

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA DE SISTEMAS

PROGRAMAÇÃO DA PRODUÇÃO E DO TRANSPORTE EM SISTEMAS FLEXÍVEIS DE MANUFATURA

Este exemplar corresponde à redação final da tese
defendida por WU XIAO BING
aprovada pela Comissão
Jugadora em 11 08 95.

Vinicius A. Armentano
Orientador

WU XIAO BING

Orientador:

Prof. Dr. Vinicius A. Armentano

Tese apresentada à Faculdade de
Engenharia Elétrica da Universidade
Estadual de Campinas - Unicamp como
parte dos requisitos exigidos para a
obtenção do título de DOUTOR EM
ENGENHARIA ELÉTRICA.

- AGOSTO 1995 -

C m.00081022.0

UNIDADE	BC
N.º CHAMADA:	T/UNICAMP
	W95p
V.	Et.
TIPO DO BC	26-228
PROZ.	433/95
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	84,11,00
DATA	02/12/95
N.º CPD	

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

W95p Wu, Xiao Bing
Programação da produção e do transporte em sistemas flexíveis de manufatura / Xiao Bing Wu.--Campinas, SP: [s.n.], 1995.

Orientador: Vinícius A. Armentano.
Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica.

1. Métodos de simulação. 2. Sistemas flexíveis de fabricação. 3. Planejamento da produção. I. Armentano, Vinícius A.. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica. III. Título.

À minha família

AGRADECIMENTOS

A todos os que colaboraram para a realização deste trabalho. Em especial:

- ao Professor Vinícius A Armentano pela orientação dedicada e ajuda fundamental em todas as etapas desse trabalho;
- ao Professor Hermano M. F. Tavares, ex-diretor da FEE/UNICAMP, pelo apoio e ajuda importante quando cheguei ao Brasil e iniciei o meu curso de doutoramento na FEE/UNICAMP;
- ao Professor Luiz F. Prestes, o diretor da FE/UPF, pelo apoio indispensável na fase de conclusão desta tese;
- aos Professores Paulo França, Chang, Ídala, Moschim da UNICAMP, Professor Jiang do Instituto de Automação de Beijing, e aos Professores Cury e Jean-Marie do LCMI/UFSC, pelo apoio e ajuda;
- aos professores e funcionários do DENSIS/FEE, CPG/FEE e IEL, pelo apoio e ajuda;
- aos amigos e colegas do DENSIS/UNICAMP, LCMI/UFSC e FE/UPF pelo apoio;
- aos amigos Márcio Dutra, Vandale, Adams pela ajuda na correção de português da versão inicial da tese;
- ao Taka, Walcir e Suze, pelo apoio computacional;
- aos amigos Anzhong, Yumei, Binbin, Jinquo, Sang, Yu, Isaac, Ana Luiza, Regina Lúcia, Paulo Valem, Vitor Bravo, Carmen, Miquel, Marcila, Rosyvan, Rogério, Andrea, pelo apoio e amizade;
- a todos os meus amigos em Guiyang e Beijing, China, pela amizade;
- à CAPES pelo apoio financeiro;
- à FE/UPF pelo apoio na autorização de meu afastamento nos últimos quatro meses da fase de conclusão da tese;
- ao LCMI/UFSC pelo apoio na utilização de seus equipamentos computacionais para a tese durante 1992-1994;
- à CGPM/DCE pelo apoio na moradia estudantil;
- à minha família, pelo carinho, paciência e confiança;
- à Xiaomei, minha irmã mais jovem, pelo incentivo e compreensão.

RESUMO

Esta tese trata dos problemas de programação da produção das peças e do transporte destas em sistemas flexíveis de manufatura. A programação da produção das peças consiste em selecionar uma rota de processamento e programar as operações nos centros de trabalho. A programação do transporte envolve a seleção de um AGV, escolha de rota e a programação e controle de transporte das peças. Uma nova metodologia é proposta para modelar e resolver estes problemas. Esta metodologia decompõe um processo complexo num conjunto de processos mais simples e resolve os problemas através da tomada de decisão nos pontos chave dos processos. Um modelo de prioridade linear multiobjetivo é construído para a tomada de decisão. Um algoritmo de busca é desenvolvido para obter soluções de melhor qualidade. Um simulador é desenvolvido para integrar e executar simultaneamente as programações da produção e do transporte para obter um programa global. Resultados computacionais são apresentados e analisados.

Palavras chave: Sistemas flexíveis de manufatura; Programação da produção e do transporte; Modelagem; Modelo de prioridade multiobjetivo; Simulação.

ABSTRACT

This work deals with the problems of machine and AGV scheduling in flexible manufacturing systems. The machine scheduling includes part routing and operation scheduling. The AGV scheduling consists of AGV selection, route selection, scheduling and control of transportation of parts. A new methodology is developed to model and solve these problems. This methodology decomposes a complex process into a set of simple processes and solve the problems through decision-making in the key points of these processes. A multiobjective priority model is built for decision-making. A search algorithm is presented to obtain solutions with better quality. A scheduler is developed to integrate the machine and AGV scheduling in order to get an overall schedule. Computational results are reported and analyzed.

Keywords: Flexible manufacturing systems; Machine and AGV scheduling, Modeling; Multiobjective priority model; Simulation.

ÍNDICE

CAPÍTULO 1 - INTRODUÇÃO	1
1.1 - Sistema Flexível de Manufatura	1
1.2 - Classificação de Sistemas Flexíveis de Manufatura	3
1.3 - Flexibilidade de Sistemas Flexíveis de Manufatura	4
1.4 - Problemas de Sistemas Flexíveis de Manufatura	6
1.5 - Objetivos da Tese	7
1.6 - Organização da Tese	8
CAPÍTULO 2 - MÉTODO ANALÍTICO DE ESTADO	10
2.1 - Introdução	10
2.2 - Um exemplo ilustrativo	12
2.3 - Conceitos Básicos do MAE	16
2.3.1 - Conceitos básicos	16
2.3.2 - Classificação	17
2.4 - Representação Gráfica	19
2.4.1 - Representação gráfica dos estados	19
2.4.2 - Representação gráfica das transições	20
2.5 - Análise do MAE	21
2.5.1 - Definições	21
2.5.2 - Propriedades	22
2.6 - Princípio de Resolução de Problema	23
2.6.1 - Idéia de resolução	23
2.6.2 - Etapas de resolução	24
2.7 - Prioridade Linear Multiobjetivo	24
2.7.1 Definição	25
2.7.2 Construção de um modelo PLM	25

CAPÍTULO 3 - PROGRAMAÇÃO DO SISTEMA DE PRODUÇÃO	33
3.1 - Introdução	33
3.2 - Revisão Bibliográfica	36
3.2.1 - Métodos ótimos	36
3.2.2 - Algoritmos heurísticos	37
3.2.3 - Outros métodos	41
3.3 - Descrição do Problema	42
3.3.1 - Conceitos básicos e hipóteses	42
3.3.2 - Programação do sistema de processamento da produção	44
3.3.3 - Os subproblemas principais	44
3.4 - Os Estados do Processo	45
3.4.1 - Descrição dos estados	45
3.4.2 - Análise dos estados	46
3.5 - As transições do Processo	47
3.6 - Modelo de Prioridade Linear Multiobjetivo	51
3.6.1 - Objetivos do modelo	51
3.6.2 - Regras de programação	53
3.6.3 - Os modelos de prioridade	60
3.7 - Resultados Computacionais	62
3.7.1 - Geração de problemas	62
3.7.2 - Variação de γ	63
3.7.3 - Variações de número de centros alternativos	71
3.7.4 - Variação dos pesos relativos dos objetivos	78
3.7.5 - Variação do número de peças processadas	80
3.7.6 - Algoritmo de busca local	80
CAPÍTULO 4 - PROGRAMAÇÃO DO SISTEMA DE TRANSPORTE DE MATERIAIS	88
4.1 - Introdução	88
4.2 - Revisão Bibliográfica	89
4.3 - Descrição de Problema	94
4.3.1 - Conceitos básicos e hipóteses	95
4.3.2 - Semelhança com um sistema de produção flexível	98
4.3.3 - Definição do problema	99
4.3.4 - Os subproblemas de programação do STM	100

4.4 - O Método Analítico de Estado para o Problema de Programação do Sistema de Transporte	101
4.4.1 - Os estados do processo	101
4.4.2 - Análise do estado	102
4.4.3 - Análise das transições	104
4.5 - Modelo de Prioridade Linear	111
4.6 - Implementação e Resultados	120
4.6.1 - Exemplo de um sistema de transporte de materiais	120
4.6.2 - Dados do sistema	122
4.6.3 - Variação do número de AGVs	124
4.6.4 - Algoritmo de busca	127
CAPÍTULO 5 - UM SIMULADOR PARA PROGRAMAÇÃO DE SISTEMAS FLEXÍVEIS DE MANUFATURA	131
5.1 - Introdução	131
5.2 - Descrição do Simulador	132
5.2.1 - Metodologia utilizado no simulador	132
5.2.2 - Estrutura geral do simulador	132
5.2.3 - Estrutura de dados e informações	135
5.2.4 - Características do simulador	135
5.3 - Simulador de SP	136
5.4 - Simulador de STM	141
5.5 - Integração dos Programadores	145
5.5.1 - Integração	145
5.5.2 - Exemplo	149
CAPÍTULO 6 - CONCLUSÕES	162
APÊNDICE: MÉTODO DA ANÁLISE HIERÁRQUICA	165
REFERÊNCIAS BIBLIOGRÁFICAS	173

CAPÍTULO 1

INTRODUÇÃO

1.1 Sistema Flexível de Manufatura

Um Sistema Flexível de Manufatura (SFM) é um sistema integrado e controlado por computador, composto de dispositivos automatizados de transporte de materiais e máquinas com controle numérico, que pode processar simultaneamente volumes médios de uma variedade de tipos de peças (Stecke 1983). O conceito de SFM nasceu na Inglaterra na década 60 e consistia inicialmente do controle por computador das máquinas e do uso destas 24 horas por dia, 16 horas sem mão de obra (Luggen 1991). O primeiro SFM foi instalado na Inglaterra em 1968 (Kusiak 1986). Deste então, foram implantados vários SFMs no mundo, especialmente no Japão e Estados Unidos.

A ênfase inicial consistia da integração de máquinas ferramenta com o sistema de transporte de materiais e sistema de controle computadorizado (Klahorst 1981), para melhorar a produtividade em sistemas com volume de produção médio. Isto é motivado pela baixa utilização dos equipamentos na manufatura. Por exemplo, nos sistemas convencionais uma peça gasta, em média, somente 5% do tempo total nas máquinas e estas operam somente 6% de tempo total (Luggen 1991). Um SFM pode melhorar estas porcentagens até 90% (Ranky 1983). Um pouco mais tarde,

foi reconhecido que o SFM também é uma forma de manufatura bem adequada à produção com lote pequeno (menor que 50) e alta variedade de produtos. Esta característica corresponde a 75% da produção nos Estados Unidos (Merchant, 1983). O SFM é uma boa forma para resolver os conflitos entre produtividade e flexibilidade, e melhorá-las num ambiente de manufatura com volume de produção pequeno ou médio.

Klahorst (1981) classifica o processo da evolução do SFM em três fases:

- Criação do SFM (1968-1973) com controle do sistema através de dois computadores, um para DNC (direct numerical control) e outro para os sistemas de transporte e informação;
- Expansão do tamanho de SFM (1974-1978) utilizando controlador programável;
- Refinamento do conceito de SFM (1979-1981) utilizando CNC (computer numerical control) para controlar as máquinas individualmente.

Embora já existissem SFMs implantados na indústria desde 1968, o interesse da comunidade acadêmica na pesquisa destes sistemas iniciou-se somente no início dos anos 80 (Kusiak 1986). Este interesse refletiu-se inicialmente pelo primeiro congresso internacional sobre SFM em 1982 e o primeiro livro de texto sobre SFM publicado em 1983 (Ranky 1983). A partir daí, o interesse pelos aspectos de projeto e operação destes sistemas tornou-se cada vez mais crescente, como demonstrado pela publicação de artigos e livros, surgimento de revistas específicas e organização de congressos anuais.

Pela definição clássica mencionada anteriormente, um Sistema Flexível de Manufatura (SFM) é composto de dois sistemas interconectados: um Sistema de Produção (SP) e um Sistema de Transporte de Materiais (STM). O sistema de produção é a parte mais importante num sistema flexível de manufatura, pois trata da produção das peças no chão de fábrica para satisfazer os objetivos desejados. Este sistema contém os seguintes componentes principais :

- Peças;
- Máquinas de controle numérico;
- Ferramentas/fixadores (fixtures);
- Magazines de ferramentas;
- Trocadores automatizados de ferramentas;
- Paletes e centros de paletes;
- Trocadores automatizados de paletes;
- Estações de entrada/saída.

O sistema de transporte de materiais é uma parte do SFM que executa o papel de ligação entre máquinas, entre estações e máquinas, com o objetivo de satisfazer as necessidades de transporte de materiais, principalmente peças, no chão de fábrica. Contém os seguintes componentes principais:

- Cargas unitárias paletizadas ou outros materiais;
- Transportadores (AGVs, esteiras, robôs.);
- Rede de transporte física no chão de fábrica;
- Paletes de transporte;
- Buffers de AGV (Automated Guided Vehicle).

Existem diversos livros que abordam diversos aspectos sobre SFMs. Como exemplo, pode-se citar Ranky (1983), Hartley (1984), Lenz (1989) e Luggen (1991).

1.2 Classificação de Sistemas Flexíveis de Manufatura

Um centro de trabalho é uma das formas básicas para organizar e operar um sistema de manufatura. É composto de uma ou mais máquinas com controle numérico, equipadas com

magazine de ferramentas, trocador automatizado de ferramentas, trocador automatizado de paletes e centro de paletes. As máquinas num mesmo centro têm funções semelhantes.

Um centro de trabalho pode ser considerado como uma "máquina imaginária" com uma capacidade equivalente às máquinas do centro. Do ponto de vista de centro de trabalho, um SFM pode ser classificado em três categorias:

- a) Centro Flexível Singular: há somente um centro de trabalho com ou sem um sistema de transporte de materiais.
- b) Sistema de Fluxo Flexível: existem vários centros de trabalho ligados por um sistema de transporte de materiais. Cada peça é processada pelos centros de trabalho na mesma seqüência.
- c) Sistema Flexível de Produção: há vários centros de trabalho ligados por um sistema de transporte de materiais. As peças têm seqüências de processamento distintas.

Há outras classificações propostas que se baseiam no conceito de células de manufatura e seus componentes tecnológicos (Browne et al. 1984; Kusiak 1985).

1.3 Flexibilidade de Sistemas Flexíveis de Manufatura

Flexibilidade é uma característica fundamental que diferencia um SFM dos sistemas de produção convencionais. A introdução de flexibilidade em manufatura mudou o conceito de produção, trazendo uma nova tecnologia de manufatura com problemas de projeto e operação bem mais complexos.

O principal motivo da introdução da flexibilidade na manufatura é a mudança de atitude dos clientes a partir da década de 70. Esta mudança é traduzida por uma crescente diversidade e

personalização de produtos, rapidez de entrega, qualidade maior e custos mais baixos. Uma consequência da maior diversidade é a diminuição do volume de produção de um dado produto. Além disso, o desenvolvimento tecnológico é cada vez mais rápido e a competição global é cada dia maior. Para poder responder eficientemente às mudanças rápidas de mercado, tecnologia, e para ser competitiva, a produção tem que ser flexível.

A flexibilidade num sistema de manufatura é a habilidade de responder efetivamente às circunstâncias variantes. Segundo Buzacott (1982) a flexibilidade pode ser usada em dois contextos. O primeiro corresponde à capacidade do sistema tomar novas ações para responder a mudanças externas e o segundo corresponde à habilidade do sistema continuar operando de maneira eficaz em situações distintas (robustez a mudanças).

Browne et al. (1984) classificaram a flexibilidade num SFM em 8 categorias. Por exemplo, a flexibilidade de máquina indica a facilidade de fazer as alterações necessárias para produzir um dado conjunto de tipos de peças; a flexibilidade de rota (seqüência de máquinas) representa a habilidade do sistema em manter a produção em caso de falhas e otimizar a utilização das máquinas através da existência de rotas alternativas de processamento.

As flexibilidades num SFM requerem que uma máquina possa processar vários tipos de peças diferentes e uma peça possa ser processada em rotas diferentes no sistema. Estas características distinguem um SFM de um sistema de produção convencional.

Outras definições e medidas de flexibilidade de um SFM que podem ser encontradas, por exemplo, em Buzacott (1982), Kusiak (1985), Gupta e Goyal (1989).

1.4 Problemas dos Sistemas Flexíveis de Manufatura

Os problemas associados a um sistema flexível de manufatura podem ser classificados em duas categorias:

- Problemas de projeto
- Problemas de operação

SFM é um sistema muito complexo tanto para o projeto quanto para a operação. Devido a essa complexidade é comum utilizar métodos hierárquicos. Um exemplo típico é a hierarquia de quatro-níveis sugerida por Kalkunte et al. (1986):

- Análise estratégica e justificativa econômica: onde é feito um planejamento estratégico de longo prazo.
- Projeto de instalação: o planejamento estratégico é convertido num projeto de instalação específico para atingir os objetivos de longo prazo.
- Planejamento de médio prazo: engloba decisões associadas ao plano mestre de produção com horizonte de alguns dias até alguns meses.
- Programação dinâmica: envolve o controle das operações no SFM minuto-a-minuto.

Note-se que os dois primeiros níveis tratam de aspectos de projeto e os dois últimos níveis tratam de aspectos de operação de um SFM. Problemas de projeto são discutidos em Kalkunte et al. (1986), Stecke (1985), Kusiak (1986), Ranky (1983).

Na literatura, os problemas de operação de um SFM são classificados em duas categorias: planejamento e programação*. Stecke (1983) definiu cinco subproblemas de planejamento para um SFM:

* Para evitar o uso da palavra *scheduling*, ela será substituída por programação da produção ou, simplesmente, programação.

- Seleção de tipos de peças: selecionar um subconjunto dos tipos de peças disponíveis para ser processadas imediatamente.
- Agrupamento de máquinas: particionar as máquinas em grupos tal que cada máquina num grupo processe o mesmo conjunto de operações.
- Determinação da taxa de produção: determinar a taxa relativa de produção dos tipos de peças selecionados no primeiro problema.
- Alocação de recurso: alocar os paletes e fixadores para os tipos de peças selecionados.
- Carregamento das máquinas: alocar as operações e as ferramentas necessárias das peças selecionadas aos grupos de máquinas sujeito a restrições tecnológicas e de capacidade.

Kusiak (1986) define que a programação do SFM consiste em dois problemas:

- Programação de máquinas (da produção): consiste em programar as operações das peças nas máquinas para satisfazer os objetivos de produção.
- Programação do sistema de transporte de materiais: aloca os transportadores e programa o transporte das cargas para satisfazer as necessidades de transporte.

Esta tese concentra-se no estudo dos problemas de programação do sistema da produção e do sistema de transporte de materiais baseado em AGV num sistema flexível de manufatura.

1.5 Objetivos da Tese

O problema da programação da produção é um problema combinatorial muito complexo que não pode ser resolvido por métodos ótimos em tempo compatível com as decisões de prazo muito curto que são características deste problema. Este problema é fundamental e é, em geral, resolvido cotidianamente no chão de fábrica por um programador através de sua experiência profissional para obter uma solução razoável. Os limites dos métodos ótimos e as necessidades industriais estimularam as pesquisas na busca de novas metodologias eficientes que incorporem a experiência

de especialistas para resolver o problema de programação, especialmente em ambiente mais complexos como os sistemas flexíveis de manufatura. Nestes sistemas, a programação envolve as atividades de produção e transporte com rotas alternativas das peças e de transporte. Neste trabalho, utiliza-se a palavra rota para denominar uma sequência de máquinas ou um caminho de transporte

Os objetivos principais desta tese são os seguintes:

- Desenvolver uma nova metodologia que possa resolver satisfatoriamente problemas de grande dimensão, com capacidade de incorporar a experiência prática e tratar múltiplos objetivos;
- Estudar os problemas de programação da produção e transporte em um SFM e aplicar o método desenvolvido para resolvê-los;
- Desenvolver um simulador computacional para programar os sistemas de produção e transporte de um SFM de forma individual e integrada

1.6 Organização da Tese

No capítulo 2 é desenvolvido um método novo e geral, chamado Método Analítico de Estado (MAE), para resolver os problemas de programações de produção e transporte em um SFM. Este método decompõe um processo complexo de um sistema em um conjunto de processos bem mais simples e utiliza conceitos de estados e transições de estado para modelar os processos. Um modelo de Prioridade Linear Multiobjetivo (PLM) é introduzido no MAE para definir as transições que requerem tomada de decisão multicritério.

No capítulo 3 o MAE é utilizado para modelar e resolver os problemas de seleção de rotas e programação de operações das peças no Sistema de Produção (SP). São construídos vários modelos de PLM para os objetivos de *makespan* e atraso total das peças. Estes modelos são combinações de fatores associados a regras de despacho. Um simulador é desenvolvido para

resolver a programação do SP. Resultados de simulação comparam os modelos de prioridade com as regras de despacho e mostram que o método é capaz de resolver problemas de grande dimensão em baixo tempo computacional. Um algoritmo de busca local é desenvolvido para obter soluções de melhor qualidade.

O capítulo 4 considera a programação do Sistema de Transporte de Materiais (STM) de um ponto de vista novo. Mostra-se que esta programação é semelhante à programação do sistema de produção e portanto, pode ser tratada de forma análoga à do capítulo 3. O MAE é aplicado para modelar e resolver os problemas de seleção de AGVs e cargas, e programação e controle de transporte das cargas de AGV. São construídos vários modelos de prioridade combinada para o objetivo de atraso total das cargas. Um simulador é utilizado para comparar os modelos de prioridade com as regras de despacho. O mesmo algoritmo de busca local desenvolvido no capítulo 3 é aplicado para obter soluções melhores.

O capítulo 5 desenvolve um simulador com base no MAE que é composto dos simuladores de SP e STM. Através deste simulador, a programação da produção e a programação de transporte são integradas e executadas simultaneamente para obter a programação completa do SFM.

O capítulo 6 apresenta as conclusões e sugestões para pesquisas futuras.

CAPÍTULO 2

MÉTODO ANALÍTICO DE ESTADO

2.1 Introdução

O Método Analítico de Estado (MAE) é uma metodologia proposta neste trabalho para a resolução de problemas de tomada de decisão multiobjetivo em sistemas que variam de forma discreta ao longo do tempo. Ele é aplicado para resolver problemas de programação em um Sistema Flexível de Manufatura (SFM). Este método utiliza os conceitos de estado e transição de estado para modelar um sistema. O estado representa as características invariantes do sistema em um período de tempo e a transição define uma mudança de um estado para outro. A determinação da transição está associada a uma tomada de decisão multiobjetivo. Uma solução para o problema de decisão é obtida quando as transições do estado inicial para o estado final são unicamente definidas.

O MAE foi desenvolvido com base em trabalhos anteriores sobre o modelo de Análise de Fluxo de Operações e a Prioridade Linear Multiobjetivo (PLM). A Análise de Fluxo de Operações é um modelo que descreve e analisa os processos das operações num sistema de manufatura. A PLM é uma prioridade baseada em múltiplos objetivos que pode ser utilizado em tomada de decisão multicritério. Ambos foram aplicados com sucesso para resolver a programação num sistema de

manufatura real do tipo *job shop* (Wu 1987). Wu e Jiang (1987) estenderam a aplicação da PLM aos sistemas de manufatura nos quais as peças processadas podem ser transportadas para as próximas máquinas, mesmo que o lote associado às peças ainda não tenha sido completado. Outra aplicação da PLM no ambiente de manufatura moderna pode ser encontrado em Wu e Armentano (1992b), para resolver a programação do Sistema de Transporte de Materiais baseado em AGV.

A PLM é baseado em alguns princípios do Método de Análise Hierárquica (MAH) proposto por Saaty em seu livro publicado em 1980 e traduzido para português em 1991 (Saaty 1991). O MAH é uma abordagem para a tomada de decisão multicritério com uma estrutura hierárquica. Uma introdução aos conceitos do MAH pode ser encontrada em Saaty (1990) e os fundamentos teóricos sobre o método estão descritos em Saaty (1986). Este método tem sido largamente aplicado a problemas de várias áreas distintas (Vargas 1990). No Brasil, este método foi aplicado em problemas de planejamento pela Embratel (Saaty 1991). Uma descrição sucinta deste método é feito no Apêndice.

O método analítico de estado utiliza a PLM para definir as transições entre estados. As idéias iniciais para o desenvolvimento do método provém das redes de Petri (Peterson 1981), especialmente em suas formas estendidas que envolvem tomada de decisão (Valavanis 1990; Wu e Armentano, 1992a). O método é simples e pode resolver problemas de grande dimensão em tempo computacional baixo. O emprego de PLM permite quantificar os fatos qualitativos e coletar o conhecimento e experiência humana para uma tomada de decisão multiobjetivo.

Este capítulo tem por objetivo apresentar o método de análise de estado. Descrevemos os conceitos básicos, as classificações dos estados e transições, propriedades do MAE e a maneira de resolução de problemas com MAE utilizando a prioridade linear multiobjetivo.

2.2 Um Exemplo Ilustrativo

Usamos um exemplo de manufatura para explicar as idéias básicas do MAE. A Figura 2.1 mostra um sistema simples, que consiste de uma máquina e duas peças. As peças P_1 e P_2 estão disponíveis e esperando para serem processadas na máquina.

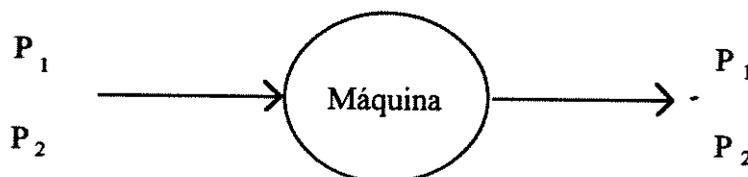


Figura 2.1. Um sistema de manufatura

Os objetos que se movimentam no processo de produção são as peças, chamados *entidades*. Este processo pode ser dividido em três períodos como mostrado na Figura 2.2. Inicialmente as peças estão esperando numa fila para serem processadas. É necessário decidir qual peça deve ser processada em primeiro lugar. A peça selecionada sai da fila e é processada pela máquina. Uma vez terminado o processamento a peça entra numa situação denominada término ao fim do qual a peça sai do sistema. O período de término pode representar, por exemplo, classificação, medição ou descarregamento. As situações “espera”, “processamento” e “término” são chamadas de *estados* do sistema. Cada estado representa uma situação na qual as características do processo não mudam. Por exemplo, a peça permanece no estado de espera durante um certo período até que ocorra uma mudança para o estado de processamento. A mudança de um estado para o outro é chamada *transição*. Todos os estados e transições formam um fluxo de produção completo da peça, ou seja, o *processo* da peça.

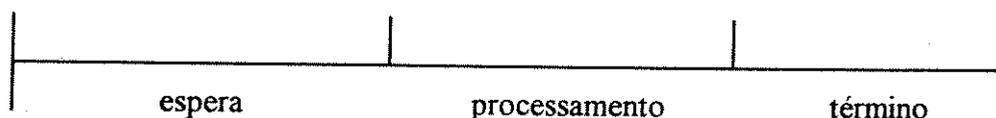
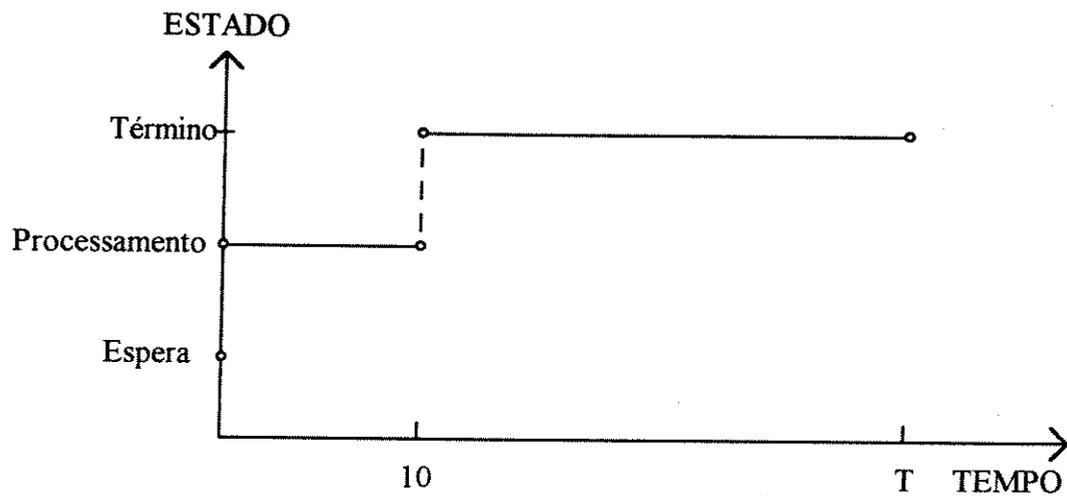


Figura 2.2 Processo produtivo de uma peça

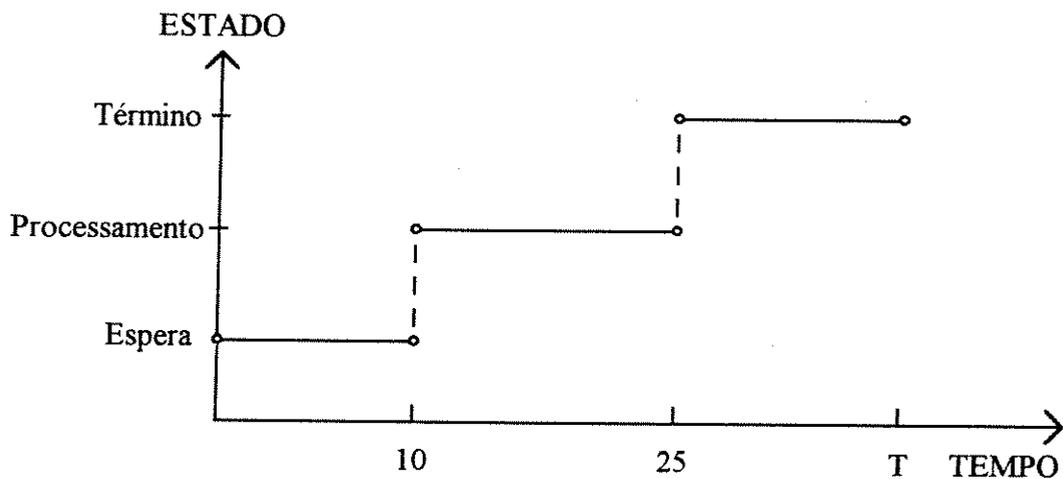
Considere os tempos de processamento das peças mostrados na Tabela 2.1 e suponha que foi decidido processar P_1 em primeiro lugar e P_2 em segundo lugar. A Figura 2.3 mostra os processos das peças e seus estados. Nesta figura, T na Figura 2.3 representa o período de programação considerada.

Tabela 2.1 Tempos de processamento das peças

Peças	tempo de processamento
P_1	10
P_2	15



(a) Processo da peça P_1



(b) Processo da peça P_2

Figura 2.3 Os processos das peças

O processo de produção das peças é um processo de transição de estados. As transições levam do estado de espera para o estado de processamento, e do estado de processamento para o estado de término. A transição do estado de espera para o estado de processamento é definido por uma tomada de decisão, isto é, processar P_1 ou P_2 em primeiro lugar. Sem decisão, a transição não está definida. No entanto, a transição do estado de processamento para o estado de término não requer decisão.

As transições envolvendo tomada de decisão são os pontos chave no processo. Diferentes decisões, em geral, levam a resultados distintos. Para ilustrar, considere as datas de entrega das peças mostradas na Tabela 2.2 e a decisão que gerou os processos mostrados na Figuras 2.3. Como resultado obtém-se um atraso zero para a peça P_1 e um atraso de 5 unidades de tempo para a peça P_2 .

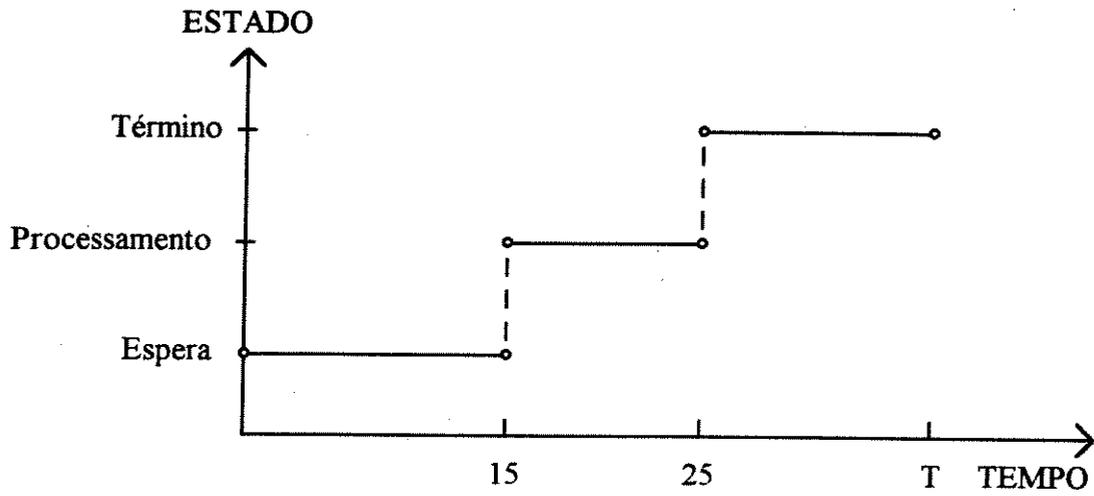
Tabela 2.2 Datas de entrega das peças

Peças	Data de entrega	Tempo de início	Tempo de término
P_1	30	0	10
P_2	20	10	25

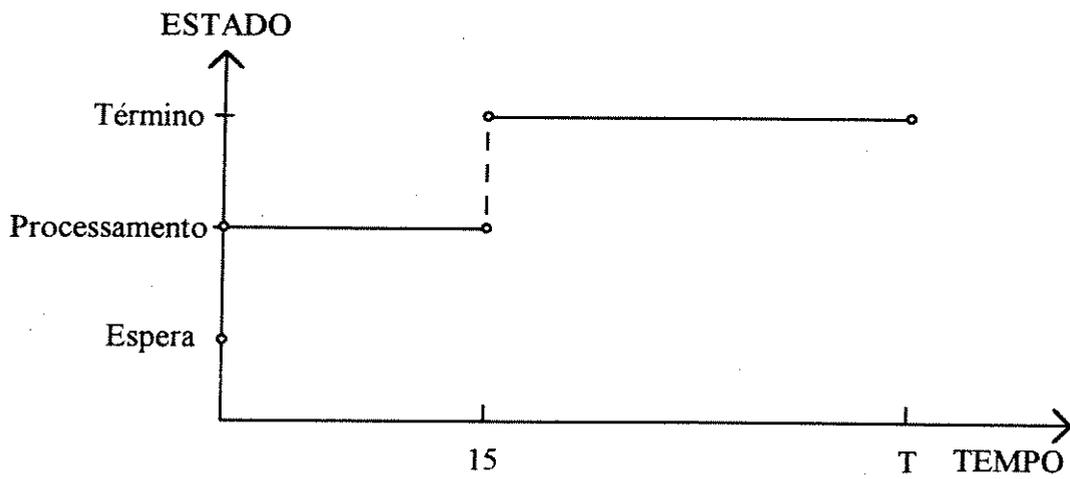
Se decidirmos processar P_2 em primeiro lugar, obtemos os processos mostrados na Figura 2.4. Note que esta decisão leva a um atraso nulo para as duas peças. Se o objetivo é minimizar o atraso total, esta é, portanto, a decisão ótima.

A partir do exemplo, podemos ressaltar os seguintes pontos:

- O processo de produção das peças é composto de estados e evolui de estado para estado através de transições;
- Algumas transições são definidas por tomada de decisão;
- As transições envolvendo tomada de decisão são responsáveis pelos resultados obtidos.



(a) Processo da peça P_1



(b) Processo produtivo da peça P_2

Figura 2.4 Os processos das peças

2.3 Conceitos Básicos do MAE

Observamos várias características do processo de produção através do exemplo apresentado na seção anterior. Na verdade, estas características estão presentes em vários sistemas que variam de forma discreta ao longo do tempo. Nesta seção apresentamos os conceitos básicos do método analítico de estado que é uma metodologia para resolver problemas destes sistemas.

2.3.1 Conceitos básicos

Entidade básica: a menor entidade considerada no sistema. No exemplo apresentado na seção anterior, as peças são as entidades básicas consideradas naquele sistema de produção com uma máquina.

Estado: representa um período do processo no qual suas características não se alteram. Um estado pode ser temporizado ou não. Um estado temporizado é aquele no qual o processo permanece durante um intervalo de tempo diferente de zero. Um estado não temporizado tem duração zero.

Transição de estado: indica a mudança de estado num processo. A transição de estados pode ser ilustrada como na Figura 2.5, no qual α e β são estados. A transição causa uma mudança do estado α para o estado β .

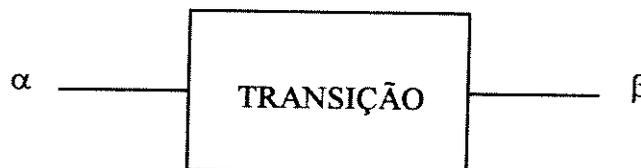


Figura 2.5 Transição de estados

Processo: indica um conjunto de estados e uma seqüência de transições destes estados associados a uma entidade que flui através do sistema.

2.3.2 Classificação

Um processo possui vários tipos de estados e transições. Os estados no MAE podem ser classificados da seguinte forma:

- a) **Estado inicial:** indica o início de um processo. É o primeiro estado do processo e, portanto, não há transição de outro estado para ele.
- b) **Estado de controle:** executa uma função lógica no processo. É um estado não temporizado.

A seleção de máquina é um exemplo de estado de controle. Num SFM há várias máquinas que podem processar uma peça. Por exemplo, na Figura 2.6, há duas máquinas disponíveis M_1 e M_2 que podem processar a peça P. O estado de seleção da máquina é um estado instantâneo que apresenta somente uma função lógica, isto é, a seleção de máquina.

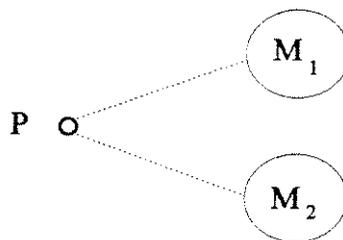


Figura 2.6 Seleção de máquina

- c) **Estado de tempo indeterminado:** é um estado temporizado e a sua duração não é conhecida previamente. Por exemplo, o estado de espera num processo de uma peça é um estado de tempo

indeterminado, porque o tempo que a peça espera depende da decisão de seleção e da situação da máquina.

d) **Estado de tempo determinado**: é um estado temporizado e sua duração é pré-definida. Um exemplo de estado de tempo determinado é o estado de processamento num processo de produção de uma peça. Normalmente o tempo de processamento de peça na máquina é conhecido “a priori”.

e) **Estado final**: O estado final indica o fim de um processo. É o último estado de um processo e, portanto, não existe transição dele para outro estado.

f) **Estado de subconjunto**: consiste de um subconjunto dos estados mencionados para descrever um processo de um subsistema. Este estado tem por objetivo facilitar a modelagem de um sistema composto de subsistemas.

Do ponto de vista de recurso, os estados podem ser classificados em:

a) **Estado sem recurso**: um estado do processo que não utiliza recurso.

b) **Estado com recurso**: um estado do processo no qual uma entidade está utilizando um recurso. Por exemplo, o estado do processamento de uma peça que utiliza o recurso máquina.

c) **Estado de espera de recurso**: um estado do processo no qual uma entidade está esperando um recurso.

d) **Estado de seleção de recurso**: um estado do processo no qual uma entidade pode escolher um recurso dentre os vários disponíveis. Na Figura 2.6, o estado de seleção de máquina da peça P é um exemplo deste tipo de estado.

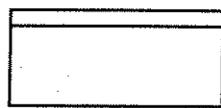
As transições no MAE podem ser de dois tipos:

- a) **Transição direta:** a transição está definida, independente de qualquer decisão.
- b) **Transição indireta:** a transição depende de alguma decisão. Sem ela, a transição não pode ser definida.

2.4 Representação Gráfica

2.4.1 Representação gráfica dos estados

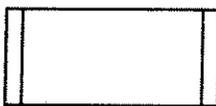
Os estados no MAE podem ser representados pelos símbolos a seguir:



Estado inicial



Estado de controle



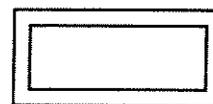
Estado de tempo determinado



Estado de tempo indeterminado



Estado final



Estado de subconjunto

Figura 2.7 Símbolos dos estados

Além disso, se um estado também for de espera de recurso, colocamos três círculos à esquerda do seu símbolo, e, se ele também for um estado de seleção de recurso, colocamos três círculos à direita do seu símbolo. Alguns exemplos destes casos são mostrados na Figura 2.8.



Estado de tempo indeterminado e de espera de recurso



Estado de controle e de seleção de recurso

Figura 2.8 Exemplos dos estados relacionados com recursos

2.4.2 Representação gráfica das transições

Os símbolos gráficos das transições são mostrados na Figura 2.9.



Transição direta



Transição indireta

Fig. 2.9 Símbolos gráficos das transições

2.5 Análise do MAE

Ao utilizar o MAE para resolver um problema, o ponto mais importante é identificar as transições indiretas e defini-las. Para facilitar a identificação das transições, introduzimos e discutimos as propriedades do MAE nesta seção.

2.5.1 Definições

Definição 1. Dado um conjunto finito de estados em um sistema, Σ , e um conjunto finito de transições, T , se $\alpha, \beta \in \Sigma$ e $t \in T$ é uma transição de α para β , então, dizemos que t muda o estado de α para β . A transição entre estados pode ser representada como:

$$t: \alpha \rightarrow \beta$$

O estado α é chamada de estado de entrada da transição t e o estado β é chamada de estado de saída da transição t .

Definição 2. Dada uma transição $t \in T$, os estados $\alpha, \beta \in \Sigma$, $t: \alpha \rightarrow \beta$ é chamada de transição definida se a transição está unicamente determinada.

Note que uma transição direta é uma transição definida. Por outro lado, uma transição indireta necessita de uma decisão para ser unicamente definida.

Definição 3. Um processo é dito definido se todas as transições são definidas.

Definição 4. Dado um conjunto de objetivos $z = \{ z_1, z_2, \dots, z_m \}$, um processo é dito ser definido com múltiplos objetivos se o processo é definido e satisfaz z .

2.5.2 Propriedades

Propriedade 1. Uma transição direta ou indireta pode ser reconhecida pelo seu estado de entrada.

Esta propriedade decorre da necessidade ou não de tomada de decisão no estado de entrada.

Propriedade 2. Se o estado de entrada for um estado de controle sem recurso, então a transição t é uma transição direta.

A duração do estado de controle é zero e como não utiliza recurso, a transição a partir dele está definida diretamente sem necessidade de decisão.

Propriedade 3. Se o estado de entrada for um estado de tempo determinado com ou sem recurso, então a transição é uma transição direta.

O instante final deste estado está determinado quando do seu início. Como o estado não requer recurso, a transição a partir dele está definida diretamente sem nenhuma decisão.

Propriedade 4. Se o estado de entrada for um estado de espera de recurso, então a transição é indireta.

O estado de espera de recurso está associado a uma decisão de alocação de recurso. A transição a partir deste estado necessita de uma decisão para ser definida.

Propriedade 5. Se o estado de entrada for um estado de seleção de recurso, então, a transição é uma transição indireta.

O estado de seleção de recurso está associado a uma decisão de escolha de recurso. A transição a partir deste estado necessita de uma decisão para ser definida.

A partir dessas propriedades, as transições, que são responsáveis pela resolução de um problema, podem ser facilmente identificadas.

2.6 Princípio de Resolução de Problema

2.6.1 Idéia de resolução

Os conceitos fundamentais do método analítico de estado para a resolução de um problema são **decomposição e ponto de decisão**.

O MAE decompõe uma entidade de um sistema nas menores entidades possíveis, que são chamadas de entidades básicas. A partir daí, decompõe-se um processo complexo de um sistema em um conjunto de processos das entidades básicas que são bem mais simples. O MAE não trata diretamente um processo complexo de um sistema, mas concentra-se nos processos de entidades básicas e utiliza os conceitos de estado e transições de estado para modelar estes processos.

Num processo de entidade básica existem alguns pontos de decisão que são responsáveis pela resolução de problema. O MAE concentra-se nestes pontos chave, conhecidos como transições indiretas. Modelos de decisão são construídos para definir as transições indiretas neste processo. Desta forma, são determinadas todas as transições a partir do estado inicial até o estado final e conseqüentemente uma solução para o problema original.

