24 - Método de seleção por roleta

- **Método do Torneio**

Quando este método é utilizado, um número de indivíduos N da população é escolhido de modo aleatório e com mesma probabilidade. O cromossomo com maior aptidão dentre estes n cromossomos é selecionado para a população intermediária. Este processo se repetirá até que a população intermediária seja preenchida. Na Figura 25 é ilustrado o método de seleção por torneio, em que a partir de uma população composta pelos cromossomos $S1$, $S2$, $S3$, $S4$ e $S5$, este método seleciona os cromossomos com maior valor de aptidão.

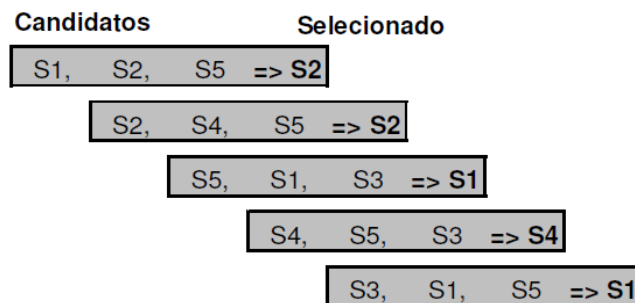


Figura 25 - Método de seleção por torneio

3.2.4 Elitismo

O elitismo é uma técnica utilizada para melhorar a convergência dos AGs. Ele foi primeiramente introduzido por Jong (1975) como uma adição aos métodos de seleção. O elitismo força os AG a reter certo número de “melhores” indivíduos em cada geração. Tais indivíduos podem ser perdidos se não forem selecionados para reprodução ou se forem destruídos por cruzamento ou mutação.

Em outras palavras, o elitismo seleciona os melhores cromossomos de uma população e transporta-os à geração seguinte. Esta técnica consiste basicamente em realizar o processo de seleção em duas etapas:

- Seleciona-se uma elite de R membros entre os melhores da população inicial, os quais são incorporados diretamente na população final;
- O restante da população final é obtida a partir dos $(N - R)$ elementos restantes da população inicial de tamanho N .

Em geral a elite tem um tamanho reduzido, com $R = 1$ ou 2 para um $N = 50$. Quando é utilizada a técnica do elitismo, o algoritmo converge mais rapidamente. Como na natureza, os indivíduos mais aptos podem, além de reproduzirem-se mais, ter uma vida mais longa, muitas vezes sobrevivendo de uma geração para a outra e se reproduzindo. O efeito negativo desta estratégia prende-se ao fato de que a população inicial pode convergir para uma população homogênea de super indivíduos, não explorando outras soluções.

3.2.5 Operadores Genéticos

Nos AGs a seleção e os operadores genéticos são utilizados para criar as novas gerações, fazendo acontecer a evolução da população.

Holland (1992) define três operadores para criar filhos diferentes dos pais: cruzamento, mutação e inversão. O objetivo final destes operadores é fazer com que os cromossomos criados durante o processo de reprodução sejam diferentes dos cromossomos dos pais.

O operador de cruzamento é responsável por combinar os cromossomos dos pais na criação dos cromossomos filhos, conforme ilustrado na Figura 26. O operador de mutação é responsável pela introdução de pequenas mudanças aleatórias nos cromossomos dos filhos, conforme ilustrado na Figura 27. O operador de inversão inverte partes da cadeia cromossômica e seu uso é indicado quando existe ordem entre os elementos da cadeia cromossômica (ICHIHARA, 1998).

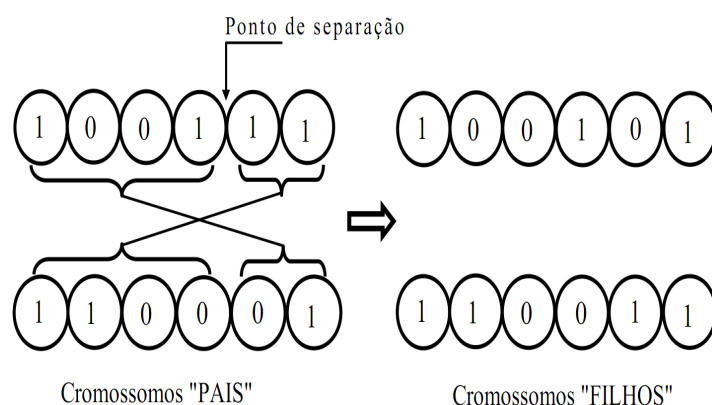


Figura 26 - Exemplo de cruzamento

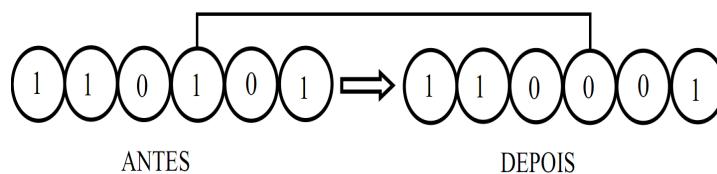


Figura 27 - Exemplo de mutação

Diversos operadores de cruzamento e mutação foram desenvolvidos por pesquisadores, alguns adequados a um tipo específico de codificação dos cromossomos, outros com intenção de serem mais genéricos.

3.2.6 Controle dos Parâmetros Genéticos

O desempenho de um Algoritmo Genético é fortemente influenciado pela definição dos parâmetros a serem utilizados, como: tamanho da população, taxa de cruzamento e taxa de mutação. Segundo Srinivas e Patnaik (1994) a escolha dos parâmetros pode tornar-se um problema complexo de otimização e, além disso, a escolha destes parâmetros está relacionada com a natureza da função objetivo. Portanto, é importante analisar como os parâmetros podem ser utilizados diante das necessidades do problema e dos recursos disponíveis. A seguir serão apresentados alguns critérios para escolha dos principais parâmetros:

Tamanho da População: O tamanho da população afeta o desempenho global e a eficiência dos Algoritmos Genéticos. Em uma população pequena, o Algoritmo Genético é mais rápido, mas seu desempenho pode cair, pois fornece uma pequena cobertura do espaço de busca.

Taxa de Cruzamento: Quanto maior a taxa, mais rapidamente novas estruturas serão introduzidas na população, mas pode perder indivíduos mais aptos. Caso esta taxa seja muito pequena, a busca pode estagnar.

Taxa de Mutação: Uma baixa taxa de mutação previne que a busca fique estagnada em sub-regiões do espaço de busca. Além disso, possibilita que qualquer ponto do espaço de busca seja atingido. A taxa muito alta torna a busca aleatória.

Dada a robustez dos AG, mesmo com uma escolha inadequada dos parâmetros genéticos, eles são capazes de convergir para uma solução otimizada. Os reflexos desta

escolha inadequada manifestam-se, principalmente, no tempo necessário para a convergência (CATARINA; BACH, 2003).

3.2.7 Sistemas *fuzzy* genéticos

Nos últimos anos, vários artigos e aplicações combinando conceitos *fuzzy* e AG surgiram na literatura e existe uma preocupação crescente sobre a integração entre eles. Muitas destas publicações exploram o uso do AG para a geração automática de um Sistema *Fuzzy*. A essas abordagens deu-se o nome de Sistemas *Fuzzy* Genéticos (SFG) (CORDÓN et al., 2004). Os SFG incluem Redes Neurais *Fuzzy* Genéticas, Algoritmos de Agrupamento *Fuzzy* Genéticos e Sistemas *Fuzzy* Genéticos Baseados em Regras.

As propriedades dos algoritmos genéticos os tornam adequados para a otimização de bases de regras *fuzzy*. Essa prática tem produzido resultados promissores (BONARINI; FIORELLATO, 2001); (DELGADO; VON ZUBEN; GOMIDE, 2001); (CORDON; HERRERA, 1997); (COOPER; VIDAL, 1994).

Para a otimização da base de regras é assumido que os parâmetros do modelo são conhecidos, isto é, é suposta a existência de uma base de dados. Normalmente é usada uma base de regras representada na forma de uma tabela de decisão. Uma base de regras consistindo de regras *fuzzy* com n variáveis de entrada e uma variável de saída pode ser representada por uma tabela de decisão *n-dimencional*, onde cada dimensão corresponde a uma variável de entrada (CORDON; HERRERA, 1997).

Cada dimensão tem um *array* associado contendo os valores linguísticos da variável de entrada particular. Uma célula na tabela de decisão contém o valor linguístico que a variável de saída toma para a combinação dos valores linguísticos das variáveis de entrada correspondentes a esta célula. Portanto, cada célula representa uma regra *fuzzy* que pode pertencer à estrutura do modelo final. Essa estrutura é codificada nos indivíduos, cromossomos, formando uma população do algoritmo genético.

3.3 Redes de Petri

As Redes de Petri (PN) foram propostas por Carl Adam Petri (1963), com o objetivo de fornecer uma ferramenta matemática e gráfica para modelagem e que fosse aplicável ao estudo de diferentes tipos de sistemas, em particular para sistemas de eventos discretos.

Desde sua proposta inicial, ela tem sido utilizada para a modelagem de sistemas em diversas áreas, como por exemplo: protocolos de comunicação, algoritmos distribuídos, arquiteturas de computadores, sistemas de auxílio à tomada de decisão, sistemas de manufatura, entre outras (JUNQUEIRA, 2006).

3.3.1 Definições de Rede de Petri

A rede de Petri chamada de lugar/transição é um grafo com dois tipos de nós: lugares (representados por círculos – elemento passivo), e transições (representado por barras – elemento ativo). Arcos orientados conectam os componentes anteriores. Os lugares podem conter um número inteiro de marcas, que são utilizadas para definir o estado do sistema modelado pela rede de Petri (MURATA, 1989). São ilustrados na tabela 1 os elementos de uma rede de Petri lugar/transição.

Tabela 1 - Elementos da Rede de Petri (MURATA, 1989).

Elemento Visual	Descrição
▮	Transição (t)
○	Lugar (p)
→	Arco
•	Marca

A rede de Petri é considerada um multigrafo direcionado bipartido, pois vários arcos orientados podem sair de um nó (lugar ou transição) para os demais, existindo a restrição de que os arcos só interligam elementos de natureza diferente (MURATA, 1989). Na Figura 28 é ilustrado um exemplo de uma rede de Petri.

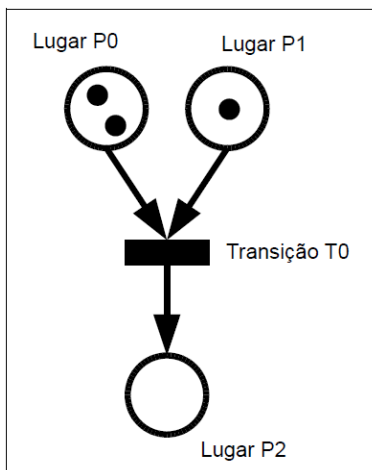


Figura 28 - Exemplo de uma rede de Petri lugar/transição

A rede de Petri lugar/transição tem a seguinte definição formal (MURATA, 1989):

Uma rede de Petri PN é uma 5 – tupla, $PN = \{P, T, F, W, M_0\}$, onde:

$P = \{p_1, p_2, \dots, p_m\}$ é um conjunto finito de lugares,

$T = \{t_1, t_2, \dots, t_m\}$ é um conjunto finito de transições,

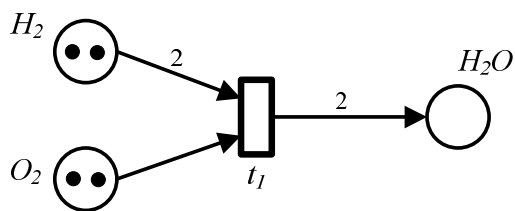
$F \subseteq (P \times T) \cup (T \times P)$ é um conjunto de arcos,

$W: F \rightarrow \{1,2,3, \dots, \}$ é uma função peso de arcos,

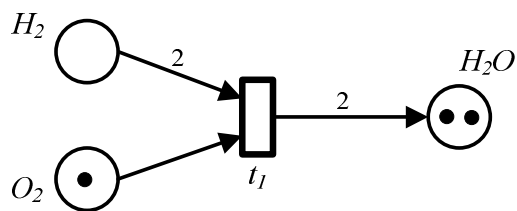
$M_0: P \rightarrow \{1,2,3, \dots, \}$ é a marcação inicial,

$P \cap T = \emptyset$ e $T \cap P = \emptyset$

Pelo exemplo da Figura 29 ainda é possível ilustrar outras duas propriedades das Redes de Petri que são os conceitos de transições habilitadas e disparo de transições. Esse modelo utiliza a conhecida composição química de dois átomos de hidrogênio e um de oxigênio que formam uma molécula de água.



(a) Transição habilitada



(b) Disparo da Transição t1

Figura 29 - Exemplos de Redes de Petri (MURATA, 1989)

No modelo, apresentado na Figura 29(a), os dois lugares que antecedem $t1$ ($H2$ e $O2$) representam duas moléculas de hidrogênio e duas de oxigênio por possuírem duas marcas cada um. Uma vez que as marcas que antecedem a transição $t1$ estão disponíveis, esta transição é considerada habilitada. Após o disparo da transição $t1$, a marcação será modificada, como mostrado na Figura 29(b), deixando a transição $t1$ desabilitada.

Como mencionado, por meio das marcas é possível observar um determinado estado do sistema modelado e de maneira a simular seu comportamento dinâmico, as marcações podem ser modificadas de acordo com as seguintes regras de disparo (MURATA, 1989):

- Uma transição t é dita habilitada se cada lugar de entrada p de t é marcado com pelo menos $w(p, t)$ marcas, onde $w(p, t)$ é o peso do arco direcionado de p a t ;
- Uma transição habilitada pode ou não disparar. (dependendo, dessa forma, de quando o evento ocorreu);
- O disparo de uma transição t retira $w(p, t)$ marcas de um lugar de entrada p de t e adiciona $w(t, p)$ marcas a cada lugar de saída p de t , onde $w(t, p)$ é peso do arco direcionado de t a p . Como por exemplo, na Figura 29(b) o disparo de $t1$ retirou duas marcas de $H2$ e uma marca de $O2$ e adicionou duas marcas em $H2O$.

3.3.2 Propriedades em Redes de Petri

Além de apresentar um grande potencial para a modelagem abrangente de diversos tipos de sistemas, as PNs fornecem dois tipos de propriedades como forma de avaliar e verificar o modelo. São as propriedades Comportamentais e Estruturais.

Propriedades comportamentais são aquelas que dependem da marcação inicial e são descritas a seguir baseando-se em Murata (1989).

- **Alcançabilidade:** esta propriedade permite averiguar se uma marcação M_i é alcançável a partir da marcação M_0 quando existir uma sequência de disparos de transições que transforma M_0 em M_i ;
- **Limitação (Boundedness):** esta propriedade indica o número máximo de marcas $M(P)$, existentes em qualquer lugar P para uma marcação alcançável a partir de M_0 . Um PN é dita k -limitada (ou simplesmente limitada) se $M(P)$ não exceder o valor k de M_i para todo $M_i \in R(M_0)$, sendo k um inteiro positivo;

- **Segurança (Safe):** uma PN é dita segura se ela for 1-limitada. Isto é o número de marcas para todos os lugares não excede 1;
- **Conservação:** a propriedade de conservação está relacionada às marcas do modelo em que durante a sua execução permanecem constantes. Dessa forma, para qualquer marcação alcançável, a soma das fichas de cada lugar para toda marcação é invariável;
- **Vivacidade (Liveness):** a propriedade de vivacidade está relacionada com a ausência de bloqueios em um sistema. A análise parte da avaliação das transições, onde uma transição $t \in T$ é dita viva se para qualquer marcação $M_i \in R(M_0)$ existir, a partir de M_i , uma sequência de disparos de transições σ contendo t ;
- **Reversibilidade (Reversibility) e Home State:** Se dada uma PN, M_0 puder ser alcançável por cada marcação $M_i \in R(M_0)$, então essa PN é dita reversível. Isso indica que em uma rede reversível para todas suas marcações é possível retornar a marcação inicial, ou ainda atribuir uma marcação M_j como sendo um estado intermediário base (home state) uma vez que a partir de M_j é possível alcançar M_0 ;
- **Cobertura (Coverability):** uma determinada marca $M_i \in R(M_0)$ em PN é dita coberta se existe uma marca $M' \in R(M_0)$ a qual $M'(P) \geq M(P)$ para cada lugar P da PN. Isso indica a possibilidade de averiguar se uma determinada marcação pode ser obtida por meio de outra;
- **Persistência (Persistence):** a propriedade de persistência determina se uma PN está livre de bloqueios. Dessa forma, se para duas transições habilitadas, o disparo de uma não desabilita o disparo da outra, a PN é persistente.

As propriedades estruturais estão relacionadas à topologia da rede possibilitando uma análise sem considerar a marcação inicial. Seguem algumas dessas propriedades baseando-se em Murata (1989).

- **Vivacidade Estrutural:** uma PN é estruturalmente viva se existe uma marcação inicial (M_0) viva;
- **Limitação Estrutural:** uma PN é estruturalmente limitada se toda marcação inicial (M_0) for limitada;
- **Conservação Estrutural:** uma PN é estruturalmente conservativa se for conservativa para qualquer marcação inicial;

- **Consistência:** uma PN é parcialmente consistente se existir uma seqüência de disparos de transições σ , a partir de MO sendo possível retornar a MO de forma que todas as transições sejam disparadas ao menos uma vez;
- **Repetitiva:** uma PN é repetitiva se houver uma marcação inicial MO e existir uma seqüência de disparos de transições σ que repita todas infinitamente;
- **Controlabilidade:** uma PN é completamente controlável se qualquer marcação for alcançável por qualquer outra marcação.

3.3.3 Métodos de Análises

Os métodos de análises permitem uma melhor investigação das propriedades do modelo de maneira a examinar suas características. Os métodos são divididos em três tipos. O primeiro é baseado na construção de todo espaço de estados que o sistema pode vir a ter. O segundo utiliza equações algébricas simples de maneira a determinar as propriedades da rede. E um terceiro método é baseado na técnica de redução dos elementos gráficos como forma de simplificar o modelo (MURATA, 1989); (AGUIRRE, 2007).

3.3.4 Extensões de Redes de Petri

Buscando tornar o poder de modelagem mais abrangente em relação às características do sistema modelado, novas extensões de Redes de Petri têm sido propostas a partir do conceito fundamental descrito anteriormente. Dentre outras extensões, estão as que incluem temporização, hierarquização, modularidade e estruturação em alto nível. Algumas das extensões são:

- Rede de Petri Colorida;
- Rede de Petri Temporizada;
- Rede de Petri Estocástica;
- Rede de Petri Virtual;
- Rede de Petri *Fuzzy*.

3.3.5 Considerações finais

Podemos portanto concluir que, a facilidade das redes de Petri para representar situações complexas, associadas com a grande flexibilidade que este tipo de rede apresenta, torna-a uma ferramenta muito útil para modelar, analisar, mapear situações de concorrência de recursos e controlar sistemas complexos de manufatura.

Redes de Petri são também muito usadas para o projeto e detecção de conflitos de veículos auto guiados (RAJU; CHETTY, 1993); (ZENG; WANG; JIN, 1991).

Capítulo 4

TRABALHOS RELACIONADOS A AGV

Veículos Auto Guiados (AGVs) são sistemas populares de manuseio de materiais em processos automatizados, sistemas de manufatura flexível e até em movimentação de *containers* em portos marítimos (QIU et al., 2002); (EVERS; KOPPERS, 1996); (YE; HSU; VEE, 2000).

Muitos trabalhos relacionados a AGV vêm sendo propostos. Existem diversos pontos inerentes ao problema que cada vez mais se busca aperfeiçoamento para uma maior flexibilidade, competitividade, qualidade, etc.

Diferentes objetivos podem ser encontrados na literatura, como minimizar os custos totais de transporte do caminho, minimizando distância de viagem ou minimizando o tempo de viagem (SEIFERT; KAY, 1995).

Neste capítulo, inicialmente serão apresentados alguns trabalhos que contemplam problemas no sistema de transporte com AGV e suas dificuldades. Posteriormente são apresentados trabalhos relacionados a regras de despacho de veículos, que é o foco do deste trabalho, incluindo desde as primeiras abordagens propostas até recentes trabalhos.

4.1 Problemas gerais relacionados aos AGVs

Peters et al. (1996) formalizam um sistema de classificação de controle, dividindo os problemas relacionados aos AGVs em três vertentes:

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

Em Le-Anh e De Koster (2006) é apresentada uma revisão sobre o projeto e controle de sistemas de veículos auto guiados. Os autores citam questões-chaves que envolvem os sistemas de veículos auto guiados, como: tipo e quantidade de veículos, roteamento, posição do veículo ocioso, gestão de bateria/combustível, resolução de *deadlock* e programação de veículos.

Nesta seção serão apresentados resumidamente trabalhos que abordam algumas das questões importantes relacionadas aos AGVs apontadas nos trabalhos citados acima.

Uma questão apresentada por Meer e Koster (1999) é a quantidade de veículos, que influencia fortemente no desempenho dos sistemas de AGV. Os AGVs são geralmente caros, e determinar o tipo e a quantidade adequada de veículos é importante. De acordo com Egbelu (1987), existem três principais fatores que afetam a quantidade necessária de veículos: *layout* do caminho, localização dos pontos de transferência de carga, e estratégias de movimentação.

Shen e Kobza (1998) apresentam um algoritmo para encontrar a quantidade mínima de veículos. O algoritmo inicia encontrando um limite superior na quantidade necessária de veículos. O primeiro passo do algoritmo pressupõe que o *deadlock* e o tempo de congestionamento dos veículos são iguais ao dobro do tempo de viagem do veículo carregado. A segunda etapa do algoritmo utiliza um modelo de filas de AGVs para determinar se o critério está satisfeito. Uma cadeia Markov discreta é utilizada para encontrar o tempo de viagem do veículo vazio, que é uma parte do tempo de serviço. Este tempo de serviço é usado em um modelo de enfileiramento para distribuição aproximada do número de solicitações de carga no sistema. O último passo do algoritmo encontra o tempo de espera para cada solicitação de carga no sistema usando um método de aproximação. A utilização de veículos com capacidade de múltiplas cargas pode reduzir a quantidade de veículos necessários ou aumentar a taxa de transferência de um sistema. O veículo multi-carga pode coletar cargas adicionais enquanto transporta uma carga anterior. O uso de

veículos multi-carga pode, portanto, reduzir a quantidade de veículos, reduzir o tempo de viagem do veículo vazio e a distância total percorrida também é susceptível a diminuir (BILGE; TANCHOCO, 1997).

Outro ponto de estudo é o roteamento de veículos. Uma vez que a decisão da programação é realizada, a missão do roteamento é atribuir uma rota adequada, (por exemplo, o caminho mais curto/distância, o caminho mais curto/espço de tempo, etc.), para cada AGV desde sua origem até o destino baseado na atual situação (DANIELS, 1988). A decisão de roteamento envolve duas questões. Primeiro, ele deve detectar se existe um caminho que poderia conduzir um veículo, desde sua origem até o destino. Em segundo lugar, a rota escolhida para o veículo deve ser viável, o planejamento de rotas dinâmicas pode ser feito de duas formas, chamadas de planejamento de rotas completas ou planejamento de rotas incremental. No planejamento de rota completa toda a rota (do seu início ao final) é determinada de uma só vez. No planejamento de rota incremental, a rota é planejada segmento por segmento até o veículo chegar a seu destino (TAGHABONIDUTTA; TANCHOCO, 1995).

Em seu trabalho, Khanmohammadi et al. (2010), propõem um método de planejamento de caminho para veículos auto guiados (AGVs). A proposta utiliza 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 em um mínimo local. O sistema *fuzzy* proposto é baseado 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.

A posição dos veículos ociosos, também é apontada com um ponto a ser tratado em sistemas de transportes. A ociosidade do veículo é inevitável em sistemas de veículos auto guiado. Ao invés de fazer com que os veículos retornem ao depósito de veículos, é melhor 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 de marcha lenta através da rede.

Egbelu (1993) usou conversão de *layout* circular para procurar a melhor posição para estacionar veículos de marcha lenta. Ele propõe modelos e métodos de solução para um único veículo em um *loop* unidirecional e bidirecional, e para vários veículos em um *loop*

unidirecional e bidirecional. O objetivo do modelo é minimizar o tempo de resposta dos veículos ociosos. Segundo o autor o local ideal para posicionar um veículo em um *loop* unidirecional coincide com a localização de uma estação de trabalho e o local ideal para posicionar um veículo em um *loop* bidirecional situa-se no ponto médio de um arco. McHaney (1995) apresentou políticas para gestão de bateria/combustível para sistemas de transportes. Segundo o autor uma política eficiente é aproveitar o tempo ocioso para o reabastecimento.

Outro tema abordado na literatura é a resolução de *deadlock*. Os autores Yamamoto, Yamada e Kato (2009) consideram dois tipos de AGV. AGV modesto, que em um cruzamento sempre cede passagem, e o AGV agressivo, que tem comportamento oposto ao AGV modesto, sempre avança no cruzamento. Nesse contexto o *deadlock* se dá quando 2 ou mais AGVs do mesmo tipo se encontram em um cruzamento. Para solução do problema, os autores apresentaram um modelo descentralizado simples baseado em duas ações, "não ir para regiões na qual existam AGV do mesmo tipo" e "vá para o destino". Foram realizadas três simulações em um Sistema Flexível de Manufatura Autônomo Descentralizado (AD-FMS) alternando o número de AGVs em 3, 4 e 5, a quantidade de tipos de produto em 3, 6 e 9, o número de máquinas em 9, 18 e 24. De acordo com os resultados das simulações, não houve nenhum tipo de colisão, e aumentou a eficiência do sistema quando comparado com sistema sem conhecimento.

Guan e Dai (2009) propuseram um método de despacho de AGV multi atributo com ajuste dinâmico de peso e sem *deadlock* denominado AWMA. No método são considerados três atributos: distância, *buffer* de entrada e *buffer* de saída. Para evitar *deadlock* foi usada a eficácia do *deadlock avoidance* baseada no conceito de capacidade restante. Os pesos dos atributos são ajustados dinamicamente. A distância de viagem reflete no transporte do carregamento, *buffer* de entrada e saída refletem no processo de carregamento. A prioridade é encontrada multiplicando o valor do atributo pelo seu peso e fazendo a soma entre eles. A avaliação do sistema foi realizada em uma simulação em um Sistema de Manufatura Reconfigurável (RMS) hipotético com oito estações de processos, uma estação de entrada, uma estação de saída e vários AGVs em um caminho unidirecional. A abordagem foi comparada com outras três encontradas na literatura: STTF apresenta simples atributo, considera o menor tempo de viagem, MOQF instancia para variável de *buffer* de saída o maior peso, posteriormente avalia o tempo mais curto de viagem e FWMA utiliza multi atributo com os mesmos atributos do AWMA, porém com pesos fixos. A avaliação mostrou que a proposta apresentou bons resultados em comparação com os outros métodos no tempo de execução das tarefas e no número de *deadlock*.

Para controlar as atividades dos AGVs, devem existir formas para designar tarefas aos AGVs de uma maneira apropriada. Existem duas possíveis arquiteturas de controle: descentralizada e centralizada (LINDGREN, 1985). Em sistemas de controle centralizado, um controlador central mantém o controle de todos os movimentos em relação ao transporte interno. Todas as informações relacionadas aos veículos são armazenadas nas bases de dados do controlador. O controlador atribui tarefas aos veículos de acordo com as regras especificadas. Sistemas descentralizados são caracterizados pela falta de controle global, tipicamente consistem de entidades autônomas que colaboraram entre si, e muitas vezes dependem de um comportamento auto organizado para atingir os requisitos (DE WOLF et al., 2005).

Segundo De Wolf et al. (2005), a experiência tem demonstrado que sistemas centralizados são eficientes quando existe um número pequeno de AGVs. A arquitetura centralizada tem problemas com escalabilidade, pois não pode lidar com muitos AGVs de forma eficiente, além disso, sistemas centralizados não são flexíveis, isto é, não podem lidar com as mudanças frequentes no problema de transporte e precisam ser personalizados e otimizados cada vez que o sistema for implantado em outro armazém.

Usando uma arquitetura descentralizada e métodos de lógica algébrica para despacho de AVGs em Sistemas Flexível de Manufatura, Bocewicz, Wojcik e Banaszak (2008) abordaram o problema de determinação das regras de coordenação local. Segundo os autores a vantagem mais importante da proposta é a possibilidade de prototipagem de sistemas livres de colisão e operações livres de *deadlock*. Os autores também afirmam que os resultados obtidos podem ser também vistos como uma contribuição para sistemas multiagente de controle distribuído, ou seja, para os casos em que os agentes atuam de forma autônoma e tem que cooperar sob restrições de recursos limitados.

A programação dos AGVs é outro ponto importante. O sistema de programação do veículo deve decidir quando, onde e como deve agir um veículo para realizar as tarefas. Se todas as tarefas são conhecidas antes do período de planejamento, o problema de programação pode ser resolvido *off-line*. Contudo, na prática, a informação exata sobre tarefas a serem executadas são geralmente conhecidas em tempo de operação. Isto faz com que a programação *off-line* seja praticamente impossível. Portanto, sistemas de programação *on-line* são necessários para controlar os veículos (LE-ANH; DE KOSTER, 2006).

Homayouni et al. (2009) propuseram um algoritmo heurístico para programação de AGVs. Um algoritmo genético é proposto para otimizar a programação simultânea de AGVs e QCs (Guindastes de Cais). Na primeira fase, um AG é usado para programar um conjunto

pré-determinado de tarefas para as gruas de cais. A função de avaliação é calculada através de um algoritmo de escalonamento heurístico que é usado na segunda etapa do método. No algoritmo de escalonamento, o conjunto de tarefas propostas pelo AG são usadas para atribuir o AGV adequado para as tarefas. O algoritmo de escalonamento encontra o AGV mais próximo para cada uma das tarefas e calcula o tempo total de funcionamento dos QCs e AGVs. O AG proposto foi configurado com a taxa de mutação e cruzamento constantes em 0,5 e 0,6 respectivamente. De acordo com os autores, os resultados mostraram que o número de AGVs influencia muito sobre o desempenho do método AG, mas o aumento em abundância do número de AGVs, não garante a redução de *makespan*. Além disso, o tempo de cálculo do método proposto é baixo devido à heurística de cálculos do algoritmo de escalonamento. Nessa circunstância o método proposto pode ser implementado em aplicações reais, onde o número de tarefas é muito grande.

Daqiang (2010) abordou o problema de programação da produção para escolher o AGV em um FMS, com o objetivo de minimizar o tempo de carregamento das peças. Foi definida uma função para determinar o tempo mínimo de carregamento englobando algumas restrições, como: adicionar penalidade se o carregamento ultrapassar o tempo previsto, garantir que a quantidade total de peso no AGV é menor ou igual à quantidade máxima, garantir que exista somente um AGV em cada caminho e que esse mesmo AGV atenda todos os pontos de carregamento desse caminho. Foi projetado um algoritmo genético com base na análise do problema proposto no qual a população inicial é gerada por método randômico, foi utilizado cruzamento baseado em ordenação e mutação de inversão simples (Etapa 1: Selecione uma sequência de genes que serão gerados pela mutação aleatória, Etapa 2: Seleciona aleatoriamente dois pontos de corte na sequência do gene, e inverta a subsequência entre esses dois pontos de corte), em que foram aplicadas as seguintes taxas: população inicial: 320, cruzamento: 80%, mutação: 10%, Quantidade de iterações: 1000. De acordo com o autor, os resultados apresentados na simulação alcançaram um ganho de 38% além de simplificar o processamento e entrega de material usando AGV em FMS.

4.2 Uso de regras de despacho para atribuição de tarefas dos AGVs

Despacho é uma questão importante para gestão e controle de AGV, conforme afirmado pelos trabalhos (LE-ANH; DE KOSTER, 2006); (VIS, 2006); (QIU et al., 2002) e que se mantém na atualidade devido a quantidade de pesquisas relacionadas encontradas

na literatura, e que algumas serão apresentadas nesse trabalho. Por ser o foco desse trabalho, há um enfoque especial para esse tópico. No início serão apresentados resumidamente trabalhos que deram origem às primeiras regras simples de despacho até chegar ao período que alguns pesquisadores começaram a propor regras ponderando vários atributos. Logo em seguida são apresentados trabalhos atuais com proposta de ponderação de ações futuras da fábrica.

Os primeiros trabalhos relacionados à regra de despacho de veículos foram motivados pela criação de regras simples de despacho que avaliavam um determinado ponto do sistema.

Em seu trabalho, Egbelu e Tanchoco (1984) apresentaram as seguintes regras:

Regras onde a estação de trabalho escolhe um veículo para o trabalho:

- Veículo Aleatório (*Random Vehicle rule – RV*);
- Veículo mais próximo (*Nearest Vehicle rule – NV*);
- Veículo mais distante (*Farthest Vehicle rule – FV*);
- Veículo disponível há mais tempo (*Longest idle vehicle rule – LIV*);
- Veículo menos utilizado (*Least Utilized Vehicle rule - LUV*).

Regras onde o veículo escolhe qual estação de trabalho será atendida:

- Estação de trabalho aleatória (*Random work center rule – RW*);
- Menor tempo de viagem / distância (*Shortest travel time / Distance rule – SST/D*);
- Maior tempo de viagem / distância (*Longest travel time / distance rule – LTT/D*);
- Menor espaço restante no *buffer* de saída (*Minimum Remaining Outgoing Queue Space rule – MROQS*). A regra MROQS é baseada em um índice que é definido pelo 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.
- Primeiro a chegar é o primeiro atendido (*First Come First Served rule – FCFS*). Na regra FCFS, o primeiro pedido de movimento é servido primeiro, mas nenhuma estação de trabalho pode ter mais do que um pedido. Isto assegura que o tempo decorrido entre o requerimento de um pedido de movimento e a satisfação do mesmo será reduzido;
- Tempo de chegada da unidade de carregamento no “chão de fábrica” (*Unit Load Shop Arrival Time rule – ULSAT*). Esta regra é útil se o objetivo é reduzir o tempo de fabricação dos produtos.

Egbelu (1987) apresentou uma regra de despacho, denominada DEMD (*Demand-Driven*), tendo com objetivo principal atender as estações de trabalhos ociosas, com o baixo número de peças em seu *buffer* de entrada e também estações de trabalho bloqueadas pelo excesso de peças em seu *buffer* de saída, reduzindo assim a ociosidade e possíveis bloqueios das estações de trabalho.

Bartholdi 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.

Han e McGinnis (1989) propuseram a regra “Atenda o mais Significativo” (*Most Significant Move – MSM*), a qual calcula o índice de prioridade de cada estação de trabalho necessitando de transporte, considerando os seguintes atributos:

- Tempo estimado até o bloqueio da estação de trabalho;
- Tempo estimado até que a estação de trabalho fique ociosa;
- A relação de carga de trabalho da estação para que haja um engarrafamento;
- O tempo de viagem entre a estação e o transportador disponível.

Taghaboni (1989) propôs a regra MCQL (*Most-Critical-output-Queue-with-Look-ahead*) a qual é uma variação da regra MROQS baseada em um índice crítico e um índice de prioridade. O índice crítico de uma estação de trabalho é a relação entre o espaço restante no *buffer* de saída e a capacidade de armazenamento do *buffer* de saída. O índice de prioridade de uma unidade de carga é a relação do número de operações completadas sobre o número total de operações. Estes dois índices são aplicados em uma tentativa de minimizar o bloqueio de máquina e o inventário.

Com o passar do tempo alguns pesquisadores perceberam que ponderando mais de um aspecto para tomada de decisão poderiam atingir melhores resultados. Klein e Kim (1996) apresentaram um estudo comparativo entre várias regras de despacho de simples e múltiplos atributos. Utilizaram para medida de desempenho os seguintes atributos: tempo de manufatura da peça, tempo de viagem de AGV, tamanho de fila no *buffer* de saída e tempo de espera por transporte. Por meio uma série de estudos e simulações, concluíram que as regras de despacho que utilizaram múltiplos atributos apresentam melhor desempenho em relação às regras que utilizam atributos simples.

Kim et al. (1999) propuseram uma regra de despacho de AGV, tendo como objetivo principal o balanceamento de cargas de trabalho entre estações de trabalho. Propuseram para isso uma equação composta de várias variáveis, envolvendo o estado corrente do “chão de fábrica”. As variáveis utilizadas na equação foram as 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.

Kim e Hwang (1999) propuseram um algoritmo para o despacho de veículos baseado em funções relacionadas a cada um dos atributos envolvidos, sendo os seguintes atributos:

- Número de trabalhos no *buffer* de entrada da estação de trabalho de destino;
- Número de trabalhos no *buffer* de saída da estação de trabalho de origem;
- Tempo de viagem / distância.

Neste trabalho, os pesquisadores mostraram que se utilizassem somente funções para cada atributo poderiam ocorrer certos problemas, como por exemplo, uma estação de trabalho é atendida pelo fato de estar próxima ao AGV, mas por outro lado, analisando-se a origem, o *buffer* de saída poderia estar com um número baixo de trabalhos, ou analisando-se o destino, o *buffer* de entrada poderia estar a ponto de exceder sua capacidade.

Tan e Tang (2000) propuseram uma regra de despacho composta por múltiplos atributos, utilizando 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. Os atributos envolvidos na regra são ligados diretamente as variáveis do “chão de fábrica”, sendo descritos a seguir.

Composição da variável PARTS_IN:

- LT_IN – Tempo antes do *buffer* de entrada da estação de trabalho de destino estar vazio;
- STD_IN – Menor distância de viagem do veículo para a estação de trabalho de origem, e para a estação de trabalho de destino;
- SLT_IN – Menor tempo antes do *buffer* de saída da estação de trabalho de origem estar cheio;

- PC_IN – Número de peças já completas pela estação de trabalho.

Composição da variável PARTS_OUT:

- SLT-OUT – Menor tempo antes do *buffer* de saída da estação de trabalho de origem estar cheio;
- STD_OUT – Menor distância de viagem do veículo para a estação de trabalho de origem, e para a estação de trabalho de destino;
- LT_OUT – Tempo antes do *buffer* de entrada da estação de trabalho de destino estar vazio;
- PC_OUT – Número de peças já completas pela estação de trabalho.

Os principais procedimentos ocorridos para a realização da regra proposta por Tan e Tang (2000) são os seguintes:

- Verificação de todas as estações de trabalho necessitando de serviço de transporte;
- Estado de todas as variáveis envolvidas no processo;
- Fuzzificação das variáveis;
- Algoritmo Genético aplicado ao resultado da fuzzificação;
- Atendimento ao serviço de transporte para a estação de trabalho que tenha obtido melhor nota.

Yamashita (2001) propôs duas regras para despacho de veículos. Na primeira regra, “O veículo mais Próximo em Tempo” (NVT – *The Nearest Vehicle in Time*), os veículos são despachados com o objetivo de minimizar o tempo de espera do primeiro item a entrar no sistema. Na segunda regra, “O Veículo mais Próximo em Distância” (NVD – *The Nearest Vehicle in Distance*), por outro lado, despacha o veículo que esteja na posição mais próxima para o item mais recentemente entrado no sistema. As duas regras foram comparadas com as regras FEFS dos autores Bartholdi e Platzman (1989), no qual as duas regras propostas obtiveram um menor tempo de espera nos pontos de carga e descarga.

Jeong e Randhawa (2001) propuseram uma regra de despacho de AGV do tipo “veículo iniciado” composta por múltiplos atributos, atribuindo pesos para cada atributo através da utilização de uma Rede Neural Artificial. Os atributos neste trabalho foram os seguintes:

- Espaço restante no local de armazenamento de saída da estação de trabalho;
- Distância entre o veículo disponível e a estação de trabalho esperando pelo serviço de transporte;

- Espaço restando no local de armazenamento de entrada da estação de trabalho.

A proposta do trabalho consiste nos seguintes procedimentos:

- Criar um conjunto contendo todas as estações de trabalho a espera de transporte;
- Utilização de um Rede Neural Artificial para obter o peso para cada um dos três atributos envolvidos na regra, utilizando para isso os valores atuais dos locais de armazenamento de entrada e saída das estações de trabalho envolvidas no processo;
- Em seguida, a nota de cada estação de trabalho é calculada baseando-se no valor e no peso de cada um dos atributos;
- Encontra-se a estação de trabalho com maior nota;
- Verifica se a estação de trabalho de destino possui espaço no seu local de armazenamento de entrada, se possuir o serviço de transporte é realizado e o processo é finalizado, caso contrário a estação de trabalho com maior nota é excluída do conjunto e o processo volta ao passo anterior.