2.6.2 Etapas de resolução

A resolução de um problema através do MAE inicia-se pela análise do sistema e pela definição dos estados e transições, até construir o modelo MAE e obter uma solução por meio de simulação. As etapas de resolução são detalhadas a seguir.

- 1) Analisar o sistema: reconhecer o problema e o processo
- 2) Determinar os estados: analisar e classificar os estados
- 3) Identificar as transições: identificar todas as transições diretas e indiretas
- 4) Construir o modelo de decisão para as transições indiretas
- 5) Simular o processo do sistema e implementá-lo num computador

Em geral, um processo pode ser simulado através de um pacote de simulação. Neste trabalho optou-se por desenvolver um sistema de simulação específico.

2.7 Prioridade Linear Multiobjetivo

A Prioridade Linear Multiobjetivo (PLM) é um modelo de tomada de decisão multiobjetivo utilizado para definir as transições indiretas (Wu 1987). É um modelo linear dos fatores que afetam a tomada de decisão e sua construção é baseada no método de análise hierárquica (MAH) (Saaty 1991).

2.7.1 Definição

Definição 2.7.1. Seja um conjunto de objetivos $z = \{ z_1, z_2, \dots, z_m \}$ e um conjunto de fatores $F = \{ F_1, F_2, \dots, F_n \}$ que influenciam os objetivos. Define-se uma função de Prioridade Linear Multiobjetivo P da seguinte forma:

a) P é uma combinação linear dos fatores F_1, F_2, \dots, F_n , isto é

$$P = \sum_{i=1}^n \alpha_i \times F_i \quad (2.1)$$

$$\text{onde, } \sum_{i=1}^n \alpha_i = 1$$

b) P satisfaz os múltiplos objetivos z_1, z_2, \dots, z_m .

2.7.2 Construção de um modelo PLM

O modelo PLM é um modelo linear que faz uso combinado de informações dos diversos fatores para uma tomada de decisão multiobjetivo. Para construir o modelo (2.1) utilizamos o Método de Análise Hierárquica que é um método para tomada de decisão multicritério e que pode quantificar fatores qualitativos tais como a experiência de especialistas. Um resumo das idéias principais do MAH é apresentado no Apêndice. Nesta seção, utilizamos um exemplo simples para ilustrar as etapas de construção de um modelo PLM.

1) Definir o problema

Considere o exemplo de programação da produção descrito na seção 2.2. Vamos construir um modelo PLM para definir as transições indiretas do problema de programação das peças. Neste modelo vamos considerar os seguintes objetivos:

- Minimizar o atraso total das peças (z_1)
- Minimizar o tempo total de produção ou *makespan* (z_2)

No modelo de prioridade, consideramos os seguintes fatores :

- Tempo de processamento (F_1) : p_i
- Tempo de folga (F_2) : $d_i - t - p_i$
- Data de entrega (F_3) : d_i

onde

i : índice das peças, $i = 1, 2$

p_i : tempo de processamento da peça i

d_i : data de entrega da peça i

t : tempo atual

Pela definição 2.7.1, o modelo PLM pode ser escrito como:

$$P = \alpha_1 \times F_1 + \alpha_2 \times F_2 + \alpha_3 \times F_3 \quad (2.2)$$

$$\text{onde, } \alpha_1 + \alpha_2 + \alpha_3 = 1$$

A prioridade é tanto maior quanto menor for o valor de P . Dá-se preferência às peças com menor tempo de processamento, menor tempo de folga e menor data de entrega com a finalidade de minimizar os objetivos.

2) Estruturar a hierarquia

Para construir um modelo PLM, precisamos estruturar uma hierarquia e definir as relações entre o problema, os objetivos e os fatores. A hierarquia é composta por três níveis:

- Nível de modelo de prioridade multiobjetivo P;
- Nível de objetivos, que representa os objetivos que desejamos considerar no modelo;
- Nível de fatores, que indica os fatores que têm influência significativa nos objetivos.

A Figura 2.10 mostra a estrutura hierárquica e a relação entre seus elementos. Por exemplo, o fator F_1 afeta os objetivos z_1 e z_2 .

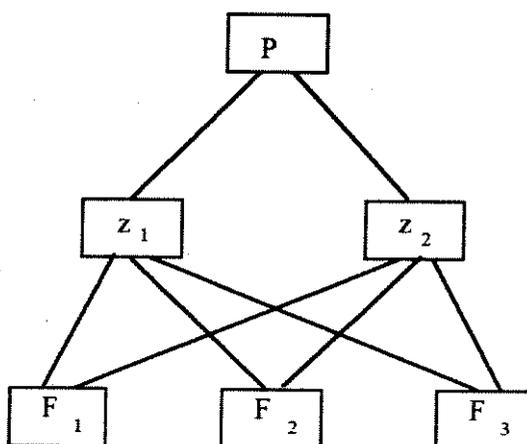


Fig. 2.10 Relações hierárquicas do modelo P

3) Construir as matrizes de comparação

Esta etapa consiste em quantificar a importância relativa dos objetivos, e a importância relativa dos fatores em cada objetivo. Isto é determinado através de comparações entre pares utilizando a escala do método MAH (Apêndice). As comparações podem ser representadas através de uma matriz denominada matriz de comparação.

A matriz de comparação dos objetivos com relação à prioridade P é construída da seguinte maneira. Os objetivos são colocados numa linha e numa coluna. Um elemento a_{ij} da matriz representa o peso ou grau de importância do objetivo z_i em relação do objetivo z_j . Esta matriz é mostrada na Figura 2.11. Por exemplo, o elemento a_{12} da matriz indica que o grau de importância

do objetivo z_1 em relação ao objetivo z_2 é 5. De acordo com a escala de Saaty, isto representa que o objetivo z_1 é de “grande importância” com relação ao objetivo z_2 .

P	z_1	z_2
z_1	1	5
z_2	1/5	1

Figura 2.11 Matriz de comparação dos objetivos

Da mesma forma, constrói-se as matrizes de comparação dos fatores F_1 , F_2 e F_3 com a relação aos objetivos z_1 e z_2 , como mostrado na Figura 2.12.

z_1	F_1	F_2	F_3
F_1	1	1/7	1/5
F_2	7	1	3
F_3	5	1/3	1

(a)

z_2	F_1	F_2	F_3
F_1	1	7	9
F_2	1/7	1	3
F_3	1/9	1/3	1

(b)

Figura 2.12 Matriz de comparação dos fatores em relação aos objetivos

4) Determinar os pesos relativos

De acordo com o método de análise hierárquica, os pesos dos objetivos e dos fatores podem ser obtidos através dos autovetores associados aos autovalores máximos das matrizes de comparação (Apêndice). Estes autovetores são obtidos através de um software para o MAH desenvolvido no Instituto de Pesquisa em Automação de Pequim que utiliza o método da potência apresentado em Saaty (1991).

Os pesos relativos dos objetivos w_1 e w_2 , os pesos relativos dos fatores $w_{F_1}^1$, $w_{F_2}^1$, $w_{F_3}^1$ referentes ao objetivo z_1 e os pesos relativos dos fatores $w_{F_1}^2$, $w_{F_2}^2$, $w_{F_3}^2$ referentes ao objetivo z_2 são mostrados na Tabela 2.3 para o exemplo considerado.

Tabela 2.3 Valores de pesos

peso	w_1	w_2	$w_{F_1}^1$	$w_{F_2}^1$	$w_{F_3}^1$	$w_{F_1}^2$	$w_{F_2}^2$	$w_{F_3}^2$
valor	0.8333	0.1667	0.0719	0.6491	0.2790	0.7854	0.1488	0.0658

Os valores dos parâmetro α_1 , α_2 e α_3 do modelo de prioridade P podem ser obtidos a partir dos pesos relativos como mostrado a seguir:

$$\alpha_1 = w_1 \times w_{F_1}^1 + w_2 \times w_{F_1}^2 = 0.8333 \times 0.0719 + 0.1667 \times 0.7854 = 0.19$$

$$\alpha_2 = w_1 \times w_{F_2}^1 + w_2 \times w_{F_2}^2 = 0.8333 \times 0.6491 + 0.1667 \times 0.1488 = 0.57$$

$$\alpha_3 = w_1 \times w_{F_3}^1 + w_2 \times w_{F_3}^2 = 0.8333 \times 0.2790 + 0.1667 \times 0.0658 = 0.24$$

Note que cada parâmetro α_i representa a contribuição do fator F_i ao modelo P através de sua influência sobre os objetivos e a influência dos objetivos em P.

Obtemos então o modelo, para decidir a seqüência de processamento das peças e programar o processo de produção, como seguinte:

$$P = 0.19 \times F_1 + 0.57 \times F_2 + 0.24 \times F_3 \quad (2.3)$$

O programa assim obtido é avaliado de acordo com o índice multiobjetivo IM:

$$IM = 0,83 \times z_1 + 0,17 \times z_2 \quad (2.4)$$

Onde 0,83 e 0,17 são os valores de w_1 , w_2 na Tabela 2.3.

A Tabela 2.4 mostra a forma geral para determinar os parâmetros α_i através dos pesos relativos.

Tabela 2.4 Síntese dos parâmetros

z	z_1	z_2	z_m	
F	w_1	w_2	w_m	
F_1	w_{F1}^1	w_{F1}^2	w_{F1}^m	$\alpha_1 = \sum_{i=1}^m w_i * w_{F1}^i$
F_2	w_{F2}^1	w_{F2}^2	w_{F2}^m	$\alpha_2 = \sum_{i=1}^m w_i * w_{F2}^i$
.....
F_n	w_{Fn}^1	w_{Fn}^2	w_{Fn}^m	$\alpha_n = \sum_{i=1}^m w_i * w_{Fn}^i$

Observe que :

$w = \{ w_1, w_2, \dots, w_m \}$ é o autovetor correspondente ao maior autovalor da matriz de comparação dos objetivos z_1, z_2, \dots, z_m , e chamado de autovetor principal

$w_{F^i} = \{ w_{F1}^i, w_{F2}^i, \dots, w_{Fn}^i \}$ é o autovetor correspondente ao maior autovalor da matriz de comparação dos fatores F_1, F_2, \dots, F_n , com relação ao objetivo z_i

$\alpha = \{ \alpha_1, \alpha_2, \dots, \alpha_n \}$ são os parâmetros tais que

$$\alpha_i = \sum_{j=1}^n w_j \times w_{F_i}^j \quad e$$

$$\sum_{i=1}^n \alpha_i = \sum_{i=1}^n \sum_{j=1}^m w_j \times w_{F_i}^j = 1$$

5) Verificar a consistência

É interessante verificar se o conjunto de comparações entre pares é consistente, isto é, se a propriedade de transitividade das comparações é razoavelmente satisfeita. Para tal, Saaty (1991) propôs um índice de consistência (IC) e uma razão de consistência (RC) para cada matriz de comparação. A partir destes índices pode-se calcular um índice de consistência e uma razão de consistência para o modelo hierárquico. Estes índices são utilizados para medir o grau de inconsistência das comparações e são apresentados no Apêndice. Um modelo é considerado aceitável quando a razão de consistência é menor ou igual a 0,1.

Para as matrizes das figuras 2.11, 2.12 (a, b), os índices de consistência são, respectivamente, zero, 0,0324 e 0,0401. A razão de consistência do modelo é igual a 0,0337.

A Tabela 2.5 mostra os valores dos fatores no instante zero (instante de decisão a ser tomada). Da expressão (2.3) obtemos os seguintes valores de prioridade das peças P_1 e P_2 .

$$\begin{aligned} P(P_1) &= 0.19 \times F_1 + 0.57 \times F_2 + 0.24 \times F_3 \\ &= 0.19 \times 10 + 0.57 \times 20 + 0.24 \times 30 \\ &= 20.5 \end{aligned}$$

$$\begin{aligned} P(P_2) &= 0.19 \times F_1 + 0.57 \times F_2 + 0.24 \times F_3 \\ &= 0.19 \times 15 + 0.57 \times 5 + 0.24 \times 20 \\ &= 10.5 \end{aligned}$$

Tabela 2.5. Os dados das peças

Fatores	Dados das peças	
	Peça P_1	Peça P_2
Tempo de Processamento (F_1)	10	15
Tempo de Folga (F_2)	20	5
Data de entrega (F_3)	30	20

Como $P(P_2) < P(P_1)$, a prioridade de P_2 é maior que P_1 e, portanto, P_2 deve ser processado em primeiro lugar. Como resultado obtemos um atraso total (\bar{z}_1) nulo e um makespan (\bar{z}_2) igual a 30. A partir da expressão (2.4), o índice multiobjetivo é dado por

$$IM = 0.83 \times \bar{z}_1 + 0.17 \times \bar{z}_2 = 0.83 \times 0 + 0.17 \times 30 = 5,1$$

Observa-se que neste caso, como existe somente uma máquina, o *makespan* é constante, independente da seqüência de produção das peças. Portanto, a satisfação dos objetivos reduz-se a minimizar o atraso total.

CAPÍTULO 3

PROGRAMAÇÃO DO SISTEMA DE PRODUÇÃO

3.1 Introdução

A programação em um sistema de produção consiste na alocação eficiente de máquinas às peças num período de tempo. Portanto, a determinação de um programa de produção é evidentemente um processo de tomada de decisão.

Os primeiros resultados na área de programação surgiram na década de 50 coincidentemente com o início do desenvolvimento da área de Pesquisa Operacional. Smith (1956) e Jackson (1955) demonstraram resultados básicos para uma máquina tais como a otimalidade das regras SPT (Shortest Processing Time) e EDD (Earliest Due Date) para critérios de fluxo médio e máximo atraso, respectivamente. Um outro resultado importante deste período deve-se a Johnson (1954) que propôs um algoritmo ótimo para o problema de minimizar o *makespan* em duas máquinas. A partir da década de 60 a área teve um grande desenvolvimento em termos de teoria, técnicas e modelos. De um modo geral, esta área é fragmentada de acordo com o ambiente de manufatura, isto é, a teoria e os algoritmos são específicos para o ambiente considerado. Por exemplo, os problemas em ambientes de *flow shop* e *job shop* são resolvidos por métodos especializados com poucos elementos em comum.

Dentre os livros textos clássicos que tratam de problemas da programação de produção, os mais citadas na literatura são Conway et al. (1967), Baker (1974) e French (1982). Recentemente, foram publicados três livros que abordam aspectos distintos da área, a saber, Blazewicz et al. (1993), Morton e Pentico (1993), e Pinedo (1995).

Os problemas de programação da produção são problemas da área de otimização combinatória. No início da década de 70, Cook (1971) desenvolveu a base da teoria de complexidade computacional para problemas de otimização combinatória considerados “intratáveis”. Um problema de otimização é considerado intratável se não existe algoritmo em tempo polinomial para sua solução, isto é, os algoritmos existentes tem complexidade exponencial com o tamanho do problema. Esta categoria de problemas formam a classe de problemas NP-hard, cuja definição formal pode ser encontrada em Garey e Johnson (1979). Subseqüentemente, Karp (1972) e Garey e Johnson (1979) demonstraram que vários problemas de programação pertencem a esta classe. A partir da década de 80 demonstrou-se que a grande maioria dos problemas de programação da produção são intratáveis (Lawler et al. 1993; Blazewicz et al. 1993).

O desenvolvimento desta teoria levou os pesquisadores a dirigir a atenção para métodos heurísticos de resolução para obter uma solução de boa qualidade em tempo computacional baixo. Isto é altamente desejável em problema de programação de produção, pois o horizonte de tomada de decisão nestes problemas é curto (tipicamente um dia, ou um turno).

O problema de programação da produção em um sistema flexível de manufatura pode ser visto como uma extensão do problema clássico de job shop. No problema de job shop clássico, uma peça consiste de um número de operações ordenadas e executadas em máquinas distintas e determinadas. Em outras palavras, a seqüência das máquinas a ser percorrida pela peça (rota da peça) é única. Em um sistema flexível de manufatura, uma operação pode ser processada em várias máquinas e, portanto, existem várias rotas, o que torna o problema mais complexo. Deve-se ressaltar que o problema de job shop clássico é considerado um dos mais complexos da área de otimização combinatória (Lawler et al. 1993).

Na prática e na literatura acadêmica, o problema de job shop é frequentemente abordado através de regras de despacho que estabelecem uma prioridade para cada peça esperando em uma fila para ser processada em uma máquina. É importante ressaltar a contribuição pioneira de Conway nesta abordagem. No período de 1960 a 1965 publicou diversos artigos (ver Conway, 1967) onde analisa o comportamento de várias regras de despacho com relação a diversos objetivos. Deste então o número de regras de despacho propostas tem crescido e o último levantamento destas regras (Panwalkar e Iskander 1977) lista mais de 100 regras. Regras de despacho também tem sido utilizadas em sistemas flexíveis de manufatura.

Este capítulo aborda o problema da programação de produção de peças em um sistema flexível de manufatura através do método analítico de estado proposto no capítulo 2. São construídos modelos lineares de prioridade (combinação linear de regras de despacho) para satisfazer simultaneamente os objetivos de *makespan* e atraso. Obtemos assim um programa de produção de peças que especifica em quais máquinas as operações devem ser executadas e quando. O programa de produção obtido não é obviamente *on-line*. Stecke (1985) destaca que em um sistema flexível de manufatura seria mais apropriado tomar decisões de programação em termos de estado atual do sistema. Assim, o programa obtido neste trabalho pode ser visto como um programa ideal que deve ser revisto periodicamente para se adequar às modificações no sistema (por exemplo, quebra de máquina). Uma versão preliminar de resolução do problema mencionado acima foi apresentada em Wu e Armentano (1994).

Este capítulo está organizado da seguinte forma. A seção 3.2 contém uma revisão bibliográfica. A seção 3.3 descreve o problema da programação de SP. Na seção 3.4 são definidos e analisados os estados das operações das peças. Na seção 3.5 são analisadas as transições de estados e um modelo MAE é construído para resolver o problema. A seção 3.6 descreve modelos de prioridade PLM para tomada de decisões e definição das transições indiretas. A seção 3.7 apresenta e analisa os resultados computacionais, e desenvolve um algoritmo de busca local para obter soluções de melhor qualidade.

3.2 Revisão Bibliográfica

Na literatura existem várias técnicas para tratar o problema de programação da produção. Nesta seção, enfatiza-se os trabalhos que utilizam métodos de otimização, especialmente os procedimentos heurísticos. Outras técnicas são citadas de forma sucinta.

3.2.1 Métodos ótimos

Devido a complexidade do problema de programação em um SFM, existem poucos trabalhos que propõem métodos para encontrar a solução ótima com relação a algum objetivo.

Hutchison et al. (1991) estudaram o problema de programação das peças e de carregamento de máquinas (alocação de operações a máquinas) num ambiente de SFM com o objetivo de minimizar o *makespan*. O problema é formulado através de um modelo de programação inteira 0-1 e resolvido por um algoritmo *branch and bound*. Além disso, é proposto uma decomposição natural do problema em um subproblema de carregamento e um subproblema de programação, resultando numa solução sub-ótima. Os autores comparam a solução ótima, a solução sub-ótima e a solução obtida pela regra de despacho SPT para problemas pequenos com 8 peças e 2 a 5 operações por peça. Os resultados deste trabalho mostram que a solução sub-ótima é de boa qualidade, e a solução obtida pela regra de despacho é de má qualidade.

Greene e Sadowski (1986) utilizam um modelo de programação inteira mista para o problema de programação das peças e de carregamento de máquinas num ambiente de múltiplas células flexíveis de manufatura. O objetivo é minimizar o *makespan*. Os autores mostram que num sistema com 5 células, 25 peças e 3 operações por peça, o número de variáveis situa-se na faixa de 550 até 14600 e o número de restrições de 950 até 28950. Isto mostra que o modelo é intratável em sistemas reais. Os autores não propõem nenhum método ou algoritmo para resolver o modelo,

mas destacam que este pode ajudar na elaboração de métodos heurísticos para a resolução do problema.

3.2.2 Algoritmos heurísticos

Devido à complexidade dos problemas de programação da produção e do curto intervalo de tempo disponível para decisões, o enfoque mais apropriado consiste na utilização de métodos heurísticos. Uma heurística é boa se possui algumas características desejáveis, tais como:

- simplicidade de implementação
- fácil de ser entendida pela gerência
- permite interação com o usuário
- fornece solução de boa qualidade em baixo tempo computacional

A maior parte das heurísticas encontradas em problemas de programação da produção podem ser classificadas em duas categorias: heurísticas de construção e heurísticas de busca em vizinhança.

Heurísticas de construção geram uma solução factível para o problema através da inclusão seqüencial de componentes individuais. Um exemplo clássico desta categoria de heurísticas é a heurística que utiliza regras de despacho e constrói um programa de produção alocando máquinas a peças de forma seqüencial, ao longo do tempo.

Heurísticas de busca em vizinhança partem de uma solução factível e procuram uma solução melhor dentro de uma vizinhança definida. Uma heurística clássica de busca em vizinhança pára quando não existe nenhum vizinho melhor que a solução corrente (ótimo local).

Heurísticas de construção

As heurísticas de construção encontradas na literatura para problemas de programação de produção em sistemas job shop e sistemas flexíveis de manufatura são predominantemente baseadas em regras de despacho ou prioridade.

Como ressaltado anteriormente, Conway (1967) foi o autor que iniciou estudos de simulação de heurísticas construtivas baseadas em regras de prioridade para sistemas job shop. Em uma série de artigos, foram avaliados os desempenhos de diversas regras para objetivos tais como makespan, atraso das peças e trabalho em processamento (work-in-process).

As regras investigadas por Conway são as regras mais simples e conhecidas, tais como SPT, EDD e outras utilizadas neste trabalho. A partir destas regras mais tradicionais houve uma proliferação de regras de prioridade propostas na literatura até os tempos atuais. Pode-se afirmar que o número de regras de despacho existentes está em torno de 200. Qual o motivo de tamanho interesse por regras de prioridade? A resposta à esta questão é fácil. A programação da produção em um sistema job shop e em um sistema flexível de manufatura é um problema extremamente complexo e as heurísticas baseadas em regras de despacho fornecem uma solução rapidamente, além de ser fácil explicar a motivação de cada regra para o usuário e a gerência. O grande problema com regras de prioridade é a imprevisibilidade do comportamento das mesmas, isto é, a qualidade da solução das regras varia bastante ao se variar os parâmetros do sistemas tais como número de peças, número de máquinas, tempo de processamento do operações e data de entrega de peças. Apesar desta desvantagem, regras de prioridade constituem o mecanismo de decisão para programação da produção mais utilizado na prática. Portanto, para um dado ambiente de manufatura deve-se testar entre várias regras e selecionar a melhor.

O primeiro trabalho de programação da produção em um sistema flexível de manufatura publicado na literatura é devido a Stecke e Solberg (1981). O sistema analisado pertence à Caterpillar com 9 máquinas, um centro de inspeção e dois carros de transporte numa trilha fixa. Os autores testaram 5 políticas de roteamento combinadas com 16 regras de prioridade e o critério utilizado para

avaliar o desempenho foi o número de peças produzido num dado período. Foram obtidos resultados superiores em relação à política de programação praticada pela Caterpillar na ocasião.

Montazeri e Van Wassenhove (1990) analisaram um sistema flexível de manufatura na Bélgica com 3 centros de máquinas (2 centros com 2 máquinas e um centro com 1 máquina), 11 buffers para peças em processamento e três carros de transporte. A produção projetada para a semana era de 199 peças. Os autores testaram 14 regras de prioridade combinadas com a política de roteamento de enviar uma peça para a máquina menos carregada. Um simulador para sistemas flexíveis de manufatura foi construído na tese de doutorado de Montazeri e utilizado no trabalho citado para avaliar o desempenho das regras. Os autores utilizaram vários critérios de desempenho, incluindo tempo de espera por peça, utilização de máquina, utilização de buffer, utilização de carro de transporte e makespan, e concluíram que não existe uma regra dominante para todos os critérios considerados.

De nosso conhecimento estes são os trabalhos mais importantes e conhecidos que abordam a programação da produção em sistemas flexíveis com resultados quantitativos. Deve-se ressaltar que nenhum dos trabalhos considera o critério de data de entrega das peças. Além disso, o trabalho de Montazeri e Van Wassenhove aponta na direção da necessidade de um modelo de prioridade que contemple vários objetivos simultaneamente.

Encontramos ainda duas heurísticas construtivas não baseadas em regras de despacho. Nasr e Elsayed (1990) propuseram uma heurística com o objetivo de minimizar o tempo de fluxo médio das peças. A cada instante de tempo onde existem operações disponíveis para processamento, aloca-se cada operação à máquina mais rápida. Em geral, isto resulta em um conflito, isto é duas ou mais operações alocadas à uma mesma máquina num mesmo intervalo de tempo. Para evitar este conflito, resolve-se um problema de designação (assignment) das operações conflitantes às máquinas em que as mesmas podem ser executadas. O custo de designação de uma operação de uma peça a uma máquina é um limitante inferior do tempo restante para completar o processamento da peça. Os autores reportam resultados para 10 peças, 10 máquinas e 10 operações por peça mas não mencionam o tempo computacional gasto pela heurística.

Chang et al. (1989) desenvolveram uma heurística com o objetivo de minimizar o makespan. A heurística utiliza um método denominado “beam search” que delimita um feixe de nós numa árvore de busca. Cada nó deste feixe corresponde a um programa parcial K . Seja $U(K)$ o conjunto de operações que podem ser executadas imediatamente após o programa K e seja K_u o programa K incluindo u , $u \in U(K)$. Para cada K_u fixado obtém-se um programa completo utilizando a regra SPT. Para a primeira operação no caminho crítico do programa completo (excluindo as operações de K_u) avalia-se o efeito de alocar esta operação a outra máquina e de resequenciar esta operação na mesma máquina (alocada pela regra SPT). O número de programas parciais guardados β é fixado a priori e correspondem aos programas parciais que geram programas completos pela regra SPT com menor makespan. Os autores realizam testes computacionais em problemas com 10 peças, 10 máquinas e 10 operações por peça. Os valores de β testados variam de 1 a 5. Os autores comparam os resultados de heurística com resultados fornecidos por algumas regras de despacho e concluem que a heurística supera a melhor regra (SPT) com ganho de 16% a 50%. O tempo computacional reportado é da ordem de 6β a 8β segundos.

Heurísticas de busca em vizinhança

O único método heurístico de busca em vizinhança conhecido na literatura para programação da produção em um sistema flexível de manufatura é devido a Brandimarte (1993). O objetivo consiste em minimizar o makespan e o enfoque adotado consiste em particionar o problema em dois subproblemas:

- um problema de roteamento que aloca operações a máquinas
- um problema de programação das operações das peças nas máquinas, que é essencialmente um problema de job shop

A solução inicial é fornecida por duas regras de despacho. A seguir aplica-se busca tabu em cada subproblema. Busca tabu é em termos simples, uma busca em vizinhança que transcende mínimos locais. O autor testa seu método em problemas envolvendo até 30 peças, 15 máquinas e 12 operações por peça e obtém reduções no makespan em relação à melhor regra de despacho que

variam de 2% a 47% com tempo computacional variando de segundos a 12 minutos para os problemas maiores em um computador PC-386.

Para o problema de programação em job shop, deve-se ressaltar a existência de vários trabalhos na literatura que utilizam procedimentos de busca no espaço de solução que superam ótimos locais, tais como busca tabu e *simulated annealing* (Aartz et al. 1994; Dell'Armico e Trubian 1993; Van Laarhoven et al. 1992). Em todos estes trabalhos o objetivo considerado é o makespan. É interessante também notar que nenhum trabalho aborda o objetivo atraso total em relação a datas de entrega.

3.2.3 Outros métodos

Na literatura, existem outras técnicas que são aplicadas para resolver os problemas de programação da produção, como Redes de Petri, Inteligência Artificial, Simulação e Redes de Filas. Uma revisão simplificada destes métodos está apresentada em seguir (nesta seção).

Rede de Petri é um modelo matemático, representado por um grafo, adequado para a descrição de sistemas concorrentes com componentes que necessitam de sincronização e compartilham recursos limitados. A descrição de sistemas através de Rede de Petri concentra-se nos conceitos de eventos e condições. Os eventos são as ações que ocorrem no sistema e as condições descrevem o estado do sistema. Uma introdução geral às redes de Petri pode ser encontrada no livro de Peterson (1981). A partir dos anos 80, as redes de Petri foram aplicadas à modelagem de sistemas de manufatura (Dubois e Stecke 1983; Valavanis 1990) e especificamente em programação da produção (Wu e Armentano 1992a; Lee e DiCesare 1994).

Sistema baseado em conhecimento é uma metodologia para resolver o problema de programação da produção através do conhecimento humano. Em geral, estes sistemas contêm três componentes principais: 1) banco de dados, que contém uma descrição simbólica do sistema real; 2) banco de conhecimento, que contém os conhecimentos específicos, geralmente na forma de lógica de

primeira ordem, regras de produção ou *frames*; 3) motor de inferência, que é responsável pelo emprego de conhecimento para resolver o problema (Shaw e Whinston 1986, Kusiak 1990). O primeiro trabalho realizado em programação de produção foi desenvolvido na tese de doutorado de M. S. Fox em 1983, que propôs o sistema ISIS (Intelligent Scheduling and Intelligent Systems) para programar um sistema de manufatura do tipo *job shop*. A partir daí, a metodologia foi utilizada na programação da produção de sistemas flexíveis de manufatura (Shaw 1988; Kusiak 1990; Kusiak e Chen 1988; Wu e Wysk 1990; Subramanyam e Askin 1986).

A simulação é uma técnica muito utilizada para modelar e avaliar o desempenho de sistemas de manufatura. O livro de Carrie (1988) trata especificamente da simulação em sistemas flexíveis de manufatura. Existem vários trabalhos que estudam a programação da produção em um ambiente flexível de manufatura através de simulação (O'Gorman et al. 1986; Wu e Wysk 1989; Montazeri e Van Wassenhove 1990; Bengu 1994). Vários pacotes gerais de simulação e simuladores específicos disponíveis no mercado são apresentados por Law e Haider (1989), Bose (1988).

3.3 Descrição do Problema

3.3.1 Conceitos básicos e hipóteses

Conceitos básicos

- **Operação:** consiste das atividades necessárias para processar uma peça numa máquina. Estas atividades são :
 - i) processamento da peça na máquina;
 - ii) carregamento e descarregamento da peça e ferramentas na máquina;
 - iii) período de espera da peça na máquina;
 - iv) interprocessamento entre duas operações adjacentes.

Para facilitar a programação, consideramos o período de espera da peça na máquina e o interprocessamento como parte da operação seguinte. Um exemplo de uma peça com duas operações é mostrado na Figura 3.1. O interprocessamento inclui as atividades entre as máquinas que executam duas operações adjacentes, tais como transporte, carregamento e descarregamento da peça em um AGV (Automated Guided Vehicle).

- **Centro de trabalho:** é um conjunto de máquinas formado por uma ou mais máquinas que têm funções de produção idênticas;
- **Lista de rotas alternativas:** é uma lista de todas as rotas possíveis para cada operação das peças;

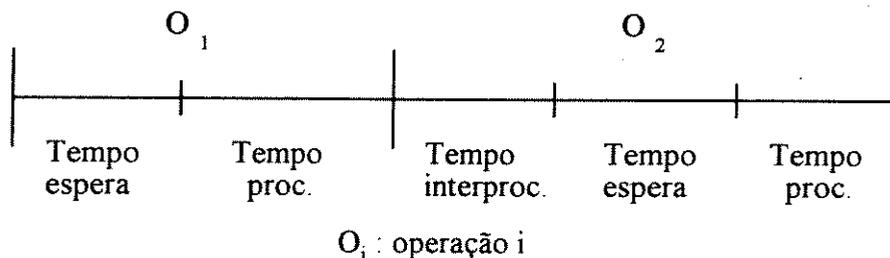


Fig. 3.1. Um exemplo de uma peça com duas operações

Hipóteses principais

- Cada operação de uma peça pode ser processada em vários centros de trabalho alternativos, os quais são descritos na lista de rotas alternativas;
- Quando uma operação começa a ser processada numa máquina, ela prossegue sem parar até o seu término;
- Cada máquina processa somente uma operação num dado instante;
- A troca de ferramentas é feita automaticamente. As ferramentas nos *magazines* dos centros de trabalho são suficientes para processar a operação selecionada;
- O tempo de interprocessamento é conhecido;

- O tempo de preparação é independente da seqüência de operações, e portanto, é adicionado ao tempo de processamento.

3.3.2 Programação do sistema de produção

A programação do Sistema de Produção (SP) envolve todos os elementos ligados diretamente ao processamento das operações, principalmente as peças e as máquinas. Neste trabalho, considera-se somente a programação de peças que pode ser descrita da seguinte maneira.

Sejam m centros de trabalho, n peças, q operações das peças e uma relação de precedência entre as operações. Cada operação pode ser processada em vários centros de trabalho. A programação de peças consiste em alocar os centros de trabalho às peças e programar todas as operações das peças nos centros de trabalho de forma a satisfazer os objetivos desejados.

3.3.3 Os subproblemas principais

A programação de peças no SP envolve dois problemas distintos, a saber:

- seleção de rota
- programação de operações

Seleção de rota:

Este problema consiste em determinar uma rota para a produção de uma dada peça dentre as várias rotas alternativas, ou seja, definir um centro de trabalho para cada operação desta peça. Com o objetivo de balancear a utilização das máquinas, o centro com menor carga será selecionado neste trabalho.

Programação

Este problema consiste em sequenciar as operações das peças em cada centro de trabalho e determinar os instantes de início e término das operações.

3.4 Os Estados do Processo

3.4.1 Descrição dos estados

Um sistema de produção processa várias peças distintas. Cada peça consiste de várias operações. Considera-se a operação como a menor entidade no sistema, ou seja, entidade básica. Baseado nos sistemas de manufatura reais, o processo das operações envolve os seguintes estados:

- Estado de bloqueio : BL
- Estado de seleção de rota: SR
- Estado de interprocessamento: IP
- Estado de espera: ES
- Estado de processamento: PR
- Estado de término: TM

1) Estado de bloqueio: representa o estado de uma operação no qual a mesma está bloqueada para processamento devido a restrições de relação de precedência. O estado é terminado quando todas as suas operações precedentes estão acabadas.

2) Estado de seleção de rota: indica o processo de selecionar um centro de trabalho para processar uma determinada operação.

- 3) Estado de interprocessamento: denota o estado de uma operação no qual a peça associada com a mesma está sendo interprocessada para ficar disponível para processamento no próximo centro de trabalho.
- 4) Estado de espera: indica o estado no qual uma operação está disponível e esperando para ser processada num centro de trabalho.
- 5) Estado de processamento: representa o estado no qual uma operação está sendo processada no seu centro de trabalho escolhido.
- 6) Estado de término: denota o estado no qual o processamento da operação já foi completado no seu centro de trabalho.

3.4.2 Análise dos estados

Estado BL: é um estado inicial de uma operação. A operação sai desse estado imediatamente depois que todas as suas operações precedentes foram terminadas. Como é um estado sem recurso, a transição a partir deste não requer nenhuma decisão.

Estado SR: num SFM, uma operação precisa selecionar um centro de trabalho para realizar o processamento. Essa seleção não gasta tempo. Por isso, o estado é de controle e de seleção de recurso.

Estado IP: para ficar disponível fisicamente no seu centro de trabalho desejado, a peça associada com uma operação nesse estado precisa de um tempo para ser interprocessada. Depois desse tempo, a operação sai do estado IP imediatamente. Neste capítulo, não consideramos a programação do sistema de transporte de materiais, e portanto, o tempo de interprocessamento é considerado constante. Por isso, o estado é de tempo determinado e sem recurso.

Estado ES: nesse estado, uma operação está competindo com outras operações pelo mesmo centro de trabalho. A operação só sai da fila de espera quando tem a maior prioridade e o recurso solicitado está disponível. Caso contrário, a operação tem que esperar na fila por um período não definido. Por isso, é um estado de tempo indeterminado e de espera de recurso.

Estado PR: uma operação precisa de um tempo constante para completar o processamento num centro de trabalho e sai do estado imediatamente depois de ser completada. Como já utiliza o recurso desejado, o estado é de tempo determinado e com recurso.

Estado TM: é o estado final de uma operação. Portanto, não há transição a partir deste.

3.5 As Transições do Processo

O processo de uma operação no SP é um processo de transição de estados. Ele muda do estado de bloqueio até o estado de término. Quando terminam todas as transições, a operação é completada. Nesta seção, analisaremos as transições entre os estados.

1) A transição do BL para o SR

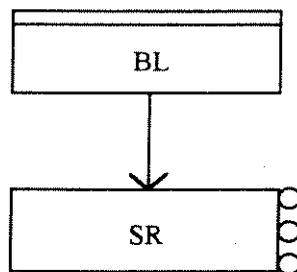


Figura 3.5.1 A transição do estado BL para SR

Nessa transição, como mostrada na Figura 3.5.1, o estado de entrada BL é o estado inicial do processo. Como discutido na seção anterior, a transição a partir dele não requer decisão, e portanto, é uma transição direta.

2). A transição do SR para o IP

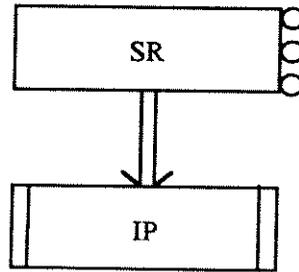


Figura 3.5.2 A transição do estado SR para IP

O estado de entrada da transição mostrada na Figura 3.5.2 é um estado de controle e de seleção de recurso. Consequentemente, a transição é indireta.

3). A transição de IP para o ES

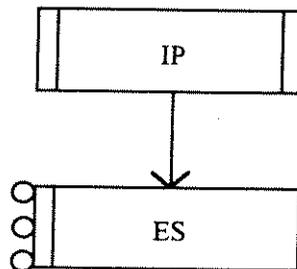


Figura 3.5.3 A transição do estado IP para ES

O estado de entrada da transição mostrada na Figura 3.5.3 é um estado de tempo determinado e sem recurso. Consequentemente, a transição é direta.

4) A transição do ES para o PR

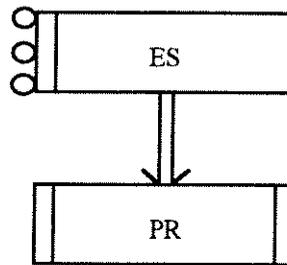


Figura 3.5.4 A transição do estado ES para PR

O estado de entrada da transição mostrada na Figura 3.5.4 é um estado de tempo indeterminado e de espera de recurso. Portanto, a transição é indireta.

5) A transição do PR para o TM

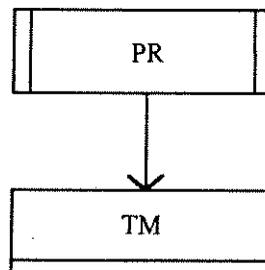


Figura 3.5.5 A transição do estado PR para TM

O estado de entrada da transição mostrada na Figura 3.5.5 é um estado de tempo determinado e com recurso. Consequentemente, a transição é direta.

Conforme a análise nesta seção, obtemos o modelo de MAE como se segue:

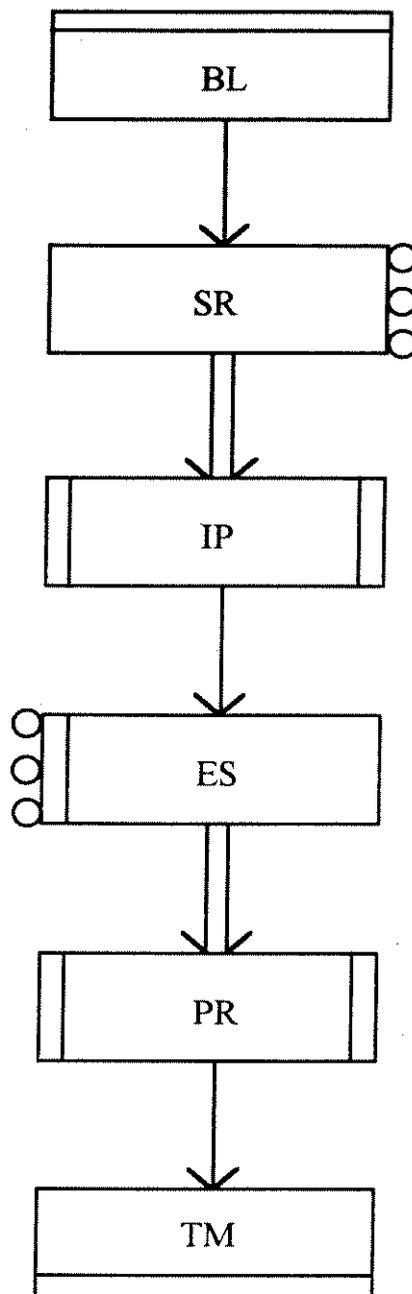


Figura 3.5.6 Modelo de MAE para a programação de peças

3.6 Modelo de Prioridade Linear Multiobjetivo

3.6.1 Objetivos do modelo

No processo de resolução da programação de produção com o MAE, a transição indireta do estado ES para o estado PR envolve uma decisão de selecionar uma das operações disponíveis num dado centro de trabalho. Esta transição será definida através de um modelo de prioridade linear multiobjetivo (PLM). Neste modelo consideram-se os seguintes objetivos:

- minimizar o tempo de atraso das peças (z_1)

$$\min z_1 = \min \left(\sum_{1 \leq i \leq n} T_i \right)$$

onde

T_i : tempo de atraso da peça i , $T_i = \max (C_i - d_i, 0)$

n : número total das peças no sistema

C_i : tempo de término da peça i

d_i : data de entrega da peça i

- minimizar o tempo total de processamento (*makespan*) (z_2)

$$\min z_2 = \min(\max_{1 \leq i \leq n} C_i)$$

Conforme o princípio de construção do modelo PLM descrito na seção 2.7, definimos a matriz de comparação para os objetivos, como mostrada na Figura 3.6.1. Consideramos que o objetivo z_1 é mais importante que o objetivo z_2 , representado pelo valor 3. Os pesos dos objetivos, o índice de consistência IC e a razão de consistência RC são mostrados na Figura 3.6.1.

	z_1	z_2	Pesos dos objetivos
z_1	1	3	$w_1 = 0,75$
z_2	1/3	1	$w_2 = 0,25$
$IC = 0, RC = 0$			

Figura 3.6.1 Matriz de comparação dos objetivos

A partir dos pesos dos objetivos, definimos um índice de avaliação de múltiplos objetivos (IM'), como mostra a equação 3.6.1. Este índice desempenha um papel semelhante à função de utilidade de múltiplos objetivos (Goicoechea et al. 1982). Uma solução com menor índice é considerada como uma solução de melhor compromisso entre os objetivos.

$$IM' = 0,75 \times z_1 + 0,25 \times z_2 \quad (3.6.1)$$

Para um dado programa, os valores de atraso e makespan são, em geral, bem distintos. Por este motivo, é conveniente utilizar, como Goicoechea et al. (1982), um índice normalizado definido na equação 3.6.2.

$$z_i' = \frac{z_i - z_i^{\min}}{z_i^{\max} - z_i^{\min}} \quad (3.6.2)$$

onde

z_i^{\min} : o menor valor do objetivo i num conjunto de programas

z_i^{\max} : o maior valor dos objetivo i num conjunto de programas

Desta forma, obtemos o índice normalizado IM :

$$IM = 0,75 \times \frac{z_1 - z_1^{\min}}{z_1^{\max} - z_1^{\min}} + 0,25 \times \frac{z_2 - z_2^{\min}}{z_2^{\max} - z_2^{\min}} \quad (3.6.3)$$

3.6.2 Regras de programação

Na literatura, existem várias regras de programação, também chamadas regras de despacho. Em geral, cada regra tem influência distinta para os objetivos mencionados anteriormente. Cada regra corresponde a um fator que pode contribuir para o modelo PLM. Utilizaremos as regras mais conhecidas na programação de produção. Para a apresentação destas regras, considere a seguinte notação.

t	Tempo atual
n	Número total das peças no sistema
i	Índice de peça
j	Índice de operação
j(t)	j-ésima operação de uma peça a ser processada no instante t
p _{ij}	Tempo de processamento da j-ésima operação da peça i
TI _{ij}	Tempo de interprocessamento da j-ésima operação da peça i (TI _{i1} = 0)
TO _i	Número total de operações da peça i
TP _i	Tempo de processamento total da peça i ($TP_i = \sum_{j=1}^{TO_i} p_{ij} + \sum_{j=1}^{TO_i} TI_{ij}$)
RO _i (t)	Número de operações restantes da peça i no instante t
RP _i (t)	Tempo de processamento restante da peça i ($RP_i = \sum_{j=j(t)}^{TO_i} p_{ij}$)
R _{ij}	Tempo que a j-ésima operação da peça i está disponível para processamento no sistema (R _{i1} = 0 para todas peças)
C _i	Tempo de término da peça i
d _i	Data de entrega da peça i
T _i	Tempo de atraso da peça i ($T_i = \max(0, C_i - d_i)$)
F _k	Fator correspondente à regra k

Consideramos as seguintes regras e seus fatores associados:

- o menor tempo restante SRT (Shortest Remaining Time) (F_1), seleciona a peça com o menor tempo restante, ou seja, seleciona o máximo F_1 onde $F_1 = -RP_i(t)$;
- o maior tempo restante LRT (Largest Remaining Time) (F_2), seleciona a peça com o maior tempo restante, ou seja, seleciona o máximo F_2 onde $F_2 = RP_i(t)$;
- o menor tempo de processamento SPT (Shortest Processing Time) (F_3), seleciona a peça com o menor tempo de processamento da operação disponível, ou seja, seleciona o máximo F_3 onde $F_3 = -P_{ij}(t)$;
- o maior tempo de processamento LPT (Longest Processing Time) (F_4), seleciona a peça com o maior tempo de processamento da operação disponível, ou seja, seleciona o máximo F_4 onde $F_4 = P_{ij}(t)$;
- o menor número de operações restantes FRO (Fewest Remaining Operations) (F_5), seleciona a peça com o menor número de operações restantes, ou seja, seleciona o máximo F_5 onde $F_5 = -RO_i(t)$;
- o maior número de operações restantes LRO (Largest Remaining Operations) (F_6), seleciona a peça com o maior número de operações restantes, ou seja, seleciona o máximo F_6 onde $F_6 = RO_i(t)$;
- a menor data de entrega EDD (Earliest Due Date) (F_7), seleciona a peça com a menor data de entrega, ou seja, seleciona o máximo F_7 onde $F_7 = -d_i$;
- o menor tempo de folga SLK (Slack Time) (F_8), seleciona a peça com o menor tempo de folga, ou seja, seleciona o máximo F_8 onde $F_8 = -(d_i - t - RP_i(t))$;
- a menor data de entrega modificada MDD (Modified Due Date) (F_9), seleciona a peça com a menor data de entrega modificada, ou seja, seleciona o máximo F_9 onde $F_9 = -\max(d_i, t + RP_i(t))$;
- FIFO (First In First Out) (F_{10}), seleciona a peça que chega em primeiro lugar, ou seja, seleciona o máximo F_{10} onde $F_{10} = -R_{ij}$;
- o maior tempo de processamento total LTT (Largest Total Processing Time) (F_{11}), seleciona a peça com o maior tempo de processamento acumulado, ou seja, seleciona o máximo F_{11} onde $F_{11} = -TP_i(t)$.