Benincasa et al. (2003) apresentaram um modelo para a definição de uma regra para despacho de AGV baseado em sistemas *fuzzy*. Foram consideradas três variáveis de entrada: distância entre o AGV e a estação de trabalho, número de nós entre o AGV e a estação de trabalho e espaço restante no *buffer* de saída da estação de trabalho requisitante. Cada variável foi dividida em 3 conjuntos triangulares *fuzzy*. Foi criada uma base de regras manualmente com todas as combinações possíveis, ou seja, três variáveis com três conjuntos por variável 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 *makespan* em todas as simulações.

A proposta de Kizil, Ozbayrak e Papadopoulou (2006) apresenta uma regra de despacho que mantém o sistema em operação em todos os momentos, ou seja, segundo os autores, deixa o sistema livre de *deadlock*. A regra desenvolvida pelos autores foi: as cargas com o número mínimo de processos (MNP). A lógica para o desenvolvimento da regra MNP é que executando os processos mais rápidos, o acúmulo de cargas à espera de transferência pode ser minimizado. Segundo os autores, após os testes de programação, a regra foi melhor que todas as outras em desempenho de tempo e fluxo.

Umashankar e Karthik (2006) propuseram uma abordagem inteligente de despacho de AGVs com base em múltiplos critérios 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: a utilização 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 através da comparação dos resultados obtidos pelo sistema com as decisões de um especialista da área. 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 utiliza um algoritmo genético (AG) para gerar uma base de regras *fuzzy* apropriada. A validação se deu na comparação com o trabalho citado anteriormente, apresentando melhores resultados.

Em pesquisas relevantes no tema, alguns autores citam que um ponto negativo das regras de despacho é que suas decisões são baseadas em informações presentes, e que se futuras informações dos processos fossem consideradas para a tomada de decisão, poderia trazer grande contribuição (NASO; TURCHIANO, 2005).

Naso e Turchiano (2005) adotaram uma estratégia de decisão multicritério para levar em conta 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 utilizado *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 um estudo de caso, apresentando bons resultados.

Hidehiko (2008) propôs um sistema de classificação que determina uma hipótese de prioridade usando *ranking*, para melhorar a eficiência do “raciocínio para antecipar o futuro” (RAF). Neste sistema, o AGV decide seus movimentos em Autonomia Descentralizada em Sistema Flexível de Manufatura (AD-FMS). O sistema inclui dados da memória das

condições de produção e ações passadas dos AGVs. Usando esses dados da memória, reordena o sistema de hipóteses, dando a maior prioridade na classificação para a hipótese que é mais provável de ser verdade. O sistema foi aplicado a um AD-FMS que foi construído em uma simulação por computador. Segundo os autores, o sistema reduziu o número de substituições de hipóteses de raciocínio até que uma hipótese mais viável foi atingida.

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 utiliza janelas de tempo em um vetor de formulário. Para cada serviço solicitado pelo supervisor, é verificada a viabilidade dos caminhos pré-candidatos. A viabilidade de um caminho particular é avaliada pela inserção de janelas de tempo adequadas e realizando teste das janelas sobrepostas, assim verificando se em um determinado tempo, o caminho não vai ser utilizado por mais de um veículo. O uso de janelas de tempo faz com que o algoritmo seja apto para programação e outros problemas de roteamento.

Os autores Chiba, Arai e Ota (2010), se baseiam na afirmação de que os problemas relacionados ao ambiente de veículos auto guiados, mais especificamente os problemas de modelagem do caminho do AGV e do roteamento de AGVs estão ligados, sendo que a decisão de um impacta no outro. Para apresentar uma solução integrada, levando em consideração a modelagem do caminho e do problema de roteamento dos AGVs, foi apresentada uma abordagem co-evolutiva cooperativa, que considera por meio de um algoritmo genético a evolução dos dois problemas em conjunto. Foram aplicados quatro métodos em uma fábrica automotiva, sendo eles: (I) Cooperativo co-evolutivo, (II) Algoritmo genético canônico (CGA), (III) Apenas a modelagem do caminho e (IV) Apenas o roteamento. O primeiro desempenho medido foi o tempo de resposta sendo que o (IV) obteve o melhor resultado. Porém, quando levado em consideração os parâmetros da função de avaliação, quantidade de veículos e a complexidade da rede criada, o (I) obteve o melhor resultado. Foi concluído que o método apresentado foi efetivo nos testes com dados reais de uma fábrica automotiva.

Capítulo 5

PROPOSTA

De acordo com os temas discutidos no capítulo de revisão bibliográfica, baseando-se nos trabalhos descritos na literatura, algumas considerações tomam parte das principais preocupações no desenvolvimento de uma abordagem eficaz para o problema de despacho de veículos em sistemas de manufatura. Uma delas é a possibilidade de considerar múltiplas variáveis sendo elas de níveis organizacionais diferentes. Outra preocupação é na possibilidade de ter mais de um objetivo final, e que esses podem se permutar em tempo de execução.

A proposta utiliza sistemas fuzzy para ponderar múltiplas variáveis e algoritmo genético para otimizar o sistema fuzzy inicial. O diferencial da proposta está na predição no despacho dos AGVs, na modelagem dos cenários em redes de Petri e na possibilidade de mudança do objetivo final, em que é possível controlar o estoque e controlar seus custos.

A maioria das empresas tem um *mix* de produção de produtos diferentes, chegando muitas vezes à casa das centenas. Dessa forma, torna-se muito difícil, efetuar uma previsão de demanda para cada um desses produtos fabricados (MARTINS; LAUGENI, 2005).

A empresa deve escolher, conforme o seu ambiente de manufatura, demanda e suas características de administração os seus níveis de produção, assim poderão ser obtidas melhorias financeiras da empresa, redução de desperdícios e a maximização dos recursos disponíveis (MARTINS; LAUGENI, 2005).

De acordo com Martins e Alt (2009), podemos classificar os custos de manter estoque em duas categorias:

Diretamente proporcional à quantidade estocada

- Armazenagem: quanto mais estoque, mais área necessária, mais custo de aluguel;
- Manuseio: quanto mais estoque, mais pessoas e equipamentos necessários para manusear os estoques, mais custo de mão-de-obra e de equipamentos;
- Perdas: quanto mais estoque, maiores as chances de perdas, mais custo decorrente de perdas;
- Obsolescência: quanto mais estoque, maiores as chances de materiais tornarem-se obsoletos, mais custos decorrentes de materiais que não mais serão utilizados;
- Furtos e roubos: quanto mais estoques, maiores as chances de materiais serem furtados e/ou roubados, mais custos decorrentes.

Inversamente proporcional à quantidade estocada

- Custo de obtenção: comprando em maior quantidade, maiores as chances de conseguir preços menores;
- Custo de preparação: produzindo-se em grande escala, usa-se os mesmos recursos existentes para produzir uma maior quantidade, reduzindo o custo unitário.

Muitas vezes, setores de uma mesma empresa possuem objetivos diferentes. Normalmente o setor financeiro está preocupado em economizar, assim sempre tentando diminuir os custos de produção. Já o setor comercial, sempre almeja melhores vantagens competitivas, e poder atender o cliente no momento que ele mais precisa é um grande diferencial. O que muitas vezes ocorre é que as empresas miram em um dos objetivos, pois alterações constantes na metodologia de produção se tornam inviáveis.

Pensando nesse problema, o método de despacho proposto nesse trabalho possui dois objetivos finais que podem ser escolhidos para coordenar a produção, que são: objetivo *makespan* e objetivo *tardiness*.

Minimizando o *makespan*, no qual se produz mais em menos tempo, tende-se a manter o estoque em altos níveis, o que significa maior probabilidade de pronto atendimento aos clientes e pode-se dar margem para criação de estoques de segurança que tem a função de proteger o sistema quando a demanda e o tempo de reposição variam ao longo do tempo. É favorável a diminuição dos custos inversamente proporcionais à quantidade

estocada. Neste caso são avaliadas as seguintes variáveis: distância, número de nós, *buffer* de entrada e *buffer* de saída.

- Distância: é a distância que o AGV tem que percorrer até a estação de trabalho que está requerendo transporte somada com a distância entre a estação de trabalho requisitante e o destino final. Considerando essa variável, dando prioridade para os veículos mais próximos se economiza tempo de produção e utilização dos AGVs;
- Número de nós: os nós são os entroncamentos ou cruzamentos existentes no *layout*. Considerando os nós pode-se fazer o controle de *deadlock* dos AGVs e também minimizar o tempo;
- *Buffer* de entrada e *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 recurso e para a produção. Se o *buffer* de saída encher por completo, a estação de trabalho fica sem espaço para alocar seus produtos acabados, causando também uma parada na produção. Controlando os níveis do *buffer* de entrada e do *buffer* de saída pode garantir menos paradas na produção, ganhando tempo. Outro ganho que essas variáveis trazem para o sistema é não deixar 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.

Trabalhando o *tardiness*, no qual se produz de acordo com a demanda, mantém-se o estoque em baixos níveis, o que significa maior chance de reduzir os índices de perdas. É favorável a diminuição dos custos diretamente proporcionais à quantidade estocada. Neste caso são avaliadas as seguintes variáveis: distância, número de nós, encadeamento de tarefas e data de entrega.

- Encadeamento de tarefas: 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 já carrega a peça B para levar para estação de trabalho 5, onde já 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 combustível

gasto e diminuindo o tráfego no *layout* consequentemente diminuindo o número de paradas nos nós;

- Data de entrega: é a data que a empresa se comprometeu para entregar o pedido ao seu cliente. Considerando essa variável, alinha-se a produção com a demanda, cumprem-se os prazos com baixo desperdício.

Devido à natureza dinâmica e incerteza do ambiente, apenas um parâmetro pode não ser o ideal, então o recálculo periódico, utilizando uma pontuação para cada parâmetro atualizado em tempo real pode trazer melhores resultados (BERMAN; SCHECHTMAN; EDAN, 2009), portanto foi adicionada ao projeto a predição, que ocorre em dois momentos.

O primeiro se dá no momento que existe uma solicitação de transporte e precisa ser designada para um AGV. Na maioria dos sistemas somente os AGVs livres entram na disputa pelo transporte, o que pode ocorrer é que um AGV ocupado irá terminar sua tarefa em um curto espaço de tempo, e quando isso ocorrer o mesmo ficará em uma posição mais viável para atender a solicitação pendente. Dessa forma a proposta incorpora na decisão esses possíveis AGVs.

O segundo momento se dá quando existe um ou mais AGVs ociosos. Ao invés dos mesmos ficarem parados até que exista uma solicitação de transporte, se faz uma verificação nos tempos de processamento das estações de trabalho, seleciona as que têm um menor tempo para finalizar a tarefa, faz-se a conta de prioridade e já encaminha o AGV antes mesmo da estação de trabalho fazer a solicitação de transportes.

A modelagem dos cenários em redes de Petri faz com que o experimento fique mais próximo do ambiente de fábrica real. Modelando em redes de Petri o cenário que será aplicado o método, configura-se que cada máquina e cada nó nas vias de transportes são recursos a serem utilizados por um veículo por vez, o que acontece em fábricas reais e muitas vezes desconsiderados nos trabalhos encontrados na literatura, nos quais se pode carregar e descarregar mais de um veículo na mesma máquina no mesmo tempo, ou dois veículos passarem por um nó (cruzamento) no mesmo tempo.

As variáveis que são consideradas em ambos objetivos possuem quantificadores linguísticos, e a ponderação das mesmas para se chegar a uma decisão final se torna um problema complexo. A lógica *fuzzy* vem sendo utilizada com sucesso em diversas abordagens, pois dentro do que já foi dito no capítulo 3, se destacam algumas características como: proporcionar uma estrutura matemática para modelagem de sistemas definidos de maneira imprecisa, supera a rigidez da lógica clássica permitindo graus de pertinência e proporciona núcleos de decisão baseados em regras descritas em linguagem

natural. Assim, quando comparada característica da técnica com as características do problema, pode-se esperar que a lógica *fuzzy* apresente bons resultados no problema de despacho de veículos.

Inicialmente buscou-se na base de conhecimento do laboratório e na literatura variáveis de maior impacto em sistemas de produção, como distância, *buffers*, data de entrega, números de nós e encadeamento de tarefas. Assim tendo definido as variáveis a serem utilizadas é criada a base de dados *fuzzy*, em que cada variável de entrada e de saída são divididas em cinco conjuntos triangulares para uma melhor representação. Todas as variáveis seguem o mesmo padrão de base de dados, porém cada uma com sua configuração de valores de domínio. Tome-se a Figura 30 como exemplo de uma base de dados de uma variável de entrada dividida em cinco conjuntos.

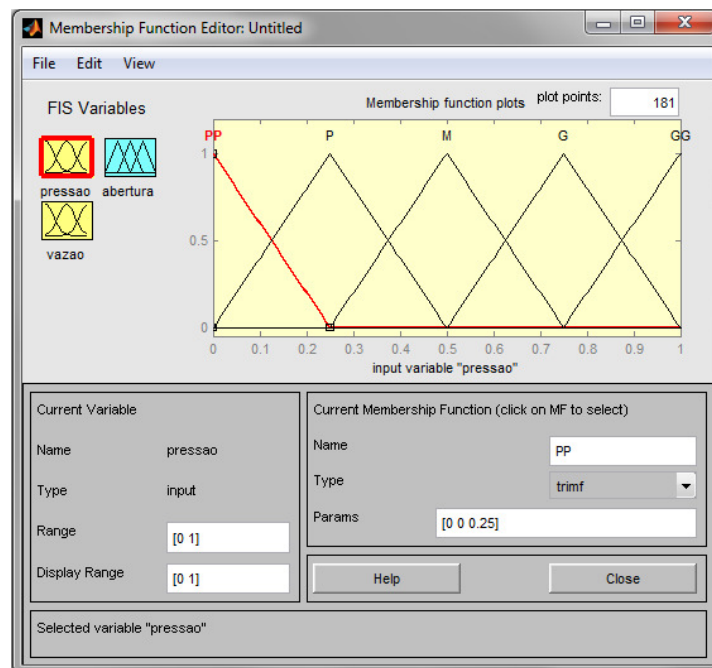


Figura 30 - Exemplo da base de dados de uma variável criada em Matlab

Também é construída a base de regras envolvendo todas as possibilidades de combinações entre os conjuntos das variáveis de entrada e usando a experiência do especialista para determinar a saída. Na Figura 31 é apresentado um exemplo de uma base de regras envolvendo algumas combinações entre duas variáveis de entrada.

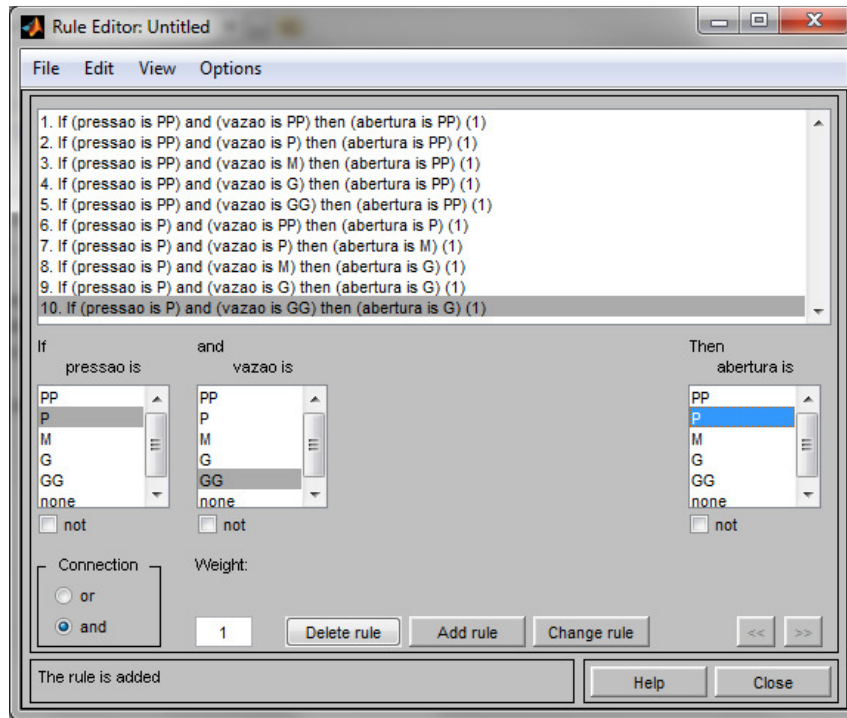


Figura 31 - Exemplo de uma base de regras criada em Matlab

Outro ponto importante é que a lógica *fuzzy* é favorável para a integração com outras técnicas de computação, tais como redes neurais e algoritmos evolucionários (AKBARZADEH-T et al., 2000). Portanto, caso a base de regras construída inicialmente não apresente resultados satisfatórios, pode-se agregar outras técnicas para refinar a base de regras e se chegar a um balanceamento entre cobertura e interpretabilidade. Por sua característica de encontrar boas soluções em pequenos períodos de tempo, e já vir apresentando bons resultados nessa linha, o método proposto usa os algoritmos genéticos para o refinamento automático da base de regra.

As regras são codificadas por números reais que representam o índice dos conjuntos fuzzy que aparecem na parte do antecedente da regra.

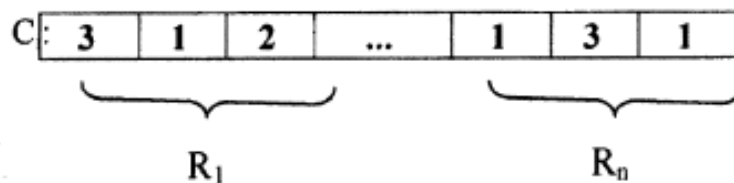


Figura 32 - Exemplo de cromossomo

Por exemplo, considere o cromossomo apresentado na Figura 32. Ele representa a base de regra a seguir, composta por n regras *fuzzy*:

R_1 : SE X_1 é A_3 e X_2 é A_1 ENTÃO B_2

R_n : SE X_1 é A_1 e X_2 é A_3 ENTÃO B_1

Onde:

- R_k , $K=1\dots n$, é o identificador da regra;
- X são variáveis de entrada;
- A_j são os conjuntos *fuzzy*;
- B_r são as variáveis de saída.

Como dito anteriormente, cada objetivo é composto por quatro variáveis de entrada, sendo cada dividida em cinco conjuntos, o que gera uma base de 625 combinações de regras para cada objetivo. Apenas como exemplo de otimização que pode ocorrer na base de regras, é quando uma variável não influencia em um determinado conjunto de regras, como no caso das regras 1, 2, 3, 4 e 5 da Figura 31. Independente do valor da vazão, quando a pressão é PP a abertura é sempre PP . Assim a base de regras otimizada ficaria como apresentado na Figura 33, ou seja, como menor número de regras que da mais agilidade no processamento, melhor interpretabilidade para os usuários e com a mesma cobertura.

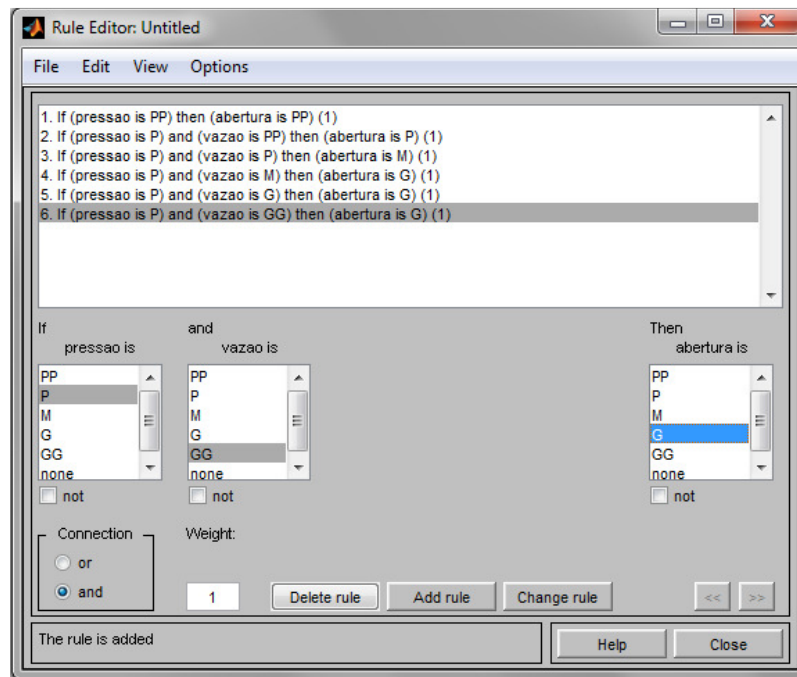


Figura 33 - Exemplo de otimização da base de regras apresentada na Figura 31.

O método proposto foi validado através de simulações utilizando os cenários descritos em Benincasa, Morandin e Kato (2003) e Reddy e Rao (2006). Os resultados obtidos no objetivo *makespan* são comparados com o trabalho Benincasa, Morandin e Kato (2003), e no objetivo *tardiness* é avaliado o comprimento dos prazos.