A partir destes fatores, contrói-se um modelo de prioridade P da seguinte:

$$P = \sum_{i=1}^r \alpha_i \times F_i \quad (3.6.4)$$

onde, $0 \leq \alpha_i \leq 1$ e r é o número de regras consideradas.

A prioridade dada pelas regras e pelo modelo P é utilizada para definir as transições indiretas a partir do estado de espera. As operações neste estado estão esperando numa fila no seu centro de trabalho determinado. Quando o centro está disponível para processamento, calcula-se a prioridade de cada operação na fila e a operação com a maior prioridade será selecionada para ser processada no centro. Neste caso, a transição muda a operação do seu estado de espera para o seu próximo estado. Caso contrário, a operação continua esperando na fila do centro.

Para poder definir o modelo de prioridade na equação 3.6.4, vamos analisar o desempenho das regras com relação ao atraso total e makespan.

A Figura 3.6.2 mostra o tempo de atraso total das peças para cada regra com a variação do fator γ , que é um parâmetro usado para definir a data de entrega das peças. Esta data é definida pela expressão (3.6.5).

$$d_i = \gamma \times TP_i ; \quad (3.6.5)$$

Foram feitas 20 simulações computacionais para cada regra e cada valor de γ , e a média foi tomada. Em cada simulação, consideram-se 100 peças, 7 centros de trabalho e até 3 rotas alternativas. A geração dos problemas é explicada na próxima seção.

Note que o fator γ torna a data de entrega mais apertada ou mais folgada e o atraso total decresce com o acréscimo de γ . As regras MDD, EDD, SRT, SLK, FRO e SPT apresentaram os menores atrasos. Dentre estas, a regra MDD é a dominante. Para os valores de $\gamma = 2, 3, 4$, as regras podem

ser ordenadas da seguinte forma (da melhor para a pior) : MDD, SRT, EDD, FRO, SLK, e SPT. Para os valores de $\gamma = 5, 6$, as regras podem ser ordenadas da seguinte forma : MDD, EDD, SRT, SLK, FRO e SPT. Para os valores de γ igual a 7, a ordem é seguinte : SLK, MDD, EDD, SRT, FRO e SPT.

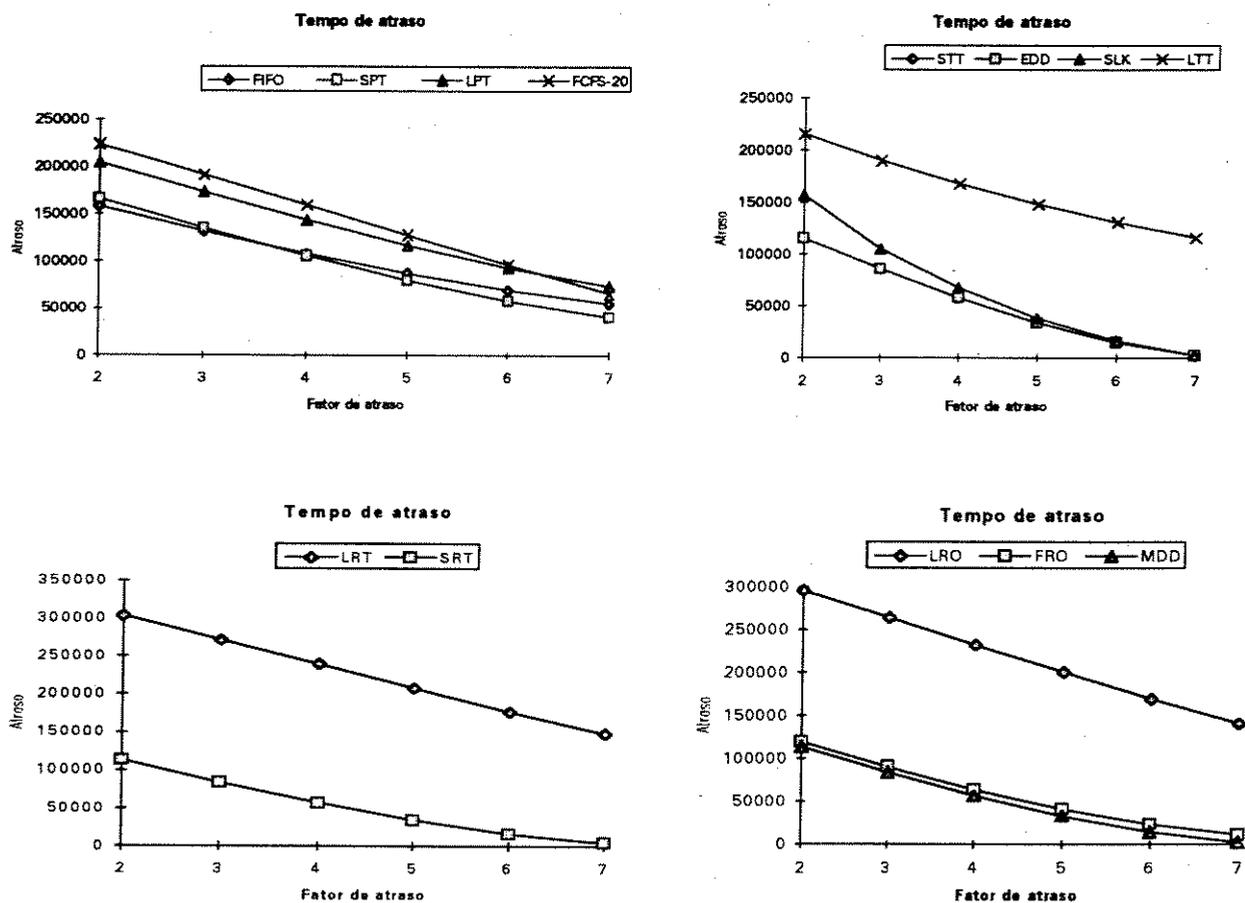


Figura 3.6.2 Tempo de atraso das regras

A Figura 3.6.3 apresenta o *makespan* associado a cada regra. As regras que dependem da data de entrega são as regras EDD, MDD e SLK. Pela expressão 3.6.5, a prioridade da regra EDD não depende do fator γ . No entanto, as prioridades das regras MDD e SLK dependem de γ . Por este motivo, o *makespan* destas regras mostrado na Figura 3.6.3 é uma média com relação aos valores de γ . Observe que as melhores são, na ordem, LRT, LRO, FIFO e LTT.

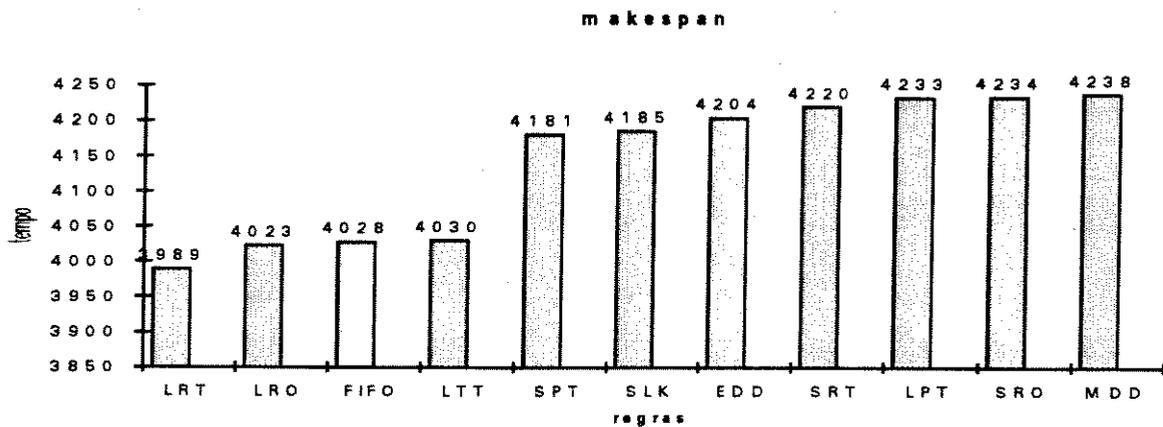


Figura 3.6.3 Makespan das regras

Foram construídos diversos modelos de prioridade de acordo com o método apresentado no capítulo 2. Cada modelo representa uma combinação de fatores com matrizes de comparação especificadas. O índice multiobjetivo é utilizado para avaliar o desempenho das diversas combinações. Para cada tipo de combinação, foram executadas 20 simulações independentes e a média é apresentada em cada célula na Tabela 3.6.1. A coluna “Média” nesta tabela representa a média sobre todos os valores de γ .

Tabela 3.6.1 Índice Multiobjetivo (IM)

Combinação dos elementos	Fator de atraso γ						Média
	2	3	4	5	6	7	
EDD+SPT	2379	2294	2314	2087	2149	2086	2223,7
EDD+SLK+SPT+LRT	2108	1949	2145	1972	2322	2377	2151,3
EDD+SLK+SRT+SPT+LRT+LTT	2525	2427	2353	2457	2532	2303	2432,8
EDD+MDD+SRT+SPT+LRT+LTT (P1)	2387	2154	2235	2351	2319	2317	2300,2
EDD+MDD+SRT+SPT+LRT+LTT	2005	2339	2075	1923	2099	2289	2127,3
EDD+MDD+SRT+LRO+LRT+LTT	2007	1975	1987	2041	2162	2182	2064,2
EDD+MDD+SRT+LRO+LRT+FIFO (P2)	1808	1911	2006	1713	1646	1954	1839,7

Os resultados na Tabela 3.6.1 mostram que o modelo P2 apresenta o melhor índice multiobjetivo. Foi observado que o modelo P1 apresenta um desempenho muito bom com relação ao objetivo atraso total. Neste trabalho, utilizamos estes dois modelos para compará-los com as regras simples.

A seguir, detalharemos a construção do modelo P1. A Figura 3.6.4 mostra a relação hierárquica entre os objetivos e os fatores neste modelo.

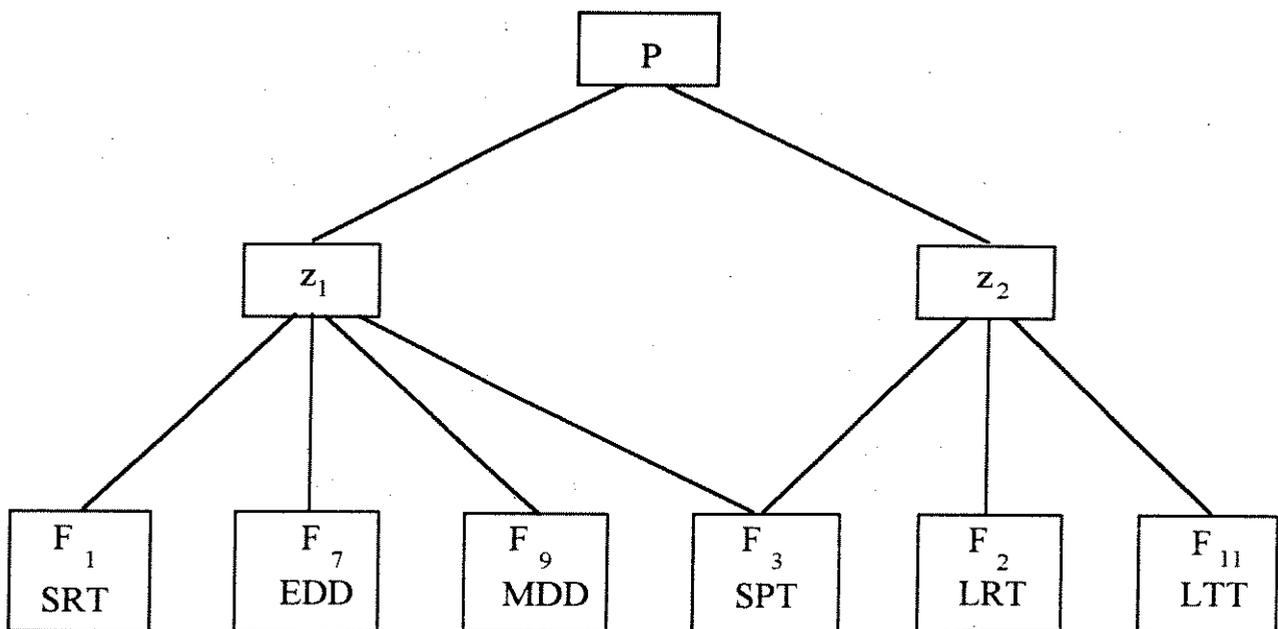


Figura 3.6.4 A relação hierárquica do modelo P1

A Figura 3.6.5 mostra as matrizes de comparação dos fatores em relação aos objetivos z_1 e z_2 no modelo P1. A definição das matrizes é baseada principalmente na análise dos resultados computacionais apresentados nas Figuras 3.6.2 e 3.6.3. Por exemplo, para o objetivo z_1 (atraso total), os graus de importância do fator F1 (SRT) em relação a F3 (SPT) e F7 (EDD) são 8 e 3 respectivamente, pois SRT apresenta um comportamento muito superior a SPT e melhor que EDD. O grau de importância de F1 (SRT) em relação a F9 (MDD) é 1/2, pois MDD é um pouco

melhor que SRT. Os outros elementos nas matrizes podem ser explicados de maneira semelhante, embora exista um grau de subjetividade na definição dos mesmos.

A Figura 3.6.5 mostra também os pesos dos fatores em relação aos objetivos que correspondem aos elementos dos autovetores principais das matrizes, bem como o índice de consistência IC_i e a razão de consistência e RC_i .

z_1 F1 F3 F7 F9	z_2 F2 F3 F11
SRT SPT EDD MDD	LRT SPT LTT

F1 1 8 3 1/2	F2 1 5 4
F3 1/8 1 1/7 1/8	F3 1/5 1 1/3
F7 1/3 7 1 1/3	F11 1/4 3 1
F9 2 8 3 1	
$w_{F1}^1=0,3293; w_{F3}^1=0,0387; w_{F7}^1=0,1684,$	$w_{F2}^2=0,6738; w_{F3}^2=0,1007; w_{F11}^2=0,2255$
$w_{F9}^1=0,4635$	
$IC_1 = 0,0598; RC_1 = 0,0664$	$IC_2 = 0,0429; RC_2 = 0,0739$

Figura 3.6.5. Matrizes de comparação do modelo P1

Através da análise dos resultados computacionais apresentados nas Figuras 3.6.2 e 3.6.3., foi elaborado o modelo P2 que contém os fatores correspondentes às melhores regras em relação aos objetivos, isto é, as regras MDD, EDD e SRT para o atraso e as regras LRT, LRO e FIFO para o makespan. A Figura 3.6.6 mostra as matrizes de comparação do modelo P2, os pesos relativos, o índice de consistência IC_i e a razão de consistência e RC_i .

z_1	F1	F7	F9		z_2	F2	F6	F10
	SRT	EDD	MDD			LRT	LRO	FIFO
F1	1	1/2	1/3		F2	1	3	3
F7	2	1	1/2		F6	1/3	1	2
F9	3	2	1		F12	1/3	1/2	1

$$w_{F1}^1 = 0,1634; w_{F7}^1 = 0,2970; w_{F9}^1 = 0,5396 \quad w_{F2}^2 = 0,5936; w_{F6}^2 = 0,2493; w_{F10}^2 = 0,1571$$

$$IC_1 = 0,0046; RC_1 = 0,0079 \quad IC_2 = 0,0268; RC_2 = 0,0462$$

Figura 3.6.6 Matrizes de comparação do modelo P2

Pode-se notar que os pontos fundamentais na definição de modelo de prioridade multiobjetivo são a escolha dos fatores e a determinação das matrizes de comparação. O conhecimento de especialistas pode ajudar a definição desses pontos fundamentais e tornar o modelo mais apropriado à situação real.

3.6.3 Os modelos de prioridade

Os valores dos parâmetros α_i nos modelos de prioridade P1 e P2 (vide equação 3.6.4) são determinados de acordo com a maneira descrita na seção 2.7. Para facilitar a apresentação, utilizamos a seguinte notação:

X_k : regra associada ao fator F_k

$F(X_k)$: fator correspondente à regra X_k

Desta forma, definimos os modelos como a seguir:

Modelo P1:

$$\alpha_1 = w_1 \times w_{F1}^1 = 0,75 \times 0,3293 = 0,2470$$

$$\alpha_7 = w_1 \times w_{F7}^1 = 0,75 \times 0,1684 = 0,1263$$

$$\alpha_9 = w_1 \times w_{F9}^1 = 0,75 \times 0,4635 = 0,3476$$

$$\alpha_2 = w_2 \times w_{F2}^2 = 0,25 \times 0,6738 = 0,1684$$

$$\alpha_3 = w_1 \times w_{F3}^1 + w_2 \times w_{F3}^2 = 0,75 \times 0,0387 + 0,25 \times 0,1007 = 0,0542$$

$$\alpha_{11} = w_2 \times w_{F11}^2 = 0,25 \times 0,2255 = 0,0564$$

$$P1 = 0,1263 \times F(EDD) + 0,3476 \times F(MDD) + 0,0786 \times F(SRT) + 0,0542 \times F(SPT) + 0,0564 \times F(LTT)$$

Modelo P2:

$$\alpha_1 = w_1 \times w_{F1}^1 = 0,75 \times 0,1634 = 0,1225$$

$$\alpha_7 = w_1 \times w_{F7}^1 = 0,75 \times 0,2970 = 0,2228$$

$$\alpha_9 = w_1 \times w_{F9}^1 = 0,75 \times 0,5396 = 0,4047$$

$$\alpha_2 = w_2 \times w_{F2}^2 = 0,25 \times 0,5936 = 0,1484$$

$$\alpha_6 = w_2 \times w_{F6}^2 = 0,25 \times 0,2493 = 0,0623$$

$$\alpha_{10} = w_2 \times w_{F10}^2 = 0,25 \times 0,1571 = 0,0393$$

$$P = 0,2228 \times F(EDD) + 0,4047 \times F(MDD) + 0,0259 \times F(LRT) + 0,0623 \times F(LRO) + 0,0393 \times F(FIFO)$$

Devido às discrepâncias numéricas dos fatores utilizados nos modelos, dividimos cada fator por seu valor máximo entre os elementos na fila de espera. Desta forma, cada fator dos elementos da fila situa-se no intervalo [0, 1].

3.7 Resultados Computacionais

Para a programação em sistemas flexíveis de manufatura, desenvolvemos um simulador em linguagem C e implementado em uma estação de trabalho SUN, modelo SPARC classic. O simulador foi construído com base no método de análise de estado apresentado no capítulo 2 e é apresentado no capítulo 5. Nesta seção, vamos utilizar o simulador para a programação do sistema de produção. Será avaliado o desempenho dos modelos de prioridade P1, P2, e das regras simples com a variação de alguns parâmetros do sistema.

3.7.1 Geração de problemas

A Tabela 3.7.1 mostra a geração dos dados dos problemas a serem simulados.

Tabela 3.7.1 Dados dos problemas

Dados	Valores
tempo de operação	distribuição uniforme, intervalo [10, 100]
centros de trabalho	7
máquinas por centro	1
número de peças	100
tempo de chegada das peças	0
data de entrega das peças	$\gamma \times$ tempo de processamento das peças
número de centros alternativos por operação	distribuição uniforme, intervalo [1, A]
tempo de interprocessamento	15
No. mínimo de operação por peça	2
No. máximo de operação por peça	igual ao n° de centros de trabalho

A Tabela 3.7.2 mostra os parâmetros que serão variados para avaliar o desempenho dos modelos e das regras.

Tabela 3.7.2 Parâmetros do sistema

Parâmetros	Variação
Fator de data de entrega γ	2, 3, 4, 5, 6, 7
N ^o máximo de centros alternativos por operação (A)	1, 2, 3, 4, 5, 6, 7
Grau de importância relativa dos objetivos	3, 1/3
Número de peças	100, 500, 1000

3.7.2 Variação de γ

Para cada valor de γ , a prioridade dos modelos e das regras é testada em 20 problemas diferentes e toma-se a média dos resultados. Nas simulações, consideramos 100 peças, 7 centros de trabalho, até 3 rotas alternativas e o objetivo atraso mais importante que o makespan, com peso 3. A Tabela 3.7.3 mostra o índice multiobjetivo (IM) com a variação de γ para as diversas prioridades e a média sobre todos valores de γ . Os gráficos correspondentes aos resultados desta tabela são mostrados nas Figuras 3.7.1 a 3.7.4.

O comportamento dos modelos de prioridade e das regras é errático, isto é, oscila com a variação de γ . Por exemplo, o índice IM da regra EDD não é crescente e nem decrescente com o acréscimo de γ . A única exceção é a regra LRT que tem um desempenho constante para todos os valores de γ .

Observa-se que dentre as regras simples, EDD é melhor para todos os valores de γ . A regra SRT fica em segundo lugar, embora tenha sido superada por SLK em 2 dos 6 valores de γ ($\gamma = 6$ e 7). A regra SLK apresenta índices melhores que MDD para γ maior ($\gamma = 5, 6$ e 7), mas tem índices piores para γ menor ($\gamma = 2, 3$ e 4). Em média, a qualidade do desempenho das melhores regras pode ser ordenada como EDD, SRT, MDD e SLK. Em geral, as regras baseadas em data de entrega (EDD, SLK e MDD) funcionam melhor, bem como as regras baseadas em trabalhos restantes (SRT e FRO). As regras restantes têm um comportamento pior.

Observa-se que o modelo P2 supera P1 e todas as regras para todos os valores de γ . P1 é superado pela regra EDD, mas é melhor que todas as outras.

Da Tabela 3.7.3 relativa ao índice multiobjetivo (IM) tem-se que a melhor regra é EDD e o melhor modelo é P2. Em média, o modelo P2 apresenta um ganho em IM de 16,22%. Em termos de atraso e makespan, o modelo P2 tem em relação a EDD, respectivamente, uma perda do 10,98% e um ganho de 1,42%.

Tabela 3.7.3: Índice multiobjetivo (IM) com as variações de γ

Prior. \ γ	2	3	4	5	6	7	Média
EDD	2293	2213	2253	2050	2185	2182	2196
SRT	2371	2304	2384	2209	2395	2434	2349,5
MDD	2400	2515	2515	2514	2437	2522	2483,8
SLK	3001	2707	2632	2414	2228	2271	2542,2
FRO	2756	2697	2803	2649	2882	2932	2786,5
SPT	4072	3963	3974	3792	3930	3898	3938,2
FIFO	4758	4706	4633	4444	4178	3666	4397,5
LTT	4462	4655	4972	5321	5801	6264	5245,8
LPT	6106	6007	6065	5873	6097	6130	6046,3
LRT	7500	7500	7500	7500	7500	7500	7500
LRO	7557	7543	7545	7500	7485	7441	7511,8
P1	2387	2154	2235	2351	2319	2317	2293,8
P2	1808	1911	2006	1713	1646	1954	1839,7

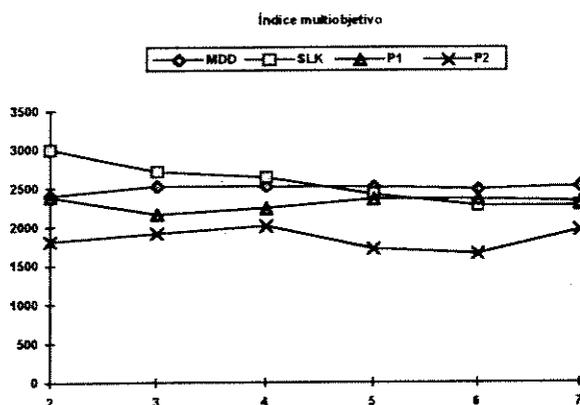


Figura 3.7.1 Desempenho dos modelos e das regras MDD e SLK

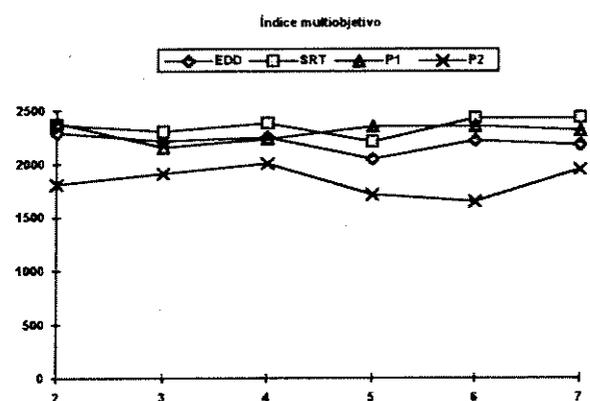


Figura 3.7.2 Desempenho dos modelos e das regras EDD e SRT

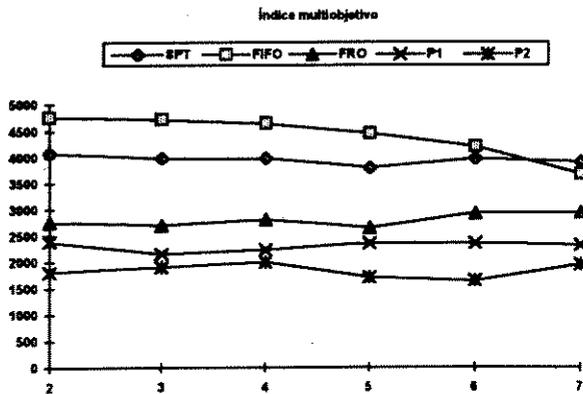


Figura 3.7.3 Desempenho dos modelos e das regras FRO, SPT e FIFO

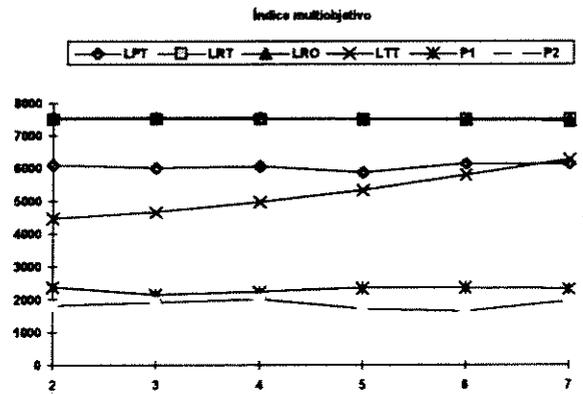


Figura 3.7.4 Desempenho dos modelos e das regras LPT, LRT, LRO e LTT

A Tabela 3.7.4 mostra os tempos de atraso total dos modelos e das regras. A representação gráfica dos resultados obtidos nesta tabela é mostrada nas Figuras 3.7.5 (a, b). Observa-se que o tempo de atraso total decresce com o acréscimo do fator γ para todas as prioridades. Isto porque o acréscimo de γ corresponde a um aumento das datas de entrega das peças.

Entre as regras simples, MDD tem o menor atraso, com exceção do valor γ igual a 7, onde é superada pela regra SLK. A regra SRT fica em segundo lugar quando γ é menor ($\gamma = 2, 3$ e 4), mas EDD é melhor que SRT quando γ é maior ($\gamma = 5, 6$ e 7). A regra SLK é dominada pelas regras MDD, SRT e EDD, com exceção de γ igual a 7. Em geral, as regras baseadas em data de entrega (EDD, SLK e MDD) funcionam melhor, bem como as regras baseadas em trabalhos restantes (SRT e FRO). A regra SPT não é boa e seu comportamento degrada com o acréscimo de γ . As demais regras têm mau desempenho.

O modelo de prioridade P1 é melhor que P2 e domina todas as regras simples. Ele é melhor que MDD em 5 valores do fator γ e melhor que as demais regras em todos os valores de γ . O modelo P2 tem um comportamento pior que MDD, SRT e EDD.

Da Tabela 3.7.4 relativa ao tempo de atraso, tem-se que a melhor regra é MDD e o melhor modelo é P1. O modelo P1 apresenta um ganho em atraso de 0,49%. Em termos de makespan, o modelo P1 tem em relação a MDD, um ganho de 0,43%.

Tabela 3.7.4: Tempos de atraso

Prior. \ γ	2	3	4	5	6	7	Média
MDD	113226	83829	56832	33274	14630	2716	50751
SRT	113265	83988	57342	34391	16105	4508	51600
EDD	115445	85681	58133	34144	15037	2730	51862
FRO	119419	90383	64123	41623	23617	11473	58440
SLK	156996	105017	67643	38529	16618	2712	64586
SPT	166577	135286	105902	79774	57831	40622	97665
LPT	204777	173661	144084	117071	93437	73901	134489
FIFO	223667	191839	160011	128189	96451	66003	144360
LTT	215624	190069	167791	148216	131157	116180	161506
LRO	296054	264227	232399	200571	169130	140451	217139
LRT	303398	271570	239742	207932	176837	148242	224620
P1	112902	83448	56451	32928	14463	2823	50503
P2	122803	93471	65988	41259	18814	2989	57554

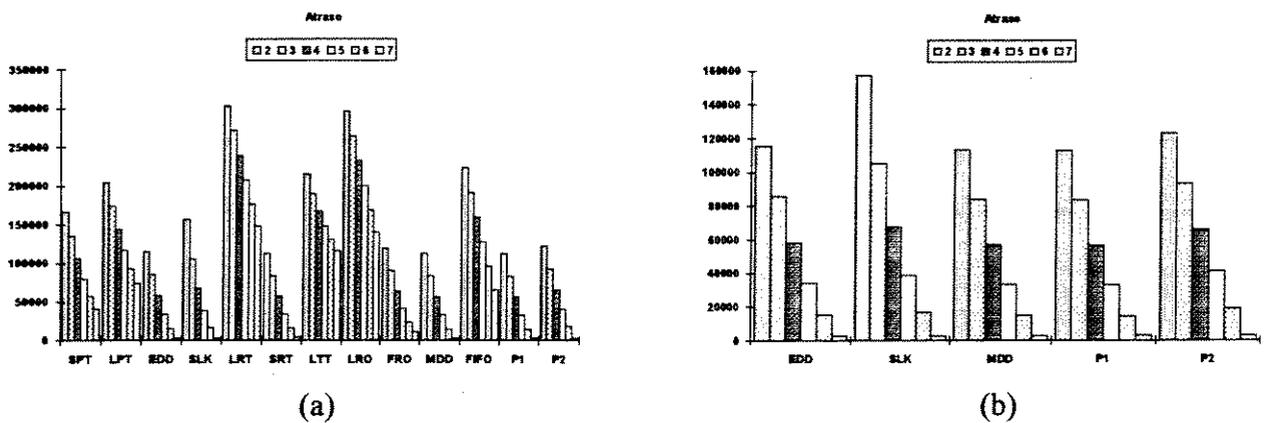


Figura 3.7.5 Tempo de atraso com a variação de γ

A Tabela 3.7.5 mostra os resultados dos modelos e das regras de prioridade em relação ao *makespan*. A representação gráfica destes resultados é mostrada na Figura 3.7.6. Observamos que a variação do *makespan* com as regras não é muito grande. Uma explicação para este fato deve-se

à política de roteamento adotada que consiste em selecionar o centro com menor carga para balancear a utilização dos centros.

Como ressaltado na seção 3.6.2, somente as regras SLK e MDD dependem do fator de atraso γ . Dentre as regras simples, a regra LRT é dominante. Em ordem crescente da média de *makespan*, as regras podem ser ordenadas em seguinte forma: LRT, LRO, FIFO, LTT, SPT, SLK, EDD, SRT, LPT, FRO e MDD. Note que as melhores regras para o atraso total são as piores para o *makespan*. Isto mostra a necessidade de combinar regras quando se tem que satisfazer os objetivos de atraso e *makespan*.

O *makespan* dos modelos de prioridade depende de γ , pois a regra MDD faz parte destes modelos. Nota-se que o modelo P2 é melhor que P1. Na média, P2 supera 7 regras, SPT, SLK, EDD, SRT, LPT, FRO, MDD, e perde para as quatro melhores regras, LRT, LRO, FIFO e LTT. O modelo P2 domina 5 regras e P1 supera 4 regras.

Da Tabela 3.7.5 relativa ao *makespan* tem-se que a melhor regra é LRT e o melhor modelo é P2. O modelo P2 apresenta uma perda em *makespan* de 3,89%. Em termos de atraso, o modelo P2 tem em relação a LRT, um ganho de 74,38%.

Tabela 3.7.5: Makespan

Prior. \ γ	2	3	4	5	6	7	Média
LRT	3989	3989	3989	3989	3989	3989	3989
LRO	4023	4023	4023	4023	4023	4023	4023
FIFO	4028	4028	4028	4028	4028	4028	4028
LTT	4030	4030	4030	4030	4030	4030	4030
SPT	4181	4181	4181	4181	4181	4181	4181
SLK	4113	4176	4203	4223	4201	4213	4188,2
EDD	4204	4204	4204	4204	4204	4204	4204
SRT	4220	4220	4220	4220	4220	4220	4220
LPT	4233	4233	4233	4233	4233	4233	4233
FRO	4234	4234	4234	4234	4234	4234	4234
MDD	4223	4242	4235	4258	4231	4238	4237,8
P1	4223	4207	4209	4242	4220	4217	4219,7
P2	4128	4142	4148	4135	4133	4180	4144,3

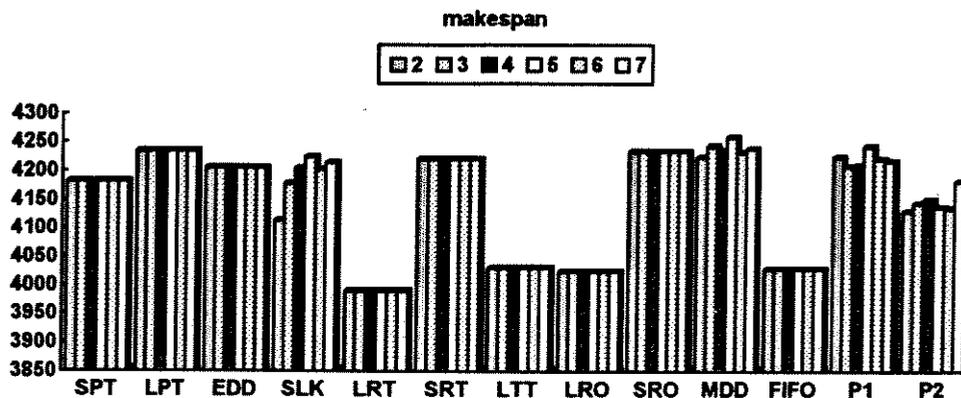


Figura 3.7.6 Makespan com a variação de γ

A Tabela 3.7.6 mostra o tempo de fluxo médio por peça com as prioridades. A representação gráfica dos resultados desta tabela é mostrada na Figura 3.7.7. Como no caso de makespan, somente as prioridades MDD, SLK, P1 e P2 variam com o fator de atraso γ . Analisando a coluna das médias, observa-se que SRT supera todas as prioridades. Em ordem crescente de tempo de fluxo médio, as regras simples são ordenadas da seguinte forma: SRT, MDD, EDD, FRO, SLK, SPT, LPT, LTT, FIFO, LRO e LRT. Note-se que as regras baseadas em data de entrega (EDD, SLK e MDD) funcionam melhor, bem como as regras baseadas em trabalho restante (SRT e FRO). As demais regras têm um desempenho pior.

O modelo de prioridade P1 é melhor que P2 para todos os níveis do fator γ . Em média, P1 tem um tempo de fluxo um pouco maior que a regra SRT, mas inferior a todas as demais regras. O modelo P2 supera 7 regras.

Tabela 3.7.6 Tempo de fluxo médio

Prior. \ γ	2	3	4	5	6	7	Média
SRT	144366	144366	144366	144366	144366	144366	144366
MDD	144470	144791	145338	145785	146255	146676	145552,5
EDD	146722	146722	146722	146722	146722	146722	146722
FRO	150361	150361	150361	150361	150361	150361	150361
SLK	188579	166734	157446	153163	150887	150025	161139
SPT	198374	198374	198374	198374	198374	198374	198374
LPT	236529	236529	236529	236529	236529	236529	236529
LTT	245628	245628	245628	245628	245628	245628	245628
FIFO	255494	255494	255494	255494	255494	255494	255494
LRO	327882	327882	327882	327882	327882	327882	327882
LRT	335225	335225	335225	335225	335225	335225	335225
P1	144113	144225	144370	144533	144648	144690	144429,8
P2	154300	155276	156217	156926	155708	154497	155487,3

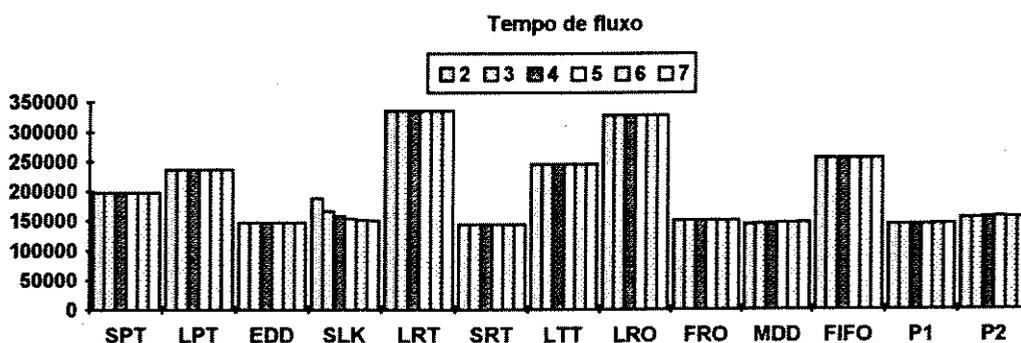


Figura 3.7.7 Tempo de fluxo médio com a variação de γ

A Tabela 3.7.7 mostra a utilização média por centro de trabalho obtida pelas prioridades. A representação gráfica dos resultados desta tabela é mostrada na Figura 3.7.8. Como no caso de makespan, somente as prioridades MDD, SLK, P1 e P2 variam com o fator de atraso γ . Observa-se que a regra LRT supera todas as prioridades. De acordo com a ordem decrescente de utilização, as regras simples são classificadas da seguinte forma: LRT, LRO, FIFO, LTT, SPT, SLK, EDD, SRT, LPT, FRO e MDD. Esta classificação é idêntica à obtida para o caso do

makespan, isto é, regras com menor makespan produzem maior utilização média. Isto se deve à política de roteamento adotada.

Observa-se também, que o modelo de prioridade P2 é melhor que P1. Em média, P2 é superado pelas quatro melhores regras, mas domina as regras restantes. O modelo P1 domina 4 regras.

Tabela 3.7.7 Utilização média dos centros

Prior. \ γ	2	3	4	5	6	7	Média
LRT	9938	9938	9938	9938	9938	9938	9938
LRO	9853	9853	9853	9853	9853	9853	9853
FIFO	9841	9841	9841	9841	9841	9841	9841
LTT	9834	9834	9834	9834	9834	9834	9834
SPT	9485	9485	9485	9485	9485	9485	9485
SLK	9636	9492	9433	9385	9438	9410	9465,7
EDD	9431	9431	9431	9431	9431	9431	9431
SRT	9393	9393	9393	9393	9393	9393	9393
LPT	9364	9364	9364	9364	9364	9364	9364
FRO	9363	9363	9363	9363	9363	9363	9363
MDD	9388	9344	9362	9312	9369	9357	9355,3
P1	9392	9424	9421	9345	9394	9401	9396,2
P2	9602	9572	9556	9586	9592	9483	9565,2

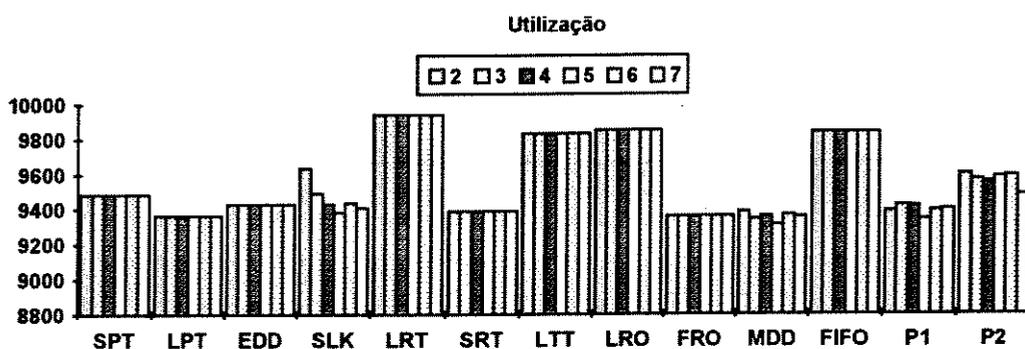


Figura 3.7.8 Utilização média dos centros com a variação de γ

3.7.3 Variações de número de centros alternativos

O número máximo (A) de centros alternativos para cada operação das peças é um parâmetro importante para medir a flexibilidade de um sistema de manufatura. Este parâmetro será variado para testar as prioridades em graus de flexibilidade distintos. Para cada valor do parâmetro, a prioridade dos modelos e das regras é testada em 20 problemas diferentes e toma-se a média dos resultados. Nas simulações, consideramos 100 peças, 7 centros de trabalho, fator de atraso γ igual a 4 e o objetivo atraso mais importante que o makespan, com peso 3. A Tabela 3.7.8 mostra o índice multiobjetivo das prioridades com a variação do número máximo de centros alternativos. As representações gráficas dos resultados desta tabela são ilustradas nas Figuras 3.7.9 a 3.7.12.

Observa-se que o comportamento das prioridades oscila com a variação de A. Por exemplo, o índice da regra MDD não é crescente e nem decrescente com o acréscimo de A. A única exceção é a regra LRT que tem um desempenho constante para todos os valores do parâmetro. Note que dentre as regras, EDD é a melhor. Ela supera as regras MDD e SLK em 5 níveis de A, domina SRT em 6 níveis e todas as demais regras em todos os níveis. MDD e SRT têm um comportamento parecido, isto é, SRT domina MDD em 4 níveis. Ambas dominam a regra SLK em 5 níveis do parâmetro A. De acordo com a média, as regras podem ser ordenadas na seguinte sequência: EDD, MDD, SRT, SLK, FRO, SPT, FIFO, LTT, LPT, LRO e LRT. Note que as regras baseadas em data de entrega (EDD, MDD e SLK) e as regras baseada no menor trabalho restante (SRT e FRO) têm os melhores índices.

Observa-se também, que o modelo de prioridade P2 é melhor que P1 e todas as regras em todos os valores de A. O modelo P1 domina as regras EDD, MDD e SLK em 6 dos 7 níveis de A e todas as demais regras em todos os níveis. Em média, os dois modelos de prioridade P1 e P2 são significativamente superiores às regras.