No próximo capítulo, os procedimentos descritos serão novamente referidos, porém tendo como base os cenários de aplicação e os resultados obtidos, em que os detalhes mais específicos serão melhores contextualizados.

Capítulo 6

APLICAÇÃO DA ABORDAGEM E ANÁLISE DE RESULTADOS

Como apresentado no Capítulo 4, trabalhos que ponderam múltiplas variáveis, alcançam melhores resultados. Porém ponderar uma quantidade exagerada de variáveis pode comprometer o método. O primeiro ponto a ser avaliado são as bases de dados e as bases de regras *fuzzy*, pois a medida que se aumenta variáveis, essas bases aumentam exponencialmente. Devido esse aumento de complexidade computacional, outro ponto afetado é o tempo de resposta do método. Visto que o método é aplicado a sistemas de produção em tempo real, em que é esperado respostas em curtos espaços de tempo.

Nesse caso, o melhor a fazer é associar as variáveis mais pertinentes ao objetivo final. Como o método proposto tem dois objetivos que podem ser permutados em tempo de operação, a partir da revisão bibliográfica e do conhecimento do especialista, foram identificadas as melhores variáveis para cada objetivo.

Objetivo 1: *makespan*

- Distância;
- Número de nós;
- *Buffer* de entrada;
- *Buffer* de Saída.

Objetivo 2: *tardiness*

- Distância;
- Número de nós;
- Encadeamento de tarefas;
- Dada de entrega.

A distância é uma variável importante. Um diferencial desse trabalho com relação à maioria dos trabalhos apresentados na literatura que considera essa variável é que os trabalhos da literatura só consideram a distância entre a posição atual do AGV e a estação de trabalho requisitante. Neste trabalho é somada também a distância entre a estação de trabalho solicitante e o destino final.

Para a criação de uma base de dados e uma base de regras que representem bem o problema e alcance bons resultados, é necessário que os cenários em que o método será aplicado sejam analisados.

Os cenários escolhidos para aplicação do método foram cenários propostos anteriormente por outros autores. Um cenário é o proposto por Benincasa, Morandin e Kato (2003) cujo layout está apresentado na Figura 34 e os outros dois são cenários propostos por Reddy e Rao (2006) cujo layout estão apresentados na Figura 35 e Figura 36.

Como nos artigos os autores não descrevem todos os detalhes referentes aos cenários, o primeiro passo foi redesenhá-los no software Automod (Student Version 11.1) e a partir dessa visualização gerar dados coerentes para cada um como, as distâncias entre as estações de trabalho, que são apresentadas nas tabelas: Tabela 2, Tabela 6, Tabela 10, os números de nós entre as estações de trabalho, que são apresentados nas tabelas: Tabela 3, Tabela 7, Tabela 11, os tempos de processamento de cada produto em cada estação de trabalho, que são apresentados nas tabelas: Tabela 5, Tabela 9,

Tabela 13 e os roteiros dos produtos, que são apresentados nas tabelas: Tabela 4, Tabela 8, Tabela 12.

Nas tabelas de distância e números de nós, na primeira coluna e na primeira linha estão os locais de carregamento e descarregamento e as estações de trabalho. A verificação segue dos itens da primeira coluna para os itens da primeira linha. Por exemplo, se localizar na Tabela 2 a linha M5 com a coluna M4 saberá que a distância para ir da estação de trabalho M5 para estação de trabalho M4 é de 78 unidades.

As tabelas de tempo de processamento são representadas pelas estações de trabalho (M1, M2, ..., M_n) e pelos produtos (P1, P2, ..., P_n).

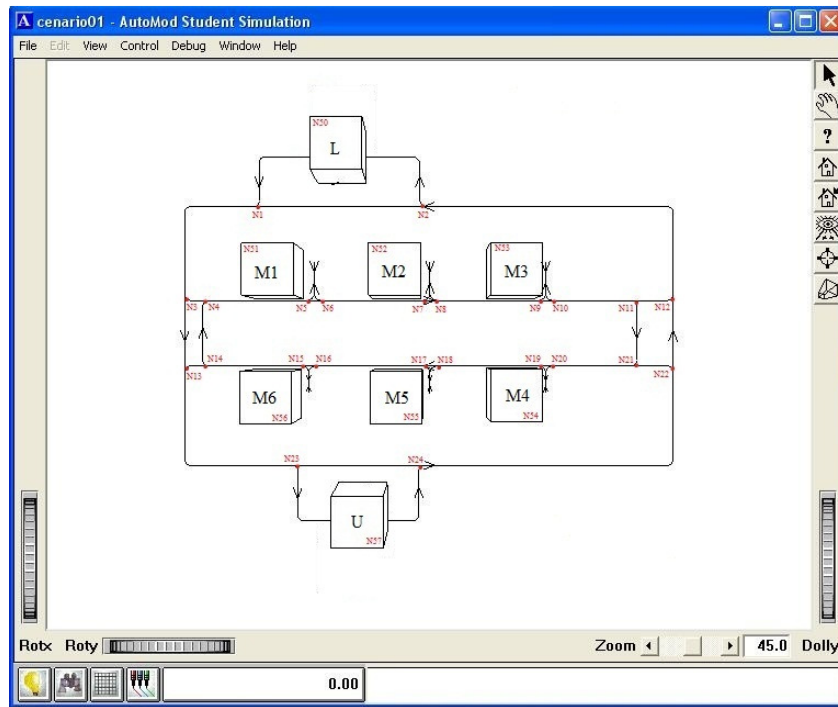


Figura 34 - Layout do cenário 1, adaptado de (BENINCASA; MORANDIN JR; KATO, 2003)

Tabela 2 - Distâncias do cenário 1

	L	M1	M2	M3	M4	M5	M6	U
L	0	33	41	49	72	80	88	56
M1	57	0	9	17	40	48	56	88
M2	49	78	0	9	32	40	48	80
M3	41	70	78	0	24	32	40	72
M4	117	40	48	56	0	9	17	49
M5	109	32	40	48	78	0	9	41
M6	101	24	32	40	70	78	0	33
U	72	101	109	117	41	49	57	0

Tabela 3 - Número de nós do cenário 1

	L	M1	M2	M3	M4	M5	M6	U
L	0	6	8	10	14	16	18	6
M1	10	0	4	6	10	12	14	18
M2	8	12	0	4	8	10	12	16
M3	6	10	12	0	6	8	10	14
M4	14	10	12	14	0	4	6	10
M5	12	8	10	12	12	0	4	8
M6	10	6	8	10	10	12	0	6
U	6	10	12	14	6	8	10	0

Tabela 4 - Roteiros de produção do cenário 1

Produto	Roteiro	Sequência				
1	1	L	1	3	5	U
1	2	L	2	3	5	U
2	1	L	2	3	4	U
2	2	L	1	2	5	U
3	1	L	1	2	6	U
3	2	L	2	4	5	U
4	1	L	1	4	5	U
4	2	L	3	4	5	U
5	1	L	1	4	6	U
5	2	L	2	4	6	U

Tabela 5 - Tempo de processamento das estações de trabalho do cenário 1

	P1	P2	P3	P4	P5
M1	130	130	130	130	130
M2	130	130	130	130	130
M3	190	190	190	190	190
M4	190	190	190	190	190
M5	250	250	250	250	250
M6	250	250	250	250	250

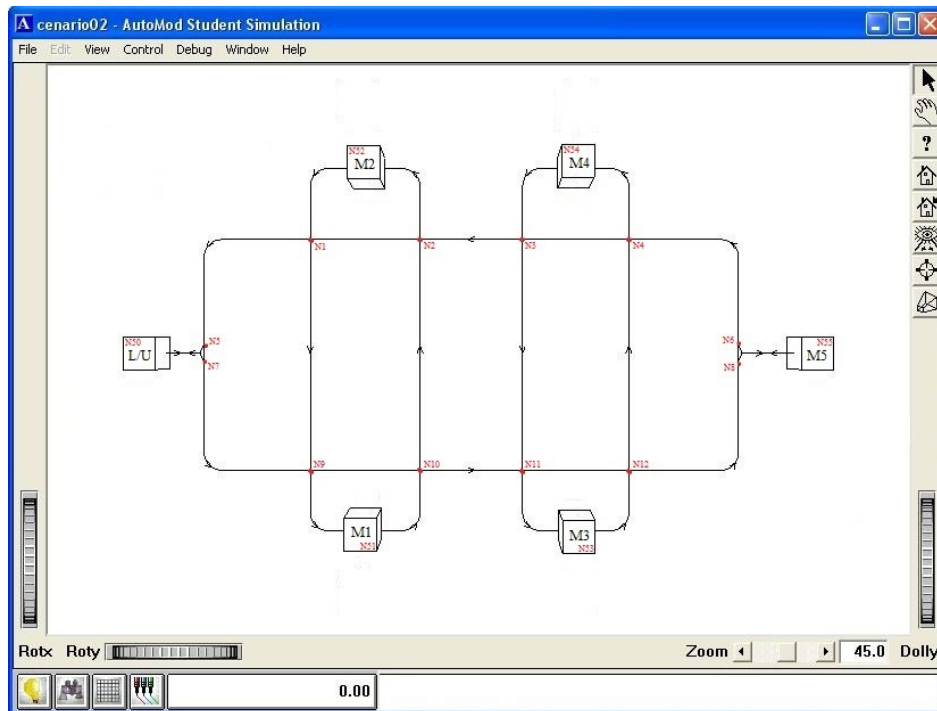


Figura 35 - Layout do cenário 2, adaptado de (REDDY; RAO, 2006)

Tabela 6 - Distâncias do cenário 2

	L/U	M1	M2	M3	M4	M5
L/U	0	25	50	41	66	58
M1	50	0	33	24	49	41
M2	25	33	0	49	74	66
M3	66	74	49	0	33	25
M4	41	49	24	33	0	50
M5	58	66	41	50	25	0

Tabela 7 - Número de nós do cenário 2

	L/U	M1	M2	M3	M4	M5
L/U	0	4	6	6	8	8
M1	6	0	4	4	6	6
M2	4	4	0	6	8	8
M3	8	8	6	0	4	4
M4	6	6	4	4	0	6
M5	8	8	6	6	4	0

Tabela 8 - Roteiros de produção do cenário 2

Produto	Roteiro	Sequência				
		L/U	1	3	5	L/U
1	1	L/U	1	3	5	L/U
1	2	L/U	2	3	5	L/U
2	1	L/U	2	3	4	L/U
2	2	L/U	1	2	5	L/U
3	1	L/U	1	2	3	L/U
3	2	L/U	3	4	5	L/U
4	1	L/U	1	4	5	L/U
4	2	L/U	3	4	5	L/U
5	1	L/U	2	3	5	L/U
5	2	L/U	2	4	5	L/U
6	1	L/U	2	3	4	L/U
6	2	L/U	1	3	4	L/U

Tabela 9 - Tempo de processamento das estações de trabalho do cenário 2

	P1	P2	P3	P4	P5	P6
M1	130	130	130	130	130	130
M2	130	130	130	130	130	130
M3	180	180	180	180	180	180
M4	190	190	190	190	190	190
M5	240	240	240	240	240	240

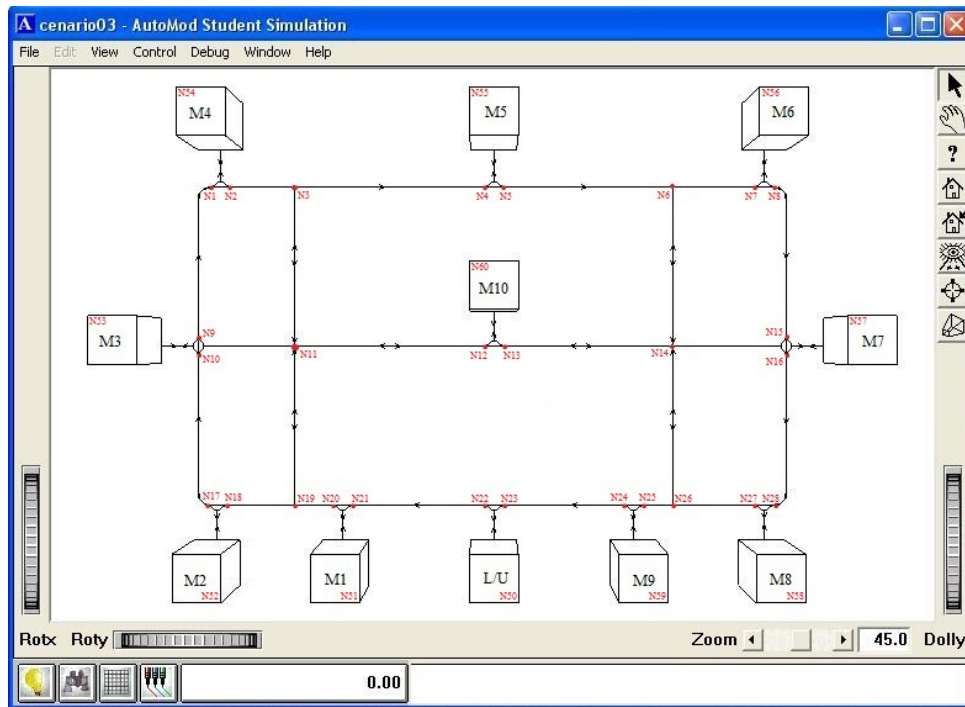


Figura 36 - Layout do cenário 3, adaptado de (REDDY; RAO, 2006)

Tabela 10 - Distâncias do cenário 3

	L/U	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
L/U	0	12	21	32	44	54	74	64	74	74	43
M1	74	0	10	21	33	43	63	53	63	63	32
M2	74	85	0	12	24	44	64	53	63	63	32
M3	64	75	22	0	13	33	53	43	53	53	22
M4	74	85	32	22	0	21	41	52	64	63	32
M5	54	65	74	62	72	0	21	32	44	43	42
M6	44	55	64	53	63	74	0	12	24	33	33
M7	33	44	53	43	53	64	22	0	13	22	23
M8	21	32	41	52	64	74	32	22	0	10	33
M9	12	23	32	43	55	65	85	75	85	0	54
M10	43	54	33	23	33	44	32	22	32	32	0

Tabela 11 - Número de nós do cenário 3

	L/U	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
L/U	0	4	7	9	11	9	12	11	12	12	8
M1	12	0	5	7	9	7	10	9	10	10	6
M2	12	14	0	4	6	9	12	9	10	10	6
M3	11	13	6	0	4	7	10	8	9	9	5
M4	12	14	7	6	0	5	8	10	12	10	6
M5	9	11	14	9	10	0	5	7	9	7	7
M6	11	13	16	9	10	10	0	4	6	9	7
M7	9	11	14	8	9	9	6	0	4	7	6
M8	7	9	12	14	16	14	7	6	0	5	7
M9	4	6	9	11	13	11	14	13	14	0	10
M10	8	10	7	6	7	7	6	5	6	6	0

Tabela 12 - Roteiros de produção do cenário 3

Produto	Roteiro	Sequência					
		L/U	1	3	4	7	L/U
1	1	L/U	1	3	4	7	L/U
1	2	L/U	2	3	5	7	L/U
2	1	L/U	1	2	5	8	L/U
2	1	L/U	1	3	6	8	L/U
3	2	L/U	6	7	9	10	L/U
3	3	L/U	5	7	8	10	L/U
4	1	L/U	3	5	8	10	L/U
4	2	L/U	3	4	8	9	L/U
5	1	L/U	2	4	6	7	L/U
5	2	L/U	2	4	5	7	L/U
6	1	L/U	1	5	6	9	L/U
6	2	L/U	1	6	8	9	L/U
7	1	L/U	3	4	8	10	L/U
7	2	L/U	3	6	8	10	L/U

Tabela 13 - Tempo de processamento das estações de trabalho do cenário 3

	P1	P2	P3	P4	P5	P6	P7
M1	130	130	130	130	130	130	130
M2	130	130	130	130	130	130	130
M3	185	185	185	185	185	185	185
M4	185	185	185	185	185	185	185
M5	230	230	230	230	230	230	230
M6	230	230	230	230	230	230	230
M7	250	250	250	250	250	250	250
M8	250	250	250	250	250	250	250
M9	260	260	260	260	260	260	260
M10	280	280	280	280	280	280	280

A partir dos dados dos cenários e da base de conhecimento do laboratório, foram criadas as bases de dados fuzzy e as bases de regras fuzzy no software Matlab (*Version 7.10.0.499 (R2010a)*).

As bases de dados foram criadas dividindo as quatro variáveis de entrada e a variável de saída de cada objetivo em cinco partições triangulares, que foram escolhidas empiricamente, conforme apresentado nas figuras: Figura 38, Figura 39, Figura 40, Figura 41, Figura 42, Figura 44, Figura 45, Figura 46, Figura 47 e Figura 48, onde:

- PP = muito pequeno;
- P = pequeno;
- M = médio;
- G = grande;
- GG = muito grande.

Juntando as figuras: Figura 37, Figura 38, Figura 39, Figura 40, Figura 41 e Figura 42 forma-se a base de dados completa do objetivo *makespan*. Na Figura 37 é apresentado o sistema fuzzy macro, com as quatro variáveis de entrada e uma de saída. Na Figura 38 é apresentada a base de dados da variável de saída: *prioridade*. Nas figuras: Figura 39, Figura 40, Figura 41 e Figura 42 são apresentadas às bases de dados das variáveis de entrada: *buffer* de saída, *distância*, *buffer* de entrada e *número de nós* respectivamente.