Da Tabela 3.7.8 relativa ao índice multiobjetivo (IM) tem-se que a melhor regra é EDD e o melhor modelo é P2. Em média, o modelo P2 apresenta um ganho em IM de 21,42%. Em termos de

atraso e makespan, o modelo P2 tem em relação a EDD, respectivamente, uma perda de 13,94% e um ganho de 2,19%.

Tabela 3.7.8 Índice multiobjetivo

Prior. \ A	1	2	3	4	5	6	7	Média
EDD	2512	2094	2253	2147	2173	2426	2496	2300,1
MDD	2394	2528	2515	2181	2446	2465	2384	2416,1
SRT	2523	2263	2384	2389	2452	2460	2452	2417,6
SLK	1834	2090	2632	2613	2795	2990	2992	2563,7
FRO	2940	2481	2803	2674	2806	2557	2763	2717,7
SPT	2831	3309	3974	3752	4048	4148	4302	3766,3
FIFO	4397	4334	4633	4572	4718	4579	4600	4547,6
LTT	5092	4986	4972	5039	5013	5013	5107	5031,7
LPT	6095	5503	6065	5853	5775	5717	5969	5853,9
LRO	7123	7487	7545	7439	7509	7474	7534	7444,4
LRT	7500	7500	7500	7500	7500	7500	7500	7500
P1	2318	1998	2235	2279	2151	2125	2142	2178,3
P2	1583	1468	2006	1847	2000	1792	1956	1807,4

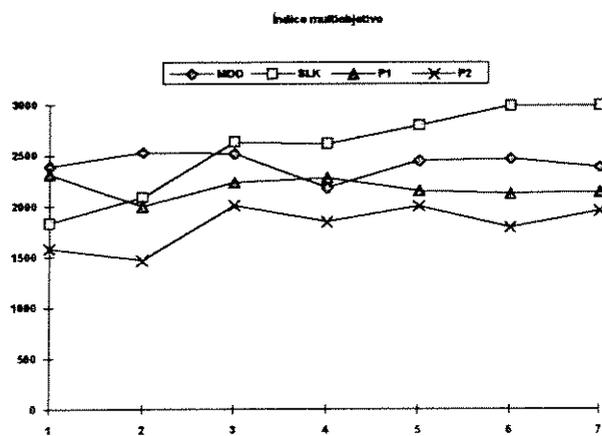


Figura 3.7.9 Desempenho dos modelos e das regras MDD e SLK

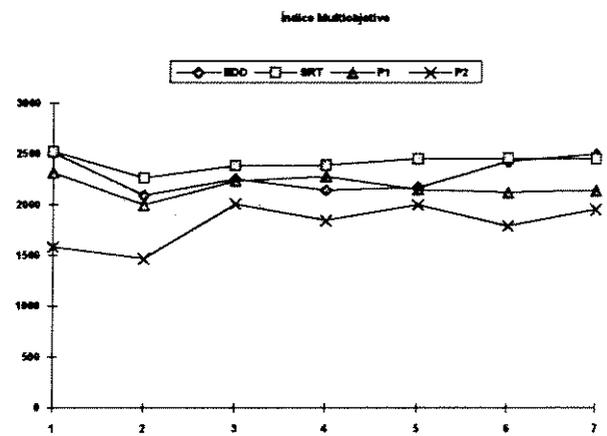


Figura 3.7.10 Desempenho dos modelos e das regras EDD e SRT

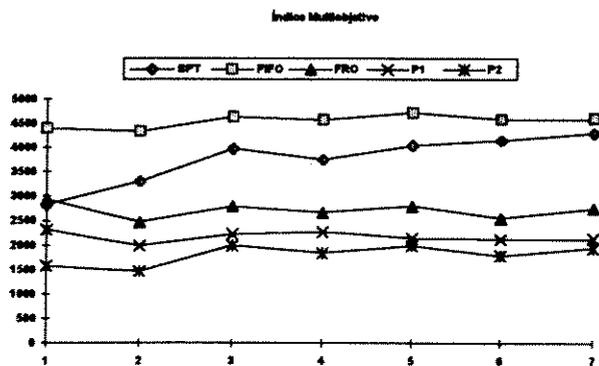


Figura 3.7.11 Desempenho dos modelos e das regras SPT, FIFO e FRO

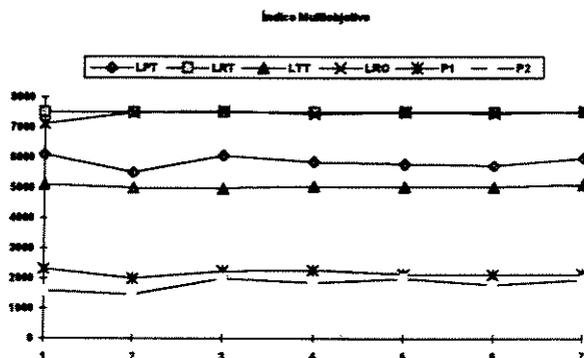


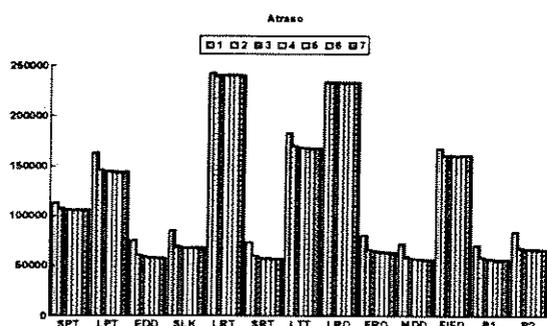
Figura 3.7.12 Desempenho dos modelos e das regras LPT, LRT, LTT e LRO

A Tabela 3.7.9 mostra os tempos de atraso total das prioridades. A representação gráfica dos resultados desta tabela é ilustrada na Figura 3.7.13. Observa-se que os tempos de atraso decrescem com o número máximo de centros alternativos A. Este decréscimo é bem marcante quando se passa de A igual a 1 para A igual a 2. Para todos os níveis de A, as melhores regras são, na ordem, MDD, SRT e EDD. Em geral, as regras baseadas em data de entrega (EDD, MDD e SLK) e as regras baseadas no trabalho restante (SRT e FRO) têm os atrasos menores.

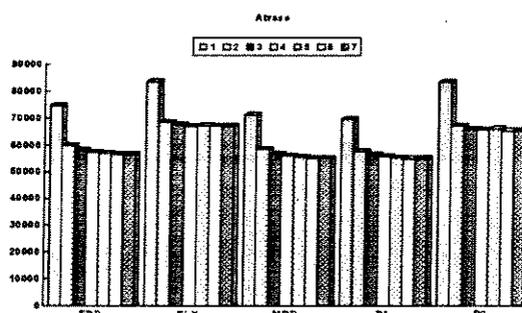
Para todos os níveis de A, o modelo P1 domina P2 e todas as regras. O modelo P2 é pior que quatro regras e domina as demais 7 regras em todos os níveis. Em relação ao atraso, tem-se que a melhor regra é MDD e o melhor modelo é P1. O modelo P1 apresenta um ganho em atraso de 0,78%. Em termos de makespan, o modelo P1 tem, em relação a MDD, um ganho de 0,29%.

Tabela 3.7.9 Tempo de atraso total

Prior. \ A	1	2	3	4	5	6	7	Média
MDD	71304	58489	56832	56068	55602	55106	55087	58355,4
SRT	72740	59315	57342	56531	56379	55617	56122	59149,4
EDD	74702	59965	58133	57310	57146	56622	56550	60061,1
FRO	79807	65799	64123	63275	62923	62628	62475	65861,4
SLK	83740	68470	67643	66925	67241	67135	67141	69756,4
SPT	112433	106853	105902	105439	105111	104901	105307	106563,7
LPT	162742	145724	144084	144097	143605	142969	143461	146668,9
FIFO	166843	159788	160011	159832	159825	159899	159920	160874
LTT	182220	169366	167791	167418	167079	166580	166868	169617,4
LRO	233141	232251	232399	232479	232454	232468	232548	232534,3
LRT	241771	239821	239742	239895	239864	239878	239858	240118,4
P1	69698	57799	56451	55896	55394	55044	55009	57898,7
P2	83541	67071	65988	65659	66042	65341	65407	68435,6



(a)



(b)

Figura 3.7.13 Tempo de atraso com a variação de A

A Tabela 3.7.10 mostra a *makespan* das prioridades. A representação gráfica dos resultados desta tabela é ilustrada na Figura 3.7.14. Como no caso de atraso, o *makespan* diminui significativamente quando A muda de 1 para 2. Observa-se que a regra LRT domina todas as regras para todos os valores de A. A regra LRO supera FIFO em 6 níveis de A e as demais regras em todos os níveis. Em geral, as melhores regras para atraso têm um *makespan* maior.

Nota-se que o modelo de prioridade P2 supera P1 em todos os níveis de A. Em relação às regras simples, o modelo P2 é superado pelas regras LRT, LRO, FIFO, LTT, e domina as demais regras.

Com relação ao makespan, observa-se que a melhor regra é LRT e o melhor modelo é P2. O modelo P2 apresenta uma perda em makespan de 3,4%. Em termos de atraso, o modelo P2 tem, em relação a LRT, um ganho de 71,5%.

Tabela 3.7.10 Makespan

Prior. \ A	1	2	3	4	5	6	7	Média
LRT	4510	4009	3989	3983	3979	3978	3980	4061,1
LRO	4510	4052	4023	4006	4004	4000	4005	4085,7
FIFO	4537	4028	4028	4015	4017	4004	4006	4090,7
LTT	4541	4065	4030	4030	4017	4017	4023	4103,3
SPT	4670	4194	4181	4148	4142	4148	4151	4233,4
SLK	4712	4246	4203	4189	4165	4178	4169	4266
LPT	4847	4279	4233	4198	4155	4150	4160	4288,9
EDD	4889	4297	4204	4182	4148	4167	4164	4293
SRT	4905	4325	4220	4208	4173	4173	4162	4309,4
FRO	4923	4318	4234	4209	4180	4158	4166	4312,6
MDD	4894	4368	4235	4190	4175	4175	4160	4313,9
P1	4893	4296	4209	4200	4152	4148	4142	4291,4
P2	4672	4165	4148	4121	4105	4088	4096	4199,3

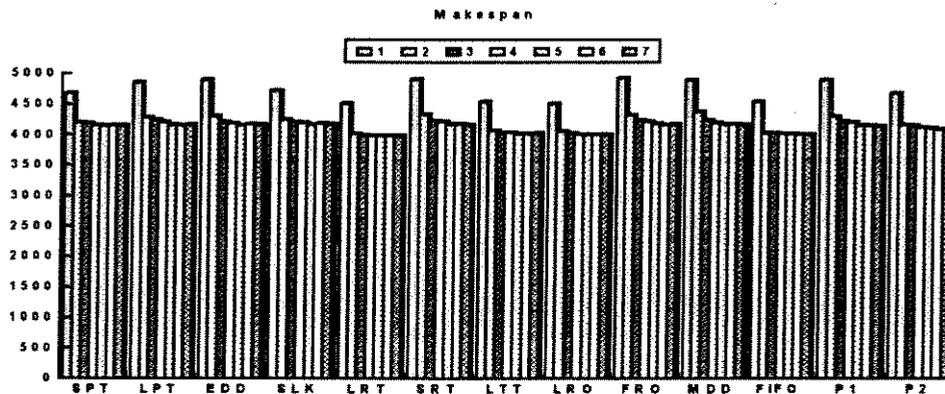


Figura 3.7.14 Makespan com a variação de A

A Tabela 3.7.11 mostra o tempo de fluxo médio por peça com as prioridades. A representação gráfica dos resultados desta tabela é mostrada na Figura 3.7.15. Como no caso de *makespan*, o maior decréscimo do tempo de fluxo médio ocorre quando A muda de 1 para 2. Para todos os níveis de A, as melhores regras são, na ordem, SRT, MDD e EDD. Em geral, as regras baseadas em data de entrega (EDD, MDD e SLK) e as regras baseadas no trabalho restante (SRT e FRO) têm os menores tempos de fluxo.

Para todos os níveis de A, o modelo P1 domina P2 e todas as regras. O comportamento do modelo P1 é melhor que a regra SRT em 4 níveis e melhor que as demais regras em todos os níveis. O modelo P2 é superado pelas regras SRT, MDD, EDD, FRO, e domina as demais.

Tabela 3.7.11 Tempo de fluxo médio

Priori.\A	1	2	3	4	5	6	7	Média
SRT	160439	146474	144366	143515	143255	142520	142810	146197
MDD	161115	147376	145338	144389	143850	143313	143314	146956
EDD	164624	148941	146722	145720	145525	144933	144912	148768
FRO	167098	152327	150361	149474	148995	148479	148293	152147
SLK	174398	158460	157446	156402	156747	156585	156930	159567
SPT	205219	199536	198374	197730	197576	197354	197759	199078
LPT	255246	238116	236529	236369	235772	235203	235692	238990
LTT	262869	248084	245628	245016	244615	243672	243985	247696
FIFO	262315	255272	255494	255315	255308	255382	255403	256356
LRO	328625	327734	327882	327962	327937	327951	328031	328017
LRT	337254	335304	335225	335378	335347	335361	335341	335601
P1	158766	146081	144370	143758	143102	142735	142762	145939
P2	174624	157297	156217	155663	156163	155345	155565	158696

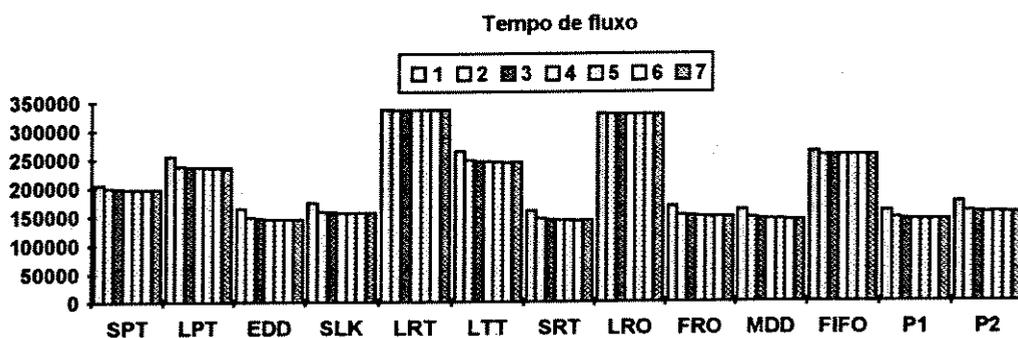


Figura 3.7.15 Tempo de fluxo médio com a variação de A

A Tabela 3.7.12 mostra a utilização média dos centros de trabalho com as prioridades. A representação gráfica dos resultados desta tabela é mostrada na Figura 3.7.16. Como no caso de *makespan*, o maior aumento de utilização ocorre quando A muda de 1 para 2. Observa-se que a regra LRT supera todas as prioridades. Em ordem decrescente da utilização média, as regras podem ser ordenadas da seguinte forma: LRT, LRO, FIFO, LTT, SPT, SLK, LPT, EDD, SRT, FRO e MDD. Em geral, as regras que favorecem o atraso apresentam menor utilização.

Observa-se também, que o modelo P2 é melhor que P1. Em média, P2 é superado pelas quatro melhores regras, mas domina as 7 regras restantes. O modelo P1 domina 4.

Tabela 3.7.12 Utilização

Prior. \ A	1	2	3	4	5	6	7	Média
LRT	8799	9888	9938	9953	9962	9964	9960	9780,6
LRO	8799	9783	9853	9894	9900	9908	9897	9719,1
FIFO	8750	9840	9841	9872	9868	9899	9893	9709
LTT	8737	9751	9834	9836	9868	9867	9852	9677,9
SPT	8502	9453	9485	9555	9569	9557	9551	9381,7
SLK	8418	9336	9433	9463	9519	9489	9511	9309,9
LPT	8191	9267	9364	9444	9541	9552	9531	9270
EDD	8119	9227	9431	9480	9558	9512	9523	9264,3
SRT	8097	9167	9393	9422	9501	9502	9525	9229,6
FRO	8057	9179	9363	9420	9485	9537	9519	9222,9
MDD	8111	9077	9362	9460	9493	9496	9530	9218,4
P1	8117	9230	9421	9440	9548	9557	9570	9269
P2	8488	9515	9556	9618	9654	9696	9678	9457,9

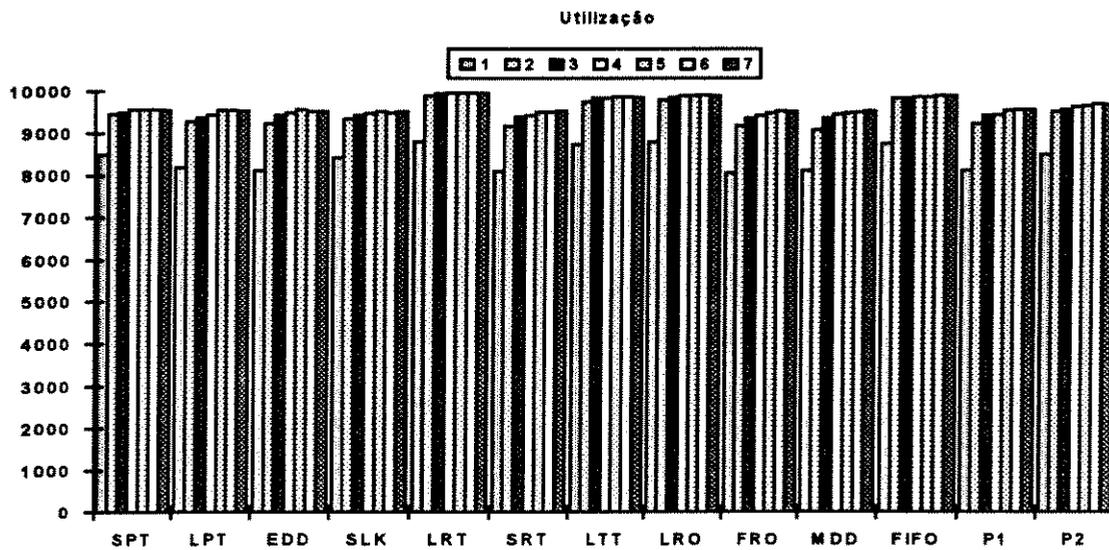


Figura 3.7.16 Utilização com a variação de A

3.7.4 Variação dos pesos relativos dos objetivos

A matriz de comparação dos objetivos, apresentada na Figura 3.7.17, representa a relação entre os pesos dos objetivos. Nos casos anteriores, foi considerado que o objetivo z_1 (tempo de atraso total) é mais importante que o objetivo z_2 (*makespan*), com peso n igual a 3. É claro que este peso representa somente um grau de preferência que pode ser alterado em função do sistema e do julgamento de especialistas. A seguir, vamos analisar o caso em que o objetivo z_1 é menos importante que o objetivo z_2 , com peso n igual a $1/3$ e avaliar esta alteração nos modelos de prioridade. Para cada valor de n , a prioridade dos modelos e das regras é testada em 20 problemas diferentes e toma-se a média dos resultados. Nas simulações, consideramos 100 peças, 7 centros de trabalho, fator de atraso igual a 4 e número máximo de centros alternativos igual a 3.

	z_1	z_2
z_1	1	n
z_2	$1/n$	1

Figura 3.7.17 Matriz de comparação dos objetivos

A Tabela 3.7.14 mostra os tempos de atraso dos modelos de prioridade e da melhor regra para o atraso, isto é a regra MDD, para os casos n igual a 3 e $1/3$. É óbvio que as regras simples, tais como MDD, não dependem de n . Observa-se que, como esperado, o tempo de atraso dos modelos de prioridade aumenta significativamente quando n igual a $1/3$, pois o objetivo de atraso é menos importante que o objetivo makespan.

Tabela 3.7.14 Tempo de atraso com a variação de n

Prioridades	Tempo de atraso	
	$n = 3$	$n = 1/3$
MDD	56832	56832
P1	56451	240001
P2	65988	241827

A Tabela 3.7.15 mostra o makespan dos modelos de prioridade e da melhor regra para o makespan, isto é a regra LRT, para os casos n igual a 3 e $1/3$. Note que o makespan dos modelos de prioridade decresce quando n igual a $1/3$ e o desempenho dos mesmos é semelhante ao da regra LRT.

Tabela 3.7.15 Makespan com a variação de n

Prioridades	Makespan	
	$n = 3$	$n = 1/3$
LRT	3989	3989
P1	4209	3982
P2	4148	3992

Os resultados das Tabelas 3.7.14 e 3.7.15 permitem concluir que os modelos de prioridade são sensíveis ao peso relativo dos objetivos. A construção do modelo mais adequado depende do grau de importância dos objetivos atribuído ao sistema prático.

3.7.5 Variação do número de peças processadas

O número total das peças no sistema é um parâmetro importante para avaliar a possibilidade de utilização do simulador/programador em problemas reais. A Tabela 3.7.16 mostra a variação dos tempos gastos para resolver problemas de programação com número de peças igual a 100, 500 e 1000. Para cada caso, são feitas três simulações e a média dos tempos é tomada. Nas simulações, consideramos 7 centros de trabalho, fator de atraso igual a 4 e número máximo de centros alternativos igual a 3. O tempo de CPU (em segundos) inclui o tempo para criar os dados dos arquivos e o tempo para obter o resultado. Observa-se que para até 1000 peças, 5049 operações, o tempo computacional é razoavelmente pequeno, o que mostra a capacidade do simulador/programador para resolver problemas reais de programação.

Tabela 3.7.16 Tempo computacional

No. de Peça	No. de Operações	Tempo de CPU
100	487	5 s
500	2540	154 s
1000	5049	716 s

3.7.6 Algoritmo de busca

Foi apresentado no capítulo 2 um método de construção do modelo de prioridade combinada. Este método é uma maneira útil e prática para a tomada de decisão com múltiplos objetivos. No entanto, o comportamento de um modelo de prioridade depende do sistema considerado, do desempenho das regras que compõem o modelo e das matrizes de julgamento. Não é possível definir um único modelo que apresente o melhor comportamento em todos os casos. Para um caso específico, é possível melhorar o modelo para obter uma solução de melhor qualidade. Para este fim, é proposto a seguir um algoritmo de busca com relação aos parâmetros α_i para um dado conjunto de fatores que definem o modelo.

Algoritmo de busca

Considere um modelo de prioridade representado pela combinação linear de n fatores dados F_1, F_2, \dots, F_n :

$$P = \sum_{i=1}^n \alpha_i \times F_i \quad (3.7.1)$$

com $\sum_{i=1}^n \alpha_i = 1$.

Os parâmetros não nulos $\alpha_1, \alpha_2, \dots, \alpha_n$ são determinados de acordo com o método apresentado no capítulo 2.

A idéia básica do algoritmo é buscar numa vizinhança dos parâmetros $\alpha_1, \alpha_2, \dots, \alpha_n$ para tentar obter uma combinação dos fatores que forneça um modelo melhor em termos de algum critério estabelecido.

Para facilitar a explicação do algoritmo, considere inicialmente um modelo de prioridade com somente dois fatores:

$$P = \alpha_1 F_1 + \alpha_2 F_2 \quad (3.7.2)$$

A seguinte notação é utilizada na descrição do algoritmo:

$z(P)$: critério utilizado para avaliar um programa obtido pelo modelo de prioridade P

τ_e : número máximo de iterações à esquerda de α_1

τ_d : número máximo de iterações à direita de α_1

δ : tamanho do passo inicial de busca, $0 < \delta, \tau_e \delta < \alpha_1, \tau_d \delta < \alpha_2$

δ_m : tamanho mínimo do passo

λ : porcentagem de redução de tamanho de passo δ

k : índice de iteração

- S_0 : lista dos elementos $(\alpha_1, \alpha_2, \delta, z)$ não pesquisados
 S_1 : lista com um único elemento sob pesquisa
 S_2 : lista de mínimos locais encontrados com relação à vizinhança dos pontos pesquisados

Algoritmo I.

Passo 1. Faça $P_0 = P$, $z_0 = z(P_0)$ e $\delta_0 = \delta$. Coloque o elemento $(\alpha_1, \alpha_2, \delta, z_0)$ na lista S_0 .

Passo 2. Se S_0 é vazia, vá para o Passo 10.

Senão, retire um elemento $(\alpha_{1,s}, \alpha_{2,s}, \delta_s, z_s) \in S_0$ da lista S_0 e coloque-o na lista S_1 . Faça $k = 0$, $z^* = z_s$, $\alpha_1^* = \alpha_{1,k} = \alpha_{1,s}$, $\alpha_2^* = \alpha_{2,k} = \alpha_{2,s}$, $\delta^* = \delta_k = \delta_s$.

Passo 3. Defina $\alpha_{1,k+1}$, $\alpha_{2,k+1}$ e obtenha um novo modelo P_{k+1} da seguinte forma:

$$\alpha_{1,k+1} = \alpha_{1,k} + \delta_k$$

$$\alpha_{2,k+1} = \alpha_{2,k} - \delta_k$$

$$P_{k+1} = \alpha_{1,k+1} F_1 + \alpha_{2,k+1} F_2$$

$$\delta_{k+1} = \delta_k$$

Avalie o tempo de atraso do novo programa e faça $z_{k+1} = z(P_{k+1})$.

Se $z_{k+1} < z^*$, faça $z^* = z_{k+1}$, $\alpha_1^* = \alpha_{1,k+1}$, $\alpha_2^* = \alpha_{2,k+1}$, $\delta^* = \delta_{k+1}$.

Passo 4. Faça $k = k + 1$.

Se $k < \tau_d$, vá para Passo 3.

Passo 5. $\delta^* = \lambda \delta^*$. Se $\delta^* > \delta_m$, coloque $(\alpha_1^*, \alpha_2^*, \delta^*, z^*)$ na lista S_0 .

Senão, coloque $(\alpha_1^*, \alpha_2^*, \delta^*, z^*)$ na lista S_2 .

Passo 6: Retire o elemento $(\alpha_{1,s}, \alpha_{2,s}, \delta_s, z_s) \in S_1$.

Faça $k = 0$, $z^* = z_s$, $\alpha_1^* = \alpha_{1,k} = \alpha_{1,s}$, $\alpha_2^* = \alpha_{2,k} = \alpha_{2,s}$, $\delta^* = \delta_k = \delta_s$.

Passo 7. Defina $\alpha_{1,k+1}$, $\alpha_{2,k+1}$ e obtenha um novo modelo P_{k+1} da seguinte forma:

$$\alpha_{1,k+1} = \alpha_{1,k} - \delta_k$$

$$\alpha_{2,k+1} = \alpha_{2,k} + \delta_k$$

$$P_{k+1} = \alpha_{1,k+1} F_1 + \alpha_{2,k+1} F_2$$

$$\delta_{k+1} = \delta_k$$

Avalie o tempo de atraso do novo programa e faça $z_{k+1} = z(P_{k+1})$.

Se $z_{k+1} < z^*$, $z^* = z_{k+1}$, $\alpha_1^* = \alpha_{1,k+1}$, $\alpha_2^* = \alpha_{2,k+1}$, $\delta^* = \delta_{k+1}$

Passo 8. $k = k + 1$.

Se $k < \tau_e$, vá para Passo 7.

Passo 9. $\delta^* = \lambda \delta^*$. Se $\delta^* > \delta_m$, coloque $(\alpha_1^*, \alpha_2^*, \delta^*, z^*)$ na lista S_0 .

Senão, coloque $(\alpha_1^*, \alpha_2^*, \delta^*, z^*)$ na lista S_2 .

Vá para Passo 2.

Passo 10. Selecione o elemento $(\alpha_1^*, \alpha_2^*, \delta^*, z^*) \in S_2$ tal que $z^* \leq z$ para qualquer $(\alpha_1, \alpha_2, \delta, z) \in S_2$. Então, α_1^* , α_2^* , representam os melhores parâmetros do modelo P da equação 3.7.2 na vizinhança procurada.

Note-se que o critério z utilizado no algoritmo pode representar um único objetivo ou um índice multiobjetivo.

Observa-se também que o algoritmo pesquisa os pontos em duas direções diferentes, a partir de um valor pré-definido do parâmetro α_1 . Primeiro, a busca é feita à direita, e depois à esquerda. Os melhores pontos encontrados em ambas as direções são colocados na lista S_0 para pesquisar-se no futuro com tamanho de passo reduzido, até chegar ao limite do tamanho mínimo do passo. Quando isto ocorre, obtém-se os mínimos locais que são colocados na lista S_2 . O melhor mínimo local na lista S_2 corresponde ao resultado final.

Neste algoritmo, o Passo 1 define as condições iniciais. No Passo 2 retira-se um ponto não pesquisado, $(\alpha_1, \alpha_2, \delta_d, z)$, da lista S_0 e coloca-se este ponto na lista S_1 para ser explorado. Nos Passos 3 e 4 pesquisam-se os pontos à direita de α_1 (à esquerda de α_2) para tentar encontrar um ponto melhor. No Passo 5 diminui-se o tamanho do passo e este é comparado com um valor mínimo pré-definido. Se for maior, coloca-se o melhor ponto obtido nos Passos 3 e 4 na lista S_0 para pesquisa futura. Caso contrário, este ponto é um mínimo local encontrado com relação à vizinhança dos pontos pesquisados e colocado na lista S_2 . Termina-se, neste passo, a pesquisa à direita de α_1 em relação ao ponto de partida e a seguir, começa-se a pesquisa à esquerda deste ponto. Isto é feito nos passos 6 a 9. No Passo 10 seleciona-se o melhor dos mínimos locais encontrados para formar o melhor modelo de prioridade com dois fatores.

O algoritmo I pode ser generalizado para o caso de n fatores (ver 3.7.1) da maneira descrita a seguir.

Algoritmo II.

Passo 1. Seja um conjunto de fatores, $F = \{F_1, F_2, \dots, F_n\}$, um conjunto de parâmetros $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ e uma lista S inicialmente vazia. Seja z o tempo de atraso, M o número máximo de iterações, e δ o tamanho do passo de busca. Faça $k = 0$, $P_0 = P$, e $\alpha_{i,0} = \alpha_i$.

Passo 2. Selecione um par de fatores $F_i, F_j \in F$ e $(F_i, F_j) \notin S$. Caso não exista, fim.

Passo 3. Utilize o algoritmo I para obter os melhores parâmetros α_i^*, α_j^* , associados aos fatores F_i, F_j . Defina $\alpha_{r,k+1}$ ($r = 1, 2, \dots, n$) e obtenha um novo modelo P_{k+1} da seguinte forma:

$$\begin{aligned}\alpha_{i,k-1} &= \alpha_i^* \\ \alpha_{j,k-1} &= \alpha_j^* \\ \alpha_{r,k-1} &= \alpha_{r,k} \quad (r \neq i, r \neq j) \\ P_{k+1} &= \sum_{i=1}^n \alpha_{i,k+1} \times F_i\end{aligned}$$

Passo 4. Adicione o par de fatores (F_i, F_j) na lista S. Faça $k = k + 1$.

Se $k \leq M$, vá ao Passo 2

Senão, fim.

Observe que o algoritmo II define o melhor par de parâmetros quando os outros estão fixados. Esta simplificação foi feita para evitar um algoritmo mais complexo que procura determinar todos os melhores parâmetros simultaneamente.

A seguir, aplica-se o algoritmo II ao modelo P1 para obter soluções de melhor qualidade com relação a um dado critério, no caso em que o fator de atraso γ é igual a 7, pois este é o único caso em que o modelo P1 não supera a regra MDD em atraso. Neste caso, o seu índice multiobjetivo IM também é superado pela regra EDD.

O algoritmo é utilizado para dois critérios: atraso total e índice multiobjetivo IM. Em ambos os casos, o tamanho mínimo do passo de busca, δ , é 0,0001.

Para o critério de atraso total, a tabela 3.7.17 apresenta os resultados originais de P1, os resultados obtidos pelo algoritmo, representados por P1*, e os resultados da melhor regra com relação ao atraso, isto é, a regra MDD. Observa-se que o algoritmo conseguiu melhorar significativamente o atraso com uma redução de 14,1% em relação a P1 e 10,71% em relação a MDD. Nota-se também que um melhor desempenho em termos de *makespan* e índice multiobjetivo.

Para o índice multiobjetivo IM, a tabela 3.7.18 mostra os resultados originais de P1, os resultados obtidos pelo algoritmo, representados por P1**, e os resultados da melhor regra com relação ao índice multiobjetivo, isto é, a regra EDD. Observa-se que o algoritmo conseguiu melhorar significativamente IM com uma redução de 18,26% em relação a P1 e 13,2% em relação a EDD. A redução de IM acarreta um acréscimo no atraso e uma redução de *makespan*.

As Figuras 3.7.25 e 3.7.26 ilustram os valores de atraso e índice multiobjetivo encontrados durante a busca. O tempo computacional para obter uma melhor solução depende do número de iterações do algoritmo II. Cada iteração implica a resolução do problema de programação. A tabela 3.7.19 mostra os valores dos parâmetros dos modelos P1, P1* e P1**.

Tabela 3.7.17 Atraso das prioridades

Prioridade	$\gamma = 7$			Redução de P1* em atraso (%)
	Atraso	Makespan	Índice IM	
P1	2823	4217	2317	14,1%
P1*	2425	4216	2286	-
MDD	2716	4238	2522	10,71%

Tabela 3.7.18 Índice IM das prioridades

Prioridade	$\gamma = 7$			Redução de P1* em índice IM (%)
	Atraso	Makespan	Índice IM	
P1	2823	4217	2317	18,26%
P1**	2998	4174	1894	-
EDD	2730	4204	2182	13,2%

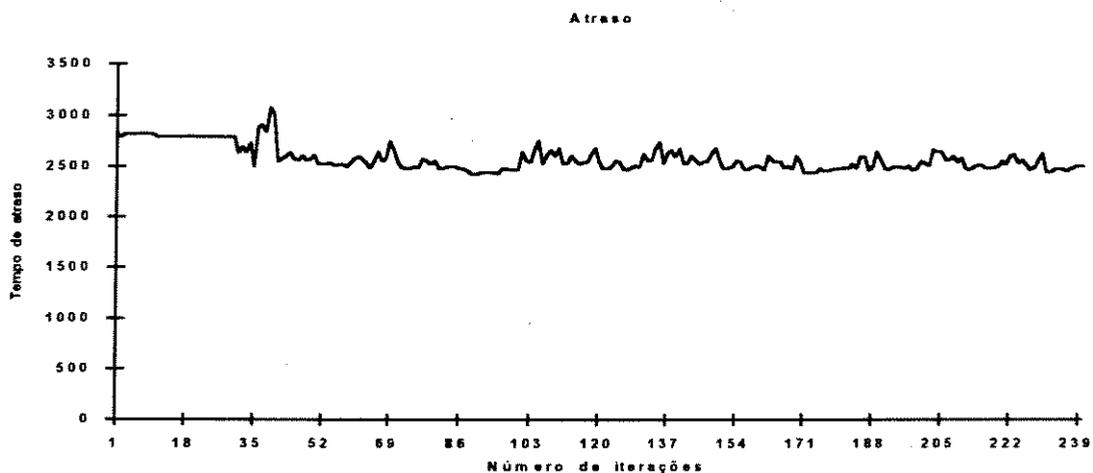


Figura 3.7.25 Atraso com o modelo P1

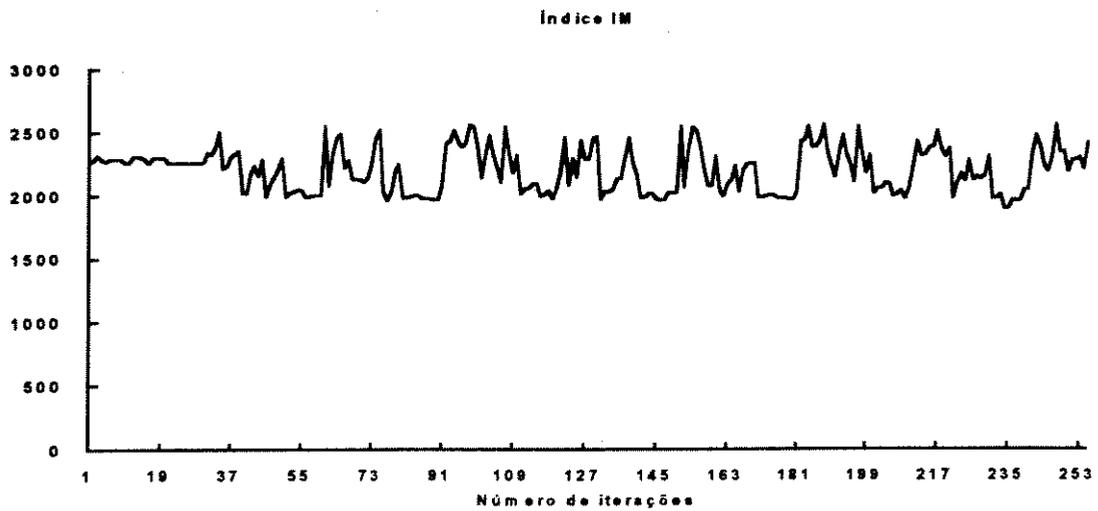


Figura 3.7.26 Índice IM com o modelo P1

Tabela 3.7.19 Variações dos parâmetros

	$\alpha(\text{MDD})$	$\alpha(\text{EDD})$	$\alpha(\text{SRT})$	$\alpha(\text{SPT})$	$\alpha(\text{LTT})$
P1	0,3476	0,1263	0,0786	0,0542	0,0564
P1*	0,4609	0,1063	0,0289	0,0542	0,0128
P1**	0,2446	0,1763	0,1291	0,0542	0,0589

CAPÍTULO 4

PROGRAMAÇÃO DO SISTEMA DE TRANSPORTE DE MATERIAIS

4.1 Introdução

O Sistema de Transporte de Materiais (STM) é uma parte importante num Sistema Flexível de Manufatura (SFM). A sua função principal é mover as peças e outros materiais necessários de um local para outro, de maneira segura, eficiente e com precisão, ou seja, material correto, em quantidade correta, no local correto. A variedade de STMs instalados nas indústrias modernas é muito grande. Detalhes sobre os diferentes tipos de STMs, como sistemas de AGV (Automated Guided Vehicle) e de esteira, podem ser encontradas em Askin e Standridge (1993), Maleki (1991), Greenwood (1988), Groover (1987), Stecke e Browne (1985), Apple (1972).

Segundo Stecke e Browne (1985), a maioria dos SFMs instalados empregam os sistemas de AGV, que são sistemas de transporte de materiais que utilizam os veículos auto-guiados como transportadores. O conceito de AGV foi primeiro aplicado a um sistema de manufatura de grandes volumes em 1974, numa fábrica da Volvo em Kalmar, Suécia (Ozden 1988). Hoje, pelas suas grandes vantagens em alta automatização e flexibilidade, os sistemas de AGV são obviamente uma

boa escolha para os SFMs. A instalação dos veículos auto-guiados melhorou significativamente o processo de transporte dos materiais nos sistemas de manufatura, trazendo muitos problemas desafiadores do ponto de vista de projeto e operação dos sistemas.

Este capítulo aborda o aspecto operacional do STM baseado em AGV. Considera-se seu problema de programação e controle de um ponto de vista novo, que consiste da semelhança com a programação das operações de um sistema flexível de produção. O método analítico de estado (MAE) é aplicado para resolver em detalhe o problema de programação e controle na rede de transporte. Modelos de prioridade linear combinada são construídos para definir as transições indiretas no processo de resolução do problema com o MAE. O algoritmo de busca apresentado no capítulo 3 é usado para obter soluções melhores em casos particulares. Finalmente, um simulador é desenvolvido para programar o sistema.

O capítulo está organizado da seguinte forma. A seção 4.2 contém uma revisão bibliográfica. A seção 4.3 descreve o problema da programação de STM. Na seção 4.4 é aplicado o método analítico de estado para modelar e resolver a programação de STM. A seção 4.5 descreve modelos de prioridade combinada para tomada de decisões. A seção 4.6 apresenta e analisa os resultados computacionais, e utiliza o algoritmo de busca apresentado no capítulo 3 para obter modelos alternativos melhores.

4.2 Revisão Bibliográfica

O sistema de transporte de materiais baseado em AGV é um sistema que utiliza somente AGVs como transportadores internos. Os movimentos dos AGVs são guiados por uma rede de transporte física. Geralmente, a rede é formada por fios embutidos no chão ou faixas pintadas sobre o chão. As rotas da rede podem ser unidirecionais (por exemplo, Egbelu e Tanchoco 1984) ou bidirecionais (por exemplo, Egbelu e Tanchoco 1986). Comparações entre estes dois tipos

mostram que os sistemas bidirecionais são significativamente mais eficientes que os sistemas unidirecionais (Egbelu e Tanchoco 1986, Kim e Tanchoco 1993).

Groover (1987) classifica os sistemas de AGVs em três tipos diferentes :

- Trem auto-dirigido: os AGVs funcionam como rebocadores e puxam um ou mais carros.
- AGV “truck pallet”: os AGVs são utilizados para movimentar as cargas paletizadas numa rota pré-definida. Sua capacidade de carga pode chegar a três toneladas.
- Transportador de carga unitária: este tipo de AGV é utilizado para movimentar cargas de uma estação para outra. Inclui os AGVs de carga leve (tipicamente até 250 kg.) e os AGVs de montagem que substituem as esteiras nos sistemas convencionais. Este trabalho concentra-se nos AGVs de carga leve.

Programar os AGVs e as cargas transportadas corretamente é um ponto chave para a eficiência e segurança do sistema de transporte. Na literatura, há duas formas básicas para tratar a programação do STM: a primeira maneira resolve a programação na etapa do projeto do sistema, antes de sua implementação (Maxwell e Muckstadt 1982, Mahadevan e Narendran 1990, Malmberg 1990); a segunda, resolve a programação na etapa de operação do sistema, baseado num *layout* da rede pré-definido (Egbelu e Tanchoco 1984, Bartholdi e Platzman 1989, Ulusoy e Bilge 1993).

Mesmo na etapa de projeto, há necessidade de se tomar várias decisões referentes à programação dos AGVs, tais como o controle de trânsito e regras de despacho dos AGVs. Estas decisões são importantes para projetar um sistema baseado em AGVs (Mahadevan e Narendram 1990). Além disso, programar e simular os processos de funcionamento do sistema projetado, pode ajudar na verificação e validação do projeto antes de sua implementação.

Maxwell e Muckstadt (1982) foram um dos pioneiros no estudo de problemas de projeto e operação dos sistemas de AGV. Discutiram o papel de despacho e estimaram o número mínimo de AGVs necessários na rede. Mahadevan e Narendram (1990) discutiram o problema de

programação no nível de projeto e operação. Compararam várias estratégias de operação e várias regras de despacho de AGV por simulação, através da linguagem GPSS/PC.

Porém, alguns problemas de operação reais podem não estar bem definidos, ou até não previstos, na época do projeto. Por isso, a programação nesta etapa é bastante simples. Quando o sistema for instalado e estiver em funcionamento, é necessário elaborar melhor a programação. Para que possamos operar um sistema caro e bem projetado de forma mais eficiente, flexível e segura, muitas pesquisas vem sendo realizadas.

Egbelu e Tanchoco (1984) foram um dos primeiros pesquisadores que analisaram os aspectos operacionais dos sistemas de transporte baseados em AGVs num ambiente de manufatura. Consideram que o problema de despacho num STM pode ser classificado em duas categorias: seleção de um dos AGVs disponíveis para atender um dado centro de trabalho e escolha de um dos centros de trabalho candidatos para ser atendido por um dado AGV disponível. Os autores desenvolveram um simulador para simular a operação de um STM unidirecional e avaliar o desempenho das regras de despacho. Testaram 3 regras, NV (Nearest Vehicle), FV (Farthest Vehicle) e LIV (Longest Idle Vehicle) para selecionar um dos AGVs disponíveis, e 5 regras, MOQS (Maximum Outgoing Queue Size), STT/D (Shortest Travel Time/Distance), LTT/D (Longest Travel Time/Distance), MROQS (Minimum Remaining Outgoing Queue Space) e MFCFS (Modified First Come First Serve) para selecionar um dos centros candidatos a ser atendido. O critério considerado é o número de cargas transportadas num intervalo de tempo. Os resultados computacionais mostram que o critério é pouco sensível com as regras de seleção de AGV. O melhor desempenho com relação à seleção dos centros candidatos é obtido pela regra MFCFS num ambiente com capacidade de buffer infinita. Os autores consideram que a prioridade das peças em um centro de trabalho é igual. Além disso, o problema de programação e controle de trânsito dos AGVs não é considerado. Egbelu e Tanchoco (1986) estenderam este trabalho para um sistema de transporte bidirecional e concluíram que este apresenta um desempenho superior quando comparado com o sistema unidirecional. Bartholdi e Platzman (1989) e Han e McGinnis (1989) consideram a programação em casos particulares. O primeiro analisa a situação em que o

AGV visita os centros sempre na mesma seqüência e o segundo considera um ambiente com um único AGV.