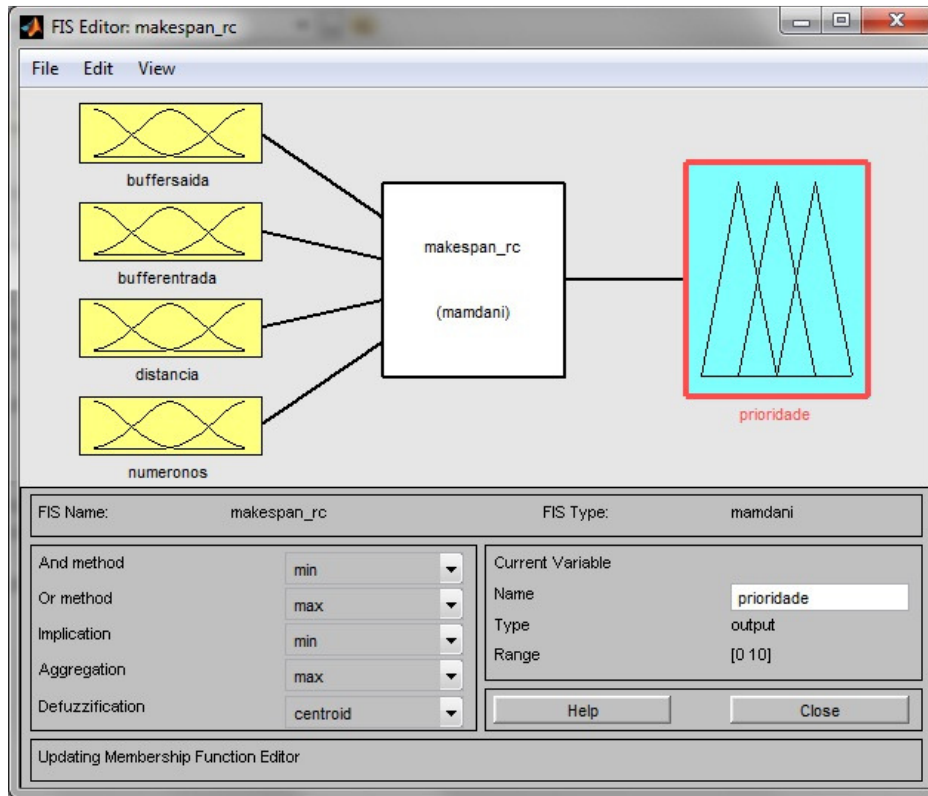


Figura 37 - Base de dados do objetivo *makespan*

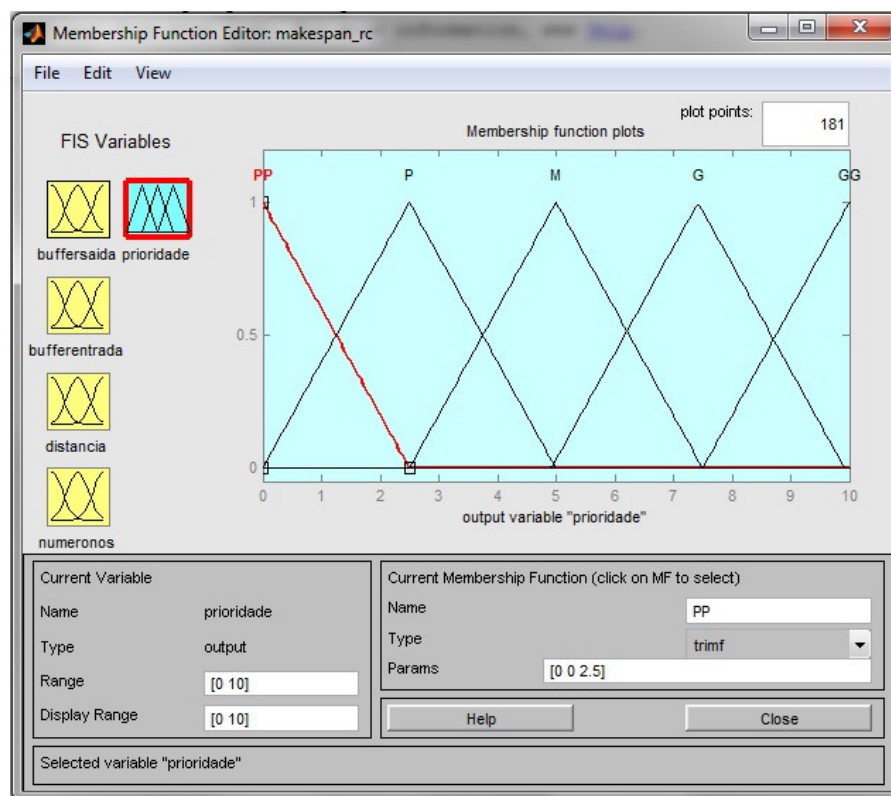


Figura 38 - Base de dados do objetivo *makespan*, variável *prioridade*

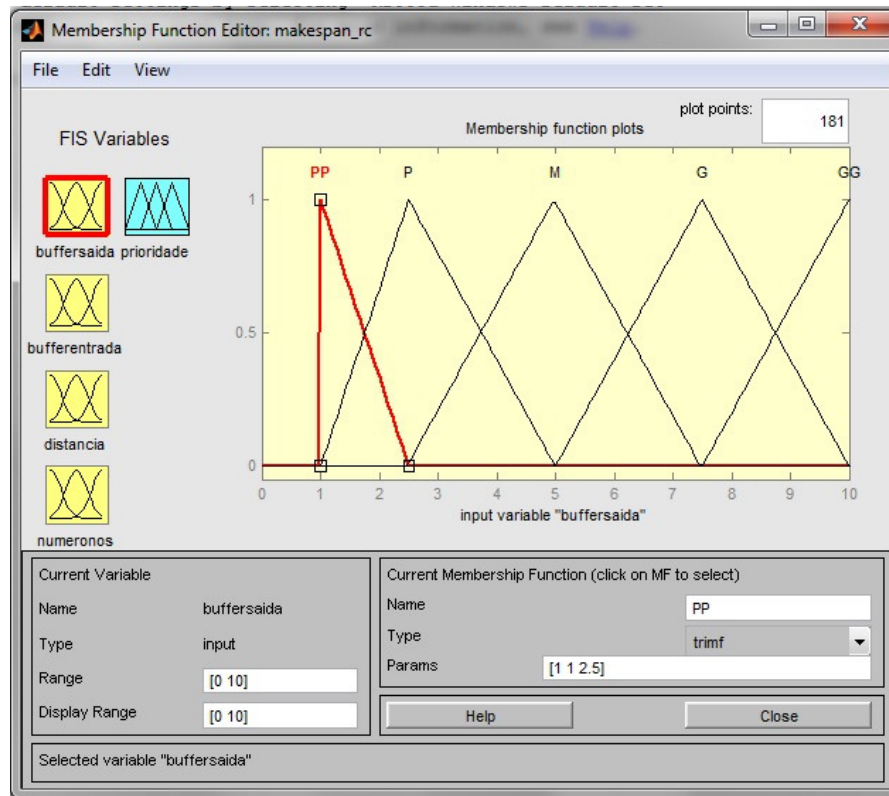


Figura 39 - Base de dados do objetivo *makespan*, variável *buffer* de saída

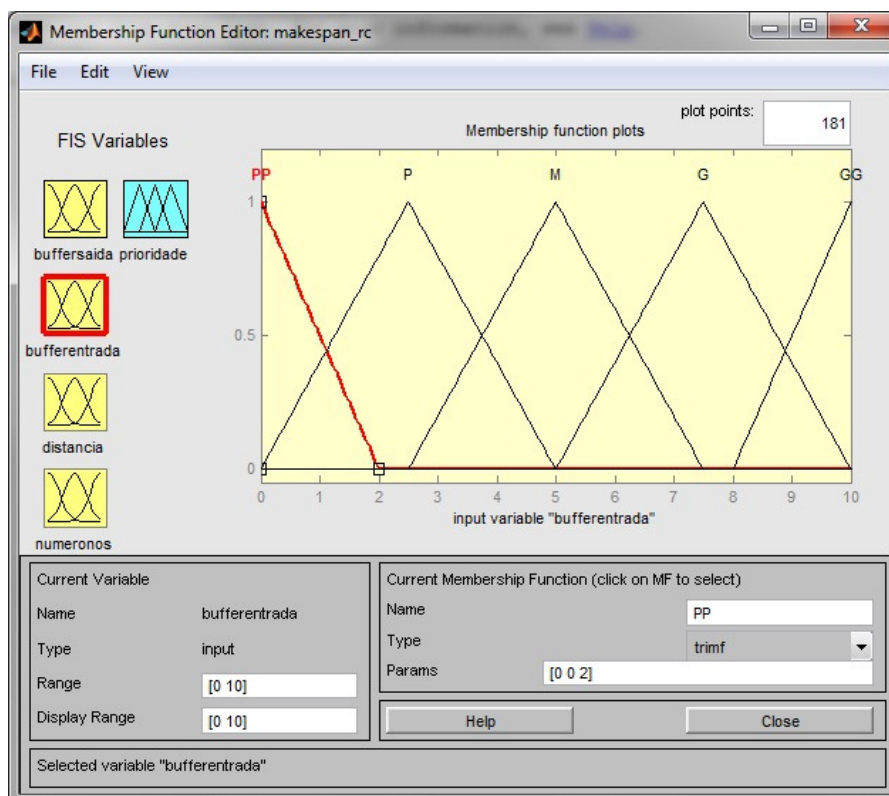


Figura 40 - Base de dados do objetivo *makespan*, variável *buffer* de entrada

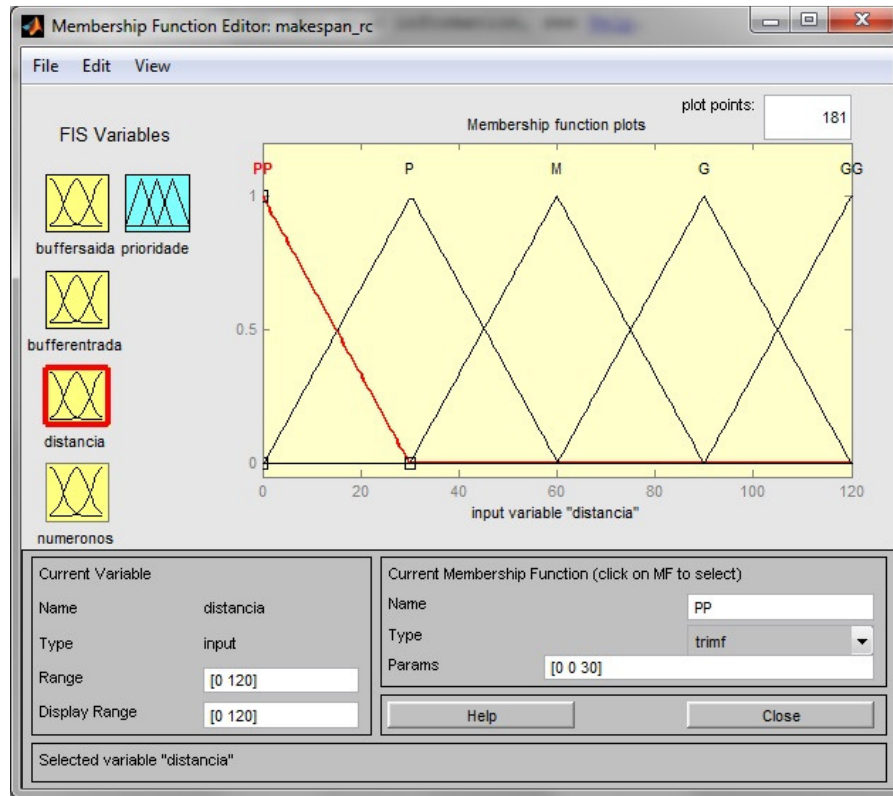


Figura 41 - Base de dados do objetivo *makespan*, variável distância

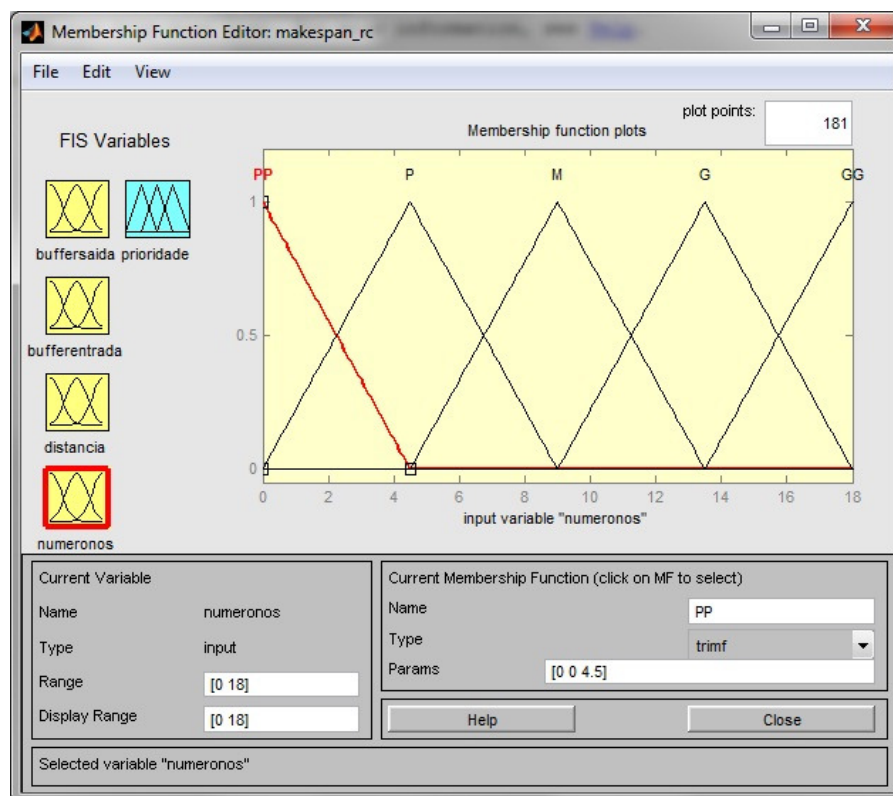


Figura 42 - Base de dados do objetivo *makespan*, variável número de nós

Juntando as figuras: Figura 43, Figura 44, Figura 45, Figura 46, Figura 47 e Figura 48 forma-se a base de dados completa do objetivo *tardiness*. Na Figura 43 é apresentado o sistema *fuzzy* macro, com as quatro variáveis de entrada e uma de saída. Na Figura 44 é apresentada a base de dados da variável de saída: prioridade. Nas figuras: Figura 45, Figura 46, Figura 47 e Figura 48 são apresentadas às bases de dados das variáveis de entrada: data de entrega, distância, encadeamento de tarefas e número de nós respectivamente.

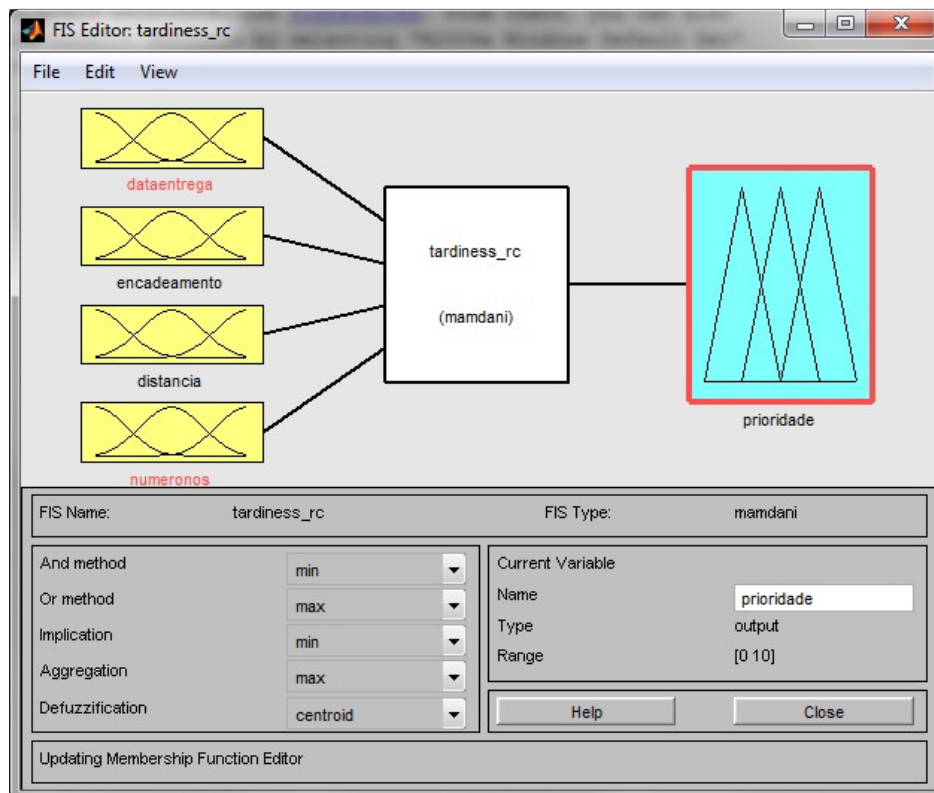


Figura 43 - Base de dados do objetivo *tardiness*

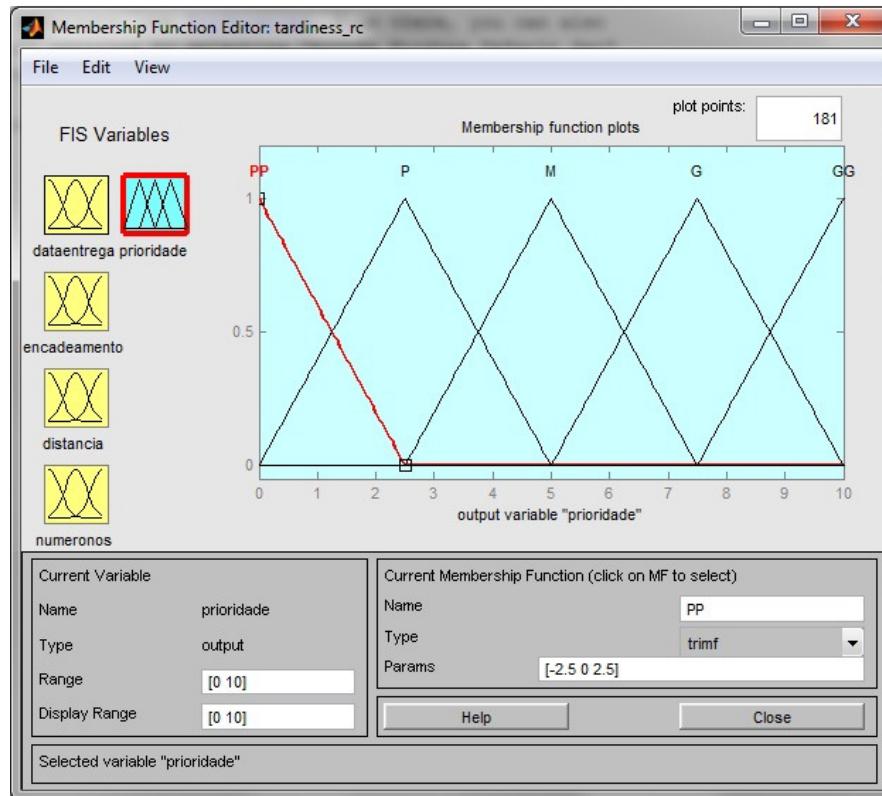


Figura 44 - Base de dados do objetivo *tardiness*, variável prioridade

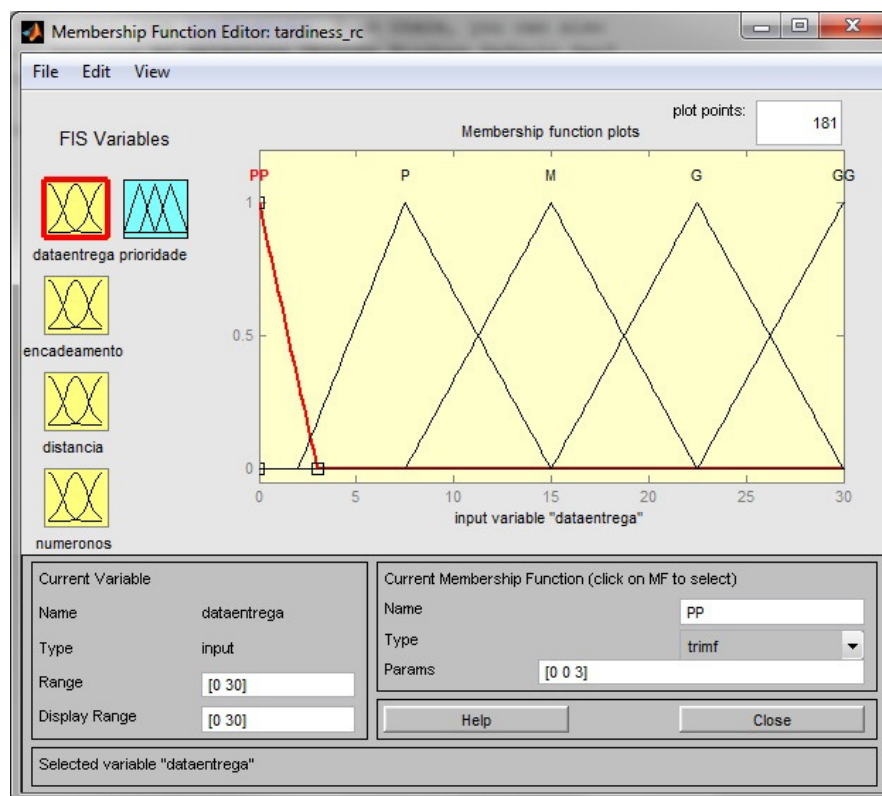


Figura 45 - Base de dados do objetivo *tardiness*, variável data de entrega

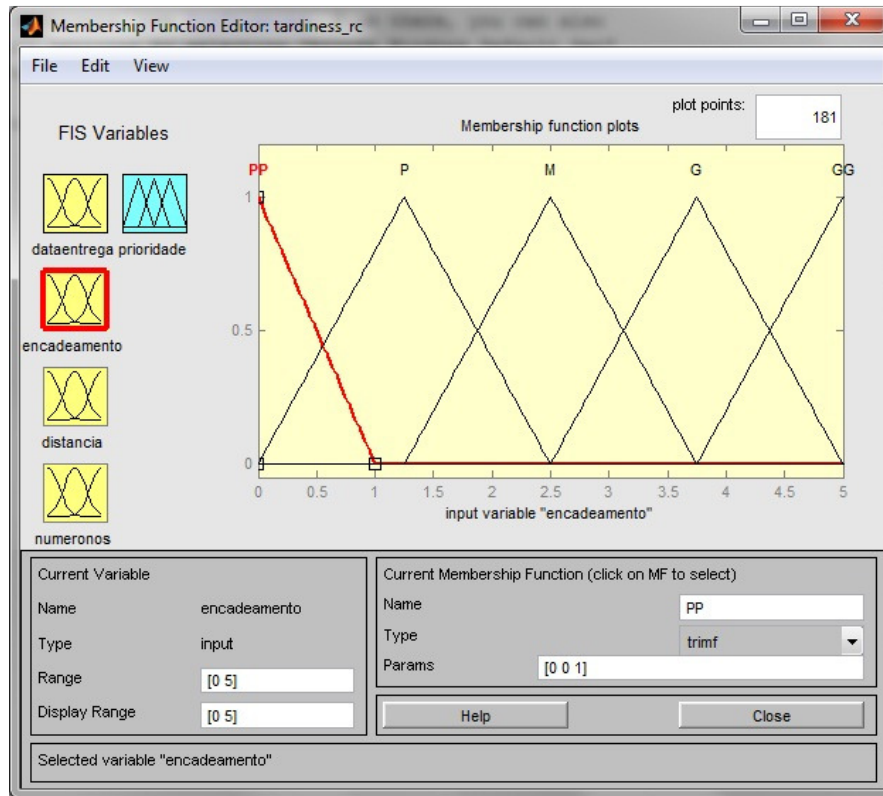


Figura 46 - Base de dados do objetivo *tardiness*, variável encadeamento de tarefas

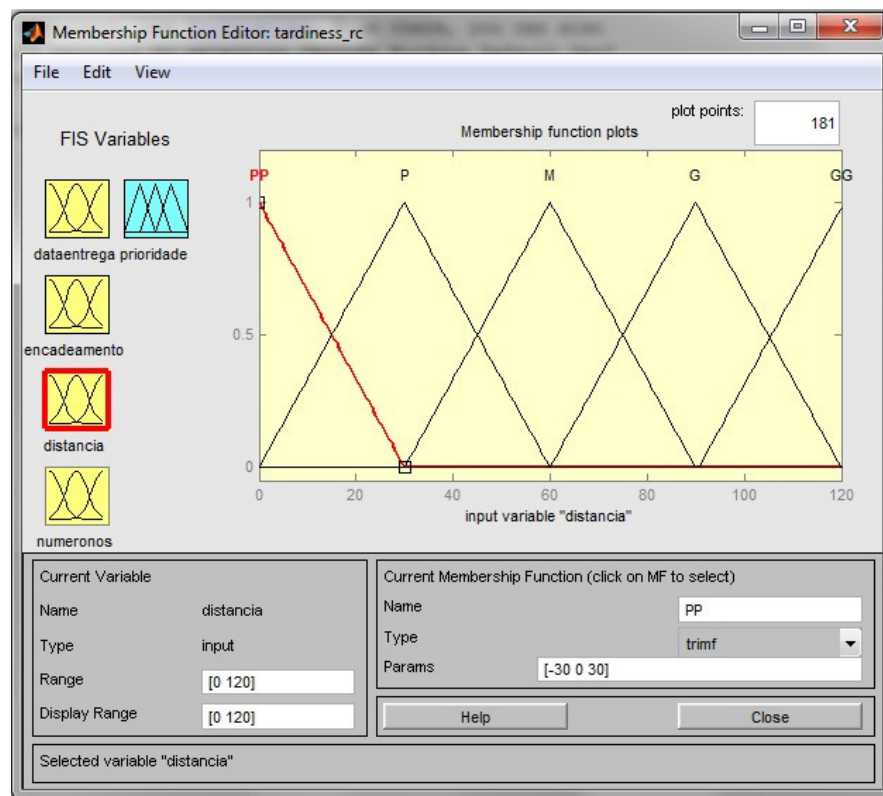


Figura 47 - Base de dados do objetivo *tardiness*, variável distância

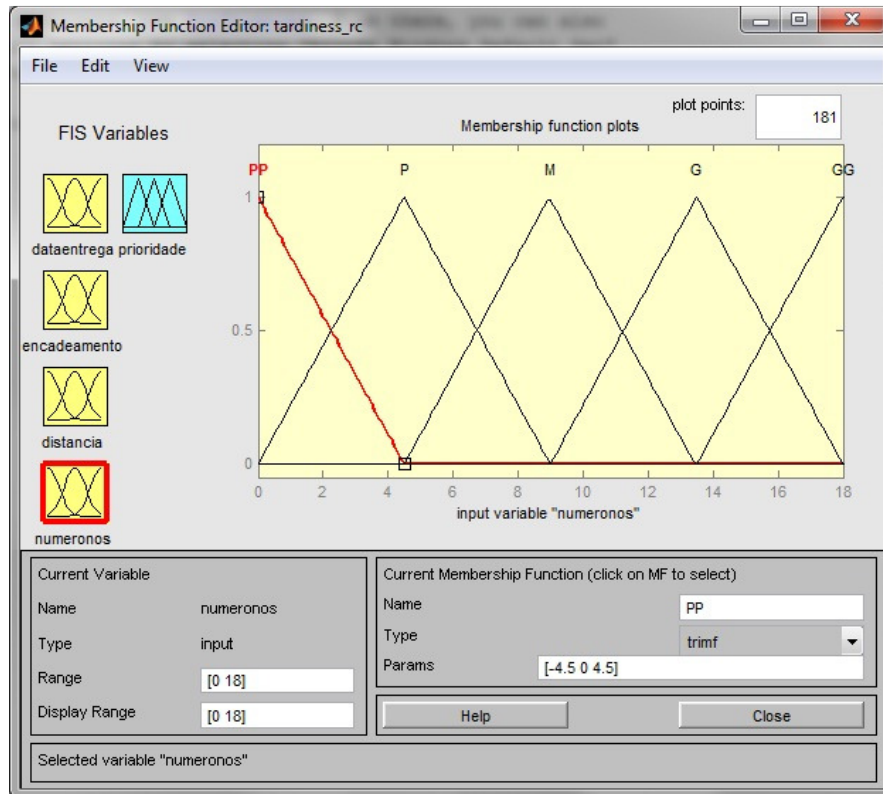


Figura 48 - Base de dados do objetivo *tardiness*, variável número de nós

Para a criação de uma base de regras *fuzzy* para esse tipo de sistema, faz-se necessário a base de conhecimento do laboratório. A base de regras inicial foi criada a partir de todas as combinações das variáveis de entrada (4 variáveis, divididas em 5 partições cada é igual a 625 regras) e a saída gerada para cada combinação foi indicada pela base de conhecimento do laboratório. Na Tabela 14 é apresentada uma amostra com as primeiras 25 das 625 regras da base de regras do objetivo *makespan*.

Tabela 14 - Amostra da base de regras do objetivo *makespan*

Buffer S.	Buffer E.	Distância	Nro de Nós	PRIORIDADE
PP	PP	PP	PP	GG
PP	PP	PP	P	GG
PP	PP	PP	M	GG
PP	PP	PP	G	GG
PP	PP	PP	GG	GG
PP	PP	P	PP	GG
PP	PP	P	P	GG
PP	PP	P	M	GG
PP	PP	P	G	GG
PP	PP	P	GG	GG
PP	PP	M	PP	GG
PP	PP	M	P	GG
PP	PP	M	M	G
PP	PP	M	G	G
PP	PP	M	GG	G
PP	PP	G	PP	G
PP	PP	G	P	G
PP	PP	G	M	G
PP	PP	G	G	M
PP	PP	G	GG	M
PP	PP	GG	PP	G
PP	PP	GG	P	G
PP	PP	GG	M	M
PP	PP	GG	G	M
PP	PP	GG	GG	M

A partir do conjunto de regras inicial que foi obtido manualmente, os AGs são usados para selecionar um subconjunto de regras. Seu objetivo é reduzir o número de regras eliminando redundâncias e conflitos melhorando o desempenho do conjunto de regras inicial e tornando-o mais equilibrado entre cobertura versus interpretabilidade.

As regras são codificadas por números reais que representam o índice dos conjuntos fuzzy que aparecem na parte do antecedente da regra. Dessa forma para cada variável

linguística dos conjuntos *fuzzy* foi atribuído um valor inteiro, e também foi adicionada a pertinência *não importa (NI)* (do inglês: *don't care*). Na Tabela 15 são apresentados os valores que foram atribuídos para cada partição.

Tabela 15 - Atribuição de valores reais

Partição <i>Fuzzy</i>	Valor Real
NI	0
PP	1
P	2
M	3
G	4
GG	5

O mínimo de otimização que se espera é que a população final seja pelo menos 50% menor que a base de regras inicial construída manualmente com todas as combinações possíveis. Dessa forma a população inicial do algoritmo genético é constituída por 312 regras aleatórias da base de regras inicial.

Cada cromossomo possui quatro genes, que correspondem às quatro variáveis de entrada e cada gene possui um valor inteiro entre 0 e 5. Na Tabela 16 é apresentada uma amostra de como são formados os cromossomos e a população inicial.

Tabela 16 – Amostra dos cromossomos do AG

C1:	1	1	1	1
C2:	1	1	1	2
C3:	1	1	1	3
C4:	1	1	1	4
C5:	1	1	1	5
C6:	1	1	2	1

C312:	3	3	3	1

A população é aplicada ao problema de despacho de AGVs e cada cromossomo recebe um índice de importância, que é medido de acordo com a quantidade de vezes que a

regra foi disparada. A população recebe como função de avaliação o valor do objetivo a ser alcançado. A função de avaliação é definida com base no erro médio quadrático, que é representada pela seguinte expressão:

$$f(C_i) = \frac{1}{2N} \sum_{y=1}^N (e_y - e'_y)^2 \tag{6.1}$$

Onde:

- N é o número total de exemplos de treinamento;
- e'_y é o valor de saída obtido a partir do sistema usando a base de regras codificada em C_i .
- e_y é o valor conhecido desejado.

Na criação da nova população aplica-se o elitismo com taxa de 20%, o cruzamento de um ponto central com taxa de 80%, a mutação se dá em seguida sobre 5% da população do cruzamento escolhendo um gene aleatório, e a seleção da população é pelo método torneio. Na Figura 49 é ilustrado o funcionamento do cruzamento e na Figura 50 é ilustrado o funcionamento da mutação.

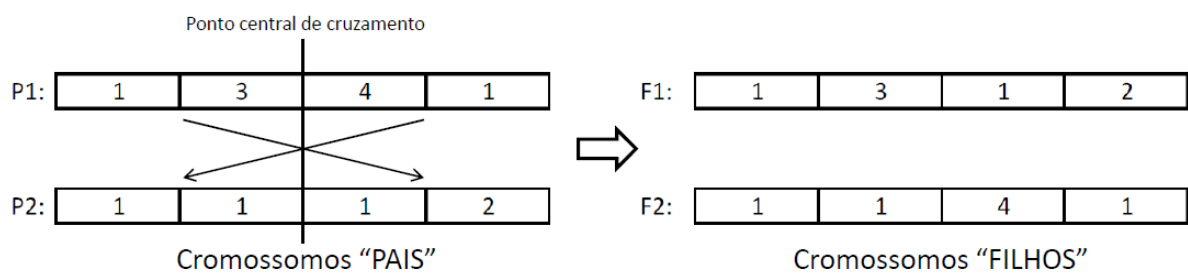


Figura 49 - Exemplo de funcionamento do cruzamento

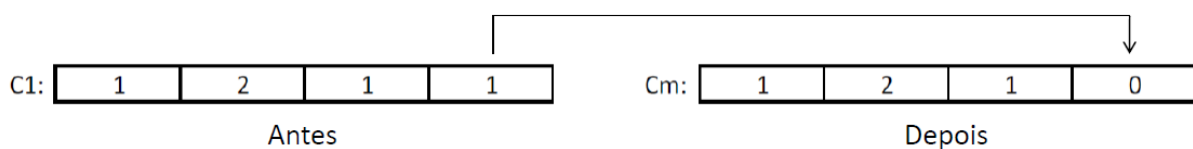


Figura 50 - Exemplo de funcionamento da mutação

Para cada nova população obtida é aplicada no problema de despacho de AGVs, e a função de avaliação obtida por cada população é armazenada. O critério de parada pode acontecer de três formas:

- Primeira: Se o valor da função de avaliação de uma determinada população for igual ou inferior ao valor pré-estabelecido.
- Segunda: Se após quinze gerações consecutivas a função de avaliação das populações não melhoraram.
- Terceira: Se for alcançado o número máximo de 100 iterações.

Assim, depois de aplicada a otimização, a base de regras final do objetivo *makespan* que inicialmente tinha 625 regras, ficou com 280 regras, e o objetivo *tardiness* ficou com 200 regras em sua base. Na Tabela 17 é apresentada uma amostra com as 16 primeiras regras da base de regras otimizada do objetivo *makespan*.

Tabela 17 - Amostra da base de regras otimizada do objetivo *makespan*

Buffer S.	Buffer E.	Distância	Nro de Nós	PRIORIDADE
*	GG	*	*	PP
PP	G	*	*	PP
PP	M	PP	*	G
PP	PP	G	PP	G
PP	PP	G	P	G
PP	PP	G	M	G
PP	PP	G	G	M
PP	PP	G	GG	M
PP	PP	GG	PP	G
PP	PP	GG	P	G
PP	PP	GG	M	M
PP	PP	GG	G	M
PP	PP	GG	GG	M
PP	P	M	PP	GG
PP	P	M	P	GG
PP	P	M	M	G

- Onde “*”, significa que a variável não influencia naquela regra.

Analisando os trabalhos da literatura, percebemos que a maioria dos trabalhos que abordam o problema de despacho de AGVs não leva em consideração o espaço físico dos AGVs, ou seja, pode existir mais de um AGV carregando ao mesmo tempo na mesma estação de trabalho, ou dois AGVs utilizando um nó ao mesmo tempo, não existe *deadlock*, etc.

Modelando os cenários sob estudo em redes de Petri, conforme apresentado nas figuras: Figura 51, Figura 52e Figura 53, conseguimos ter uma visão melhor dos recursos a serem utilizados.

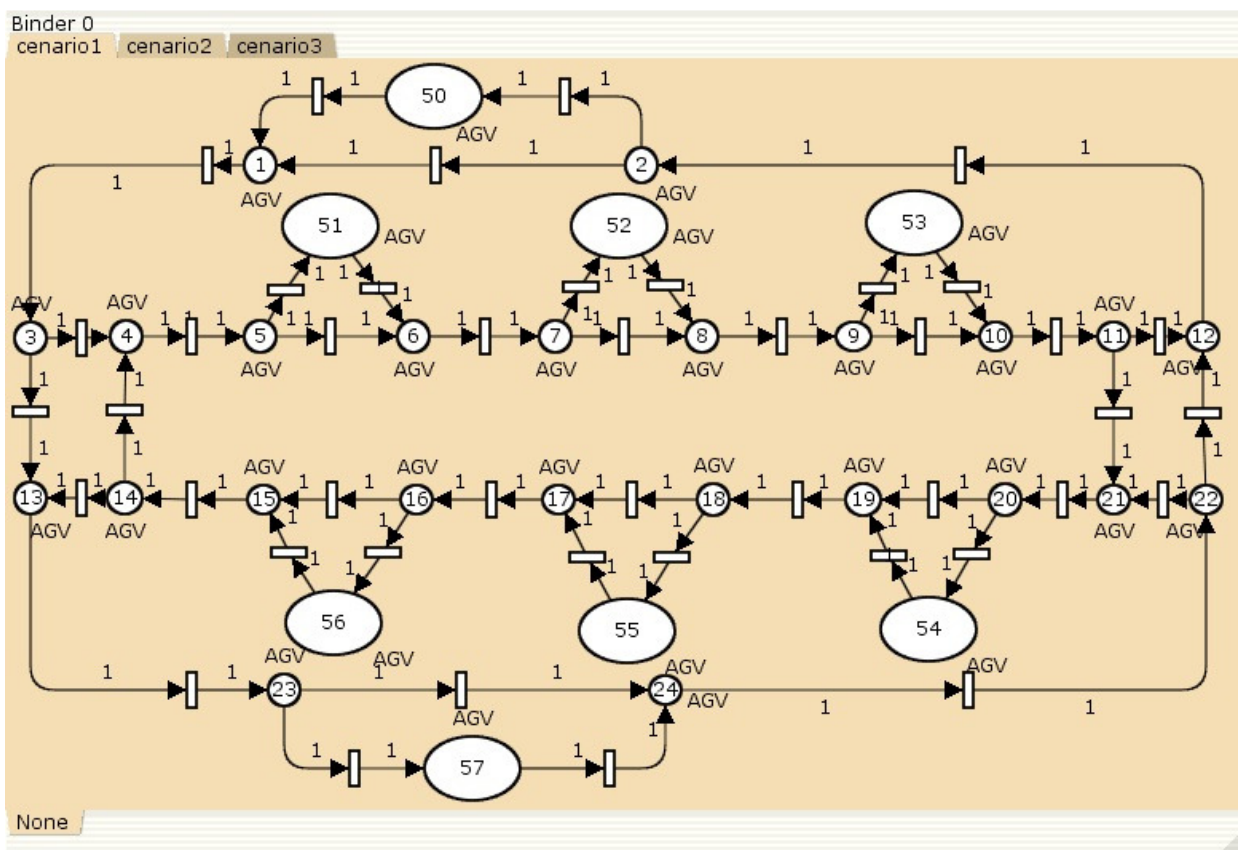


Figura 51 - Modelo em redes de Petri do cenário 1

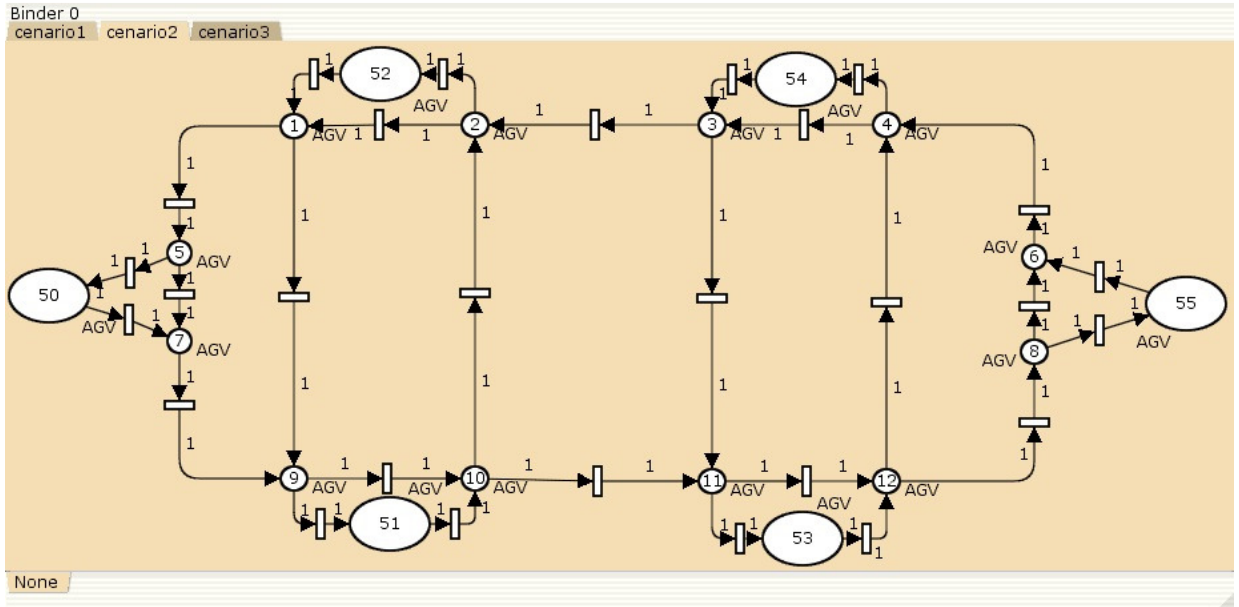


Figura 52 - Modelo em redes de Petri do cenário 2

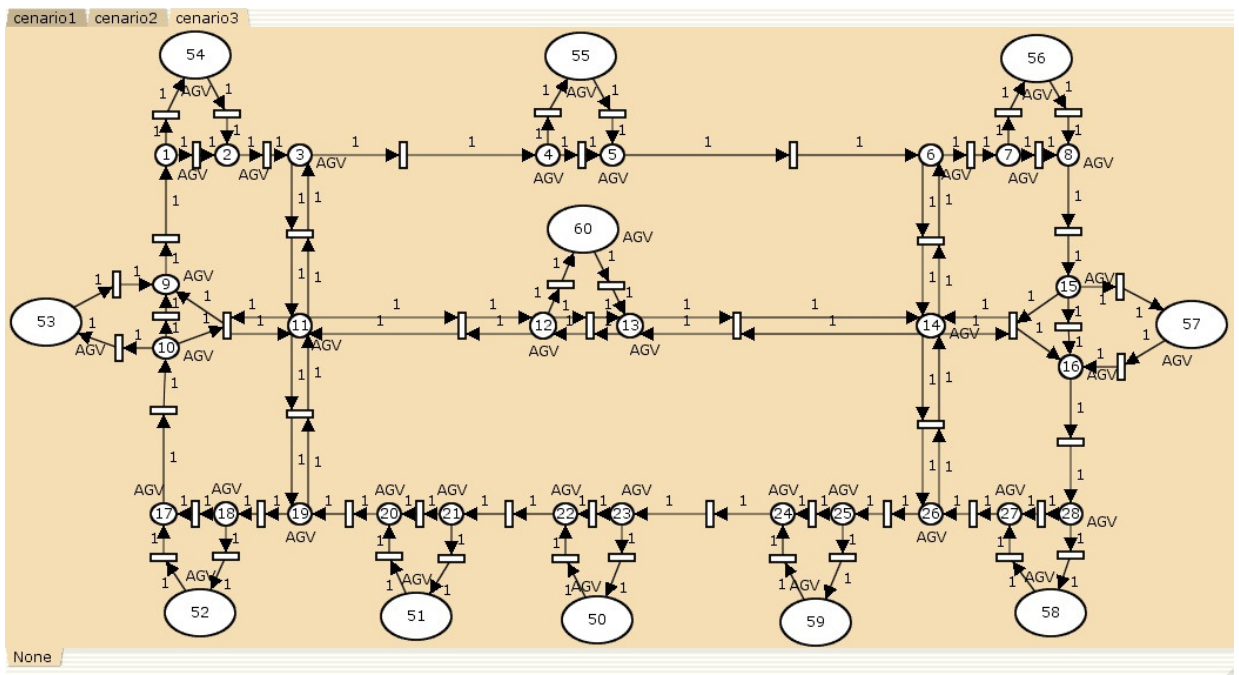


Figura 53 - Modelo em redes de Petri do cenário 3

Dessa forma, todas as estações de trabalho e todos os nós do caminho são considerados um recurso a ser usado por um único AGV em um determinado período de tempo, o que acontece em fábricas reais.