Nos últimos anos, o sistema de transporte de materiais começou a ser considerado na programação de produção. Raman et al. (1986) ligaram a programação de AGV à programação de máquina. No seu trabalho, eles supõem que os AGVs sempre voltam para a estação de carga/descarga depois de completar o transporte de uma carga unitária entre duas máquinas. O tempo de transporte entre as duas estações é considerado como o tempo de uma operação numa máquina fictícia. O problema de programação integrada (transporte e produção) é formulado como um problema de programação inteira e o objetivo é minimizar o atraso total das peças. Como o modelo de programação inteira é intratável, os autores propõem a utilização da regra de despacho MOD (Modified Operation Due Date) para as peças. Em seu experimento computacional, os autores consideram 20 tipos de peça, 5 máquinas, um AGV e 80% de utilização do sistema. Dois casos são comparados: máquinas balanceadas e máquinas não balanceadas. Os resultados de simulação com 4200 peças mostram que é melhor balancear máquinas.

Sabuncuoglu e Hommertzheim (1989) estenderam o trabalho de Egbelu e Tanchoco (1984) para incluir o processo de produção ao processo de transporte de AGVs. Utilizam um SFM imaginário para analisar o desempenho de regras de despacho das peças e dos AGVs. Uma regra de despacho de AGV seleciona uma máquina. O sistema contém 6 máquinas, uma estação de lavagem, uma estação de inspeção, e dois AGVs. A simulação é feita pela linguagem de simulação SIMAN. Um período de *warm-up* com 300 peças e um período de operação de 3000 peças adicionais são considerados e o objetivo consiste em minimizar o tempo de fluxo médio. As regras consideradas para as máquinas são SPT, SPT*TOT (tempo de processamento total da peça), SRT, FCFS e RANDOM. As regras para os AGVs são FCFS, STD (shortest travel distance), LQS (Largest Queue Size) e LOQS (Largest Outgoing Queue Size). A regra LQS considera a fila de entrada e saída das máquinas. Os resultados mostram que SPT domina as outras regras quando a regra FCFS é utilizado para despachar os AGVs. Dentro as regras de AGV, LQS domina as demais.

Sabuncuoglu e Hommertzheim (1992a) utilizaram o mesmo SFM imaginário e elaboram um procedimento *on-line* complexo para tratar o problema de despacho dos AGVs e das peças. Um algoritmo é utilizado para alocar um AGV a uma peça quando o AGV completa o transporte da peça anterior, e alocar uma máquina a uma peça quando a máquina termina a operação da peça anterior. Várias informações do sistema e regras de despacho simples, como STD (Shortest Travel Distance), SRT e EDD, são ordenadas em uma hierarquia e utilizadas, em uma ordem definida, para alocar peças nas filas de saída das máquinas aos AGVs. A regra MOD é utilizada para despachar as peças nas filas de entrada das máquinas quando o objetivo é minimizar o atraso das peças. Se o objetivo é minimizar o tempo de fluxo, utiliza-se uma combinação da regra SPT com o tempo de espera previsto na próxima máquina. O algoritmo foi implementado na linguagem de simulação SIMAN. Resultados computacionais mostram que o algoritmo é superior quando comparado com duas regras de alocação dos AGVs (SDT e LQS) e duas regras de despacho das máquinas (SPT e MOD). Comparações entre regras simples no mesmo SFM mencionado acima podem ser encontradas em Sabuncuoglu e Hommertzheim (1992b).

Ulusoy e Bilge (1993) tratam o problema de programação da produção e de transporte por AGVs de forma integrada. Os autores propõem um método heurístico que decompõe o problema em dois subproblemas, programação de máquina e programação de veículo (*vehicle scheduling problem*), que são resolvidos separadamente. A programação de máquina é resolvido através do algoritmo de geração de programas ativos proposto por Giffler e Thompson em 1960 e do algoritmo de geração de programas sem atraso (*non-delay*, ver Baker 1974), utilizando três regras de despacho, SPT, SRT e LRT para reduzir o número de programas pesquisados. O programa obtido desta forma em ambos os casos é uma solução sub-ótima com relação ao *makespan* dentro das classes dos programas ativos e sem atraso. A programação das máquinas determina janelas de tempo tais que o instante de início mais cedo da viagem do AGV corresponde ao término de uma operação prévia e o instante de início mais tarde é tal que a operação sucessora seja completada no tempo programado. Portanto, a programação dos AGVs é resolvida como um problema de programação de veículos com janelas de tempo, que originou-se do estudo da programação de ônibus escolar (ver, por exemplo, Desrosiers et al. 1986, Ferland e Fortin 1989). Um procedimento iterativo entre os sub-problemas é proposto com o objetivo de obter uma solução melhor. Os autores

consideram 4 *layouts* de transporte com 4 máquinas, dois AGVs, e 5 conjuntos de peça com até 6 peças por conjunto nos exemplos. Testam 22 combinações de layouts de transporte e conjuntos das peças para comparar os dois algoritmos citados acima, com a variação das três regras de despacho mencionadas, em um procedimento não iterativo (*single pass*). O melhor resultado para cada combinação obtida nesta fase é comparado com a melhor solução obtida através do procedimento iterativo. Concluem que o procedimento iterativo é melhor que o não iterativo.

Observa-se que na literatura de programação de STM, a pesquisa é concentrada em sistemas onde a rota de transporte de um ponto para outro é fixa. De nosso conhecimento, não existe trabalho que aborde sistemas com rotas alternativas. Além disso, a maioria dos trabalhos trata o problema de programação de STM como uma questão de alocação de AGVs aos centros de trabalho e não considera o controle do tráfego dos AGVs na rede de transporte. O método mais utilizado para tratar a programação de STM é simulação e específica para um sistema particular.

Neste trabalho, o problema de programação de STM é tratado como um problema de programação da produção e resolvido pelo Método Analítico de Estado (MAE), proposto no capítulo 2. Consideramos sistemas de transporte dinâmicos, isto é, a escolha da rota de cada viagem de AGV é feita dinamicamente ao longo do seu caminho. Aplicamos o MAE para programar e controlar em detalhe os movimentos dos AGVs na rede de transporte. Alocamos os AGVs diretamente para as cargas e não simplesmente para os centros de trabalho. As idéias iniciais desta abordagem foram apresentadas em Wu e Armentano (1992b).

4.3 Descrição de Problema

A programação do STM é um problema novo e importante num ambiente de manufatura automatizado e informatizado. A operação deste sistema deve ser eficiente para justificar o alto investimento envolvido. A programação do STM trata do seguintes problemas:

- seleção de AGVs disponíveis
- seleção das cargas que estão esperando nos centros de trabalho
- seleção da rota de transporte
- programação e controle de transporte das cargas

A seguir, introduzimos alguns conceitos básicos e restrições necessárias para a definição do problema de programação do STM.

4.3.1 Conceitos básicos e hipóteses

Conceitos básicos

Carga unitária: é uma peça típica ou um lote de peças que é tratado como uma unidade.

Carga de AGV: é uma carga unitária que precisa ser transportada de um ponto de retirada para um ponto de entrega através de um AGV. Os pontos de retirada e entrega podem ser um centro de trabalho ou uma estação de entrada/saída.

Note que uma carga de AGV é definida a partir do instante que a peça solicita um AGV para transporte até o ponto desejado. Caso uma peça seja processada em mais de um centro de trabalho, é necessário solicitar mais de uma vez um AGV para transportá-la, e cada solicitação gera uma carga de AGV.

Zona de controle: é uma pequena vizinhança de um ponto de retirada, de entrega, ou de interseção de segmentos da rede de transporte. A função da zona é facilitar o controle de tráfego e evitar colisão de veículos.

Segmento: é um caminho na rede de transporte que liga dois pontos adjacentes dentre os mencionados acima. Um exemplo simples com cinco zonas e quatro segmentos é mostrado na Figura 4.1.

Carga de Segmento: em geral, um AGV com uma carga percorre diversos segmentos entre o seu ponto inicial e final. Uma carga de segmento é definida num dado segmento do seu caminho. Caso uma carga de AGV passe por n segmentos, dizemos que esta carga é dividida em n cargas de segmento.

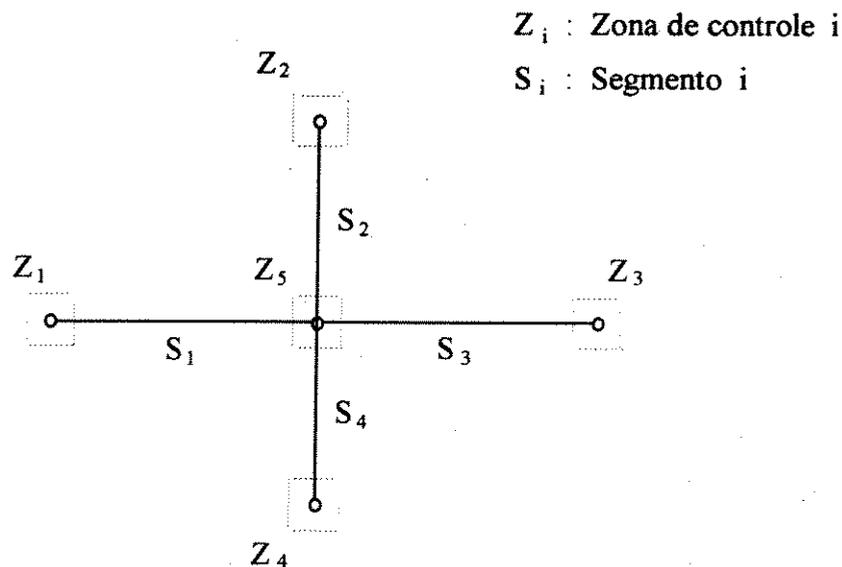


Figura 4.1 Exemplo de zonas de controle e segmentos

Hipóteses principais

Consideramos as seguintes hipóteses para o STM:

- Somente uma carga de AGV é permitida atravessar uma zona de controle para evitar conflito dos AGVs e simplificar o controle de trânsito. Em geral, o tempo para atravessar uma zona de controle é bem menor que o tempo de transporte num segmento.

- As zonas de controle têm seus buffers próprios, com capacidade suficiente para armazenar as cargas de AGV bloqueadas. Qualquer carga de AGV pode ser liberada dos buffers em primeiro lugar. Um exemplo deste tipo de buffer é mostrado na Figura 4.2.
- A viagem sem carga de um AGV é tratada como uma carga de AGV normal.
- Quando um AGV é alocado a uma carga, ele transporta a carga do ponto de retirada até o ponto de entrega desejado.
- Quando um AGV começa a movimentar-se num segmento da rede de transporte, ele continua o movimento sem parar até o ponto final do segmento.
- Quando dois AGVs estão se movimentando no mesmo sentido de um segmento, há um intervalo de tempo entre eles para evitar conflitos.
- A rede de transporte é bidirecional e consiste de dois segmentos unidirecionais (Egbelu e Tanchoco, 1986).
- Os intervalos de tempo entre pedidos de transporte são variáveis aleatórias independentes com distribuição exponencial.
- O ponto de estacionamento de cada AGV é o ponto de entrega da última carga atendida. No início, todos os AGVs estão num ponto reservado do sistema.

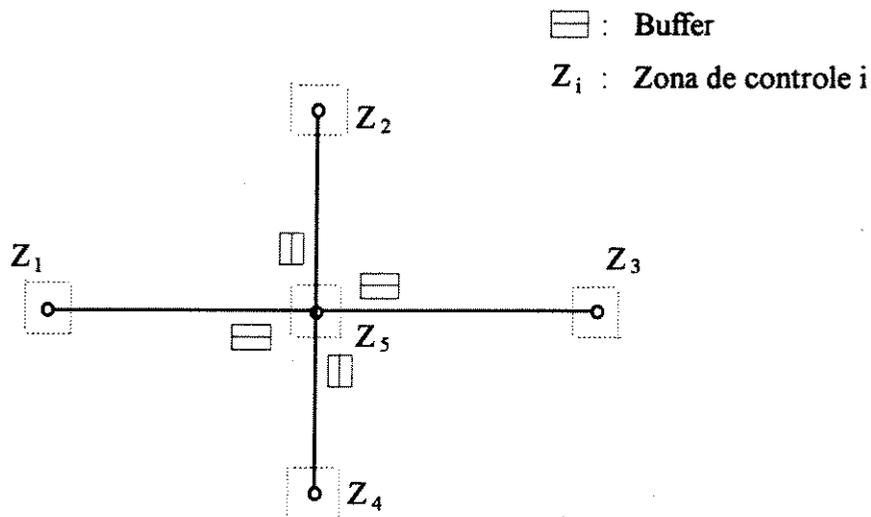
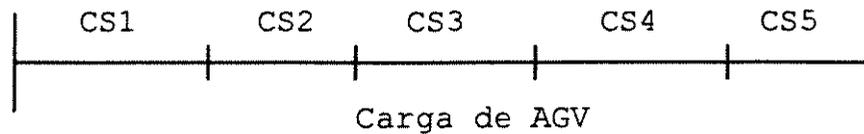


Figura 4.2. Um exemplo de zona de controle com buffers

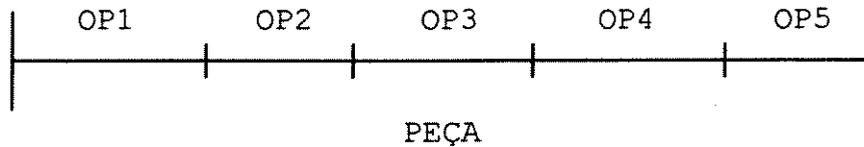
4.3.2 Semelhança com um sistema de produção flexível

A Figura 4.3 (a) mostra uma carga de AGV composta das cargas de segmento (CS1, CS2, ..., CS5). Esta seqüência ordenada das cargas de segmento consiste de uma das possíveis rotas entre os pontos de retirada e entrega de uma carga. A carga de AGV pode ser vista como uma peça e cada carga de segmento pode ser vista como uma operação da peça num sistema de produção flexível, onde existem rotas alternativas de processamento. A Figura 4.3 (b) mostra a seqüência ordenada de operações (OP1, OP2, ..., OP5) da peça.

A programação de cargas de segmento consiste em seqüenciar as cargas de segmento em cada zona de controle e determinar os instantes de início e término de transporte das cargas na zona e no seu próximo segmento. Isto é equivalente a programar as operações das peças num centro de trabalho, ou seja, determinar a seqüência de processamento das operações e o início e término das mesmas.



(a) Uma carga de AGV e suas cargas de segmento



(b) Uma peça e suas operações

Figura 4.3 Associação entre a carga de segmento e operação de uma peça

4.3.3 Definição do problema

Num ambiente de manufatura, existem diversas solicitações de AGV para movimentar as peças de um ponto para outro na rede de transporte. Os pedidos podem ocorrer em pontos e instantes distintos. É necessário portanto, programar o sistema de transporte de materiais. Este problema pode ser descrito da seguinte forma.

Sejam n cargas de AGV, m segmentos, l zonas de controle, g AGVs e p pontos de retirada, de entrega ou de interseção num sistema de transporte. Cada carga de AGV precisa ser transportada de um ponto de retirada até um ponto de entrega, podendo haver vários caminhos para realizar o transporte. Para cada caminho, a carga de AGV pode ser dividida em várias cargas de segmento, havendo uma relação de precedência entre elas, isto é uma carga só pode ser transportada num dado segmento se tiver percorrido todos os seus segmentos predecessores. O problema de programação de STM envolve a alocação de AGVs às peças, determinação do caminho de transporte e programação das cargas de segmento.

Neste trabalho, vamos considerar que os intervalos de tempo entre pedidos de transporte são variáveis aleatórias e independentes com distribuição exponencial. Isto caracteriza um processo dinâmico de pedidos. No entanto, o método aqui proposto também pode ser utilizado para o caso em que os tempos entre os pedidos de transporte são constantes.

4.3.4 Os subproblemas de programação do STM

A programação do STM envolve os seguintes subproblemas:

- seleção da carga;
- seleção de AGV;
- seleção de caminho (rota) de transporte;
- programação de carga de segmento.

Seleção da carga: consiste em selecionar uma das cargas que estão solicitando um AGV para transporte. Neste trabalho, a seleção de carga é feita por um modelo de prioridade combinada que será apresentado na próxima seção.

Seleção do AGV: consiste em selecionar um dos AGVs disponíveis para atender a carga selecionada. Utilizamos a regra do veículo mais próximo ao ponto de pedido, como em Egbelu e Tanchoco (1984).

Seleção do caminho: consiste em selecionar um dos caminhos alternativos para transportar a carga de AGV na rede de transporte. A definição do caminho é feita de forma dinâmica, isto é, o próximo segmento de transporte de uma carga de AGV é determinado no fim do segmento anterior. A determinação de cada segmento é feita através de uma lista de rotas alternativas pré-ordenadas. O próximo segmento de uma carga é determinado por sua primeira alternativa. Caso ocupado, seleciona-se a segunda alternativa, e assim por diante. Se todas rotas alternativas estão ocupadas, a carga tem que esperar até que uma rota fique disponível.

Programação de carga de segmento: consiste em seqüenciar as cargas de segmento em cada zona de controle e determinar os instantes de início e término de transporte das cargas na zona e no seu próximo segmento.

4.4 O Método Analítico de Estado Para o Problema de Programação do Sistema de Transporte

Como descrito anteriormente, uma carga de AGV consiste, em geral, de várias cargas de segmento. Cada carga de segmento corresponde ao transporte da carga num segmento definido. A partir da programação das cargas de segmento, obtemos a programação das cargas de AGV, bem como a programação completa do STM. Por isso, a carga de segmento é uma entidade básica do sistema de transporte. Nesta seção, aplicamos o método analítico de estado para tratar a programação do STM. Definimos e analisamos os estados, as transições de estado no processo de transporte das cargas de segmento, e construímos o modelo MAE para obter a resolução do problema.

4.4.1 Os estados do processo

Conforme o processo de transporte, as cargas de segmento envolvem os seguintes estados:

1) Estado de bloqueio (BL): representa o estado de uma carga de segmento no qual a mesma está bloqueada para transporte devido a restrições de relação de precedência. O estado é terminado quando todas as suas cargas precedentes estão completadas.

2) **Estado de espera (ES)**: indica o estado no qual uma carga de segmento está pronta e esperando ser atendida por um dos AGVs disponíveis para transporte. Este estado ocorre somente para a primeira carga de segmento de uma carga de AGV.

3) **Estado de AGV**: representa o processo de escolha de um dos AGVs disponíveis e chegada do AGV selecionado.

4) **Estado de buffer (BF)**: representa o estado no qual uma carga de segmento está esperando ser selecionada para atravessar uma zona de controle.

5) **Estado de zona (ZN)**: indica o estado no qual uma carga de segmento está atravessando a zona de controle e bloqueando a mesma.

6) **Estado de transporte I (TP1)**: denota o estado no qual uma carga de segmento está sendo transportada num dado segmento durante um intervalo de tempo no início de segmento. Este intervalo corresponde à distância mínima pré-especificada entre dois AGVs adjacentes. Neste intervalo, outras cargas são proibidas de entrar neste segmento.

7) **Estado de transporte II (TP2)**: representa o estado no qual uma carga de segmento está sendo transportada num dado segmento e este se encontra disponível para receber outras cargas.

8) **Estado de chegada (CG)**: indica o estado no qual uma carga de segmento já chegou ao ponto final do mesmo. O processo de transporte desta carga de segmento é terminado.

4.4.2 Análise dos estados

Estado BL: é um estado inicial de uma carga de segmento. Esta carga sai desse estado no momento em que todas as suas cargas de segmento precedentes foram terminadas. Como é um estado sem recurso, a transição a partir deste não requer nenhuma decisão.

Estado ES: neste estado, a carga de segmento está competindo com outras cargas pelos AGVs. A carga de segmento só sai da fila de espera quando tem a maior prioridade e há um AGV disponível. Caso contrário, a carga tem que esperar na fila por um período não definido. Portanto, é um estado de tempo indeterminado e de espera de recurso.

Estado AGV: neste estado, é necessário escolher um dos AGVs disponíveis para atender a carga de segmento selecionada. Quando o AGV é determinado, a carga deve esperar a chegada do AGV a partir do seu ponto de estacionamento. O movimento do AGV vazio é considerado como uma carga de AGV normal e o tempo deste movimento depende da situação da rede de transporte. Logo, é um estado de tempo indeterminado e de seleção de recurso.

Estado BF: neste estado, a carga de segmento está competindo com outras cargas pela mesma zona de controle. A carga de segmento só pode atravessar a zona quando tem a maior prioridade e a zona está disponível. Caso contrário, a carga tem que esperar na fila por um período não definido. Por isso, é um estado de tempo indeterminado e de espera de recurso.

Estado ZN: uma carga de segmento leva um tempo pequeno e constante para atravessar uma zona de controle. Depois deste tempo, a carga sai do estado imediatamente. Como a carga ocupa um AGV e uma zona, o estado é de tempo determinado e com recurso.

Estados TP1 e TP2: uma carga de segmento fica um tempo constante em cada um dos dois estados e sai destes imediatamente depois de terminar os tempos correspondentes. Como utiliza um AGV e um segmento de transporte, os estados TP1, TP2 são de tempo determinado e com recurso.

Estado CG: é o estado final de uma carga de segmento. Portanto, não há transição a partir deste.

4.4.3 Análise das transições

O processo de transporte de uma carga de segmento num STM é considerado como um processo de transição de estados. As transições mudam o estado de bloqueio da carga de segmento até o seu estado de chegada. Quando terminam-se todas as transições, o transporte da carga neste segmento é completado. Nesta seção, vamos analisar, em detalhe, cada transição de estado.

1) Transição do BL para o ES

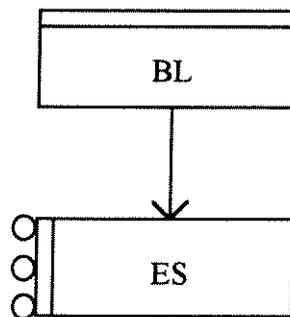


Figura 4.4.1 Transição do estado BL para ES

Nesta transição, como mostrada na Figura 4.4.1, o estado de entrada BL é o estado inicial do processo. Como discutido na seção anterior, a transição a partir dele não requer decisão, e portanto, é uma transição direta.

2) Transição do BL ao BF

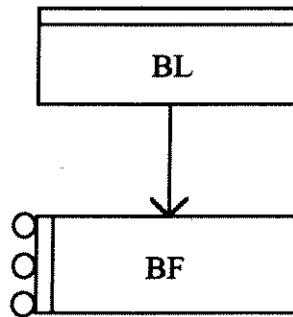


Figura 4.4.2 Transição do estado BL para BF

A transição está apresentada na Figura 4.4.2. Como no caso anterior, a transição a partir do estado BL não requer decisão, e por isso, é uma transição direta.

3) Transição do ES para o AGV

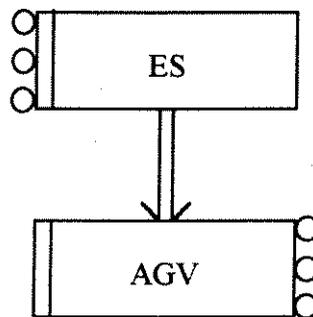


Figura 4.4.3 Transição do estado ES para AGV

O estado de entrada da transição mostrada na Figura 4.4.3 é um estado de tempo indeterminado e de espera de recurso. Portanto, a transição é indireta.

4). Transição do AGV para o BF

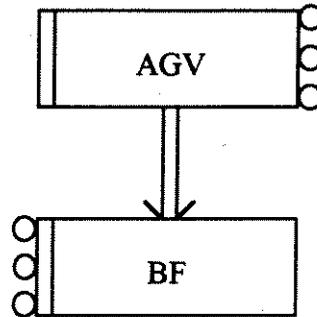


Figura 4.4.4 Transição do estado AGV para BF

Nesta transição, como mostrada na Figura 4.4.4, o estado de entrada AGV é um estado de tempo indeterminado e de seleção de recurso. Por isso, a transição é uma transição indireta.

5) Transição do BF para o ZN

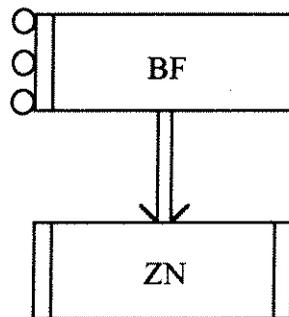


Figura 4.4.5 Transição do estado BF para ZN

O estado de entrada da transição mostrada na Figura 4.4.3 é um estado de tempo indeterminado e de espera de recurso. Conseqüentemente, a transição é indireta.

6). Transição do ZN para o TP1

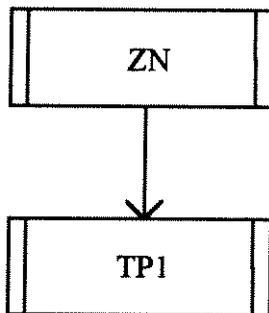


Figura 4.4.6 Transição do estado ZN para TP1

Nesta transição, como mostrada na Figura 4.4.6, o estado de entrada ZN é de tempo determinado e com recurso. Portanto, a transição é direta.

7) Transição do TP1 para o TP2

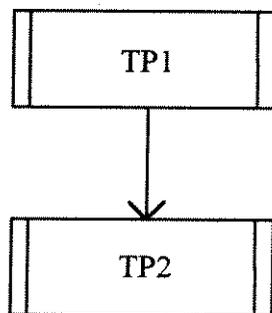


Figura 4.4.7 Transição do estado TP1 para TP2

O estado de entrada da transição mostrada na Figura 4.4.7 é de tempo determinado e com recurso. Consequentemente, a transição é direta.

7) Transição do TP2 para o CG

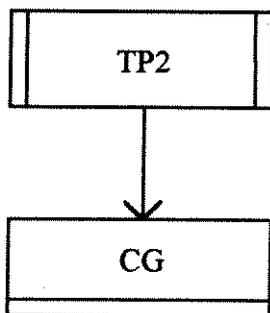


Figura 4.4.8 Transição do estado TP2 para CG

Nesta transição, como mostrada na Figura 4.4.8, o estado de entrada TP2 é de tempo determinado e com recurso. Portanto, a transição é direta.

Conforme as análises dos estados e transições, obtemos um modelo analítico de estado (MAE) para a primeira carga de segmento das cargas de AGV e um modelo para as cargas de segmento restantes. Os modelos correspondentes estão mostrados nas Figuras 4.4.9 e 4.4.10. Note que a diferença entre os modelos reside no fator de que o primeiro requer um AGV.

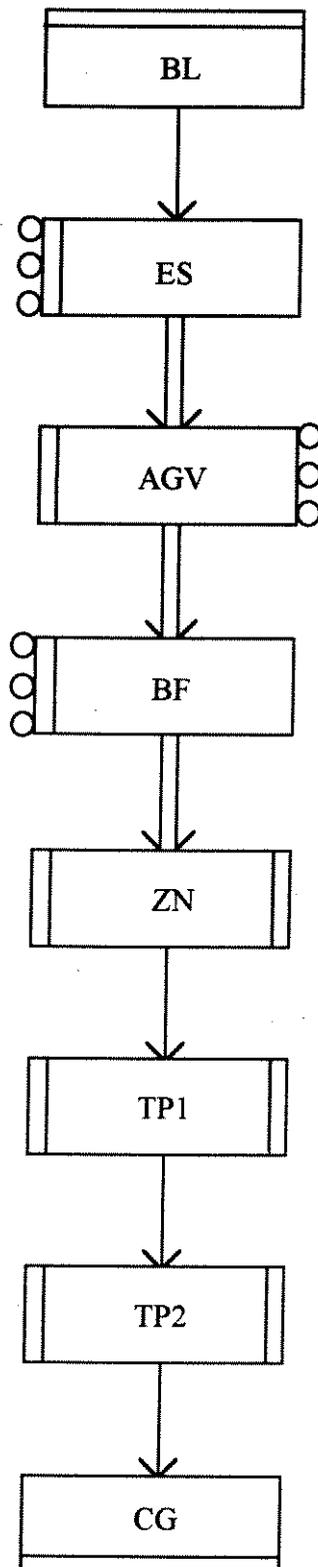


Figura 4.4.9 Modelo MAE para a primeira carga de segmento na programação de STM

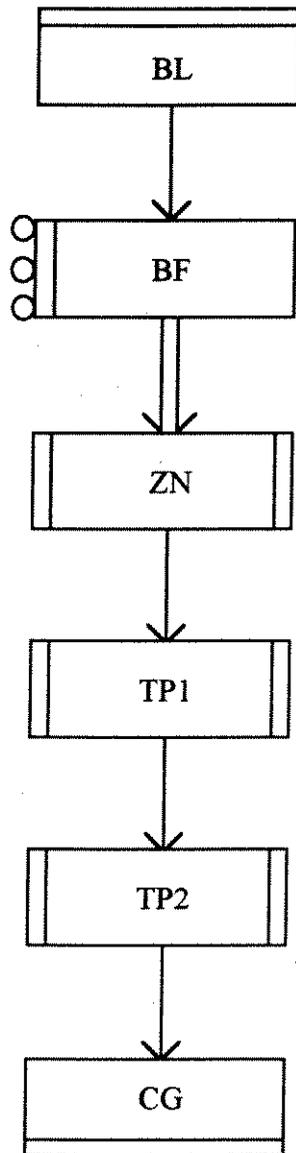


Figura 4.4.10 Modelo MAE para as cargas restantes na programação de STM

4.5 Modelo de Prioridade Linear

No processo de resolução do problema da programação do STM através do MAE, existem várias transições indiretas que serão definidas através de um modelo linear de prioridade que combina diversas regras de despacho. Neste modelo, vamos considerar um único objetivo que consiste em minimizar o tempo de atraso de todas as cargas de AGV, como mostrado em 4.5.1. Desta forma, mostramos também, a possibilidade de resolver problemas com um único objetivo através do MAE com um modelo de prioridade linear.

$$\min z = \left(\sum_{i=1}^n \max(0, C_i - d_i) \right) \quad (4.5.1)$$

onde

n : número das cargas de AGV em cada período de simulação

C_i : tempo de chegada da carga de AGV i no seu ponto de entrega

d_i : tempo máximo para a carga de AGV i chegar no seu ponto de entrega (data de entrega da carga de AGV i)

A natureza dinâmica dos pedidos de AGV requer uma simulação de longo prazo com vários períodos diferentes. Por este motivo, o *makespan* não é um critério adequado e não é considerado neste caso. Observamos também, pelas experiências de simulação nos exemplos considerados neste capítulo, que outros objetivos tais como tempo de transporte médio por carga e utilização média dos AGVs apresentam pouca variação com as regras de despacho e o modelo de prioridade.

Existem várias regras de despacho ou prioridade que têm por objetivo minimizar o atraso total. Cada regra constitui um fator de influência para um modelo de prioridade composto de uma combinação linear das regras. Neste capítulo, considera-se regras semelhantes às utilizadas no capítulo 3 e que são comparadas com os modelos de prioridade construídos. Para a apresentação destas regras, utilizamos a seguinte notação.

t	Tempo atual
n	Número total das cargas de AGV sendo transportadas em cada período de simulação
N	Número total dos AGVs no sistema
i	Índice de carga de AGV
j	Índice de carga de segmento
j(t)	j(t)-ésima carga de segmento disponível de uma carga de AGV no instante t, ou seja, todas as cargas de segmento j, $1 \leq j < j(t)$, estão terminadas
p _{ij}	Tempo de transporte da j-ésima carga de segmento da carga de AGV i
t _z	Tempo para um carga atravessar uma zona de controle; t _z é nulo no ponto de chegada da carga
NS _i	Número total de cargas de segmento associadas à carga de AGV i
TT _i	Tempo de transporte total da carga de AGV i ($TT_i = \sum_{j=1}^{NS_i} (p_{ij} + t_z)$)
RS _i (t)	Número total das cargas de segmento restantes da carga de AGV i no instante t
RT _i (t)	Tempo de transporte restante da carga de AGV i no instante t
	$(RT_i(t) = \sum_{j=j(t)}^{NS_i} (p_{ij} + t_z)$
R _{ij}	Tempo em que a j-ésima carga de segmento da carga de AGV i está disponível para transporte no sistema; R _{i1} corresponde ao instante de pedido de um AGV.
C _i	Tempo de chegada da carga de AGV i no seu ponto de entrega
d _i	Data de entrega da carga de AGV i
T _i	Tempo de atraso da carga de AGV i ($T_i = \max(0, C_i - d_i)$)
F _k	Fator correspondente à regra k

As seguintes regras são consideradas:

- o menor tempo de transporte restante SRT (Shortest Remaining Transportation Time) (F₁), seleciona a carga com o menor tempo de transporte restante, ou seja, seleciona o máximo F₁ onde $F_1 = -RT_i(t)$, $i = 1, 2, \dots, n$;

- o maior tempo de transporte restante LRT (Largest Remaining Transportation Time) (F_2), seleciona a carga com o maior tempo de transporte restante, ou seja, seleciona o máximo F_2 onde $F_2 = RT_i(t)$, $i = 1, 2, \dots, n$;
- o menor tempo de transporte STT (Shortest Transportation Time) (F_3), seleciona a carga com o menor tempo de transporte da carga de segmento disponível, ou seja, seleciona o máximo F_3 onde $F_3 = - p_{ij}(t)$, $i = 1, 2, \dots, n$;
- o maior tempo de transporte LTT (Longest Transportation Time) (F_4), seleciona a carga com o maior tempo de transporte da carga de segmento disponível, ou seja, seleciona o máximo F_4 onde $F_4 = p_{ij}(t)$, $i = 1, 2, \dots, n$;
- o menor número das cargas de segmento restantes FRS (Fewest Remaining Segment Load Number) (F_5), seleciona a carga com o menor número de cargas de segmento restantes, ou seja, seleciona o máximo F_5 onde $F_5 = - RS_i(t)$, $i = 1, 2, \dots, n$;
- o maior número das cargas de segmento restantes LRS (Largest Remaining Segment Load Number) (F_6), seleciona a carga com o maior número de cargas de segmento restantes, ou seja, seleciona o máximo F_6 onde $F_6 = RS_i(t)$, $i = 1, 2, \dots, n$;
- a menor data de entrega EDD (Earliest Due Date) (F_7), seleciona a carga com a menor data de entrega, ou seja, seleciona o máximo F_7 onde $F_7 = - d_i$, $i = 1, 2, \dots, n$;
- o menor tempo de folga SLK (Slack Time) (F_8), seleciona a carga com o menor tempo de folga, ou seja, seleciona o máximo F_8 onde $F_8 = - (d_i - t - RT_i(t))$, $i = 1, 2, \dots, n$;
- a menor data de entrega modificada MDD (Modified Due Date) (F_9), seleciona a carga com a menor data de entrega modificada, ou seja, seleciona o máximo F_9 onde $F_9 = - \max(d_i, t + RT_i(t))$, $i = 1, 2, \dots, n$;
- FIFO (First In First Out) (F_{10}), seleciona a carga que chega em primeiro lugar na fila, ou seja, seleciona o máximo F_{10} onde $F_{10} = - R_{ij}$, $i = 1, 2, \dots, n$;

A partir destes fatores, podemos construir um modelo de prioridade P da seguinte forma:

$$P = \sum_{i=1}^r a_i \times F_i \quad (4.5.2)$$

onde

$$0 \leq \alpha_i \leq 1$$

r : número total dos fatores considerados no modelo

A prioridade dada pelas regras e pelo modelo P é utilizada para definir as transições indiretas a partir dos estados de espera e de buffer. Cada carga de segmento nestes estados está esperando numa fila de uma certa zona de controle para ser alocada a um AGV ou liberada para atravessar a zona. Quando um AGV está disponível para transporte, calcula-se a prioridade de cada carga de segmento na fila de espera e aquela com maior prioridade é selecionada para ser alocada ao AGV. Neste caso, a transição muda a carga do seu estado de espera para o seu próximo estado. Caso contrário, a carga continua esperando na fila. Da mesma forma, quando uma zona de controle está livre, define-se um valor de prioridade para cada carga de segmento nos buffers desta zona e a aquela com maior prioridade é liberada para atravessar a zona. Ou seja, a transição muda a carga do seu estado de buffer para o seu próximo estado.

Para poder definir o modelo de prioridade representado na equação 4.5.2, vamos analisar o desempenho das regras com relação ao tempo de atraso das cargas, através de simulação computacional. Para tal, utiliza-se um sistema de transporte proposto por Egbelu e Tanchoco (1986) descrito na seção 4.6.

Considera-se que o número de AGVs, N, no sistema varia de 6 até 9. Para cada regra e cada valor de N, inclui-se um período de *warm-up* e 40 períodos de simulação computacional. O período de *warm-up* é um período inicial da simulação que é descartado na avaliação do desempenho da programação para evitar a instabilidade no início do processo. 1000 cargas iniciais são simuladas e descartadas neste período. Em cada um dos 40 períodos seguintes, consideram-se 500 cargas de AGV e 80% de utilização dos AGVs. O número total das cargas de AGVs no sistema é controlado para evitar sobrecarga no sistema. No final, a média de desempenho nos 40 períodos é calculada.

O número de AGVs no sistema é um parâmetro importante tanto para o desempenho do sistema quanto para a determinação dos outros parâmetros no sistema. A equação (4.5.3) mostra a definição da utilização do sistema de transporte que é análoga à definição da utilização de um sistema de produção (Conway, 1967).

$$\rho = \frac{\lambda \times (\bar{p} + \bar{p}_v)}{N}, \quad (4.5.3)$$

onde

\bar{p} : tempo médio de transporte das cargas de AGV no sistema

\bar{p}_v : tempo médio de transporte dos AGVs vazios no sistema

ρ : utilização do sistema de transporte

λ : taxa de chegada dos pedidos, que caracteriza a distribuição exponencial dos tempos entre as chegadas dos pedidos.

A taxa de chegada λ e os tempos \bar{p} e \bar{p}_v foram obtidos através de várias simulações para se obter um nível de utilização de 80%, para um dado número de AGVs N.

A data de entrega de cada carga de AGV é definida a partir do seu tempo de transporte total (TT_i) e do fator de data de entrega (γ), como mostra a expressão 4.5.4.

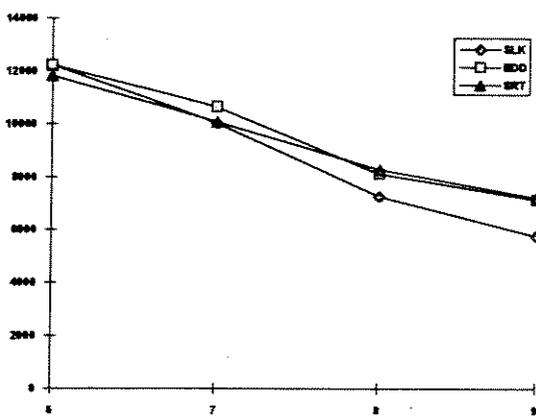
$$d_i = \gamma \times TT_i ; \quad (4.5.4)$$

A Tabela 4.5.1 mostra o tempo de atraso total das regras, onde cada célula representa a média do atraso nos 40 períodos de simulação. A coluna “Média” representa a média sobre todos os números de AGVs. As tabelas restantes neste capítulo são organizadas em forma semelhante. As Figuras 4.5.1 (a, b) ilustram o desempenho das melhores regras. Observa-se, como esperado, que os tempos de atraso diminuem com o acréscimo do número de AGVs. Nota-se também, que a regra MDD domina todas as outras regras, exceto quando o número de AGVs é igual a 9, onde é superada pela regra SLK. A regra SLK é a segunda melhor regra, pois supera as regras FRS e

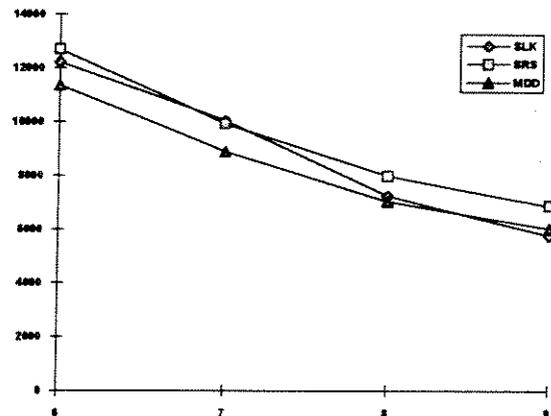
SRT em 3 dos 4 níveis de N e as demais regras (exceto MDD) em todos os níveis. A regra FRS fica em terceiro lugar, dominando SRT e EDD em 3 níveis de N.

Tabela 4.5.1 Desempenho das regras

Prior. \ N	6	7	8	9	Média
STT	20396,7	17077,9	13828,9	11512,1	15703,9
LTT	16332,8	13441,4	11751,7	8879,6	12601,4
EDD	12223,5	10638,8	8114,6	7140,1	9529
SLK	12215,4	10030,9	7259,7	5774,6	8820,2
LRT	29343,6	23500,8	17251,8	14957,9	21263,5
SRT	11822,5	10085,0	8288,8	7223,6	9355
LRS	29284,7	21385,5	17394,3	16004,3	21017,2
FRS	12682,5	9928,2	7987,8	6885,4	9371
MDD	11350,6	8905,3	7046,9	6035,3	8334,5
FIFO	15671,5	11408,3	10128,0	7915,6	11280,8



(a)



(b)

Figura 4.5.1 Tempo de atraso das regras

Foram construídos diversos modelos de prioridade de acordo com o método apresentado no capítulo 2. Cada modelo representa uma combinação distinta das melhores regras identificadas acima. Para cada combinação, foi executada uma simulação com um período de *warm-up* e 40 períodos consecutivos, e a média é apresentada em cada célula na Tabela 4.5.2.

Tabela 4.5.2 Desempenho das combinações

Combinações de regras	Número de AGV (N)				
	6	7	8	9	Média
MDD+SLK	11990,3	9789,8	7111,1	5684,7	8647
MDD+EDD	11543,1	8895,0	7194,5	5951,4	8396
MDD+SRT	11657,6	10241,6	8145,5	7180,7	9306,3
MDD+FRS	12342,0	9440,4	7995,1	6594,4	9093
MDD+SLK+EDD	11990,3	9789,8	7111,1	5684,7	8644
MDD+SLK+FRS	12098,0	8807,0	7576,6	6270,4	8688
SLK+FRS+EDD	12153,8	8807,0	7576,6	6126,4	8666
MDD+SLK+FRS+EDD	12086,1	9031,4	7162,9	5265,1	8386,4
MDD+SLK+EDD+SRT	12405,3	9455,4	7211,1	5524,9	8649,2
MDD+SLK+EDD+SRT+FRS	13109,4	9090,6	7104,4	5633,1	8734,4
MDD+SRT+SLK (P2)	11220,0	9585,2	7454,9	5573,4	8458,4
MDD+SLK+FRS (P1)	12215,5	8889,1	6975,9	5295,1	8343,9

Os resultados na Tabela 4.5.2 mostram que o modelo P1 apresenta o melhor desempenho na média. Este modelo apresenta um bom desempenho quando N é igual a 7, 8 e 9; e mau desempenho quando N é igual a 6. Embora o modelo P2 não seja o segundo melhor em média, ele tem o menor atraso quando N é igual a 6. Por este motivo, utilizamos estes dois modelos para compará-los com as regras simples.

A seguir, mostramos os detalhes da construção dos modelos. A Figura 4.5.2 mostra, como um exemplo, a relação hierárquica entre o objetivo e os fatores (regras) no modelo P1.

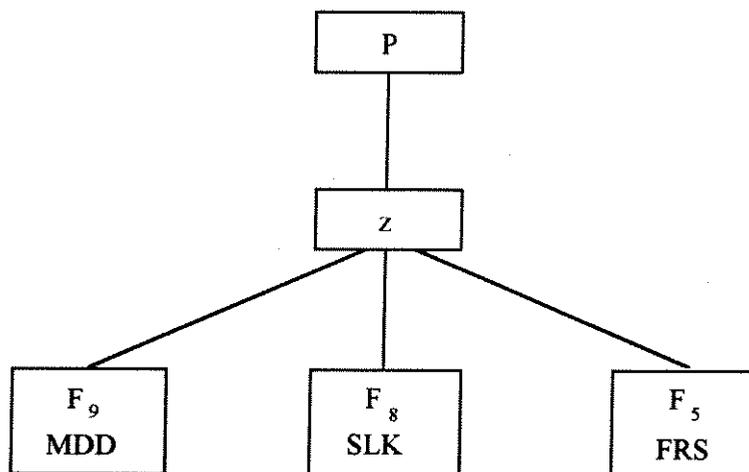


Figura 4.5.2 A relação hierárquica do modelo P1

A Figura 4.5.3 mostra a matriz de comparação dos fatores em relação ao modelo P1. A definição da matriz é baseada principalmente na análise dos resultados computacionais apresentados nas Figuras 4.5.1 (a, b). Por exemplo, o grau de importância do fator F9 (MDD) em relação a F8 (SLK) e F5 (FRS) é 3 e 5 respectivamente, pois a regra MDD é melhor que a regra SLK e muito melhor que a regra FRS. O grau de importância de F8 (SLK) em relação a F5 (FRS) é 3, pois a regra SLK apresenta um comportamento superior à regra FRS.

A Figura 4.5.3 mostra, também, os pesos dos fatores em relação ao objetivo atraso total, bem como o índice de consistência IC e a razão de consistência RC.

	MDD F9	SLK F8	FRS F5	
MDD	1	3	5	$w_{F9} = 0,6370$
SLK	1/3	1	3	$w_{F8} = 0,2583$
FRS	1/5	1/3	1	$w_{F5} = 0,1047$
IC = 0,0193; RC = 0,0332				

Figura 4.5.3 Matriz de comparação do modelo P1

Da mesma maneira, obtemos a matriz de comparação, os pesos dos fatores, o índice de consistência IC e a razão de consistência RC para o modelo P2 como mostra a Figura 4.5.4.

	MDD F9	SRT F1	SLK F8	
MDD	1	3	4	$w_{F9} = 0,6144$
SRT	1/3	1	3	$w_{F1} = 0,2684$
SLK	1/4	1/3	1	$w_{F8} = 0,1172$
IC = 0,0367; RC = 0,0633				

Figura 4.5.4 Matriz de comparação do modelo P2

Observa-se pela tabela 4.5.1 que a regra SRT supera a regra SLK quando o número de AGVs é igual a 6. Isto explica porque o modelo P2 domina o modelo P1 neste caso.

Deve-se observar que os pontos fundamentais na definição do modelo de prioridade são a escolha dos fatores e a determinação da matriz de comparação. O conhecimento de especialistas pode ajudar a definição destes pontos fundamentais e tornar o modelo mais apropriado à situação real.

A partir dos pesos dos fatores, pode-se determinar os valores dos parâmetros α_i nos modelos de prioridade P1 e P2 (vide equação 4.5.2) de acordo com a maneira descrita na seção 2.7.

Modelo P1:

$$\alpha_9 = 1 * 0,6370 = 0,6370$$

$$\alpha_8 = 1 * 0,2583 = 0,2583$$

$$\alpha_5 = 1 * 0,1047 = 0,1047$$

$$P1 = 0,6370 * F(MDD) + 0,2583 * F(SLK) + 0,1047 * F(FRS)$$

Modelo P2:

$$\alpha_9 = 1 * 0,6144 = 0,6144$$

$$\alpha_1 = 1 * 0,2684 = 0,2684$$

$$\alpha_8 = 1 * 0,1172 = 0,1172$$

$$P2 = 0,6144 * F(MDD) + 0,2684 * F(SRT) + 0,1172 * F(SLK)$$

Devido às discrepâncias numéricas dos fatores utilizados nos modelos, dividimos cada fator por seu valor máximo dentre os elementos na fila de espera.

4.6 Implementação e Resultados

Para a programação do Sistema de Transporte de Materiais (STM) num sistema flexível de manufatura, foi desenvolvido um simulador em linguagem C e implementado na estação de trabalho SUN, modelo SPARC classic. O simulador foi construído com base no método analítico de estado apresentado no capítulo 2. O simulador é apresentado no capítulo 5. Nesta seção, vamos utilizar o simulador para programar o STM e avaliar o desempenho dos modelos de prioridade P1, P2, e das regras simples com a variação do número de AGVs no sistema. Além disso, é utilizado o algoritmo de busca com relação aos parâmetros α_i com o objetivo de encontrar soluções de melhor qualidade.

4.6.1 Exemplo de um sistema de transporte de materiais

Utilizamos um sistema típico de STM proposto por Egbelu e Tanchoco (1986) para resolver o problema de programação através do método analítico de estado e das regras e modelos de prioridade descritos nas seções anteriores. O sistema é composto por 8 departamentos. Cada departamento tem um ponto de entrega (E_i) e um de retirada (R_i). Cada ponto tem seu próprio buffer de AGV. Os segmentos que ligam os pontos são bidirecionais. Os AGVs localizam-se inicialmente numa área reservada no sistema. A Figura 4.6.1 mostra o *layout* do sistema.

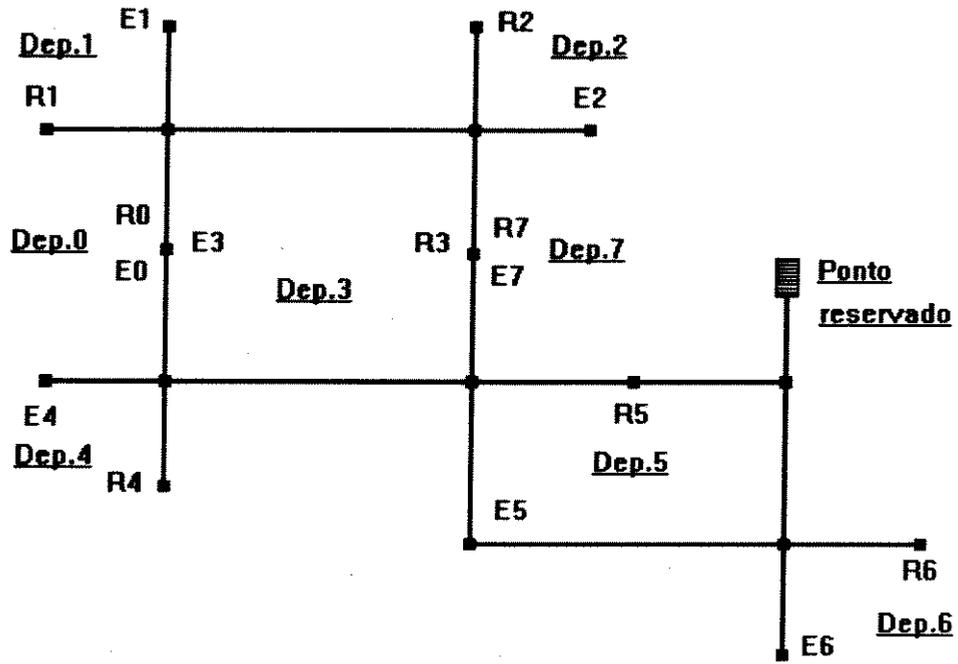


Figura 4.6.1 Layout de um STM

Para facilitar o estudo, renumeramos todos os pontos de entrega, de retirada e de interseção, e definimos um tempo de transporte para cada segmento, como mostra a Figura 4.6.2.

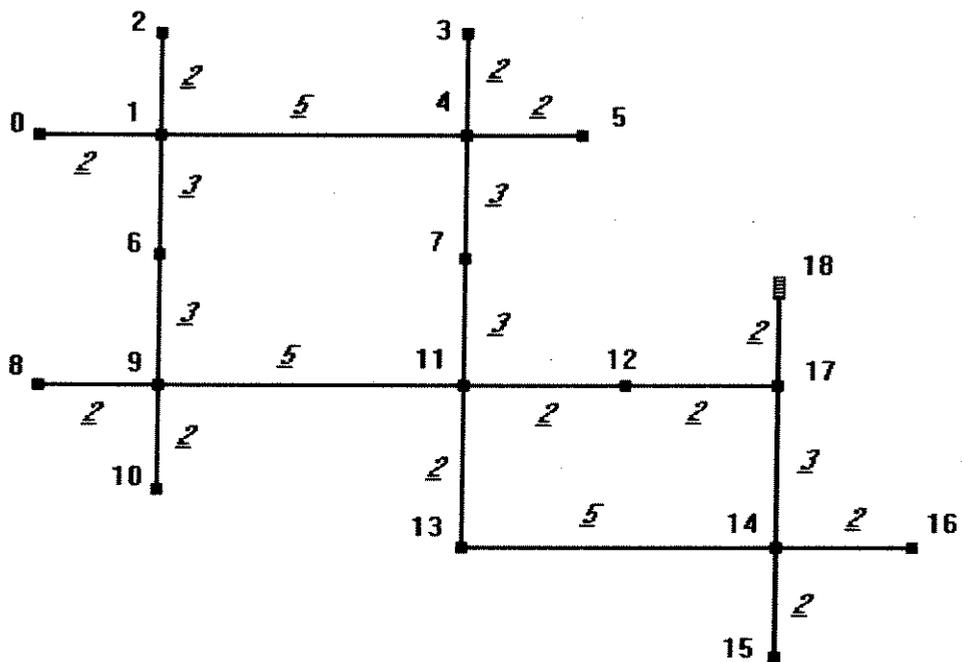


Figura 4.6.2 Numeração e tempo de transporte de um STM

Os segmentos bidirecionais no STM são representados por arcos orientados. A Figura 4.6.3 apresenta a rede de fluxo do STM.

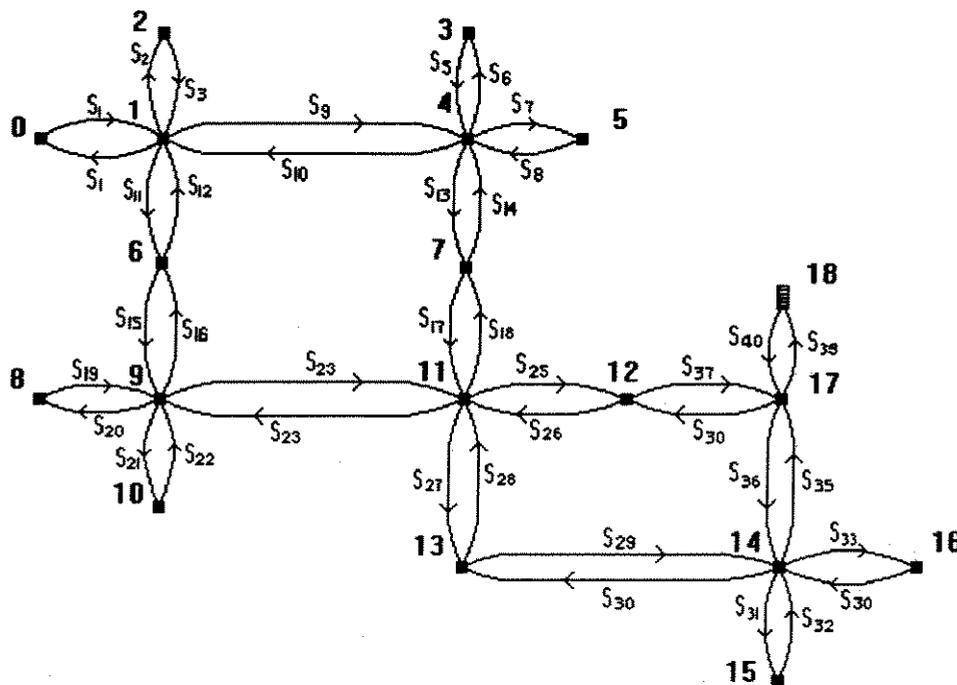


Figura 4.6.3 Rede de fluxo do STM

4.6.2 Dados do sistema

A partir da Figura 4.6.2, obtemos uma matriz que contém o tempo de transporte mínimo entre qualquer dois pontos na rede do STM, como mostra a Figura 4.6.4. Note que os elementos na diagonal são nulos e a matriz é simétrica.

i \ j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	2	4	9	7	9	5	10	10	8	10	13	15	15	20	22	22	17	19
1	2	0	2	7	5	7	3	8	8	6	8	11	13	13	18	20	20	15	17
2	4	2	0	9	7	9	5	10	10	8	10	13	15	15	20	22	22	17	19
3	9	7	9	0	2	4	10	5	15	13	15	8	10	10	15	17	17	12	14
4	7	5	7	2	0	2	8	3	13	11	13	6	8	8	13	15	15	10	12
5	9	7	9	4	2	0	10	5	15	13	15	8	10	10	15	17	17	12	14
6	5	3	5	10	8	10	0	11	5	3	5	8	10	10	15	17	17	12	14
7	10	8	10	5	3	5	11	0	10	8	10	3	5	5	10	12	12	7	9
8	10	8	10	15	13	15	5	10	0	2	4	7	9	9	14	16	16	11	13
9	8	6	8	13	11	13	3	8	2	0	2	5	7	7	12	14	14	9	11
10	10	8	10	15	13	15	5	10	4	2	0	7	9	9	14	16	16	11	13
11	13	11	13	8	6	8	8	3	7	5	7	0	2	2	7	9	9	4	6
12	15	13	15	10	8	10	10	5	9	7	9	2	0	4	5	7	7	2	4
13	15	13	15	10	8	10	10	5	9	7	9	2	4	0	5	7	7	6	8
14	20	18	20	15	13	15	15	10	14	12	14	7	5	5	0	2	2	3	5
15	22	20	22	17	15	17	17	12	16	14	16	9	7	7	2	0	4	5	7
16	22	20	22	17	15	17	17	12	16	14	16	9	7	7	2	4	0	5	7
17	17	15	17	12	10	12	12	7	11	9	11	4	2	6	3	5	5	0	2
18	19	17	19	14	12	14	14	9	13	11	13	6	4	8	5	7	7	2	0

Figura 4.6.4 Matriz de tempos mínimos

A Tabela 4.6.1 mostra uma lista das rotas alternativas dos AGVs. As rotas são apresentadas através do ponto final de seu próximo segmento. Por exemplo, do ponto 1 ao ponto 11, o valor superior, 4, encontrado na célula (1, 11) da lista, representa que o segmento (1, 4) é o próximo segmento a ser percorrido de 1 a 11 na sua primeira rota alternativa. Os demais segmentos desta rota podem ser obtidos da mesma forma. O elemento inferior, 6, na célula (1, 11) da lista indica que o segmento (1, 6) é o próximo segmento a ser percorrido de 1 a 11 na sua segunda rota alternativa.

Tabela 4.6.1 Rotas alternativas

i \ j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0		2	4	4	4	6	4	6	6	6	4	4	4	4	4	4	4	4
2	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	4	4	4		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	1	1	1	3		5	1	7	1	1	1	7	7	7	7	7	7	7	7
5	4	4	4	4	4		4	4	4	4	4	4	4	4	4	4	4	4	4
6	1	1	1	1	1	1		1	9	9	9	9	9	9	9	9	9	9	9
7	4	4	4	4	4	4	4		11	11	11	11	11	11	11	11	11	11	11
8	9	9	9	9	9	9	9	9		9	9	9	9	9	9	9	9	9	9
9	6	6	6	6	6	6	6	11	8		10	11	11	11	11	11	11	11	11
10	9	9	9	9	9	9	9	9	9	9		9	9	9	9	9	9	9	9
11	9	9	9	7	7	7	9	7	9	9	9		12	13	13	13	13	13	12
12	11	11	11	11	11	11	11	11	11	11	11	11		11	17	17	17	17	17
13	11	11	11	11	11	11	11	11	11	11	11	11	11		14	14	14	11	11
14	13	13	13	13	13	13	13	13	13	13	13	13	17	13		15	16	17	17
15	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14		14	14	14
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14		14	14
17	12	12	12	12	12	12	12	12	12	12	12	12	12	12	14	14	14		18
18	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	

4.6.3 Variação do número de AGVs

O número de AGVs, N , é um parâmetro significativo num sistema de transporte. Neste exemplo, simulamos e avaliamos o desempenho das prioridades para N igual a 6, 7, 8 e 9. Para cada valor de N e cada prioridade, a simulação é feita por um período de *warm-up* e 40 períodos consecutivos. A Tabela 4.6.2 mostra o atraso com a variação de N para as diversas prioridades. Os gráficos correspondentes aos resultados desta tabela são mostrados nas Figuras 4.6.5 a 4.6.8.

Dentre as regras, MDD domina todas as demais, exceto quando N é igual a 9, onde é superada pela regra SLK. A regra SLK é a segunda colocada, pois é superior às regras FRS e SRT em três valores de N e superior às demais (exceto MDD) para todos os valores de N.

Observa-se que o modelo P1 é melhor que P2 em três dos quatro valores de N. Em relação às regras, P1 é melhor que as regras MDD, SLK e SRT em 3 valores e melhor que as demais regras em todos os valores. Em relação à melhor regra, MDD, o modelo P1 tem um ganho de 0,18% até 12,26% para N igual a 7, 8 e 9, e uma perda de 7,62% para N igual a 6. Na média, o modelo P1 e a regra MDD apresentam o melhor comportamento. Deve-se ressaltar que o modelo P2 domina o modelo P1 e todas as regras quando N é igual a 6.

Tabela 4.6.2 Tempo de atraso das prioridades

Prioridades	Atraso com a variação de N				
	6	7	8	9	Média
MDD	11350,6	8905,3	7046,9	6035,3	8334,5
SLK	12215,4	10030,9	7259,7	5774,6	8820,2
SRT	11822,5	10085,0	8288,8	7223,6	9355
FRS	12682,5	9928,2	7987,8	6885,4	9371
EDD	12223,5	10638,8	8114,6	7140,1	9529
FIFO	15671,5	11408,3	10128,0	7915,6	11280,8
LTT	16332,8	13441,4	11751,7	8879,6	12601,4
STT	20396,7	17077,9	13828,9	11512,1	15703,9
LRS	29284,7	21385,5	17394,3	16004,3	21017,2
LRT	29343,6	23500,8	17251,8	14957,9	21263,5
P1	12215,5	8889,1	6975,9	5295,1	8343,9
P2	11220,0	9585,2	7454,9	5573,4	8458,4

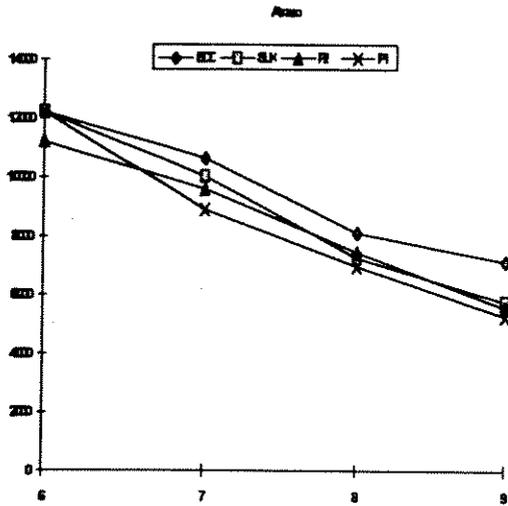


Figura 4.6.5 Tempo de atraso dos modelos e das regras EDD e SLK

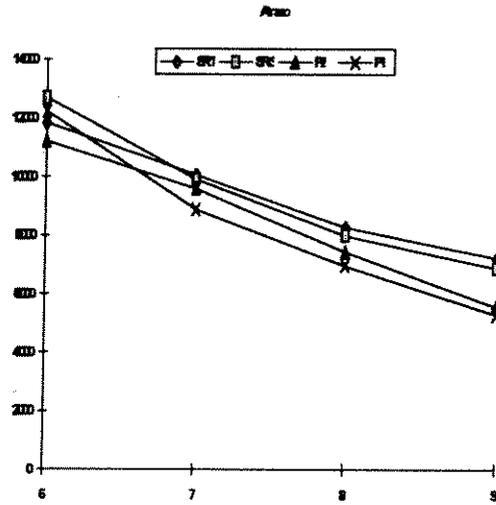


Figura 4.6.6 Tempo de atraso dos modelos e das regras FRS e SRT

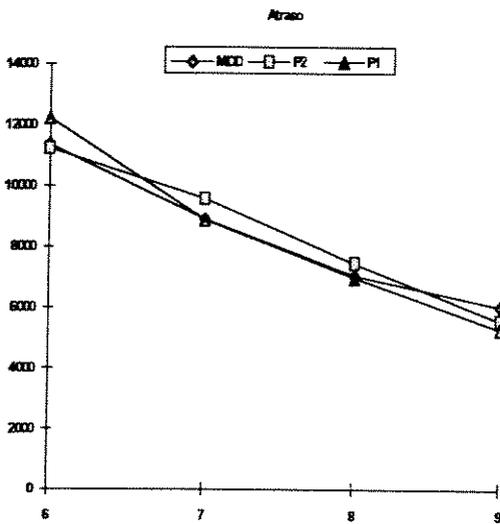


Figura 4.6.7 Tempo de atraso dos modelos e da regra MDD

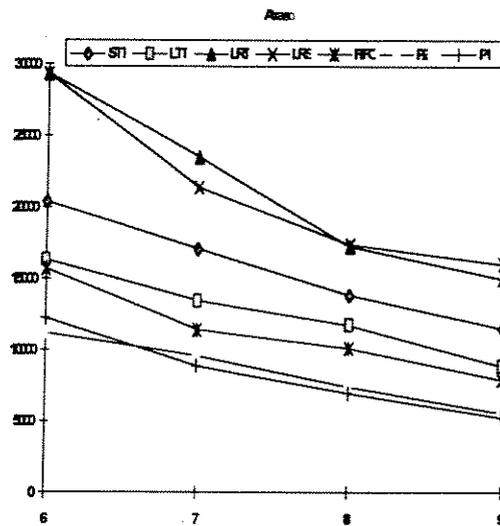


Figura 4.6.8 Tempo de atraso dos modelos e das regras LTT, LRT, LRO e FIFO

4.6.4 Algoritmo de busca

Como no capítulo 3, não é possível definir um único modelo que apresente o melhor comportamento em todos os casos. Por este motivo, aplica-se o algoritmo de busca proposto na seção 3.7.6 do capítulo 3 para melhorar os modelos de prioridade P1 e P2, construídos na seção anterior, no caso em que o número de AGVs (N) é igual a 6, pois este é o único caso em que o modelo P1 não supera as regras e o modelo P2. O tamanho mínimo do passo de busca, δ , utilizado é 0,0001. A tabela 4.6.3 mostra os resultados originais de P1 e P2, representados como “Atraso de PR”, e os resultados obtidos pelo algoritmo de busca, representados como “Atraso de AL”. As Figuras 4.6.9 e 4.6.10 ilustram os valores de atraso obtidos durante a busca. Observa-se que o algoritmo conseguiu melhorar significativamente o modelo P1, com uma redução de 9,07% no atraso. O novo valor de atraso está 2,14% abaixo do atraso obtido pela melhor regra simples, MDD. O algoritmo não conseguiu melhorar o resultado original do modelo P2, o que indica este modelo tem um bom comportamento para N igual a 6.

Tabela 4.6.3 Atraso das prioridades

Prioridade	N = 6	
	Atraso de PR	Atraso de AL
P1	12215,5	11108,1
P2	11220,0	11220,0
MDD	11350,6	11350,6

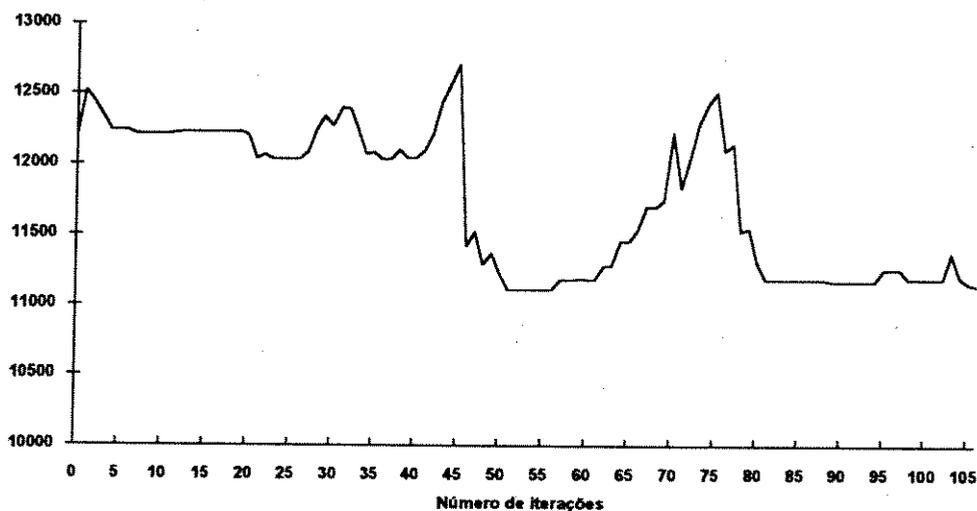


Figura 4.6.9 Atrasos com o modelo P1

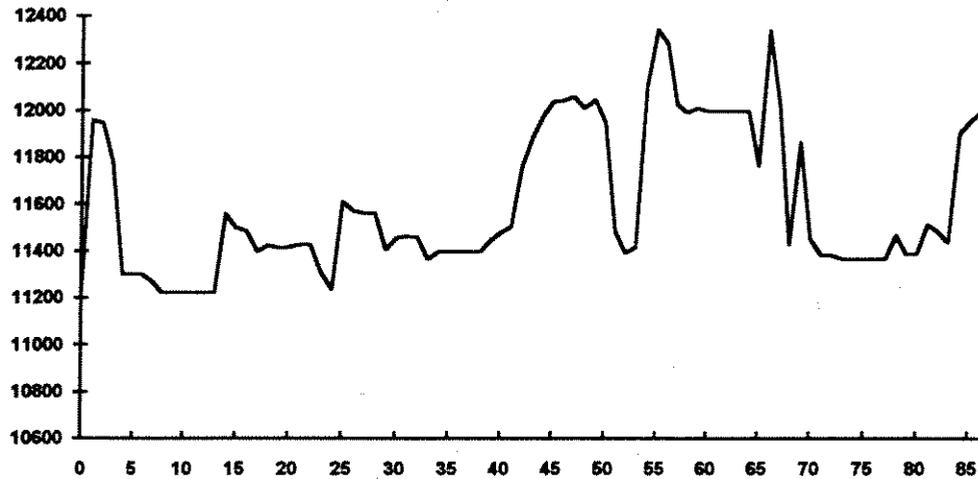


Figura 4.6.10 Atrasos com o modelo P2

A seguir, utiliza-se o algoritmo I para obter modelos de prioridade com dois fatores. O objetivo é ilustrar como é o desempenho da combinação de dois fatores.

As equações 4.6.3 a 4.6.6 mostram 4 modelos de prioridade resultantes da combinação de duas regras. Os gráficos correspondentes aos atrasos obtidos na busca são representados na Figuras 4.6.11 a 4.6.14.

$$P_{C1} = \alpha \times F_7 + (1 - \alpha) \times F_9 \quad (4.6.3)$$

$$P_{C2} = \alpha \times F_1 + (1 - \alpha) \times F_9 \quad (4.6.4)$$

$$P_{C3} = \alpha \times F_8 + (1 - \alpha) \times F_9 \quad (4.6.5)$$

$$P_{C4} = \alpha \times F_3 + (1 - \alpha) \times F_9 \quad (4.6.6)$$

Os fatores F_1 , F_3 , F_7 , F_8 e F_9 correspondem às regras SRT, SPT, EDD, SLK e MDD. Para obter o ponto inicial de α , consideramos os casos em que α é igual a 0,1; 0,2; ..., 0,9. A partir do melhor ponto em relação ao atraso, utilizamos o algoritmo I para tentar obter uma combinação melhor. Além disso, exploramos também os melhores pontos nos intervalos inicial e final do valor de α , isto é, (0; 0,1) e (0,9; 1) respectivamente, pois estes intervalos correspondem ao início da transição de uma regra simples para uma combinação de regras.

As Figuras 4.6.11 a 4.6.14 mostram o comportamento do atraso com a variação de α para os modelos representados em 4.6.3 - 4.6.6. Observa-se que o atraso da combinação entre as regras

MDD e EDD varia muito com o acréscimo do parâmetro α . As demais combinações apresentam um comportamento de atraso mais estável com relação a α . Os resultados computacionais mostram que a combinação entre as regras SRT e MDD fornecer o menor atraso. O segundo melhor atraso é obtido pela combinação das regras EDD e MDD.

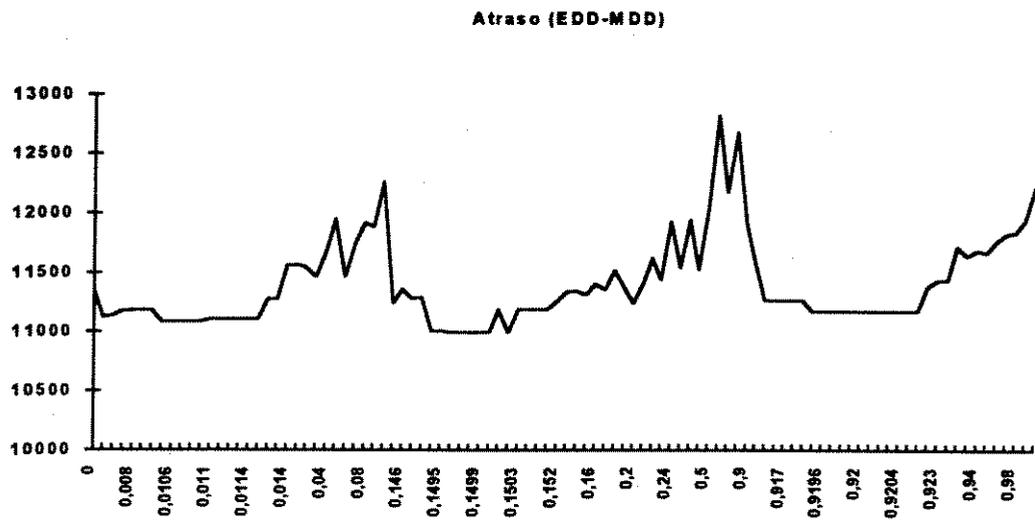


Figura 4.6.11 Atraso com a variação de combinação entre as regras MDD e EDD

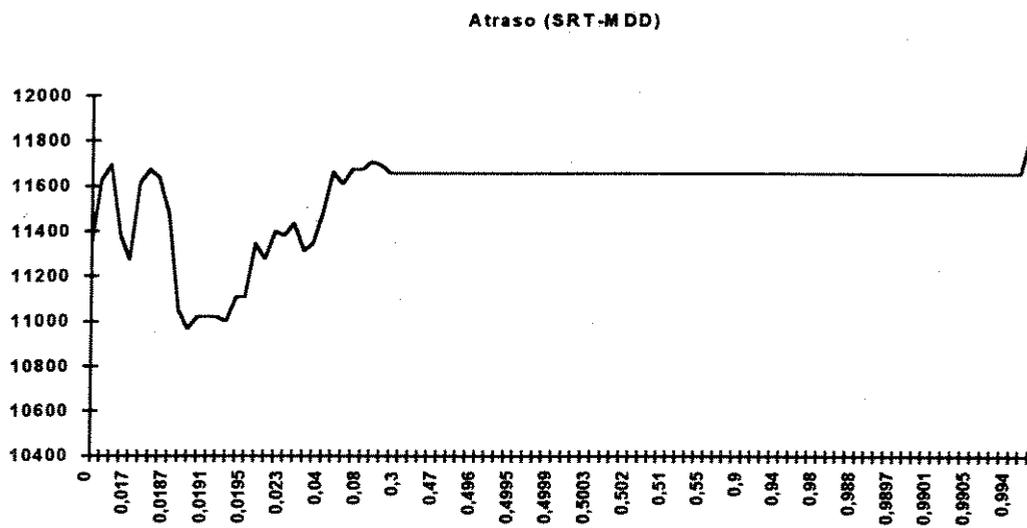


Figura 4.6.12 Atraso com a variação de combinação entre as regras MDD e SRT

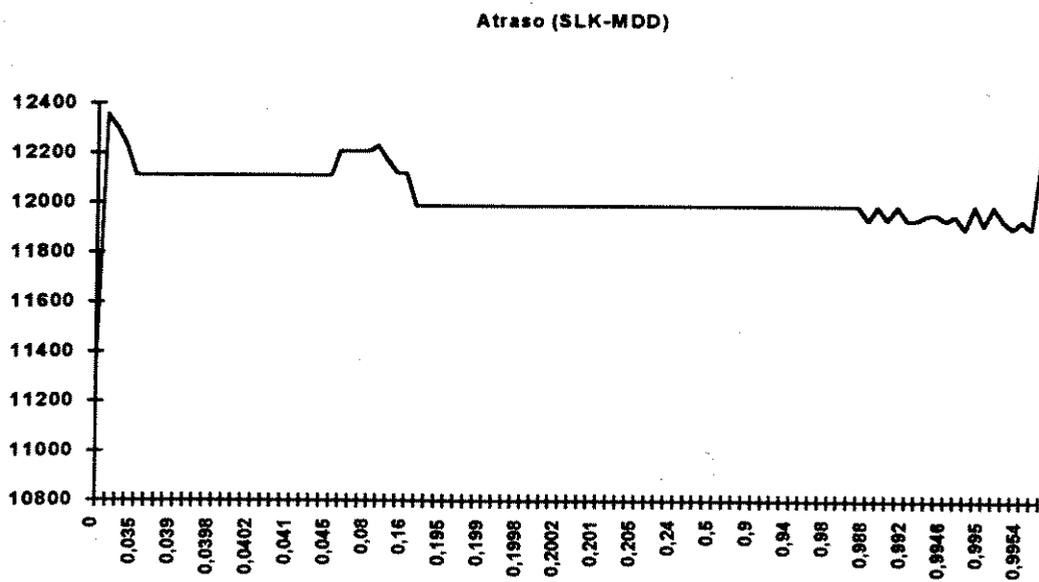


Figura 4.6.13 Atraso com a variação de combinação entre as regras MDD e SLK

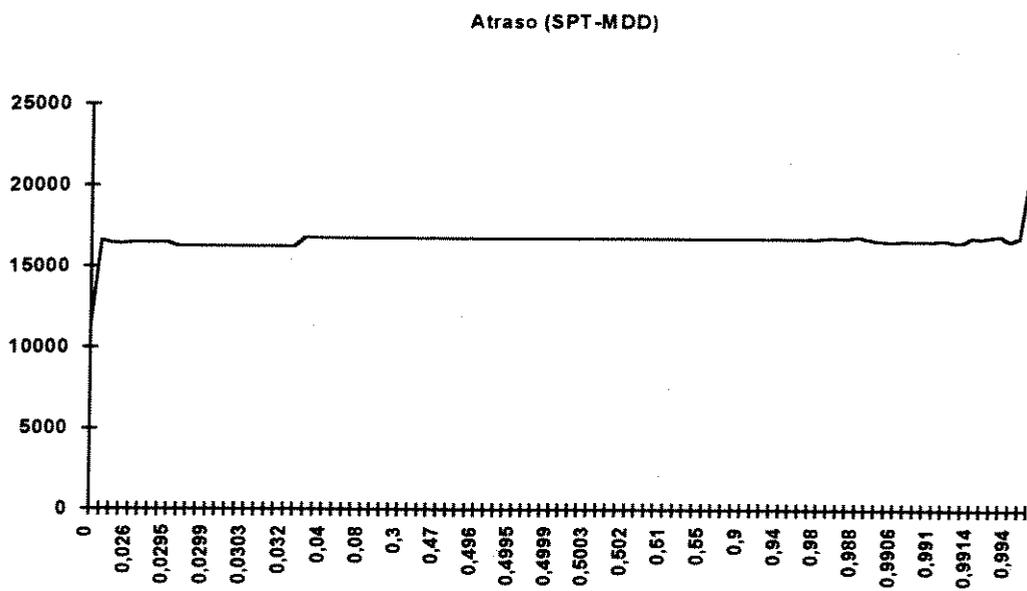


Figura 4.6.14 Atraso com a variação de combinação entre as regras MDD e SPT

CAPÍTULO 5

UM SIMULADOR PARA PROGRAMAÇÃO DE SISTEMAS FLEXÍVEIS DE MANUFATURA

5.1 Introdução

Para resolver o problema de programação dos sistemas de produção e transporte em um SFM, foi desenvolvido um Simulador para Programação de Sistemas Flexíveis de Manufatura. Este é um simulador genérico, dirigido por evento discreto e orientado para o usuário. Devido a estas características, este simulador tem potencial para utilização em pesquisas acadêmicas e aplicações industriais. O simulador consiste da integração dos simuladores de programação do Sistema de Produção (SP) e do Sistema de Transporte de Materiais (STM) desenvolvidos nos capítulos 3 e 4. Este capítulo apresenta a estrutura do simulador e um exemplo é utilizado para ilustrar o seu funcionamento.

5.2 Descrição do Simulador

O simulador é um sistema flexível e genérico para programação num ambiente de manufatura flexível. Foi desenvolvido em linguagem C, implementado em uma estação de trabalho SUN Classic, e contém um programa fonte de 620 kb e um código executável de 697 kb. Tem seus próprios editores e geradores de dados, simuladores de programação do SP e do STM e analisadores de desempenho. Esta seção descreve os aspectos gerais do simulador. Aspectos mais detalhados são apresentados nas seções seguintes.

5.2.1 Metodologia utilizado no simulador

O simulador foi desenvolvido com base no Método Analítico de Estado apresentado no capítulo 2. Este método divide o processo operacional de um sistema em um conjunto de processos das entidades básicas neste sistema. O processo operacional de cada entidade básica é simulado por seus estados e transições de estados. O simulador contém um banco de prioridades, incluindo os modelos PLM, para tomar decisões e definir as transições indiretas. A seqüência das transições de estados no processo de cada entidade básica define a sua programação, bem como a programação global do sistema.

5.2.2 Estrutura geral do simulador

Um SFM é composto principalmente de duas partes importantes: um Sistema de Produção e um Sistema de Transporte de Materiais. Devido a esta estrutura natural, o simulador é composto de dois simuladores de programação:

- Simulador de programação de SP
- Simulador de programação de STM

A relação entre estes simuladores é apresentada na Figura 5.2.1. Cada simulador pode ser executado individualmente para obter somente uma programação de um subsistema do SFM. Alternativamente, ambos os simuladores podem ser executados de forma integrada para obter uma programação completa num SFM. A interligação dos dois simuladores é feita por uma interface através da qual são trocadas informações.

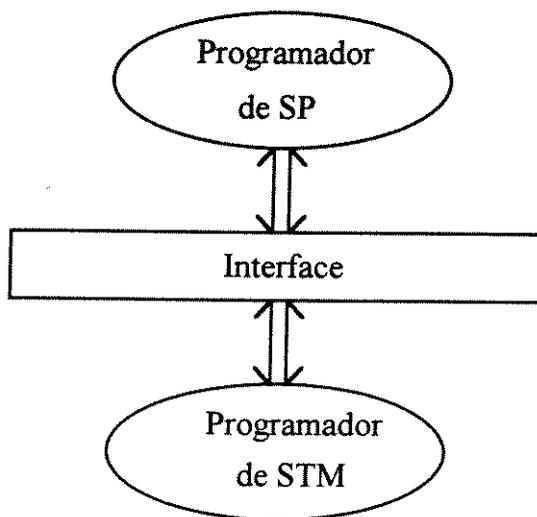


Figura 5.2.1. Relação entre os simuladores

A estrutura geral do simulador é mostrada na Figura 5.2.2. Os dados principais são definidos nos arquivos. Os dados nestes arquivos podem ser gerados diretamente pelo usuário ou automaticamente por computador. A partir dos arquivos de dados, pode-se programar os sistemas através dos programadores. Ao final, são produzidos relatórios para apresentar os programas obtidos aos usuários. Os analisadores do simulador podem ser executados separadamente para analisar e avaliar o desempenho das programações obtidas. Relatórios de desempenho também são gerados para os usuários. Quando os simuladores são executados simultaneamente para obter a programação integrada do SFM, utiliza-se uma interface para trocar informações entre os sistemas. Um relógio comum é empregado para coordenar os eventos nos sistemas.

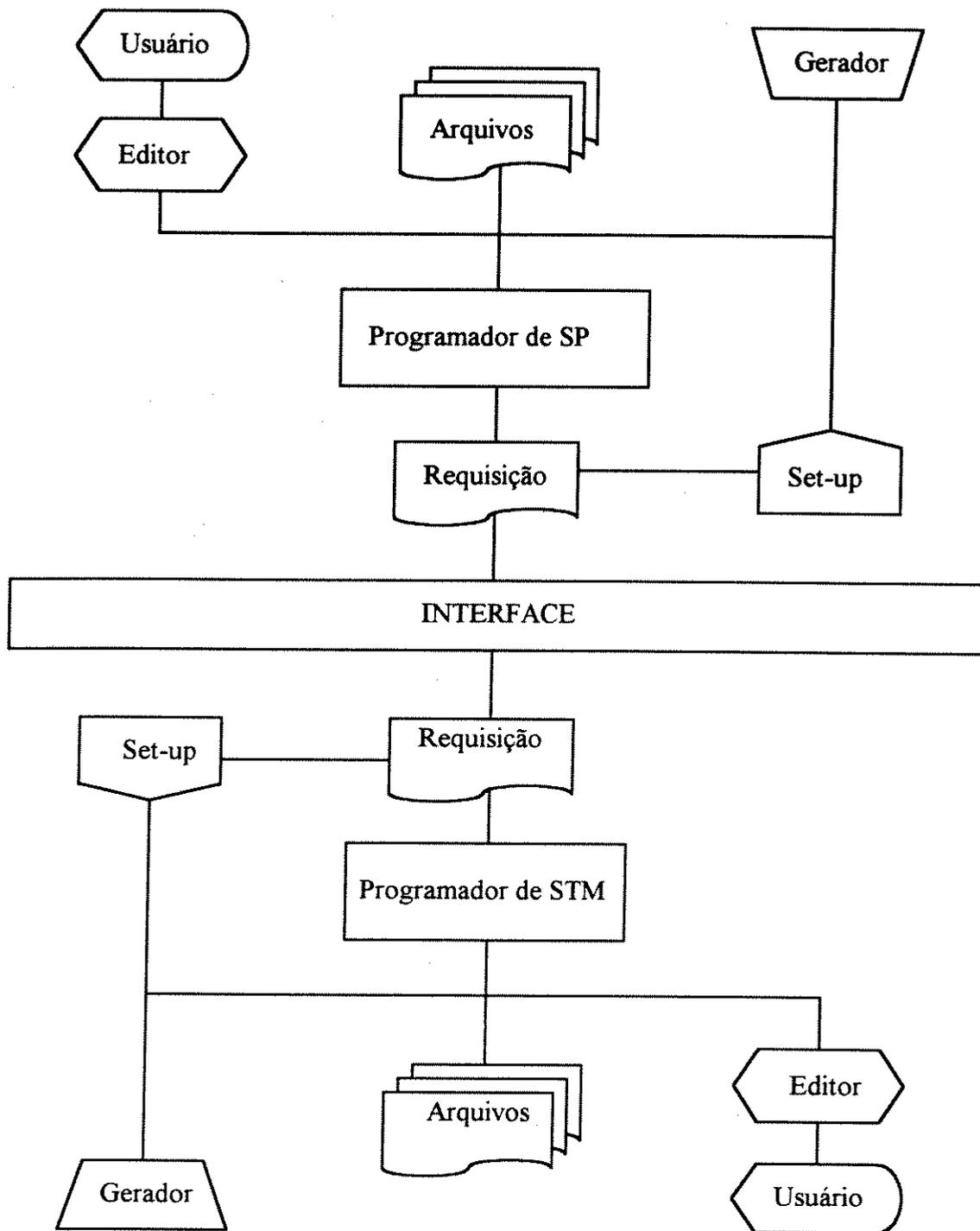


Figura 5.2.2 Estrutura do simulador

5.2.3 Estrutura de dados e informações

No simulador, existe uma grande quantidade de dados e informações, incluindo os de descrição de sistema, de informações temporárias e de resultado final. Os dados e informações principais são organizados das seguintes maneiras:

- **Variáveis globais:** representam o sistema e condições de simulação, tais como número de peças, máquinas, AGVs, seleção de regras, fator de atraso, tempo de simulação e número de replicações.
- **Variáveis locais:** provêm os espaços de trabalho temporários para auxiliar a resolução do problema de programação.
- **Matrizes:** contêm as informações para definir o *layout* do sistema, rotas alternativas, tempo/distância de transporte.
- **Arquivos:** coletam as informações e características associadas às operações, peças, requisições, centros de trabalho, cargas de transporte e AGVs.

5.2.4 Características do simulador

As seguintes características podem ser destacadas:

- **Descentralização:** o simulador divide a programação completa do SFM em uma programação do SP e uma programação do STM. Cada programação tem seu próprio programador. Esta divisão facilita a resolução da programação do SFM.
- **Dirigido por evento:** o simulador é desenvolvido com base no MAE, ou seja, simula os processos com base nos estados e transição de estados.
- **Flexibilidade:** o simulador é flexível. Pode ser utilizado para programar o SP ou o STM individualmente; ou ambos integradamente. A definição do *layout* do sistema é feita nas

matrizes que são separadas do programa fonte. Isto facilita o tratamento de sistemas com *layouts* diferentes.

- **Decisão:** o simulador tem um banco de regras de prioridade contendo regras de despacho e modelos de prioridade combinada para a tomada de decisões no processo de programação. O usuário pode selecionar a regra de programação desejada e comparar o desempenho de regras distintas.
- **Amigável:** o simulador contém um editor (entrada de dados pelo usuário) e um gerador (entrada de dados criada por computador) para cada um dos simuladores de SP e STM. O simulador contém ainda várias interfaces entre o computador e usuário. Estas interfaces permitem que o usuário defina e modifique as condições de simulação. O simulador tem também um módulo de *set-up* para reinicializar os dados e estados do sistema, e facilitar as repetições de simulação.