A partir dessas informações o sistema pode criar Tabelas de Alocação dinâmica (TAD), alocando um recurso para um determinado AGV em um período de tempo, e esse recurso não poderá ser usado por outro AGV nesse mesmo período de tempo. Incorporando todo o método no Matlab também conseguiu-se suprir uma outra característica do CPNtools. Na modelagem em redes de Petri, se existe um caminho que leva 10 unidades de tempo para ser percorrido, e um AGV está usando-o, o sistema só libera o uso desse caminho para outro AGV, quando o primeiro percorreu-o por completo. Já na simulação no Matlab, o caminho de 10 unidades de tempo é “dividido” em 10 partes, então se o AGV1 está na parte 2 do caminho, já libera para o AGV2 entrar na parte 1, e assim vai sendo incrementando.

Outro ponto importante que é adicionado ao método, é o monitoramento constante dos tempos dos AGVs e dos tempos das estações de trabalho. Com essas informações é possível fazer algumas predições na tomada de decisão, que geram ganhos significativos no resultado final. Uma das predições ocorre quando existe uma solicitação de transporte, mas não só os AGVs livres entram na disputa pelo atendimento. Os AGVs que de acordo com a tabela de alocação dinâmica de recursos, estão indo para uma região favorável ao atendimento dessa solicitação, e de acordo com o monitoramento dos tempos falta 12 ou menos unidades de tempo para o termino do seu trabalho, esse AGV também entra na disputa pelo atendimento. Outra possível predição ocorre quando um ou mais AGVs estão livres e não existe solicitação de transportes, através dos tempos de processamento das estações de trabalho verifica-se as que irão terminar primeiro, assim o AGV já vai se deslocando para aquela determinada estação de trabalho.

Os autores Benincasa, Morandin e Kato (2003), tiveram como objetivo em seu trabalho a redução do *makespan*. O objetivo *makespan* proposto nesse trabalho foi simulado nos três cenários descritos anteriormente e comparado com os a proposta do trabalho citado. A seguir na Tabela 19 e Tabela 20 são apresentados os resultados das simulações.

Tabela 19 - Resultados do objetivo *makespan*

PRODUÇÃO DE 5 LOTES				PRODUÇÃO DE 10 LOTES			
Quantidade de AGVs	Makespan (Benincasa)	Makespan (Proposta)	% de ganho da proposta	Quantidade de AGVs	Makespan (Benincasa)	Makespan (Proposta)	% de ganho da proposta
cenário 01				cenário 01			
1	6086	5060	16,86	1	12418	9487	23,60
2	4720	4392	6,95	2	9360	8193	12,47
3	4576	4268	6,73	3	9199	7980	13,25
cenário 02				cenário 02			
1	8056	7169	11,01	1	15260	13596	10,90
2	5288	4678	11,54	2	9896	8851	10,56
3	5194	4622	11,01	3	9752	8807	9,69
cenário 03				cenário 03			
1	9244	8213	11,15	1	16934	15066	11,03
2	6223	5538	11,01	2	12151	10814	11,00
3	5747	5092	11,40	3	11874	10567	11,01
4	5722	5039	11,94	4	11226	10438	7,02

Tabela 20 - Resultados do objetivo *makespan*

PRODUÇÃO DE 25 LOTES				PRODUÇÃO DE 50 LOTES			
Quantidade de AGVs	Makespan (Benincasa)	Makespan (Proposta)	% de ganho da proposta	Quantidade de AGVs	Makespan (Benincasa)	Makespan (Proposta)	% de ganho da proposta
cenário 01				cenário 01			
1	29198	25300	13,35	1	56782	48660	14,30
2	23298	18373	21,14	2	45996	38874	15,48
3	22443	18306	18,43	3	44572	35798	19,69
cenário 02				cenário 02			
1	37496	32996	12,00	1	74666	66578	10,83
2	23553	20962	11,00	2	46362	41816	9,81
3	23382	20809	11,00	3	46144	41262	10,58
cenário 03				cenário 03			
1	41882	37256	11,05	1	81663	72653	11,03
2	27992	24912	11,00	2	52779	46973	11,00
3	28131	25036	11,00	3	54451	47886	12,06
4	27004	24742	8,38	4	52733	46932	11,00

Para o objetivo *tardiness*, é usado o *takt-time*³ dos produtos, e também é simulado a criação de uma programação alocando os produtos como um gráfico de Gantt que é construído de trás para frente acrescentando um tempo de segurança, para que os produtos sejam fabricados o mais próximo possível da data de entrega. Na Figura 54 é ilustrado como seria uma programação da produção do objetivo *tardiness*.

³ O *takt-time* é definido a partir da demanda do mercado e do tempo disponível para produção; é o ritmo de produção necessário para atender a demanda. Matematicamente, resulta da razão entre o tempo disponível para a produção e o número de unidades a serem produzidas (ALVAREZ; ANTUNES JR., 2001).

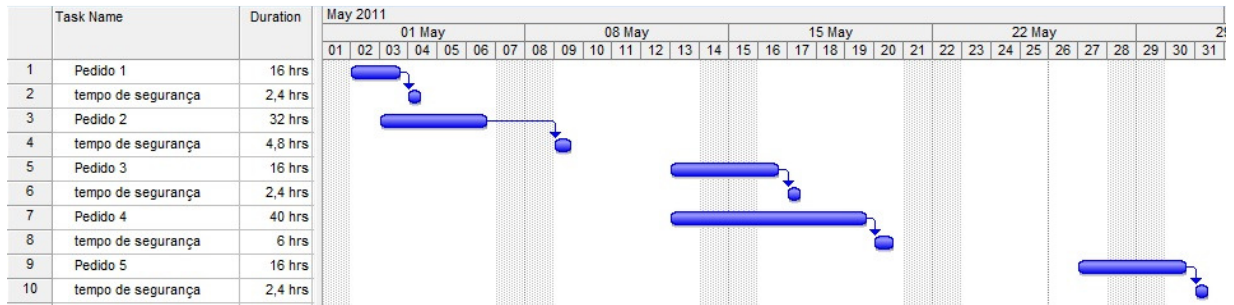


Figura 54 - Amostra de gráfico para alocação da produção

O tempo de segurança é de 15% do tempo total que o lote precisa para ser produzido. Dessa forma os pedidos são produzidos o mais próximo possível da sua data de entrega, evitando custo de estoque e atrasos na entrega. Nos períodos de tempo sem pedidos para produção a fábrica fica literalmente parada, evitando desgaste dos veículos e das máquinas, esses tempos podem ser usados para agendamento de manutenção preventiva e também para incluir pedidos de urgência.

O objetivo *tardiness* também foi simulado nos três cenários citados anteriormente, e para validação foi verificado se os pedidos eram produzidos nas datas esperadas. Na Figura 55 é apresentado o resultado de uma simulação, em que, se existe uma demanda igual ou inferior a quantidade que a indústria é capaz de produzir, todos os pedidos são produzidos na data correta.

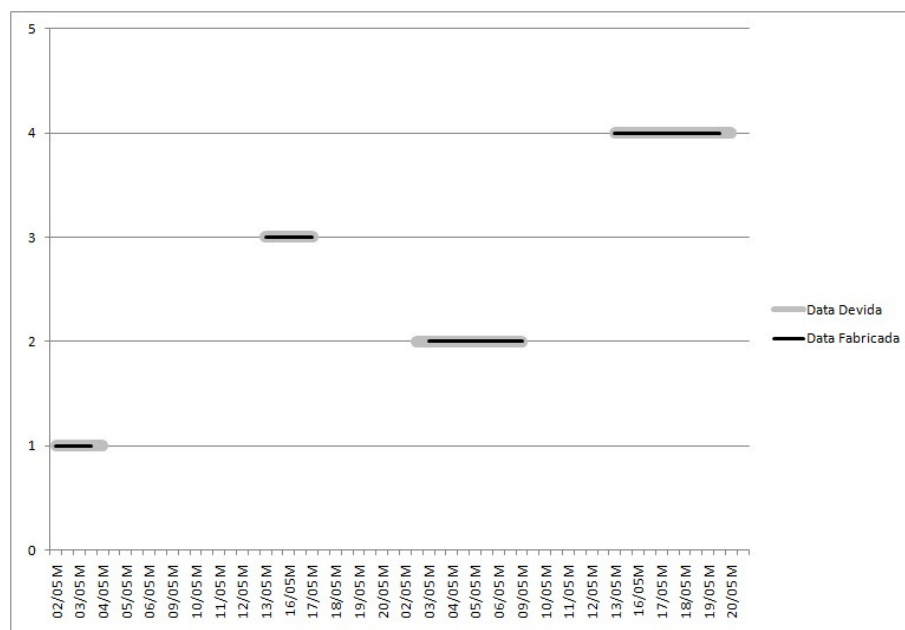
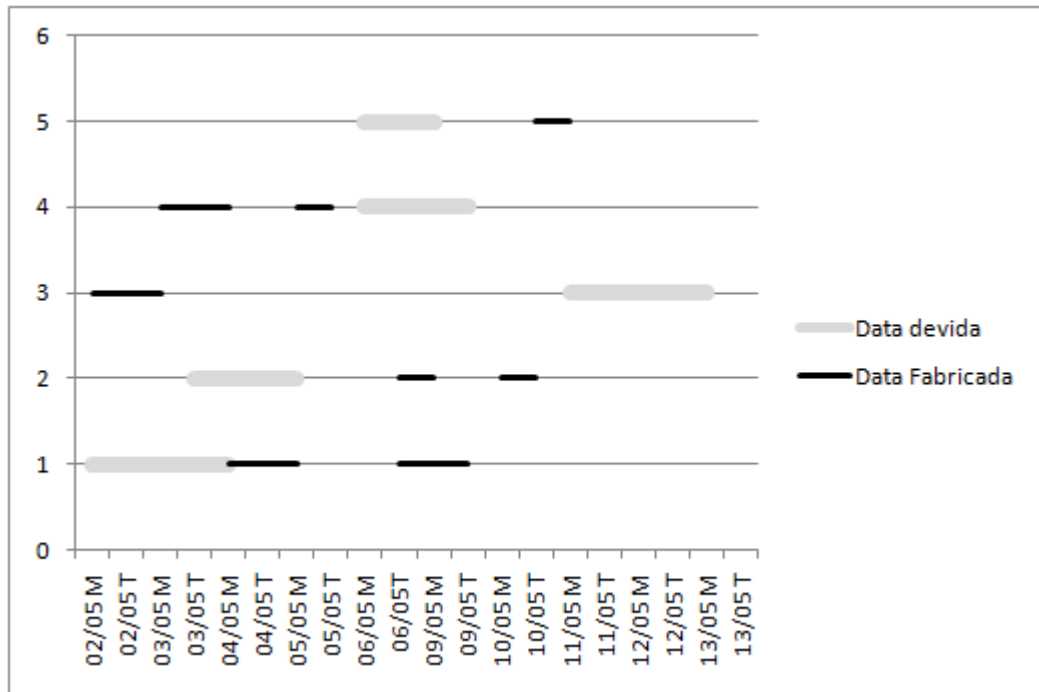
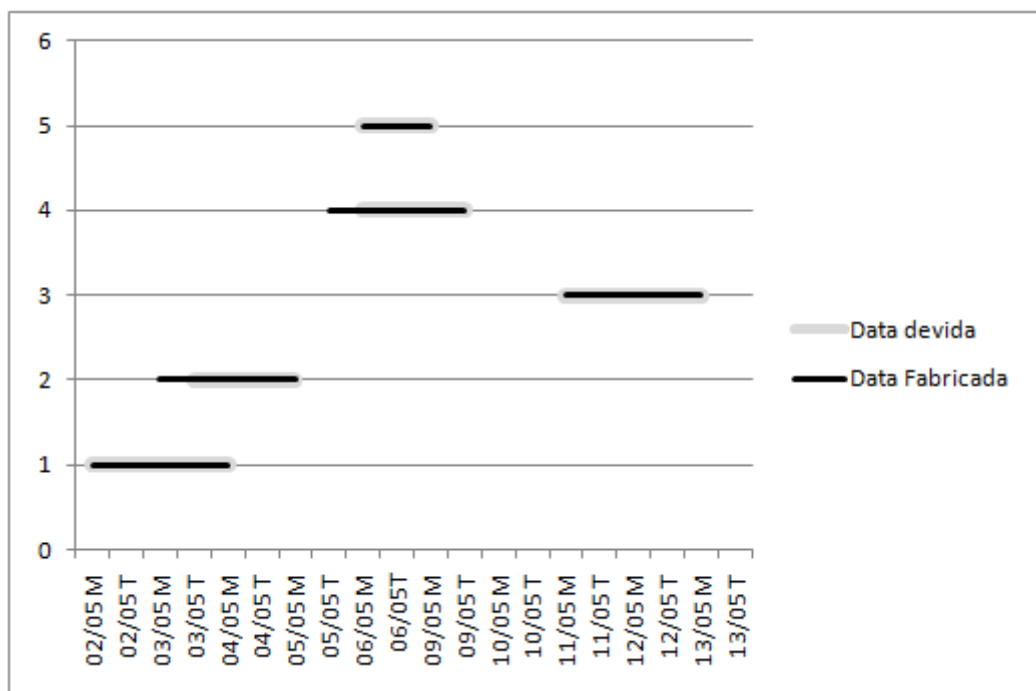


Figura 55 - Simulação do objetivo *tardiness*

Na Figura 56 é apresentada uma comparação entre o resultado o objetivo *makespan* e objetivo *tardiness*.



(a) Produção com o objetivo *makespan*



(b) Produção com o objetivo *tardiness*

Figura 56 - Comparação entre produções do objetivo *makespan* e do objetivo *tardiness*

Analisando a Figura 56(a) é possível verificar que em uma produção de nove dias e meio, o objetivo *makespan* finaliza a produção em sete dias e meio, obtendo um adiantamento de dois dias ou 11% do tempo total, porém não atende aos pedidos, em que alguns são entregues muito adiantados, outros muito atrasados. Já o objetivo *tardiness*, apresentado na Figura 56(b) usa todo o período de nove dias e meio para produção e atende corretamente todos os pedidos.

Dessa forma fica a critério da empresa decidir qual ganho é melhor para os negócios em determinados períodos de produção de determinados produtos.

Em todas as situações simuladas do objetivo *tardiness* não houve pedidos produzidos com atraso. Na maioria das vezes, os pedidos foram produzidos entre 2% e 15% antes do prazo, porém com tempo de adiantamento pouco significativo para geração de grandes estoques.

Capítulo 7

CONCLUSÃO

O processo de evolução dos sistemas de manufatura é caracterizado por um índice elevado de competitividade, em que ambientes cada vez mais complexos buscam atuar em níveis estratégicos frente às exigências de mercado. O controle da produção é uma das atividades específicas da área de manufatura e que tem um impacto expressivo no resultado geral da empresa. Devido às inúmeras vantagens que os Veículos Auto Guiados proporcionam, cada vez mais os AGVs são usados para tarefa de transporte em indústrias, armazéns, portos, etc. Dentre os problemas existentes no gerenciamento dos AGVs, um dos principais em sistemas de produção é o despacho.

Esse trabalho apresenta um método de despacho de AGVs, usando técnicas de inteligência artificial, para ponderação de múltiplas variáveis e melhoramento no sistema de decisão. Também foi usada predição de tarefas, modelagem de recursos e simulações, tornando o modelo mais realista; também foi adicionada a opção de troca de objetivos, tornando o método mais robusto. Todas essas características trouxeram algumas vantagens para o modelo proposto.

A constante mudança do mercado e as diferentes metas almejadas por cada setores das empresas, torna a troca do objetivo uma valiosa vantagem.

Quando o objetivo é minimizar o *makespan*, é utilizado todo o processamento possível. Tende-se a diminuir o custo por peça, já que se produz mais utilizando a mesma quantidade de recursos como máquinas, veículos e mão de obra. A desvantagem é que isso não garante que o prazo será cumprido, e uma produção adiantada pode gerar custos de estocagem.

Quando o objetivo é o *tardiness*, somente é produzido aquilo que é devido, e tenta-se ao máximo finalizar a produção no prazo correto. O *tardiness* normalmente é mais interessante se os produtos a serem produzidos são caros. No tempo que não existe trabalho a produção literalmente para, o que para alguns pode ser visto como prejuízo. Porém com isso não se tem gastos desnecessários, nem desgaste de máquinas e veículos. Outro ponto positivo é que se por ventura aparecer uma ordem de produção com urgência, é possível encaixar a mesma nessa janela de tempo livre, e não comprometer o prazo de entrega dos outros produtos.

A troca de objetivos pode beneficiar quando a demanda está alta, então é decidido por produzir focando no objetivo *makespan*, porém em um determinado momento alguns pedidos podem ser cancelados, para não usar recursos em uma produção que ficará estocada, nesse momento pode-se trocar pelo *tardiness*. Em outro momento, quando existe uma produção controlada usando *tardiness*, porém chegam vários pedidos com um prazo curto de entrega, isso pode ser alterado trocando o objetivo para *makespan*.

Outro ganho para troca de objetivos é quando uma indústria trabalha com um leque grande de produtos similares, porém alguns de alto consumo e outros de consumo esporádico. Por exemplo, em uma fábrica de embreagens, em que a produção de embreagens de veículos populares tem uma demanda muito maior do que veículos de luxo. Então, por serem similares, os produtos podem usar os mesmos recursos produtivos da indústria, porém o sistema de produção pode ser configurado de acordo com o produto.

A ponderação de múltiplas variáveis para cada objetivo torna o sistema de decisão mais completo e flexível. O controle dessas variáveis com o sistema fuzzy otimizado pelos algoritmos genéticos faz com que possamos mapear o processo de tomada de decisões do problema abordado de forma fundamentada e consistente em torno de regras linguísticas.

A modelagem de recursos trouxe uma visão diferenciada para o trabalho, fazendo com que o método avalie situações muitas vezes não mencionadas nos trabalhos da literatura, mais que estão presentes nos sistemas produtivos.

A predição de tarefas conseguiu ganhos significativos no resultado final, principalmente no objetivo *makespan*. A predição contribui quando existe a situação de AGVs ociosos, adiantando o percurso do atendimento e quando um AGV ocupado estará em situação favorável quando terminar a tarefa.

As simulações realizadas validam que o método proposto produziu resultados consistentes. Os dois objetivos foram simulados em três cenários de FMS propostos por outros trabalhos da literatura, um por Benincasa, Morandin e Kato (2003) e outros dois propostos por Reddy e Rao (2006). Os resultados obtidos no objetivo *makespan* foram

comparados com o trabalho de Benincasa, Morandin e Kato (2003) que tinha o mesmo objetivo, em média o método proposto foi 12% melhor. Para testar a eficácia do objetivo *tardiness* foi verificado se o mesmo conseguiu cumprir a programação e fabricar os pedidos nas datas devidas. Nas simulações realizadas o mesmo não produziu nenhum pedido em atraso, os pedidos foram produzidos com adiantamento de 2% a 15%, o que não ocasiona acúmulo de estoque.