5.3 Simulador de SP

O simulador de SP é o módulo que programa o sistema de produção de peças. Trata dos problemas de alocação de máquinas e programação de operações. É composto por um programador, um editor, um gerador, um analisador, um módulo de *set-up* e arquivos de dados associados. A sua estrutura é mostrada na Figura 5.3.1.

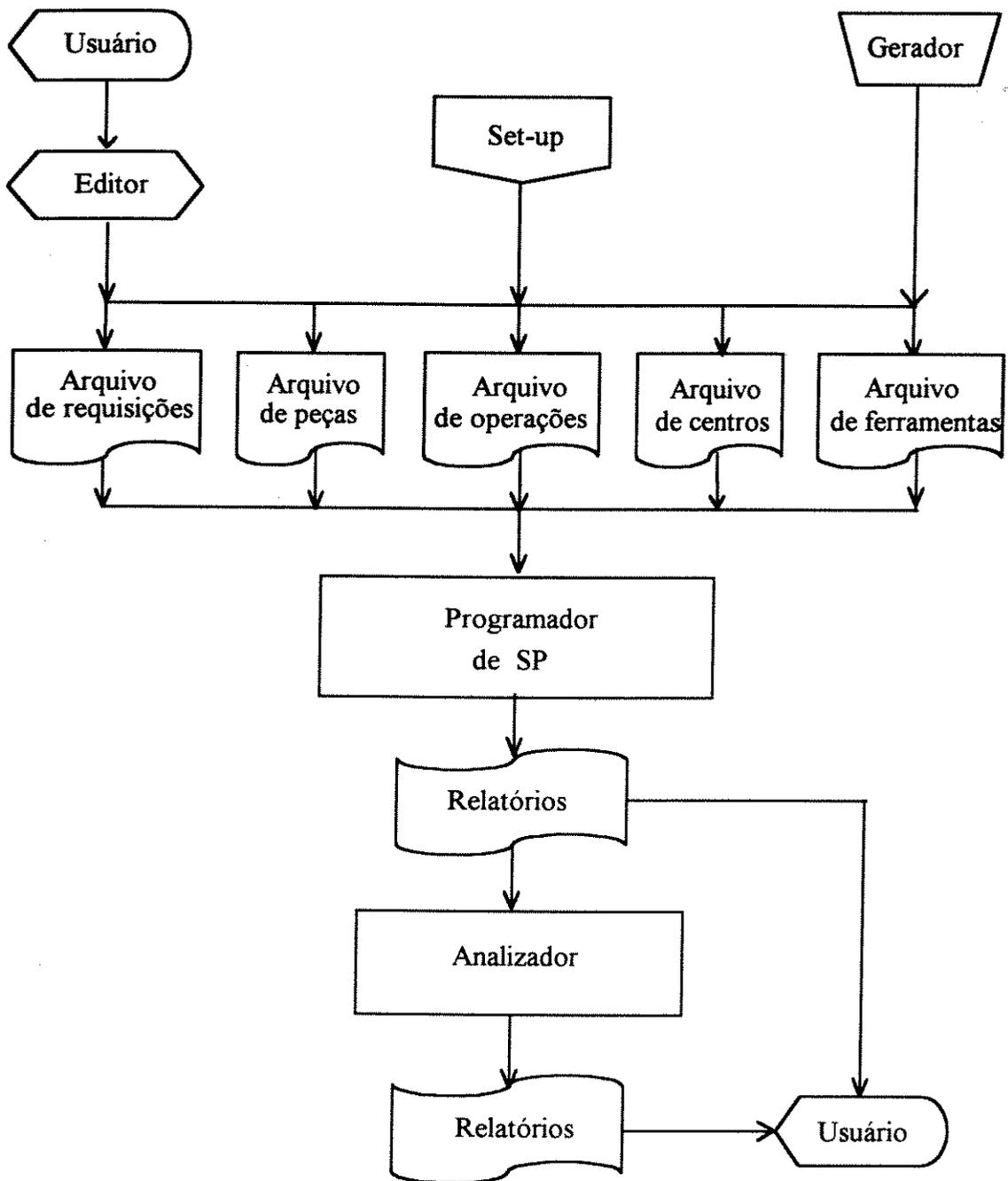


Figura 5.3.1 Estrutura do simulador de SP

Os arquivos de dados neste simulador são apresentados na tabela a seguir:

Arquivos	Informações principais
Arquivo de requisições	1. Informações dos pedidos das peças para serem processados no SP; 2. Informações dos pedidos de transporte das peças ao STM; 3. Informações dos pedidos de transporte das peças atendidos pelo STM.
Arquivo de peças	Informação geral das peças, incluindo identificação, estado, tempos e localização.
Arquivo de operações	Informação geral das operações, incluindo identificação, estado, centros e tempos.
Arquivo de centros de trabalho	Informação geral dos centros de trabalho, contendo identificação, estado, capacidade, carregamento e localização.
Arquivo de ferramentas e fixadores	Informação geral de ferramentas/fixadores, incluindo identificação, estado, quantidade disponível e ocupada.

Para preencher os dados dos arquivos, o simulador de SP contém um editor e um gerador. O editor foi desenvolvido para que os usuários insiram os dados manualmente. Suas funções principais são as seguintes:

- entrada de um novo registro em um arquivo;
- modificação de um registro de um arquivo;
- localização de um registro em um arquivo pelo nome ou número de identificação do registro;
- cancelamento de um registro em um arquivo.

O gerador foi desenvolvido para criar os dados diretamente por computador. É uma maneira prática para gerar casos de simulação diferentes e independentes. O gerador contém também uma interface entre o computador e o usuário. O usuário pode controlar a criação dos casos de simulação através dos valores de parâmetros, tais como número de peças, número de centros de trabalho, fator de data de entrega, etc.

O programador de SP foi desenvolvido com base no método analítico de estado e é o módulo chave deste simulador. Utiliza transições de estado nos processos para simular o funcionamento do sistema, e registra a seqüência das transições para definir a programação. Relatórios da programação obtida são gerados para os usuários. A Figura 5.3.2 mostra em detalhe o diagrama de fluxo do programador de SP. As transições de estado nesta figura são apresentadas em detalhe no capítulo 3.

O analisador é a parte que avalia o desempenho da programação obtida. Ele analisa medidas, tais como tempo de atraso, *makespan* e utilização, obtidas através de várias simulações independentes. Os resultados obtidos são relatados para os usuários.

O módulo de *set-up* é desenvolvido para inicializar as variáveis e os estados do sistema com o fim de facilitar múltiplas simulações.

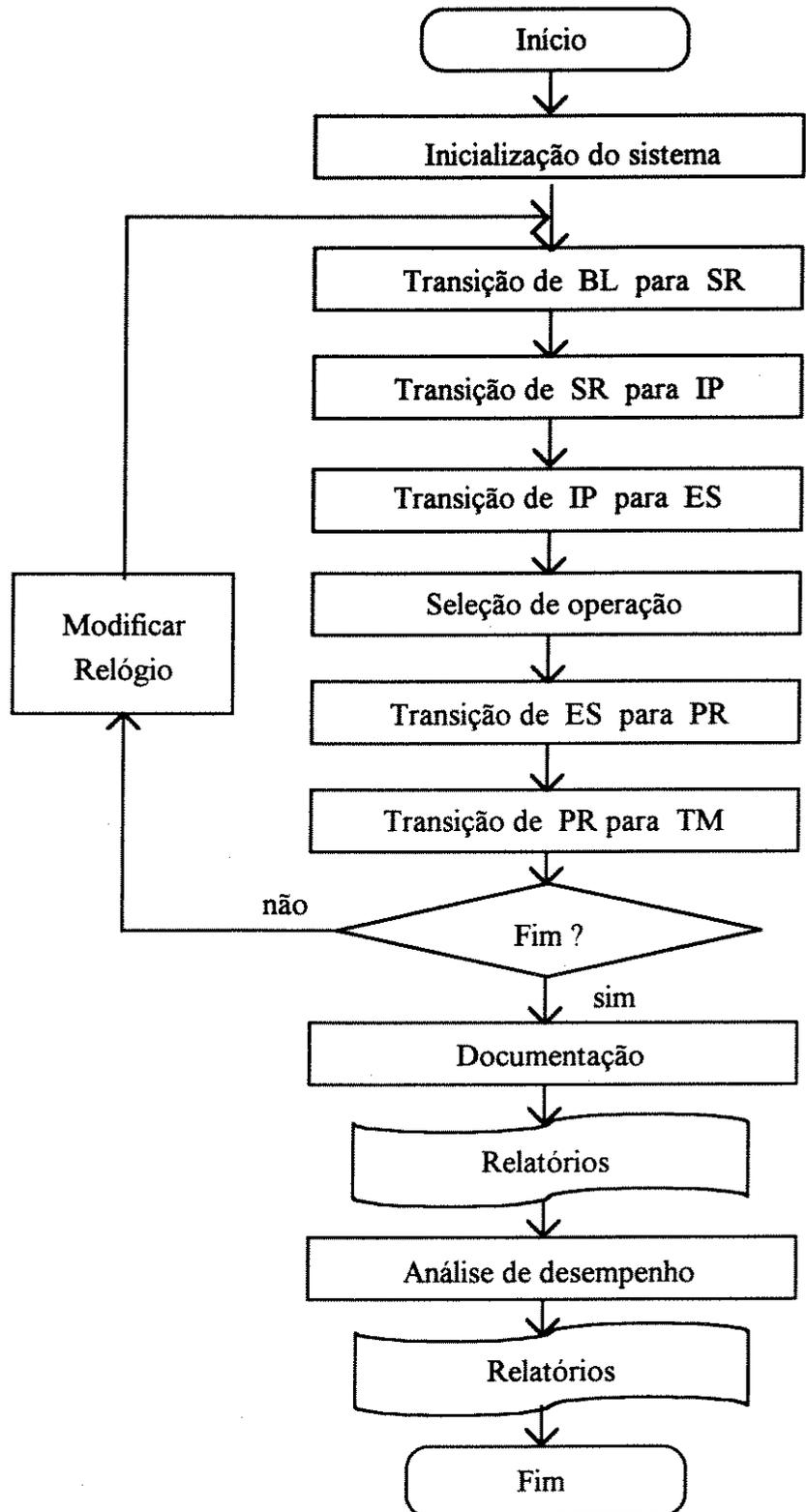


Figura 5.3.2 Diagrama de fluxo do simulador de SP

5.4 Simulador de STM

O simulador de STM é o módulo que programa o sistema de transporte de materiais. Trata os problemas de alocação de AGVs e programação de transporte de cargas. É composto por um programador, um editor, um gerador, um analisador, um módulo de *set-up* e arquivos de dados associados. A sua estrutura é mostrada na Figura 5.4.1.

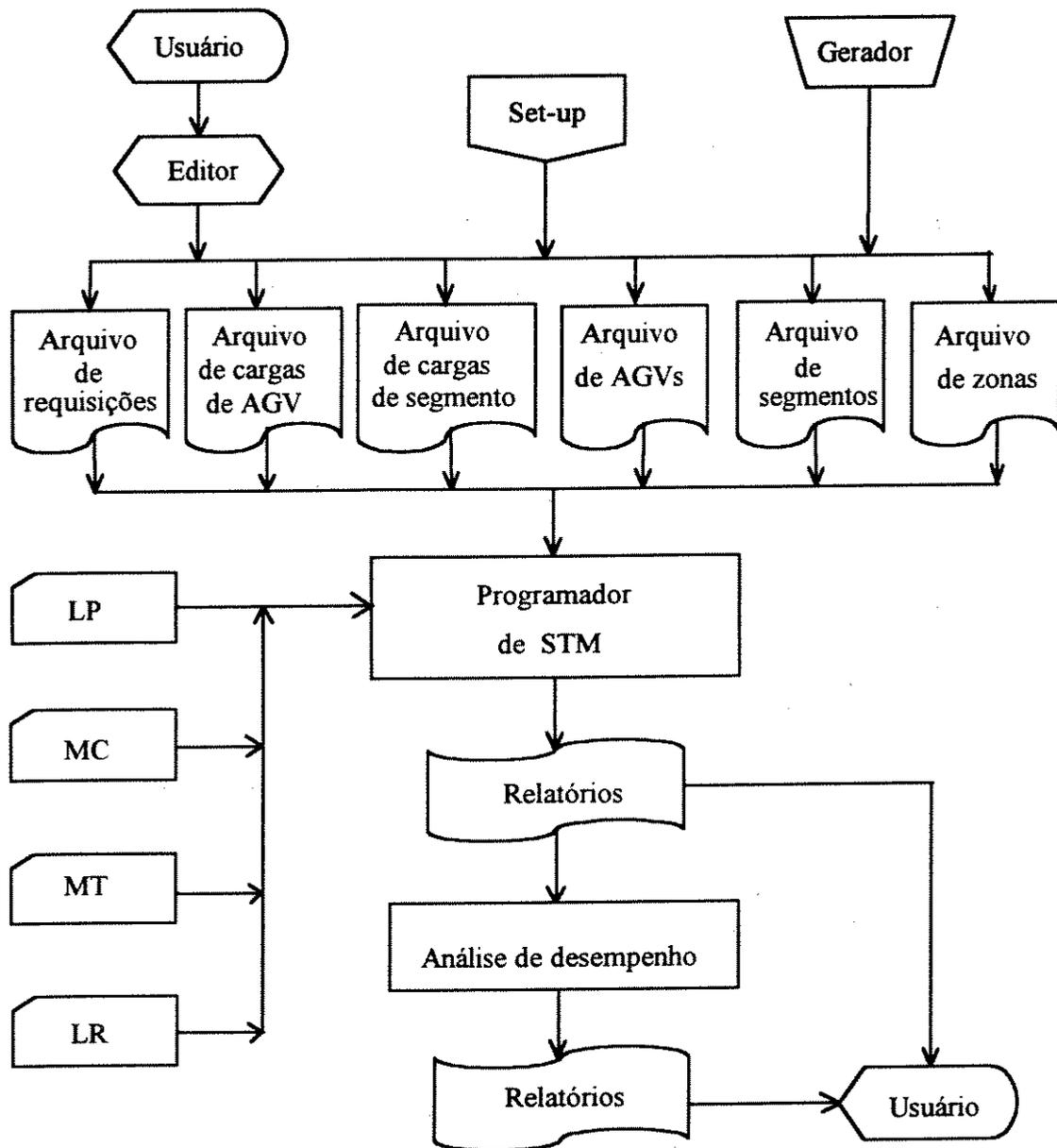


Figura 5.4.1 Estrutura do simulador de STM

Os arquivos de dados neste simulador são apresentados na seguinte tabela:

Arquivos	Informações principais
Arquivo de requisições	1. Informações dos pedidos de transporte das peças ao STM; 2. Informações dos pedidos de AGV vazio.
Arquivo de cargas de AGV	Informação geral das cargas de AGV, incluindo identificação, estado, AGV associado, tempos e localização.
Arquivo de cargas de segmento	Informação geral das cargas de segmento, incluindo identificação, estado, AGV associado, localização e tempos.
Arquivo de AGVs	Informações geral dos AGVs, contendo identificação, estado e localização.
Arquivo de zonas de controle	Informação geral de zonas de controle, incluindo identificação, estado e localização.
Arquivo de segmentos	Informação geral dos segmentos, contendo identificação, estado e localização.

Além dos arquivos, o simulador de STM contém matrizes e listas, LP, MC, MT e LR, para definir o *layout* de um sistema de transporte.

- LP: lista que define todos os pontos de retirada, entrega e interseção;
- MC: matriz que define as conexões dos pontos em LP;
- MT: matriz que define os tempos mínimos de transporte entre qualquer dois pontos em LP;
- LR: lista que define as rotas de transporte alternativas entre quaisquer dois pontos em LP.

Para preencher os dados dos arquivos, o simulador de STM contém um editor e um gerador. O editor foi desenvolvido para que os usuários insiram os dados manualmente. Suas funções principais são as seguintes:

- entrada de um novo registro em um arquivo;
- modificação de um registro de um arquivo;
- localização de um registro em um arquivo pelo nome ou número de identificação do registro;
- cancelamento de um registro em um arquivo.

O gerador foi desenvolvido para criar os dados diretamente por computador. É uma maneira prática para gerar casos de simulação diferentes e independentes. O gerador contém também uma interface entre o computador e o usuário. O usuário pode controlar a criação dos casos de simulação através dos valores de parâmetros, tais como números de AGVs, número de cargas de AGV, períodos de simulação, etc.

O programador de STM foi desenvolvido com base no método analítico de estado e é o módulo chave deste simulador. Utiliza transições de estados nos processos para simular o funcionamento do sistema e registra a seqüência das transições para definir a programação. Relatórios da programação obtida são gerados para os usuários. A Figura 5.4.2 mostra em detalhe o diagrama de fluxo do programador de STM. As transições de estado nesta figura são apresentadas em detalhe no capítulo 4.

O analisador é a parte que avalia o desempenho da programação obtida. Ele analisa medidas, tais como tempo de atraso e utilização de AGVs, obtidas através de vários períodos de simulação num horizonte longo. Os resultados obtidos são relatados aos usuários.

O módulo de *set-up* é desenvolvido para inicializar as variáveis e os estados do sistema com o objetivo de facilitar múltiplas simulações.

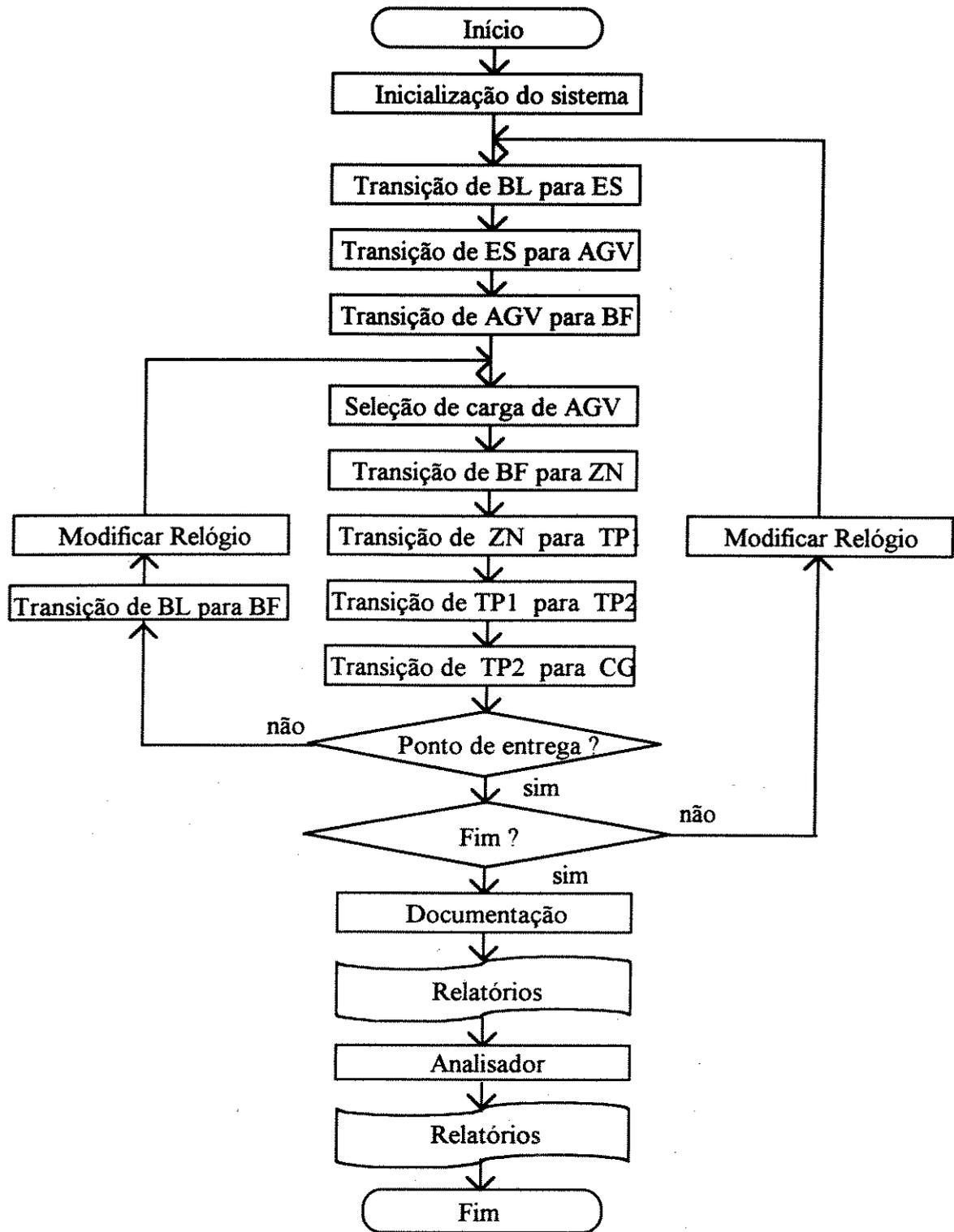


Figura 5.4.2. Diagrama de fluxo do simulador de STM

5.5 Integração dos Programadores

5.5.1 Integração

Os programadores dos sistemas da produção e do transporte podem ser integrados e executados conjuntamente. A idéia básica da integração é seguinte. Quando precisa-se processar uma operação de uma peça, o programador de SP seleciona um centro de trabalho para a operação da peça. A seguir, envia-se um pedido de transporte ao programador de STM para transportar a peça para o centro selecionado. O pedido gera uma tarefa de transporte no STM, conhecida como carga de AGV, através da interface entre os dois programadores. O programador de STM programa o processo de transporte para levar a peça em questão da sua posição atual para o seu próximo centro de trabalho. Após completar o transporte da carga de AGV, o programador de STM passa informações para o programador de SP através da interface. O programador de SP continua a programar o processo de produção a partir do instante em que a peça chegar no centro de trabalho. O processo é repetido até completar o processamento de todas as peças.

Utilizamos o Método Analítico de Estado (MAE) para realizar a integração dos programadores. O modelo construído para esta integração é mostrado na Figura 5.5.1 e é semelhante ao da programação da produção apresentado no capítulo 3. Note que a programação do STM é dependente da programação da produção. Observa-se também, que o estado STM no modelo é um estado de subconjunto, que representa o processo completo da programação do transporte. O detalhamento da programação do transporte é apresentado no capítulo 4. A Figura 5.5.2 mostra a fluxo da programação integrada na parte do sistema de produção. A parte correspondente ao sistema de transporte é apresentado na seção anterior (ver Figura 5.4.2).

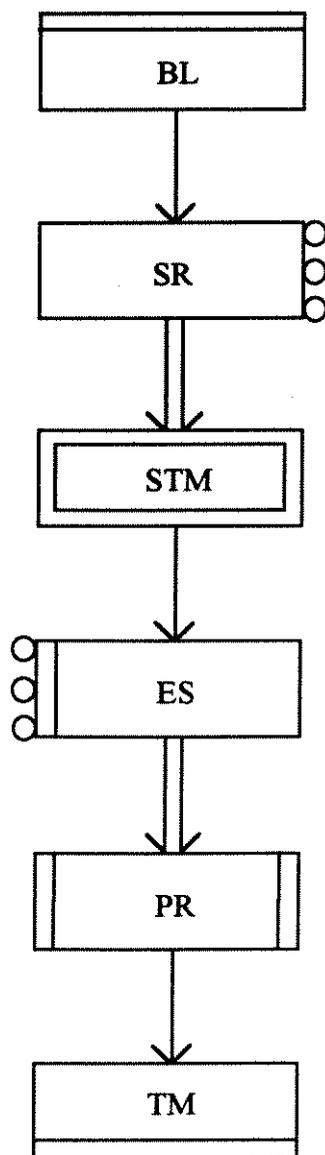


Figura 5.5.1 Modelo de MAE para a programação integrada

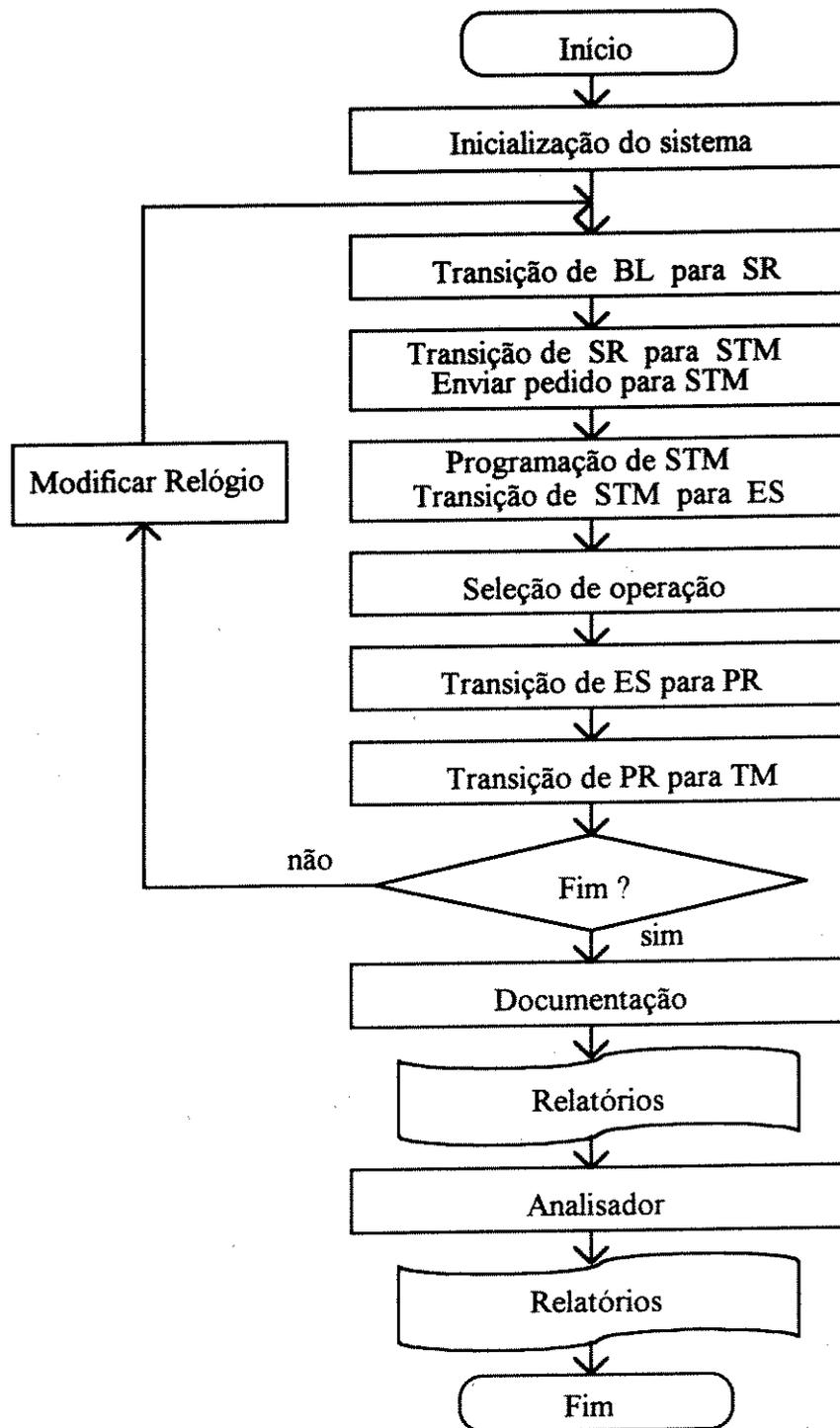


Figura 5.5.2 Diagrama de fluxo integrado do simulador

Para garantir que o início da próxima operação da peça não seja atrasado por causa de transporte, o programador de STM deve programar a entrega da peça no seu centro desejado antes de um certo instante de tempo, denominado data de entrega da carga de AGV. A determinação desta data é feita através da definição da data de entrega da operação associada com esta carga, que foi proposta por Baker (1984) :

$$d_{ij} = d_{ij-1} + (d_i - r_i) p_{ij} / \sum_j p_{ij}$$

$$d_{i0} = r_i$$

onde

d_{ij} : data de entrega da j-ésima operação da peça i

p_{ij} : tempo de processamento da j-ésima operação da peça i

r_i : data de chegada da peça i

d_i : data de entrega da peça i

A data de entrega de uma carga de AGV no seu ponto de entrega é definida da seguinte forma:

$$d_{C_k} = t + T_a + (d_{ij} - t - T_a - p_{ij}) / 2$$

$$= (d_{ij} - p_{ij} + t + T_a) / 2$$

$$= (IT_{ij} + IC_{C_k}) / 2$$

onde

t : tempo atual

d_{C_k} : data de entrega da carga de AGV C_k associada com a j-ésima operação da peça i

T_a : tempo de transporte total da carga de AGV C_k

IT_{ij} : instante de início mais tarde do processamento da j-ésima operação da peça i

IC_{C_k} : instante de chegada mais cedo da carga de AGV C_k

Note que a definição da data de entrega da carga de AGV é tal que metade do tempo de folga $(d_{ij} - t - T_a - p_{ij})$ é alocado ao transporte e a outra metade é alocada à produção.

5.5.2 Exemplo

Para ilustrar a programação integrada entre os sistemas de produção e transporte, utilizamos o *layout* apresentado no capítulo 4, e que está representado na Figura 5.5.1. No *layout* existem 8 centros de trabalho com uma máquina, denominados WC0 a WC7. Consideramos um exemplo simples com 5 AGVs e 5 peças. Os dados foram gerados de modo semelhante ao descrito nos capítulos 3 e 4. A condição inicial é que os AGVs estão no ponto de estacionamento reservado no sistema e as peças estão disponíveis para processamento no seus centros de trabalho desejados. As tarefas principais de programação integrada são programar as operações das peças nos centros de trabalho atuais, definir os próximos centros de trabalho para continuar o processamento e programar os AGVs para transportar as peças dos centros atuais até seus próximos centros.

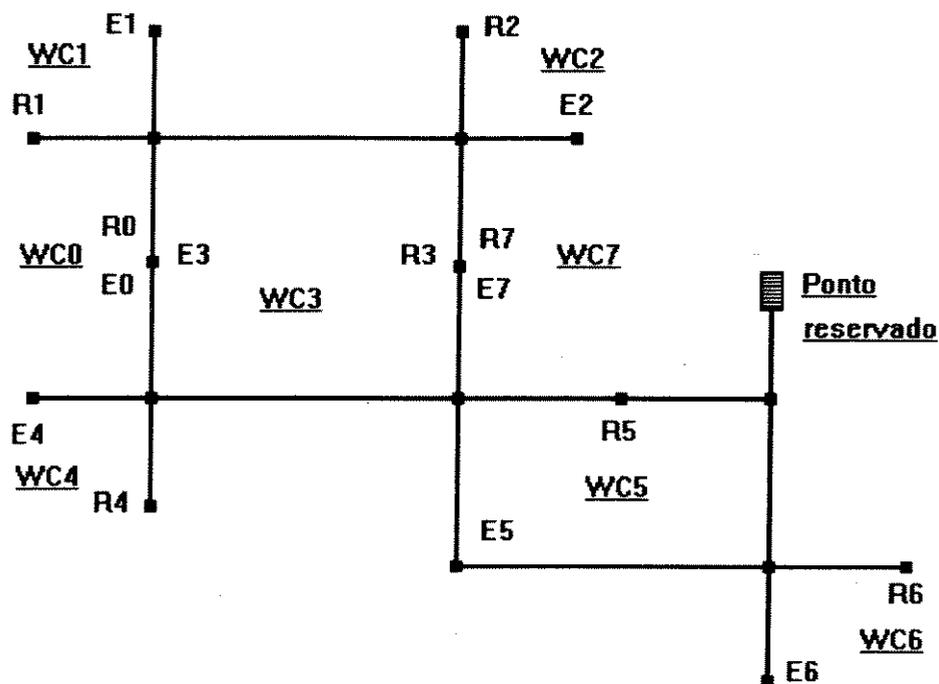


Figura 5.5.1 Exemplo de um sistema integrado

O problema de programação integrada foi resolvido pelo simulador apresentado anteriormente. As informações básicas da programação integrada são apresentadas nos seguintes relatórios.

1. Relatório dos tempos de atraso das peças.
2. Relatório do programa de produção ordenado pelas peças: contém em detalhe todos os programas das operações das peças.
3. Relatório do programa de produção ordenado pelos tempos: apresenta o mesmo programa acima ordenado pelos tempos de início das operações das peças.
4. Relatório de carregamento nos centros de trabalho: contém capacidades, carregamentos e taxa de utilização dos centros.
5. Relatório dos pedidos de transporte e tempos de atraso das cargas.
6. Relatório do programa de transporte das cargas ordenado pelos tempos: contém todos os movimentos das cargas de segmento ordenadas pelos tempos de início de transporte.
7. Relatório do programa de transporte das cargas ordenado pela seqüência de pedidos de transporte.
8. Relatório do programa de transporte dos AGVs: contém todos os movimentos dos AGVs utilizados.

Os primeiros quatro relatórios apresentam informações da programação do sistema de produção e os demais são relativos ao sistema de transporte. Nos relatórios de produção, o código de AGV "NUL" corresponde à primeira operação de uma peça, e que não necessita serviço de transporte, como suposto neste exemplo. Quando o ponto de saída do centro de trabalho da operação predecessora é o mesmo ponto de entrada do centro da operação atual, não há necessidade de transporte. Esta situação é representada pelo código de AGV "NO". Neste caso, o código de segmento da carga e a zona de término deste segmento nos relatórios de transporte são denominados "NUL". Nos relatórios de transporte, o código "AGV" da carga de AGV representa a viagem vazia de um AGV. Uma carga de AGV CA-x é associada à operação com número x. Por exemplo, a carga de AGV CA-19 do primeiro pedido de transporte R-1 no relatório 5 é gerada pela operação número 19, cujo código é P4-O2 como mostrado no relatório 2.

A seguir descrevemos um exemplo de programação da produção e transporte através dos relatórios correspondentes. Observa-se através dos relatórios 2 e 3 que a peça P4 está inicialmente disponível no centro de trabalho WC1. O início da primeira operação P4-O1 está programado para o instante 0 e o término para o instante 383. A operação seguinte P4-O2 é alocada para o centro WC7. O programador de SP gera um pedido de transporte ao programador de STM para transportar P4 de Z-0, ponto de retirada do centro atual WC1, até Z-7, ponto de entrega do próximo centro WC7. O relatório 8 mostra que o programador de STM programa o AGV A-1 (vazio) para movimentar-se do seu ponto de estacionamento Z-18, no instante 383, até a zona Z-0, no instante 580, através de 7 cargas de segmento. Neste instante, P4 é carregada no AGV A-1 e transportada para o ponto de entrega Z-7 do centro WC7 no instante 683. A partir daí, o relatório 3 mostra que o programador de SP programa o início da operação P4-O2 para o instante 683 e término no instante 1176. A seguir, o programador da produção aloca a próxima operação P4-O3 para o centro de trabalho WC-5 e faz outro pedido de transporte ao programador de STM para transportar a peça do ponto Z-7 até o ponto Z-13. O relatório 8 mostra que o programador de STM envia o mesmo AGV A-1, atualmente situado no ponto Z-13, no instante 1176, para a zona Z-7, lá chegando no instante 1228. Neste instante a peça é carregada e o AGV A-1 transporta-a até o ponto de entrega Z-13 do centro de trabalho WC-5 no instante 1280. Como tem fila em WC5, o relatório 3 mostra que a operação P4-O3 espera até o instante 1726 para começar seu processamento que é completado no instante 2420. Neste instante, termina o processamento da peça P4. Observa-se através do relatório 8 que somente três AGVs foram utilizados neste exemplo.

Relatório 1.**RELATÓRIO DOS TEMPOS DE ATRASO
DAS PEÇAS NO SISTEMA DE PRODUÇÃO**

Nome de peça	Nº total de oper.	Data de chegada	Data de entrega	Tempo de início	Tempo de término	Tempo de atraso
P-1	7	0	11074	0	5198	0
P-2	8	0	11204	406	5940	0
P-3	2	0	2470	0	1274	0
P-4	3	0	3940	0	2420	0
P-5	2	0	3460	0	1726	0

Relatório 2.**RELATÓRIO DO PROGRAMA DE PRODUÇÃO
ORDENADO PELAS PEÇAS**

Código da oper.	Nº da oper.	Nome da peça	j-ésima oper.	Centro de trab.	Zona de entrada	Zona de saída	Código de AGV	Tempo de proc.	Tempo de início	Tempo de término
P1-O1	1	P-1	1	WC-0	Z-6	Z-6	NUL	406	0	406
P1-O2	2	P-1	2	WC-3	Z-6	Z-7	NO	902	406	1308
P1-O3	3	P-1	3	WC-2	Z-5	Z-3	A-1	635	1412	2047
P1-O4	4	P-1	4	WC-4	Z-8	Z-10	A-1	453	2244	2697
P1-O5	5	P-1	5	WC-6	Z-15	Z-16	A-1	910	2904	3814
P1-O6	6	P-1	6	WC-2	Z-5	Z-3	A-1	726	4032	4758
P1-O7	7	P-1	7	WC-1	Z-2	Z-0	A-1	305	4893	5198
P2-O1	8	P-2	1	WC-0	Z-6	Z-6	NUL	967	406	1373
P2-O2	9	P-2	2	WC-7	Z-7	Z-7	A-2	110	1538	1648
P2-O3	10	P-2	3	WC-7	Z-7	Z-7	NO	199	1648	1847
P2-O4	11	P-2	4	WC-7	Z-7	Z-7	NO	896	1847	2743
P2-O5	12	P-2	5	WC-3	Z-6	Z-7	A-2	205	2856	3061
P2-O6	13	P-2	6	WC-6	Z-15	Z-16	A-3	776	3814	4590
P2-O7	14	P-2	7	WC-4	Z-8	Z-10	A-3	367	4797	5164
P2-O8	15	P-2	8	WC-3	Z-6	Z-7	A-3	682	5258	5940
P3-O1	16	P-3	1	WC-2	Z-5	Z-3	NUL	484	0	484
P3-O2	17	P-3	2	WC-1	Z-2	Z-0	A-2	551	723	1274
P4-O1	18	P-4	1	WC-1	Z-2	Z-0	NUL	383	0	383
P4-O2	19	P-4	2	WC-7	Z-7	Z-7	A-1	493	683	1176
P4-O3	20	P-4	3	WC-5	Z-13	Z-12	A-1	694	1726	2420
P5-O1	21	P-5	1	WC-4	Z-8	Z-10	NUL	732	0	732
P5-O2	22	P-5	2	WC-5	Z-13	Z-12	A-1	798	928	1726

Relatório 3.**RELATÓRIO DO PROGRAMA DE PRODUÇÃO
ORDENADO PELOS TEMPOS**

Código da oper.	Nº da oper.	Nome da peça	j-ésima oper.	Centro de trab.	Zona de entrada	Zona de saída	Cód. de AGV	Tempo de proc.	Tempo de início	Tempo de término
P1-O1	1	P-1	1	WC-0	Z-6	Z-6	NUL	406	0	406
P3-O1	16	P-3	1	WC-2	Z-5	Z-3	NUL	484	0	484
P4-O1	18	P-4	1	WC-1	Z-2	Z-0	NUL	383	0	383
P5-O1	21	P-5	1	WC-4	Z-8	Z-10	NUL	732	0	732
P1-O2	2	P-1	2	WC-3	Z-6	Z-7	NO	902	406	1308
P2-O1	8	P-2	1	WC-0	Z-6	Z-6	NUL	967	406	1373
P4-O2	19	P-4	2	WC-7	Z-7	Z-7	A-1	493	683	1176
P3-O2	17	P-3	2	WC-1	Z-2	Z-0	A-2	551	723	1274
P5-O2	22	P-5	2	WC-5	Z-13	Z-12	A-1	798	928	1726
P1-O3	3	P-1	3	WC-2	Z-5	Z-3	A-1	635	1412	2047
P2-O2	9	P-2	2	WC-7	Z-7	Z-7	A-2	110	1538	1648
P2-O3	10	P-2	3	WC-7	Z-7	Z-7	NO	199	1648	1847
P4-O3	20	P-4	3	WC-5	Z-13	Z-12	A-1	694	1726	2420
P2-O4	11	P-2	4	WC-7	Z-7	Z-7	NO	896	1847	2743
P1-O4	4	P-1	4	WC-4	Z-8	Z-10	A-1	453	2244	2697
P2-O5	12	P-2	5	WC-3	Z-6	Z-7	A-2	205	2856	3061
P1-O5	5	P-1	5	WC-6	Z-15	Z-16	A-1	910	2904	3814
P2-O6	13	P-2	6	WC-6	Z-15	Z-16	A-3	776	3814	4590
P1-O6	6	P-1	6	WC-2	Z-5	Z-3	A-1	726	4032	4758
P2-O7	14	P-2	7	WC-4	Z-8	Z-10	A-3	367	4797	5164
P1-O7	7	P-1	7	WC-1	Z-2	Z-0	A-1	305	4893	5198
P2-O8	15	P-2	8	WC-3	Z-6	Z-7	A-3	682	5258	5940

Relatório 4.

RELATÓRIO DE CARREGAMENTO NOS CENTROS DE TRABALHO

Código do centro de trab.	Nº do centro de trab.	Nº de máq. no centro	Zona de entrada	Zona de saída	Capacidade total	Carga total	Taxa de util.
WC-0	0	1	Z-6	Z-6	5940	1373	23.11
WC-1	1	1	Z-2	Z-0	5940	1239	20.86
WC-2	2	1	Z-5	Z-3	5940	1845	31.06
WC-3	3	1	Z-6	Z-7	5940	1789	30.12
WC-4	4	1	Z-8	Z-10	5940	1552	26.13
WC-5	5	1	Z-13	Z-12	5940	1492	25.12
WC-6	6	1	Z-15	Z-16	5940	1686	28.38
WC-7	7	1	Z-7	Z-7	5940	1698	28.59

Relatório 5.

RELATÓRIO DOS PEDIDOS DE TRANSPORTE E TEMPOS DE ATRASO DAS CARGAS

Código do pedido	Código da carga de AGV	Cód. de AGV	Zona de entrada Z _E	Zona de saída Z _S	Data de chegada	Data de entrega	Tempo de início em Z _E	Tempo de término em Z _S	Tempo de atraso
R-1	CA-19	A-1	Z-0	Z-7	383	1071	580	683	0
R-2	AGV	A-1	Z-18	Z-0	383	971	383	580	0
R-3	CA-2	NO	Z-6	Z-6	406	1260	406	406	0
R-4	CA-17	A-2	Z-3	Z-2	484	1246	630	723	0
R-5	AGV	A-2	Z-18	Z-3	484	1156	484	630	0
R-6	CA-22	A-1	Z-10	Z-13	732	1742	835	928	0
R-7	AGV	A-1	Z-7	Z-10	732	1652	732	835	0
R-8	CA-20	A-1	Z-7	Z-13	1176	2236	1228	1280	0
R-9	AGV	A-1	Z-13	Z-7	1176	2186	1176	1228	0
R-10	CA-3	A-1	Z-7	Z-5	1308	2704	1360	1412	0
R-11	AGV	A-1	Z-13	Z-7	1308	2654	1308	1360	0
R-12	CA-9	A-2	Z-6	Z-7	1373	1963	1425	1538	0
R-13	AGV	A-2	Z-2	Z-6	1373	1853	1373	1425	0
R-14	CA-10	NO	Z-7	Z-7	1648	2400	1648	1648	0
R-15	CA-11	NO	Z-7	Z-7	1847	3247	1847	1847	0
R-16	CA-4	A-1	Z-3	Z-8	2047	3868	2089	2244	0
R-17	AGV	A-1	Z-5	Z-3	2047	3718	2047	2089	0
R-18	CA-5	A-1	Z-10	Z-15	2697	5079	2739	2904	0
R-19	AGV	A-1	Z-8	Z-10	2697	4919	2697	2739	0
R-20	CA-12	A-2	Z-7	Z-6	2743	4501	2743	2856	0
R-21	CA-13	A-3	Z-7	Z-15	3061	5355	3155	3279	0
R-22	AGV	A-3	Z-18	Z-7	3061	5235	3061	3155	0
R-23	CA-6	A-1	Z-16	Z-5	3814	6661	3856	4032	0
R-24	AGV	A-1	Z-15	Z-16	3814	6491	3814	3856	0
R-25	CA-14	A-3	Z-16	Z-8	4590	6911	4632	4797	0
R-26	AGV	A-3	Z-15	Z-16	4590	6751	4590	4632	0
R-27	CA-7	A-1	Z-3	Z-2	4758	7808	4800	4893	0
R-28	AGV	A-1	Z-5	Z-3	4758	7718	4758	4800	0
R-29	CA-15	A-3	Z-10	Z-6	5164	7868	5206	5258	0
R-30	AGV	A-3	Z-8	Z-10	5164	7818	5164	5206	0

Relatório 6.