7.1 Trabalhos futuros

Como propostas de trabalhos futuros para gerar novas contribuições para o método de despacho proposto, pode-se citar primeiramente a introdução de novas variáveis, como por exemplo, a margem de contribuição dos produtos; testar outras técnicas de geração de conhecimento como redes neurais e redes *Bayesianas* e criar visualização gráfica da simulação.

O desenvolvimento de outras abordagens pode agregar grande valor ao trabalho como:

- Uma política para controle de manutenção e abastecimento dos AGVs, bem como a melhor posição no cenário para isso.
- Um método de roteamento, assim complementando esse trabalho, pois não só decidiria quem o veículo irá atender, mas qual a melhor rota que ele deve tomar. Caminhos de duplo sentido podem melhorar o sistema de transporte, mas pode não haver espaço físico para duplicar todas as rotas. Pode-se verificar os melhores pontos para se duplicar. Com a verificação dos caminhos com a modelagem em redes de Petri, mesmo em caminhos unidirecionais, pode-se verificar se aquele caminho está livre, então em alguns casos usar o caminho na “contramão” ganhando tempo.
- Pode-se usar uma arquitetura descentralizada, utilizando sistemas multiagente e sistemas holônicos para comunicação e colaboração entre os AGVs.
- Considerar vários tipos de veículos, como veículos de carga unitária, veículos de múltiplas cargas, etc.

REFERÊNCIAS BIBLIOGRÁFICAS

AGUIRRE, L. A. **Enciclopédia de automática: controle e automação**. São Paulo: Blucher, 2007. 450 p.

AKBARZADEH-T, M. R.; KUMBLA, K.; TUNSTEL, E.; JAMSHIDI, M. Soft computing for autonomous robotic systems. **Computers and Electrical Engineering**, v. 26, n. 1, p. 5-32, 2000.

ALVAREZ, R. D. R.; ANTUNES JR., J. A. V. Takt-time: conceitos e contextualização dentro do Sistema Toyota de Produção. **Gestão & Produção**, v. 8, p. 1-18, 2001.

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 JR, O.; KATO, E. R. R. Reactive fuzzy dispatching rule for automated guided vehicles. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS, 2003, Washington, DC, United States, p. 4375-4380, 2003.

BERMAN, S.; SCHECHTMAN, E.; EDAN, Y. Evaluation of automatic guided vehicle systems. **Robotics and Computer-Integrated Manufacturing**, v. 25, n. 3, p. 522-528, 2009.

BILGE, U.; TANCHOCO, J. M. A. AGV systems with multi-load carriers: basic issues and potential benefits. **Journal of Manufacturing Systems**, v. 16, n. 3, p. 159-174, 1997.

BISHOP, C. M. **Neural networks for pattern recognition**. England, Oxford University Press, 1996. 482 p.

BOCEWICZ, G.; WOJCIK, R.; BANASZAK, Z. Agvs distributed control subject to imprecise operation times. In: International conference on Agent and multi-agent systems: technologies and applications, Incheon, Korea. **Proceedings**. Springer-Verlag, p. 421-430, 2008.

BONARINI, A.; FIORELLATO, F. Evaluation of genetic-fuzzy systems in the configuration space. In: IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th, 2001, p. 1235-1240, vol.2, 2001.

CARVALHO, A. C. P. L. F.; BRAGA, A. P.; LUDERMIR, T. B. Computação evolutiva. In: REZENDE, S. O. (Ed.). **Sistemas inteligentes: fundamentos e aplicações**. Barueri: Editora Manole, p. 225-248, 2003.

CASTRO, J. P. **Um algoritmo evolucionário para geração de planos de rotas.** Dissertação de Mestrado – Universidade Federal de Santa Catarina, Florianópolis, 1999.

CATARINA, A. S.; BACH, S. L. Estudo do efeito dos parâmetros genéticos sobre a solução otimizada e sobre o tempo de convergência em algoritmos genéticos com codificações binária e real. **Acta Scientiarum Technology**, v.25, n. 2, p. 147-152, 2003.

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.

COOPER, M. G.; VIDAL, J. J. Genetic design of fuzzy controllers: the cart and jointed-pole problem. In: Fuzzy Systems, IEEE World Congress on Computational Intelligence., p. 1332-1337, vol.2, 1994.

CORDÓN, O.; GOMIDE, F.; HERRERA, F.; HOFFMANN, F.; MAGDALENA, L. Ten years of genetic fuzzy systems: current framework and new trends. **Fuzzy Sets and Systems**, v. 141, n. 1, p. 5-31, 2004.

CORDON, O.; HERRERA, F. Three-stage evolutionary process for learning descriptive and approximate fuzzy-logic-controller knowledge bases from examples. **International Journal of Approximate Reasoning**, v. 17, n. 4, p. 369-407, 1997.

DANIELS, M. **Description of distributed systems.** In: St Lucia, Aust. Univ of Queensland, p. 40-55, 1988.

DAQIANG, C. Genetic algorithm for production order picking schedule in an AGV-based FMS. In: APPLIED MECHANICS AND MATERIALS, Macao, China. **Proceedings.** Trans Tech Publications, p. 283-287, 2010.

DAVIS, L. Adapting operator probabilities in genetic algorithms. In: THIRD INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 1989, George Mason University, United States. **Proceedings.** Morgan Kaufmann Publishers, 1989.

DE WOLF, T.; SAMAIEY, G.; HOLVOET, T.; ROOSE, D. Decentralised autonomic computing: Analysing self-organising emergent behaviour using advanced numerical methods. In: SECOND INTERNATIONAL CONFERENCE ON AUTONOMIC COMPUTING (ICAC), 2005, Seattle, United States. **Proceedings.** Institute of Electrical and Electronics Engineers Computer Society, p. 52-63, 2005.

DELGADO, M. R.; VON ZUBEN, F.; GOMIDE, F. Local and global estimation of Takagi-Sugeno consequent parameters in genetic fuzzy systems. In: IFSA World Congress and 20th NAFIPS International Conference, p. 1247-1252, vol.3, 2001.

DUBOIS, D. Fuzzy-set-based method for the optimization of machining operations. In: INTERNATIONAL CONFERENCE ON CYBERNETICS AND SOCIETY, 1981, Atlanta, Ga, USA, p. 331-334, 1981.

DUDA, R. O.; HART, P. E. Pattern classification and scene analysis. **Wiley-Interscience, Information and Control**, 1973. 482 p.

DUINKERKEN, M. B.; OTTJES, J. A.; LODEWIJKS, G. Comparison of routing strategies for AGV systems using simulation. In: WINTER SIMULATION CONFERENCE, 2006, Monterey, CA, United states., p. 1523-1530, 2006.

DYCKHOFF, H.; PEDRYCZ, W. Generalized means as model of compensative connectives. **Fuzzy Sets and Systems**, v. 14, n. 2, p. 143-154, 1984.

DYCUS, J. P. Kansas City: flexible manufacturing system. **Assembly Automation**, v. 18, p. 39 - 47, 1998.

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.

_____. Positioning of automated guided vehicles in a loop layout to improve response time. **European Journal of Operational Research**, v. 71, n. 1, p. 32-44, 1993.

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.

ELMEKKAWY, T. Y.; LIU, S. A new memetic algorithm for optimizing the partitioning problem of tandem AGV systems. **International Journal of Production Economics**, v. 118, n. 2, p. 508-520, 2009.

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.

FERNANDES, F. C. F.; FILHO, M. G. **Planejamento e Controle da Produção: dos fundamentos ao essencial**. São Paulo: Atlas, 2010.

GOETZ JR, W. G.; EGBELU, P. J. Guide path design and location of load pick-up/drop-off points for an automated guided vehicle system. **International Journal of Production Research**, v. 28, n. 5, p. 927-941, 1990.

GOLDBERG, D. E. **Genetic algorithms in search, optimization and machine learning**. Boston: Addison-Wesley Longman Publishing Co., 1989. 372 p.

GROOVER, M. P. **Automation, production systems, and computer integrated manufacturing**. New Jersey: Prentice-Hall, 2003. 808 p.

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.

GUPTA, Y. P.; GOYAL, S. Flexibility of manufacturing systems: concepts and measurements. **European Journal of Operational Research**, v. 43, n. 2, p. 119-135, 1989.

HAMMOND, G. **AGVS at Work**, Automated Guided Vehicle System. Luton: Bartham Press Ltd, 1986.

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.

HERRERA, F.; LOZANO, M.; MORAGA, C. Hybrid distributed real-coded genetic algorithms. In: Proceedings of PPSN, v. 1498, p. 603-612, 1998.

HIDEHIKO, Y. Addition and deletion of agent memories to realize autonomous decentralized manufacturing systems. In: (Eds.). **New Frontiers in Applied Artificial Intelligence**. Springer Berlin / Heidelberg, p. 738-748, 2008. (Lecture Notes in Computer Science; v. 5027)

HOLLAND, J. H. **Adaptation in natural and artificial systems**. University of Michigan Press, 1975.

_____. Genetic algorithms. **Scientific American**, v. 267, n. 1, p. 66-66, 1992.

HOMAYOUNI, S. M.; TANG, S. H.; ISMAIL, N.; ARIFFIN, M. K. A. M.; SAMIN, R. A hybrid genetic-heuristic algorithm for scheduling of automated guided vehicles and quay cranes in automated container terminals. In: COMPUTERS AND INDUSTRIAL ENGINEERING, 2009, Troyes, France. **Proceedings**. Elsevier Ltd, p. 96-101, 2009.

HUTCHINSON, G. K. Flexibility is key to economic feasibility of automating small batch manufacturing. **Industrial Engineering**, v. 21, p. 77-86, 1984.

ICHIHARA, J. A. **Um método de solução heurístico para a programação de edifícios dotados de múltiplos pavimentos-tipo**. Tese de Doutorado – Universidade Federal de Santa Catarina, Florianópolis, 1998.

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, New Jersey: Prentice-Hall, 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.

JONG, K. A. D. **An analysis of the behavior of a class of genetic adaptive systems**. University of Michigan, 1975.

JOSHI, S. B.; SMITH, J. S. **Computer control of flexible manufacturing systems: research and development**. Chapman & Hall, 1994. 427 p.

JUNQUEIRA, F. **Modelagem e Simulação Distribuída de Sistemas**. Tese de Doutorado – Escola politécnica, Universidade de São Paulo, São Paulo, 2006.

KECMAN, V.; ARTHANARI, T.; HADZIC, I. LP and QP based learning from empirical data. In: Washington, DC, United states. **Proceedings**. Institute of Electrical and Electronics Engineers, p. 2451-2455, 2001..

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

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, S. H.; HWANG, H. Adaptive dispatching algorithm for automated guided vehicles based on an evolutionary process. **International Journal of Production Economics**, v. 60, p. 465-472, 1999.

KING, R. E.; WILSON, C. A review of automated guided-vehicle systems design and scheduling. **Production Planning & Control: The Management of Operations**, v. 2, n. 1, p. 44 - 51, 1991.

KITCHENHAM, B. A. Procedures for Performing Systematic reviews. In: Joint Technical Report Software Engineering Group, Keele University, United Kingdom and Empirical Software Engineering, 2004.

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.

KLEIN, C. M.; KIM, J. AGV dispatching. **International Journal of Production Research**, v. 34, n. 1, p. 95-110, 1996.

KLEMENT, E. P.; MESIAR, R.; PAP, E. Triangular norms. **Fuzzy Sets and Systems**, v. 123, n. 3, p. 399-403, 2001.

KLIR, G. J.; YUAN, B. **Fuzzy sets and fuzzy logic: theory and applications**. Prentice Hall, 1995. 574 p.

KULWIEC, R. Trends in automatic guided vehicle systems. **Plant Engineering**, Barrington, Illinois, v. 38, n. 23, p. 66-73, 1984.

LACERDA, E. G. M.; CARVALHO, A. C. P. L. F. Introdução aos algoritmos genéticos. **Anais... XIX Congresso Nacional da Sociedade Brasileira de Computação, Rio de Janeiro**, p. 51-126, p. 1999.

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.

LINDGREN, H. Centralized and decentralized control of AGV S-Advantages and disadvantages. In: International Conference on Automated Guided Vehicle Systems Stockholm, Swed., p. 209-218, 1985.

LUCASIUS, C. B.; KATEMAN, G. Application of genetic algorithms in chemometrics. In: 3RD INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 1989, **Proceedings**. Morgan Kaufmann Publishers, 1989.

LUGER, G. **Artificial intelligence: structures and strategies for complex problem solving**. 5. ed. Addison Wesley Longman, 2004. 928 p.

MAMDANI, E. H. Application of fuzzy logic to approximate reasoning using linguistic synthesis. **IEEE Transactions on Computers**, v. C-26, n. 12, p. 1182-1191, 1977.

MARTINS, P. G.; ALT, P. R. C. **Administração de materiais e recursos patrimoniais**. São Paulo: Saraiva, 2009.

MARTINS, P. G.; LAUGENI, F. P. **Administração da produção**. São paulo: Saraiva, 2005.

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.

MCNEILL, F. M.; THRO, E. **Fuzzy logic: a practical approach**. Academic Press Professional, 1994. 292 p.

MEER, J. R. V. D.; KOSTER, R. D. Using multiple load vehicles for internal transport with batch arrivals of loads. In: GRAZIA SPERANZA, M., STAHLY, N. P. (Ed.). **Advances in distribution logistics**. Berlin: Springer, p. 197-214, 1999.

MORANDIN JR, O.; CASTRO, P. A. D.; KATO, E. R. R.; CAMARGO, H. A. A genetic fuzzy system for defining a reactive dispatching rule for AGVs. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS, 2007, Taipei, Taiwan, p. 56-61, 2007.

MORANDIN, O. J. **Metodologia de modelagem de sistemas automatizados de manufatura, utilizando redes de petri virtual**. Tese de Doutorado – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 1999.

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

MURATA, T. Petri Nets: properties, analysis and applications. In: Of the IEEE, 1989, **Proceedings**. vol. 77, n. 4, 1989.

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.

NICOLETTI, M. D. C.; CAMARGO, H. A. **Fundamentos da teoria de conjuntos fuzzy**. São Carlos: Editora UFSCar, 2004.

OLIVEIRA, J. R. F. **O uso de algoritmos genéticos na decomposição morfológica de operadores invariantes em translação aplicados a imagens digitais**. Tese de Doutorado – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2000.

ORTEGA, N. R. S. **Aplicação da teoria de conjuntos fuzzy a processos de biomedicina**. Tese de Doutorado – Universidade de São Paulo, São Paulo, 2001.

OZDEN, M. Simulation study of multiple-load-carryng automated guided vehicles in a flexible manufacturing system. **International Journal of Production Research**, v. 26, n. 8, p. 1353-1366, 1988.

PAI, M.; MCCULLOCH, M.; GORMAN, J. D.; PAI, N.; ENANORIA, W.; KENNEDY, G.; THARYAN, P.; COLFORD, J. M. Systematic reviews and meta-analyses: An illustrated, step-by-step guide. **The National Medical Journal of India**, v. 17, 2004.

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.

PETRI, C. A. Fundamentals of a theory of asynchronous information flow. In: IFPI Congress 62, 1963, p. 386-390, 1963.

PMBOK, P. M. I. **Um guia do conhecimento em gerenciamento de projetos: guia PMBOK**. Pennsylvania, 2008.

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.

QUINLAN, J. R. Simplifying decision trees. **International Journal of Man-Machine Studies**, v. 27, n. 3, p. 221-234, 1987.

RAJU, K. R.; CHETTY, O. V. K. Design and evaluation of automated guided vehicle systems for flexible manufacturing systems: An extended timed Petri net-based approach. **International Journal of Production Research**, v. 31, n. 5, p. 1069-1096, 1993.

RAOUF, A.; BEN-DAYA, M. **Flexible manufacturing systems: recent developments**. Elsevier, 1995. 310 p.

REDDY, B. S. P.; RAO, C. S. P. A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS. **International Journal of Advanced Manufacturing Technology**, v. 31, n. 5-6, p. 602-613, 2006

SCHNEIDER, F. Automated material handling in FMS and FAS. In: HOLLIER, R. H. (Ed.). **Automated guided vehicle systems**. IFS (Publications) Lts., Springer Verlag, p. 171-192, 1987.

SEIFERT, R. W.; KAY, M. G. Evaluation of AGV routing strategies using hierarchical simulation. In: WINTER SIMULATION CONFERENCE 1995, Arlington, VA, USA., p. 850-856, 1995.

SETHI, A. K.; SETHI, S. P. Flexibility in manufacturing: a survey. **International Journal of Flexible Manufacturing Systems**, v. 2, n. 4, p. 289-328, 1990.

SHEN, Y. C.; KOBZA, J. E. Dispatching-rule-based algorithm for automated guided vehicle systems design. **Production Planning and Control**, v. 9, n. 1, p. 47-59, 1998.

SLACK, N.; CHAMBERS, S.; HARLAND, C.; HARRISON, A.; JOHNSTON, R. **Administração da produção**. São Paulo: Atlas, 1997. 726 p.

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.

SRINIVAS, M.; PATNAIK, L. M. Adaptive probabilities of crossover and mutation in genetic algorithms. **IEEE Transactions on Systems, Man and Cybernetics**, v. 24, n. 4, p. 656-667, 1994.

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.

SULE, R. **Manufacturing facilities**. PWS-KENT Publishing Company, 1988.

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**. Ph. D. Thesis – Purdue University, West Lafayette, Indiana, 1989.

TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its applications to modeling and control. **IEEE Transactions on Systems, Man and Cybernetics**, v. SMC-15, n. 1, p. 116-132, 1985.

TAN, K. K.; TANG, K. Z. Simulation of an evolutionary tuned fuzzy dispatching system for automated guided vehicles. In: WINTER SIMULATION CONFERENCE 2000, Orlando, FL, USA., p. 1339-1343, 2000.

TEMPELMEIER, H.; KUHN, H. **Flexible manufacturing systems**. John Wiley and sons, 1993.

UMASHANKAR, N.; KARTHIK, V. N. Multi-criteria intelligent dispatching control of automated guided vehicles in FMS. In: IEEE CONFERENCE ON CYBERNETICS AND INTELLIGENT SYSTEMS, 2006, Bangkok, Thailand. **Proceedings**. Inst. of Elec. and Elec. Eng. Computer Society, 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.

YAGER, R. R. Ordered weighted averaging aggregation operators in multicriteria decisionmaking. **IEEE Transactions on Systems, Man and Cybernetics**, v. 18, n. 1, p. 183-190, 1988.

YAMAMOTO, H.; YAMADA, T.; KATO, S. AGV Mind Model and Its Usage for Autonomous Decentralized FMS by Change of Mind. In: (Eds.). **Agent and Multi-Agent Systems: Technologies and Applications**. Springer Berlin / Heidelberg, p. 744-753, 2009. (Lecture Notes in Computer Science; v. 5559).

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.

YE, R.; HSU, W.-J.; VEE, V.-Y. Distributed routing and simulation of automated guided vehicles. In: IEEE REGION 10 ANNUAL INTERNATIONAL CONFERENCE, PROCEEDINGS/TENCON, 2000, Kuala Lumpur, Malaysia., p. II-315-II-320, 2000.

ZADEH, L. A. Fuzzy sets. **Information and Control**, v. 8, n. 3, p. 338-353, 1965.

_____. Outline of a new approach to the analysis of complex systems and decision processes. **IEEE Transactions on Systems, Man and Cybernetics**, SMC-3, n. 1, p. 28-44, 1973.

_____. Fuzzy sets as a basis for a theory of possibility. **Fuzzy Sets and Systems**, Berkeley, CA, v. 1, p.3-28, 1978.

ZENG, L.; WANG, H.-P.; JIN, S. Conflict detection of automated guided vehicles. A Petri net approach. **International Journal of Production Research**, v. 29, n. 5, p. 865-879, 1991.

ZHOU, M. **Modeling, simulation, and control of flexible manufacturing systems: a petri net approach**. World Scientific, 1998. 377 p.

ZIMMERMANN, H.; ZYSNO, P. Latent connectives in human decision making. **Fuzzy Sets and Systems**, v. 4, p. 37-51, 1980.

ZIMMERMANN, H.; ZYSNO, P. Latent connectives in human decision making. **Fuzzy Sets and Systems**, v. 4, p. 37-51, 1980.

ZIMMERMANN, J. H. **Fuzzy set theory and its applications**. 2. ed. Boston, USA: Kluwer Academic Publishers, 1991. 399 p.