RELATÓRIO DO PROGRAMA DE TRANSPORTE DAS CARGAS ORDENADO PELOS TEMPOS

Código da carga de seg.	Carga de seg.	Código da carga de AGV	Cód. de AGV	Zona de entrad. Z _E	Zona de saída Z _S	Tempo de início em Z _E	Tempo de término em Z _E	Cód. de seg.	Tempo de transp.	Tempo de início de transp.	Tempo de término de transp.
R2-S1	1	AGV	A-1	Z-18	Z-17	383	384	SG-40	20	383	404
R2-S2	2	AGV	A-1	Z-17	Z-12	404	405	SG-37	20	404	425
R3-S1	1	CA-2	NO	Z-6	NUL	0	0	NUL	0	406	406
R2-S3	3	AGV	A-1	Z-12	Z-11	425	426	SG-27	20	425	446
R2-S4	4	AGV	A-1	Z-11	Z-9	446	447	SG-24	50	446	497
R5-S1	1	AGV	A-2	Z-18	Z-17	484	485	SG-40	20	484	505
R2-S5	5	AGV	A-1	Z-9	Z-6	497	498	SG-18	30	497	528
R5-S2	2	AGV	A-2	Z-17	Z-12	505	506	SG-37	20	505	526
R5-S3	3	AGV	A-2	Z-12	Z-11	526	527	SG-27	20	526	547
R2-S6	6	AGV	A-1	Z-6	Z-1	528	529	SG-13	30	528	559
R5-S4	4	AGV	A-2	Z-11	Z-7	547	548	SG-23	30	547	578
R2-S7	7	AGV	A-1	Z-1	Z-0	559	560	SG-2	20	559	580
R5-S5	5	AGV	A-2	Z-7	Z-4	578	579	SG-15	30	578	609
R1-S1	1	CA-19	A-1	Z-0	Z-1	580	581	SG-1	20	580	601
R1-S2	2	CA-19	A-1	Z-1	Z-4	601	602	SG-4	50	601	652
R5-S6	6	AGV	A-2	Z-4	Z-3	609	610	SG-9	20	609	630
R4-S1	1	CA-17	A-2	Z-3	Z-4	630	631	SG-7	20	630	651
R4-S2	2	CA-17	A-2	Z-4	Z-1	651	652	SG-8	50	651	702
R1-S3	3	CA-19	A-1	Z-4	Z-7	652	653	SG-11	30	652	683
R4-S3	3	CA-17	A-2	Z-1	Z-2	702	703	SG-3	20	702	723
R7-S1	1	AGV	A-1	Z-7	Z-11	732	733	SG-16	30	732	763
R7-S2	2	AGV	A-1	Z-11	Z-9	763	764	SG-24	50	763	814
R7-S3	3	AGV	A-1	Z-9	Z-10	814	815	SG-20	20	814	835
R6-S1	1	CA-22	A-1	Z-10	Z-9	835	836	SG-22	20	835	856
R6-S2	2	CA-22	A-1	Z-9	Z-11	856	857	SG-21	50	856	907
R6-S3	3	CA-22	A-1	Z-11	Z-13	907	908	SG-26	20	907	928
R9-S1	1	AGV	A-1	Z-13	Z-11	1176	1177	SG-29	20	1176	1197
R9-S2	2	AGV	A-1	Z-11	Z-7	1197	1198	SG-23	30	1197	1228
R8-S1	1	CA-20	A-1	Z-7	Z-11	1228	1229	SG-16	30	1228	1259
R8-S2	2	CA-20	A-1	Z-11	Z-13	1259	1260	SG-26	20	1259	1280
R11-S1	1	AGV	A-1	Z-13	Z-11	1308	1309	SG-29	20	1308	1329
R11-S2	2	AGV	A-1	Z-11	Z-7	1329	1330	SG-23	30	1329	1360
R10-S1	1	CA-3	A-1	Z-7	Z-4	1360	1361	SG-15	30	1360	1391
R13-S1	1	AGV	A-2	Z-2	Z-1	1373	1374	SG-6	20	1373	1394
R10-S2	2	CA-3	A-1	Z-4	Z-5	1391	1392	SG-10	20	1391	1412
R13-S2	2	AGV	A-2	Z-1	Z-6	1394	1395	SG-5	30	1394	1425
R12-S1	1	CA-9	A-2	Z-6	Z-1	1425	1426	SG-13	30	1425	1456
R12-S2	2	CA-9	A-2	Z-1	Z-4	1456	1457	SG-4	50	1456	1507
R12-S3	3	CA-9	A-2	Z-4	Z-7	1507	1508	SG-11	30	1507	1538
R14-S1	1	CA-10	NO	Z-7	NUL	0	0	NUL	0	1648	1648

R15-S1	1	CA-11	NO	Z-7	NUL	0	0	NUL	0	1847	1847
R17-S1	1	AGV	A-1	Z-5	Z-4	2047	2048	SG-12	20	2047	2068
R17-S2	2	AGV	A-1	Z-4	Z-3	2068	2069	SG-9	20	2068	2089
R16-S1	1	CA-4	A-1	Z-3	Z-4	2089	2090	SG-7	20	2089	2110
R16-S2	2	CA-4	A-1	Z-4	Z-1	2110	2111	SG-8	50	2110	2161
R16-S3	3	CA-4	A-1	Z-1	Z-6	2161	2162	SG-5	30	2161	2192
R16-S4	4	CA-4	A-1	Z-6	Z-9	2192	2193	SG-14	30	2192	2223
R16-S5	5	CA-4	A-1	Z-9	Z-8	2223	2224	SG-19	20	2223	2244
R19-S1	1	AGV	A-1	Z-8	Z-9	2697	2698	SG-17	20	2697	2718
R19-S2	2	AGV	A-1	Z-9	Z-10	2718	2719	SG-20	20	2718	2739
R18-S1	1	CA-5	A-1	Z-10	Z-9	2739	2740	SG-22	20	2739	2760
R20-S1	1	CA-12	A-2	Z-7	Z-4	2743	2744	SG-15	30	2743	2774
R18-S2	2	CA-5	A-1	Z-9	Z-11	2760	2761	SG-21	50	2760	2811
R20-S2	2	CA-12	A-2	Z-4	Z-1	2774	2775	SG-8	50	2774	2825
R18-S3	3	CA-5	A-1	Z-11	Z-13	2811	2812	SG-26	20	2811	2832
R20-S3	3	CA-12	A-2	Z-1	Z-6	2825	2826	SG-5	30	2825	2856
R18-S4	4	CA-5	A-1	Z-13	Z-14	2832	2833	SG-30	50	2832	2883
R18-S5	5	CA-5	A-1	Z-14	Z-15	2883	2884	SG-32	20	2883	2904
R22-S1	1	AGV	A-3	Z-18	Z-17	3061	3062	SG-40	20	3061	3082
R22-S2	2	AGV	A-3	Z-17	Z-12	3082	3083	SG-37	20	3082	3103
R22-S3	3	AGV	A-3	Z-12	Z-11	3103	3104	SG-27	20	3103	3124
R22-S4	4	AGV	A-3	Z-11	Z-7	3124	3125	SG-23	30	3124	3155
R21-S1	1	CA-13	A-3	Z-7	Z-11	3155	3156	SG-16	30	3155	3186
R21-S2	2	CA-13	A-3	Z-11	Z-13	3186	3187	SG-26	20	3186	3207
R21-S3	3	CA-13	A-3	Z-13	Z-14	3207	3208	SG-30	50	3207	3258
R21-S4	4	CA-13	A-3	Z-14	Z-15	3258	3259	SG-32	20	3258	3279
R24-S1	1	AGV	A-1	Z-15	Z-14	3814	3815	SG-35	20	3814	3835
R24-S2	2	AGV	A-1	Z-14	Z-16	3835	3836	SG-33	20	3835	3856
R23-S1	1	CA-6	A-1	Z-16	Z-14	3856	3857	SG-36	20	3856	3877
R23-S2	2	CA-6	A-1	Z-14	Z-13	3877	3878	SG-31	50	3877	3928
R23-S3	3	CA-6	A-1	Z-13	Z-11	3928	3929	SG-29	20	3928	3949
R23-S4	4	CA-6	A-1	Z-11	Z-7	3949	3950	SG-23	30	3949	3980
R23-S5	5	CA-6	A-1	Z-7	Z-4	3980	3981	SG-15	30	3980	4011
R23-S6	6	CA-6	A-1	Z-4	Z-5	4011	4012	SG-10	20	4011	4032
R26-S1	1	AGV	A-3	Z-15	Z-14	4590	4591	SG-35	20	4590	4611
R26-S2	2	AGV	A-3	Z-14	Z-16	4611	4612	SG-33	20	4611	4632
R25-S1	1	CA-14	A-3	Z-16	Z-14	4632	4633	SG-36	20	4632	4653
R25-S2	2	CA-14	A-3	Z-14	Z-13	4653	4654	SG-31	50	4653	4704
R25-S3	3	CA-14	A-3	Z-13	Z-11	4704	4705	SG-29	20	4704	4725
R25-S4	4	CA-14	A-3	Z-11	Z-9	4725	4726	SG-24	50	4725	4776
R28-S1	1	AGV	A-1	Z-5	Z-4	4758	4759	SG-12	20	4758	4779
R25-S5	5	CA-14	A-3	Z-9	Z-8	4776	4777	SG-19	20	4776	4797
R28-S2	2	AGV	A-1	Z-4	Z-3	4779	4780	SG-9	20	4779	4800
R27-S1	1	CA-7	A-1	Z-3	Z-4	4800	4801	SG-7	20	4800	4821
R27-S2	2	CA-7	A-1	Z-4	Z-1	4821	4822	SG-8	50	4821	4872
R27-S3	3	CA-7	A-1	Z-1	Z-2	4872	4873	SG-3	20	4872	4893
R30-S1	1	AGV	A-3	Z-8	Z-9	5164	5165	SG-17	20	5164	5185
R30-S2	2	AGV	A-3	Z-9	Z-10	5185	5186	SG-20	20	5185	5206
R29-S1	1	CA-15	A-3	Z-10	Z-9	5206	5207	SG-22	20	5206	5227
R29-S2	2	CA-15	A-3	Z-9	Z-6	5227	5228	SG-18	30	5227	5258

Relatório 7.

**RELATÓRIO DO PROGRAMA DE TRANSPORTE
DAS CARGAS ORDENADO PELA SEQUÊNCIA DE
PEDIDOS DE TRANSPORTE**

Código da carga de seg.	Carga de seg.	Código da carga de AGV	Cód. de AGV	Zona de entrad. Z _E	Zona de saída Z _S	Temp o de início em Z _E	Temp o de término em Z _E	Cód. de seg.	Temp o de transp.	Tempo de início de transp.	Tempo de término de transp.
R1-S1	1	CA-19	A-1	Z-0	Z-1	580	581	SG-1	20	580	601
R1-S2	2	CA-19	A-1	Z-1	Z-4	601	602	SG-4	50	601	652
R1-S3	3	CA-19	A-1	Z-4	Z-7	652	653	SG-11	30	652	683
R2-S1	1	AGV	A-1	Z-18	Z-17	383	384	SG-40	20	383	404
R2-S2	2	AGV	A-1	Z-17	Z-12	404	405	SG-37	20	404	425
R2-S3	3	AGV	A-1	Z-12	Z-11	425	426	SG-27	20	425	446
R2-S4	4	AGV	A-1	Z-11	Z-9	446	447	SG-24	50	446	497
R2-S5	5	AGV	A-1	Z-9	Z-6	497	498	SG-18	30	497	528
R2-S6	6	AGV	A-1	Z-6	Z-1	528	529	SG-13	30	528	559
R2-S7	7	AGV	A-1	Z-1	Z-0	559	560	SG-2	20	559	580
R3-S1	1	CA-2	NO	Z-6	NUL	0	0	NUL	0	406	406
R4-S1	1	CA-17	A-2	Z-3	Z-4	630	631	SG-7	20	630	651
R4-S2	2	CA-17	A-2	Z-4	Z-1	651	652	SG-8	50	651	702
R4-S3	3	CA-17	A-2	Z-1	Z-2	702	703	SG-3	20	702	723
R5-S1	1	AGV	A-2	Z-18	Z-17	484	485	SG-40	20	484	505
R5-S2	2	AGV	A-2	Z-17	Z-12	505	506	SG-37	20	505	526
R5-S3	3	AGV	A-2	Z-12	Z-11	526	527	SG-27	20	526	547
R5-S4	4	AGV	A-2	Z-11	Z-7	547	548	SG-23	30	547	578
R5-S5	5	AGV	A-2	Z-7	Z-4	578	579	SG-15	30	578	609
R5-S6	6	AGV	A-2	Z-4	Z-3	609	610	SG-9	20	609	630
R6-S1	1	CA-22	A-1	Z-10	Z-9	835	836	SG-22	20	835	856
R6-S2	2	CA-22	A-1	Z-9	Z-11	856	857	SG-21	50	856	907
R6-S3	3	CA-22	A-1	Z-11	Z-13	907	908	SG-26	20	907	928
R7-S1	1	AGV	A-1	Z-7	Z-11	732	733	SG-16	30	732	763
R7-S2	2	AGV	A-1	Z-11	Z-9	763	764	SG-24	50	763	814
R7-S3	3	AGV	A-1	Z-9	Z-10	814	815	SG-20	20	814	835
R8-S1	1	CA-20	A-1	Z-7	Z-11	1228	1229	SG-16	30	1228	1259
R8-S2	2	CA-20	A-1	Z-11	Z-13	1259	1260	SG-26	20	1259	1280
R9-S1	1	AGV	A-1	Z-13	Z-11	1176	1177	SG-29	20	1176	1197
R9-S2	2	AGV	A-1	Z-11	Z-7	1197	1198	SG-23	30	1197	1228
R10-S1	1	CA-3	A-1	Z-7	Z-4	1360	1361	SG-15	30	1360	1391
R10-S2	2	CA-3	A-1	Z-4	Z-5	1391	1392	SG-10	20	1391	1412
R11-S1	1	AGV	A-1	Z-13	Z-11	1308	1309	SG-29	20	1308	1329
R11-S2	2	AGV	A-1	Z-11	Z-7	1329	1330	SG-23	30	1329	1360
R12-S1	1	CA-9	A-2	Z-6	Z-1	1425	1426	SG-13	30	1425	1456
R12-S2	2	CA-9	A-2	Z-1	Z-4	1456	1457	SG-4	50	1456	1507
R12-S3	3	CA-9	A-2	Z-4	Z-7	1507	1508	SG-11	30	1507	1538
R13-S1	1	AGV	A-2	Z-2	Z-1	1373	1374	SG-6	20	1373	1394
R13-S2	2	AGV	A-2	Z-1	Z-6	1394	1395	SG-5	30	1394	1425
R14-S1	1	CA-10	NO	Z-7	NUL	0	0	NUL	0	1648	1648

R15-S1	1	CA-11	NO	Z-7	NUL	0	0	NUL	0	1847	1847
R16-S1	1	CA-4	A-1	Z-3	Z-4	2089	2090	SG-7	20	2089	2110
R16-S2	2	CA-4	A-1	Z-4	Z-1	2110	2111	SG-8	50	2110	2161
R16-S3	3	CA-4	A-1	Z-1	Z-6	2161	2162	SG-5	30	2161	2192
R16-S4	4	CA-4	A-1	Z-6	Z-9	2192	2193	SG-14	30	2192	2223
R16-S5	5	CA-4	A-1	Z-9	Z-8	2223	2224	SG-19	20	2223	2244
R17-S1	1	AGV	A-1	Z-5	Z-4	2047	2048	SG-12	20	2047	2068
R17-S2	2	AGV	A-1	Z-4	Z-3	2068	2069	SG-9	20	2068	2089
R18-S1	1	CA-5	A-1	Z-10	Z-9	2739	2740	SG-22	20	2739	2760
R18-S2	2	CA-5	A-1	Z-9	Z-11	2760	2761	SG-21	50	2760	2811
R18-S3	3	CA-5	A-1	Z-11	Z-13	2811	2812	SG-26	20	2811	2832
R18-S4	4	CA-5	A-1	Z-13	Z-14	2832	2833	SG-30	50	2832	2883
R18-S5	5	CA-5	A-1	Z-14	Z-15	2883	2884	SG-32	20	2883	2904
R19-S1	1	AGV	A-1	Z-8	Z-9	2697	2698	SG-17	20	2697	2718
R19-S2	2	AGV	A-1	Z-9	Z-10	2718	2719	SG-20	20	2718	2739
R20-S1	1	CA-12	A-2	Z-7	Z-4	2743	2744	SG-15	30	2743	2774
R20-S2	2	CA-12	A-2	Z-4	Z-1	2774	2775	SG-8	50	2774	2825
R20-S3	3	CA-12	A-2	Z-1	Z-6	2825	2826	SG-5	30	2825	2856
R21-S1	1	CA-13	A-3	Z-7	Z-11	3155	3156	SG-16	30	3155	3186
R21-S2	2	CA-13	A-3	Z-11	Z-13	3186	3187	SG-26	20	3186	3207
R21-S3	3	CA-13	A-3	Z-13	Z-14	3207	3208	SG-30	50	3207	3258
R21-S4	4	CA-13	A-3	Z-14	Z-15	3258	3259	SG-32	20	3258	3279
R22-S1	1	AGV	A-3	Z-18	Z-17	3061	3062	SG-40	20	3061	3082
R22-S2	2	AGV	A-3	Z-17	Z-12	3082	3083	SG-37	20	3082	3103
R22-S3	3	AGV	A-3	Z-12	Z-11	3103	3104	SG-27	20	3103	3124
R22-S4	4	AGV	A-3	Z-11	Z-7	3124	3125	SG-23	30	3124	3155
R23-S1	1	CA-6	A-1	Z-16	Z-14	3856	3857	SG-36	20	3856	3877
R23-S2	2	CA-6	A-1	Z-14	Z-13	3877	3878	SG-31	50	3877	3928
R23-S3	3	CA-6	A-1	Z-13	Z-11	3928	3929	SG-29	20	3928	3949
R23-S4	4	CA-6	A-1	Z-11	Z-7	3949	3950	SG-23	30	3949	3980
R23-S5	5	CA-6	A-1	Z-7	Z-4	3980	3981	SG-15	30	3980	4011
R23-S6	6	CA-6	A-1	Z-4	Z-5	4011	4012	SG-10	20	4011	4032
R24-S1	1	AGV	A-1	Z-15	Z-14	3814	3815	SG-35	20	3814	3835
R24-S2	2	AGV	A-1	Z-14	Z-16	3835	3836	SG-33	20	3835	3856
R25-S1	1	CA-14	A-3	Z-16	Z-14	4632	4633	SG-36	20	4632	4653
R25-S2	2	CA-14	A-3	Z-14	Z-13	4653	4654	SG-31	50	4653	4704
R25-S3	3	CA-14	A-3	Z-13	Z-11	4704	4705	SG-29	20	4704	4725
R25-S4	4	CA-14	A-3	Z-11	Z-9	4725	4726	SG-24	50	4725	4776
R25-S5	5	CA-14	A-3	Z-9	Z-8	4776	4777	SG-19	20	4776	4797
R26-S1	1	AGV	A-3	Z-15	Z-14	4590	4591	SG-35	20	4590	4611
R26-S2	2	AGV	A-3	Z-14	Z-16	4611	4612	SG-33	20	4611	4632
R27-S1	1	CA-7	A-1	Z-3	Z-4	4800	4801	SG-7	20	4800	4821
R27-S2	2	CA-7	A-1	Z-4	Z-1	4821	4822	SG-8	50	4821	4872
R27-S3	3	CA-7	A-1	Z-1	Z-2	4872	4873	SG-3	20	4872	4893
R28-S1	1	AGV	A-1	Z-5	Z-4	4758	4759	SG-12	20	4758	4779
R28-S2	2	AGV	A-1	Z-4	Z-3	4779	4780	SG-9	20	4779	4800
R29-S1	1	CA-15	A-3	Z-10	Z-9	5206	5207	SG-22	20	5206	5227
R29-S2	2	CA-15	A-3	Z-9	Z-6	5227	5228	SG-18	30	5227	5258
R30-S1	1	AGV	A-3	Z-8	Z-9	5164	5165	SG-17	20	5164	5185
R30-S2	2	AGV	A-3	Z-9	Z-10	5185	5186	SG-20	20	5185	5206

Relatório 8.

RELATÓRIO DO PROGRAMA DE TRANSPORTE DOS AGVs

Código de AGV	Código da carga de seg.	Carga de seg.	Código da carga de AGV	Zona de entrad. Z _E	Zona de saída Z _S	Tempo de início em Z _E	Tempo de término em Z _E	Cód. de seg.	Tempo de transp.	Tempo de início de transp.	Tempo de término de transp.
A-1	R2-S1	1	AGV	Z-18	Z-17	383	384	SG-40	20	383	404
A-1	R2-S2	2	AGV	Z-17	Z-12	404	405	SG-37	20	404	425
A-1	R2-S3	3	AGV	Z-12	Z-11	425	426	SG-27	20	425	446
A-1	R2-S4	4	AGV	Z-11	Z-9	446	447	SG-24	50	446	497
A-1	R2-S5	5	AGV	Z-9	Z-6	497	498	SG-18	30	497	528
A-1	R2-S6	6	AGV	Z-6	Z-1	528	529	SG-13	30	528	559
A-1	R2-S7	7	AGV	Z-1	Z-0	559	560	SG-2	20	559	580
A-1	R1-S1	1	CA-19	Z-0	Z-1	580	581	SG-1	20	580	601
A-1	R1-S2	2	CA-19	Z-1	Z-4	601	602	SG-4	50	601	652
A-1	R1-S3	3	CA-19	Z-4	Z-7	652	653	SG-11	30	652	683
A-1	R7-S1	1	AGV	Z-7	Z-11	732	733	SG-16	30	732	763
A-1	R7-S2	2	AGV	Z-11	Z-9	763	764	SG-24	50	763	814
A-1	R7-S3	3	AGV	Z-9	Z-10	814	815	SG-20	20	814	835
A-1	R6-S1	1	CA-22	Z-10	Z-9	835	836	SG-22	20	835	856
A-1	R6-S2	2	CA-22	Z-9	Z-11	856	857	SG-21	50	856	907
A-1	R6-S3	3	CA-22	Z-11	Z-13	907	908	SG-26	20	907	928
A-1	R9-S1	1	AGV	Z-13	Z-11	1176	1177	SG-29	20	1176	1197
A-1	R9-S2	2	AGV	Z-11	Z-7	1197	1198	SG-23	30	1197	1228
A-1	R8-S1	1	CA-20	Z-7	Z-11	1228	1229	SG-16	30	1228	1259
A-1	R8-S2	2	CA-20	Z-11	Z-13	1259	1260	SG-26	20	1259	1280
A-1	R11-S1	1	AGV	Z-13	Z-11	1308	1309	SG-29	20	1308	1329
A-1	R11-S2	2	AGV	Z-11	Z-7	1329	1330	SG-23	30	1329	1360
A-1	R10-S1	1	CA-3	Z-7	Z-4	1360	1361	SG-15	30	1360	1391
A-1	R10-S2	2	CA-3	Z-4	Z-5	1391	1392	SG-10	20	1391	1412
A-1	R17-S1	1	AGV	Z-5	Z-4	2047	2048	SG-12	20	2047	2068
A-1	R17-S2	2	AGV	Z-4	Z-3	2068	2069	SG-9	20	2068	2089
A-1	R16-S1	1	CA-4	Z-3	Z-4	2089	2090	SG-7	20	2089	2110
A-1	R16-S2	2	CA-4	Z-4	Z-1	2110	2111	SG-8	50	2110	2161
A-1	R16-S3	3	CA-4	Z-1	Z-6	2161	2162	SG-5	30	2161	2192
A-1	R16-S4	4	CA-4	Z-6	Z-9	2192	2193	SG-14	30	2192	2223
A-1	R16-S5	5	CA-4	Z-9	Z-8	2223	2224	SG-19	20	2223	2244
A-1	R19-S1	1	AGV	Z-8	Z-9	2697	2698	SG-17	20	2697	2718
A-1	R19-S2	2	AGV	Z-9	Z-10	2718	2719	SG-20	20	2718	2739
A-1	R18-S1	1	CA-5	Z-10	Z-9	2739	2740	SG-22	20	2739	2760
A-1	R18-S2	2	CA-5	Z-9	Z-11	2760	2761	SG-21	50	2760	2811
A-1	R18-S3	3	CA-5	Z-11	Z-13	2811	2812	SG-26	20	2811	2832
A-1	R18-S4	4	CA-5	Z-13	Z-14	2832	2833	SG-30	50	2832	2883
A-1	R18-S5	5	CA-5	Z-14	Z-15	2883	2884	SG-32	20	2883	2904
A-1	R24-S1	1	AGV	Z-15	Z-14	3814	3815	SG-35	20	3814	3835
A-1	R24-S2	2	AGV	Z-14	Z-16	3835	3836	SG-33	20	3835	3856

A-1	R23-S1	1	CA-6	Z-16	Z-14	3856	3857	SG-36	20	3856	3877
A-1	R23-S2	2	CA-6	Z-14	Z-13	3877	3878	SG-31	50	3877	3928
A-1	R23-S3	3	CA-6	Z-13	Z-11	3928	3929	SG-29	20	3928	3949
A-1	R23-S4	4	CA-6	Z-11	Z-7	3949	3950	SG-23	30	3949	3980
A-1	R23-S5	5	CA-6	Z-7	Z-4	3980	3981	SG-15	30	3980	4011
A-1	R23-S6	6	CA-6	Z-4	Z-5	4011	4012	SG-10	20	4011	4032
A-1	R28-S1	1	AGV	Z-5	Z-4	4758	4759	SG-12	20	4758	4779
A-1	R28-S2	2	AGV	Z-4	Z-3	4779	4780	SG-9	20	4779	4800
A-1	R27-S1	1	CA-7	Z-3	Z-4	4800	4801	SG-7	20	4800	4821
A-1	R27-S2	2	CA-7	Z-4	Z-1	4821	4822	SG-8	50	4821	4872
A-1	R27-S3	3	CA-7	Z-1	Z-2	4872	4873	SG-3	20	4872	4893
A-2	R5-S1	1	AGV	Z-18	Z-17	484	485	SG-40	20	484	505
A-2	R5-S2	2	AGV	Z-17	Z-12	505	506	SG-37	20	505	526
A-2	R5-S3	3	AGV	Z-12	Z-11	526	527	SG-27	20	526	547
A-2	R5-S4	4	AGV	Z-11	Z-7	547	548	SG-23	30	547	578
A-2	R5-S5	5	AGV	Z-7	Z-4	578	579	SG-15	30	578	609
A-2	R5-S6	6	AGV	Z-4	Z-3	609	610	SG-9	20	609	630
A-2	R4-S1	1	CA-17	Z-3	Z-4	630	631	SG-7	20	630	651
A-2	R4-S2	2	CA-17	Z-4	Z-1	651	652	SG-8	50	651	702
A-2	R4-S3	3	CA-17	Z-1	Z-2	702	703	SG-3	20	702	723
A-2	R13-S1	1	AGV	Z-2	Z-1	1373	1374	SG-6	20	1373	1394
A-2	R13-S2	2	AGV	Z-1	Z-6	1394	1395	SG-5	30	1394	1425
A-2	R12-S1	1	CA-9	Z-6	Z-1	1425	1426	SG-13	30	1425	1456
A-2	R12-S2	2	CA-9	Z-1	Z-4	1456	1457	SG-4	50	1456	1507
A-2	R12-S3	3	CA-9	Z-4	Z-7	1507	1508	SG-11	30	1507	1538
A-2	R20-S1	1	CA-12	Z-7	Z-4	2743	2744	SG-15	30	2743	2774
A-2	R20-S2	2	CA-12	Z-4	Z-1	2774	2775	SG-8	50	2774	2825
A-2	R20-S3	3	CA-12	Z-1	Z-6	2825	2826	SG-5	30	2825	2856
A-3	R22-S1	1	AGV	Z-18	Z-17	3061	3062	SG-40	20	3061	3082
A-3	R22-S2	2	AGV	Z-17	Z-12	3082	3083	SG-37	20	3082	3103
A-3	R22-S3	3	AGV	Z-12	Z-11	3103	3104	SG-27	20	3103	3124
A-3	R22-S4	4	AGV	Z-11	Z-7	3124	3125	SG-23	30	3124	3155
A-3	R21-S1	1	CA-13	Z-7	Z-11	3155	3156	SG-16	30	3155	3186
A-3	R21-S2	2	CA-13	Z-11	Z-13	3186	3187	SG-26	20	3186	3207
A-3	R21-S3	3	CA-13	Z-13	Z-14	3207	3208	SG-30	50	3207	3258
A-3	R21-S4	4	CA-13	Z-14	Z-15	3258	3259	SG-32	20	3258	3279
A-3	R26-S1	1	AGV	Z-15	Z-14	4590	4591	SG-35	20	4590	4611
A-3	R26-S2	2	AGV	Z-14	Z-16	4611	4612	SG-33	20	4611	4632
A-3	R25-S1	1	CA-14	Z-16	Z-14	4632	4633	SG-36	20	4632	4653
A-3	R25-S2	2	CA-14	Z-14	Z-13	4653	4654	SG-31	50	4653	4704
A-3	R25-S3	3	CA-14	Z-13	Z-11	4704	4705	SG-29	20	4704	4725
A-3	R25-S4	4	CA-14	Z-11	Z-9	4725	4726	SG-24	50	4725	4776
A-3	R25-S5	5	CA-14	Z-9	Z-8	4776	4777	SG-19	20	4776	4797
A-3	R30-S1	1	AGV	Z-8	Z-9	5164	5165	SG-17	20	5164	5185
A-3	R30-S2	2	AGV	Z-9	Z-10	5185	5186	SG-20	20	5185	5206
A-3	R29-S1	1	CA-15	Z-10	Z-9	5206	5207	SG-22	20	5206	5227
A-3	R29-S2	2	CA-15	Z-9	Z-6	5227	5228	SG-18	30	5227	5258

CAPÍTULO 6

CONCLUSÕES

Nesta tese abordou-se os problemas da programação do Sistema de Produção (SP) e a programação do Sistema de Transporte de Materiais (STM) em Sistemas Flexíveis de Manufatura (SFM).

Uma nova metodologia denominada Método Analítico de Estado (MAE), foi proposta para a resolução dos problemas acima mencionados. Através do conceito de entidade básica, o método decompõe um processo complexo de um sistema em um conjunto de processos bem mais simples e utiliza estados e transições de estados para modelar os processos. A resolução dos problemas é obtida através da definição das transições indiretas nos processos. Um modelo de Prioridade Linear Multiobjetivo (PLM) é utilizado para definir as transições indiretas e obter uma solução que satisfaz múltiplos critérios. Um algoritmo de busca é proposto para obter modelos alternativos que fornecem soluções de melhor qualidade em caso particular.

O método analítico de estado foi aplicado para modelar e resolver a programação do sistema de produção. O modelo construído seleciona dinamicamente as rotas de processamento das peças e programa as operações destas peças nos centros de trabalho. Foram construídos modelos de prioridade linear multiobjetivo que combinam diversos fatores associados a regras de despacho.

Os modelos obtidos são comparados com as regras de despacho através de um simulador desenvolvido neste trabalho. Os resultados obtidos pelo melhor modelo indicam que este fornece soluções melhores em termos do índice dos objetivos *makespan* e atraso total quando comparados com os resultados obtidos pelas regras isoladas. Os baixos tempos computacionais mostram a capacidade do método para resolver problemas práticos com grande dimensão. Um algoritmo de busca é proposto para obter modelos de prioridade alternativos com soluções de melhor qualidade em casos particulares.

A programação do sistema de transporte de materiais é analisada e comparada com a do sistema de produção. A partir da semelhança nos dois sistemas, a programação do STM é tratada de forma análoga à programação de SP. Um modelo de MAE é construído para selecionar os AGVs, escolher as rotas de viagem dos AGVs dinamicamente e programar em detalhe o transporte das cargas nos segmentos e nas zonas de controle. São construídos modelos de prioridade para o objetivo atraso total e comparado com várias regras de despacho através de um simulador. Os resultados mostram que as soluções do melhor modelo são superiores às soluções obtidas por regras individuais na maioria dos casos. O mesmo algoritmo de busca é utilizado para obter modelos de prioridade melhores em casos particulares.

Um simulador genérico dirigido por evento é desenvolvido com base no método analítico de estado para obter resultados computacionais. O simulador é composto por dois programadores: o programador de SP e o programador de STM. Através deste simulador, integram-se e executam-se as programações de produção e transporte simultaneamente para obter uma programação completa nos SFMs. No entanto, cada programador pode ser utilizado individualmente para fornecer um programa de somente um sistema particular. O simulador possui várias interfaces para facilitar o seu uso. A definição do *layout* do sistema e do modelo são separadas para facilitar a simulação de sistemas distintos. O simulador é amigável e possui características de descentralização, decisão e flexibilidade.

Direções de pesquisas futuras podem incluir:

- A programação de sistemas através de prioridade tem grande vantagem em termos de rapidez e simplicidade na resolução dos problemas. É especificamente útil para a programação *on-line* ou tempo real. Este trabalho propõe duas maneiras para construir modelos de prioridade multiobjetivo com combinação linear dos fatores associados a regras de despacho. A primeira consiste em construir o modelo através do método de análise hierárquica e a segunda através de um algoritmo de busca. Um tópico de pesquisa interessante seria investigar outros métodos de combinação dos fatores, incluindo métodos de combinação dinâmica, onde os pesos são definidos de acordo com a situação do sistema.
- Utilização de técnicas de Inteligência Artificial no método analítico de estado para a tomada de decisão nas transições indiretas.
- Um sistema flexível de manufatura é um sistema composto de vários elementos cuja programação global é muito complexa. Seria interessante considerar a programação de forma distribuída, isto é, tratar os elementos individualmente e desenvolver um mecanismo para coordenar as programações individuais através de comunicação interna. A estrutura de troca de informação através de requisições dos programadores utilizados no simulador desenvolvido nesta tese pode indicar uma possibilidade para a construção do simulador de programação distribuído. O método analítico de estado aqui proposto pode fornecer uma metodologia rápida e prática para cada programação individual e a coordenação global.

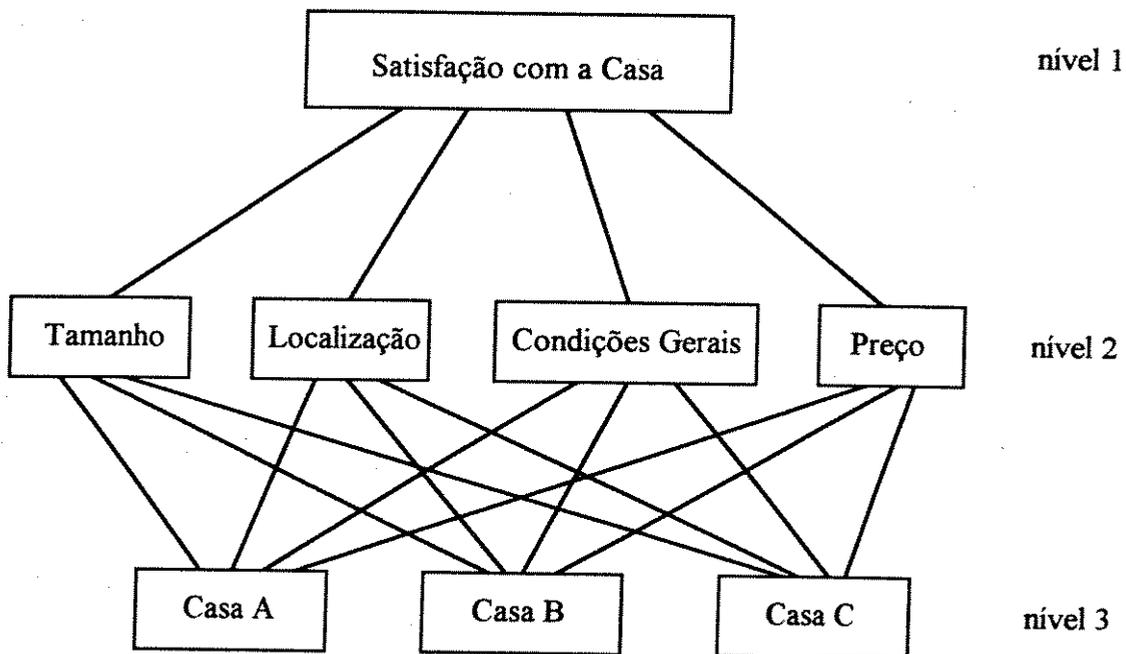
APÊNDICE

MÉTODO DA ANÁLISE HIERÁRQUICA

A.1 Introdução

O método da análise hierárquica é uma abordagem para tomada de decisão multicritério proposto por Saaty (1991). Os fatores que afetam a decisão são arranjados em uma estrutura hierárquica onde no nível mais alto está o objetivo global e em níveis inferiores encontram-se os critérios, subcritérios e as alternativas.

O exemplo a seguir ilustra essa estrutura hierárquica (Saaty 1990) para a decisão de compra de uma casa.



No nível 1 encontra-se a meta global “Satisfação com a Casa”. No nível 2 encontram-se quatro critérios que contribuem para a meta global e no nível 3 estão as três casas candidatas (alternativas) que devem ser avaliadas em termos dos critérios no segundo nível.

O método da análise hierárquica (MAH) faz dois tipos de medidas, absoluta e relativa, para obter prioridades (grau de importância) dos critérios com relação à meta global. A medida absoluta consiste da comparação entre pares de critérios no mesmo nível para determinar a importância relativa dos mesmos com relação aos critérios do nível superior. No exemplo, deve-se comparar cada par dos quatro critérios para obter a importância relativa dos mesmos com relação à meta global. Por exemplo, qual a prioridade do preço da casa em relação ao seu tamanho com respeito à meta global “Satisfação com a Casa”? A medida relativa consiste da comparação entre pares de elementos situados em um certo nível, incluindo o nível das alternativas, com relação a cada critério situado num nível adjacente superior.

De acordo com o MAH existem duas questões fundamentais para uma boa decisão multiobjetivo: consistência do conjunto de comparações entre pares e a definição de uma escala adequada que permita expressar quantitativamente a comparação entre pares.

A.2 Consistência

Considere um conjunto de n objetos A_1, A_2, \dots, A_n com pesos conhecidos w_1, w_2, \dots, w_n positivos. Seja a matriz

$$A = \begin{bmatrix} w_1 / w_1 & w_1 / w_2 & \dots & w_1 / w_n \\ w_2 / w_1 & w_2 / w_2 & \dots & w_2 / w_n \\ \dots & \dots & \dots & \dots \\ w_n / w_1 & w_n / w_2 & \dots & w_n / w_n \end{bmatrix}$$

Observações sobre a matriz A

- i) A linha i de A consiste das razões do peso do objeto i em relação aos pesos de todos os objetos.
- ii) Seja $A = \{a_{ij}\}$, $a_{ij} = w_i/w_j$, $i, j = 1, 2, \dots, n$. Note que $a_{ij} = 1/a_{ji}$, isto é, se o objeto i é, por exemplo, 4 vezes mais pesado que o objeto j , então, automaticamente, o objeto j tem $1/4$ do peso do objeto i . A matriz A é denominada recíproca.
- iii) Note que $a_{ik} = a_{ij} * a_{jk}$, $i, j, k = 1, 2, \dots, n$. Esta relação é chamada de **consistência** da matriz A e expressa uma propriedade de **transitividade**. Por exemplo, se o objeto i é 2 vezes mais pesado que o objeto j e j é 5 vezes mais pesado que o objeto k então i é 10 vezes mais pesado que k .
- iv) Note que cada linha i é obtida ao se multiplicar a primeira linha por w_i/w_1

Seja $w^T = (w_1, w_2, \dots, w_n)$. Segue-se então que

$$Aw = nw$$

isto é, w é um autovetor de A com autovalor associado igual a n . Da observação iv) acima, conclui-se que $\text{rank } A = 1$ e, portanto, todos os autovalores de A são nulos, com exceção de um autovalor. Como a soma dos autovalores é igual ao traço da matriz A , que por sua vez é igual a n , segue-se que n é o maior autovalor de A (único autovalor não nulo). Este autovalor é denominado autovalor principal. O autovetor normalizado $w / \sum_{i=1}^n w_i$, é chamado de autovetor principal.

Em um ambiente de tomada de decisão geral os valores exatos de w_i / w_j não são conhecidos com precisão. Tem-se apenas uma estimativa destes valores obtidos por especialistas que podem fazer erros de julgamento. Estes valores estimados podem conduzir a uma matriz A inconsistente. A questão que se põe é a seguinte: qual grau de inconsistência pode ser aceito? Esta questão conduz ao problema de autovalor

$$Aw = \lambda_{\max} w$$

onde λ_{\max} é o autovalor principal de A, w é o autovetor principal correspondente a λ_{\max} , e A é uma matriz recíproca mas não necessariamente consistente. O problema consiste em saber até que ponto o autovetor w reflete a opinião real do especialista. Note que se w é a solução do problema acima, então a matriz formada com elementos w_i / w_j é consistente e uma aproximação de A.

Para $i \neq j$, seja $a_{ij} = (1 + \delta_{ij}) w_i / w_j$, $\delta_{ij} > -1$ uma perturbação de w_i / w_j , onde w é o autovetor principal de A. Os seguintes teoremas são enunciados em Saaty (1990).

Teorema 1. $\lambda_{\max} \geq n$

Teorema 2. A é consistente se e somente se $\lambda_{\max} = n$

Destes resultados conclui-se que $\lambda_{\max} - n$ é uma medida da inconsistência de A. Como traço

$A = n = \sum_{i=1}^n \lambda_i$, onde λ_i , $i = 1, 2, \dots, n$, são os autovalores de A e $\lambda_1 = \lambda_{\max}$, segue-se que

$$\lambda_{\max} - n = - \sum_{i=2}^n \lambda_i.$$

O índice de consistência (IC) é definido como $(\lambda_{\max} - n) / (n - 1)$, é a média negativa dos n-1 autovalores de A e representa o desvio médio da consistência. Este valor é comparado com o mesmo índice \overline{IC} obtido como uma média sobre um número grande de matrizes recíprocas de mesma dimensão com elementos gerados aleatoriamente.

A razão de consistência (RC) é definida como IC/\overline{IC} . Se $RC \leq 0,1$, aceita-se a estimativa de w. Caso contrário, tenta-se melhorar a consistência. Este valor limiar 0,1 de consistência é fundamentado pela análise do trabalho de Vargas (1982).

A.3 Escala

Para comparar dois fatores e obter uma matriz de julgamento ou comparação de fatores, Saaty (1991) propôs a seguinte escala de medida mostrada na Tabela A.1.

Tabela A1 Escala Fundamental

Intensidade de importância em escala absoluta	Definição	Explicação
1	Mesma importância	Duas atividades contribuem igualmente para o objetivo
3	Importância moderada de uma sobre a outra	A experiência e o julgamento favorecem levemente uma atividade sobre a outra
5	Importância grande ou essencial	A experiência e o julgamento favorecem fortemente uma atividade sobre a outra
7	Importância muito grande	Uma atividade é fortemente favorecida em relação à outra e sua dominância é demonstrada na prática
9	Importância absoluta	A evidência favorece uma atividade em relação a outra com o mais alto grau de certeza
2, 4, 6, 8	Valores intermediários entre dois julgamentos adjacentes	Quando se procura um compromisso
Recíprocos	Se um dos números acima é atribuído a uma atividade i quando comparada com a atividade j, o valor recíproco (inverso) é atribuído à atividade j quando comparada com i	

A escala proposta por Saaty é baseada em trabalhos sobre variação de estímulos e respostas e também em trabalhos de psicologia onde se detetou que a mente humana pode lidar simultaneamente com poucos fatos (sete mais ou menos dois).

A.4 Exemplo

Vamos aplicar a metodologia proposta por Saaty ao problema de decisão de compra de uma casa descrito na seção A.1. Para facilitar a apresentação, os critérios são numerados como:

- 1 - Tamanho
- 2 - Localização
- 3 - Condições gerais
- 4 - Preço

A matriz de comparação dos critérios (dada pelos compradores da casa) é mostrada na Tabela A2. O vetor de prioridades ou pesos é o autovetor principal da matriz e representa a prioridade relativa dos critérios. Neste caso, o preço tem a prioridade mais alta com influência de 49,5 %.

Tabela A2 Matriz de comparação para o nível 1

	1	2	3	4	Vetor de prioridades w
1	1	5	1/3	1/4	$w_1 = 0,152$
2	1/5	1	1/5	1/7	$w_2 = 0,051$
3	3	5	1	1/2	$w_3 = 0,302$
4	4	7	2	1	$w_4 = 0,495$

$$\lambda_{\max} = 4.1659, IC = 0.0553, RC = 0.0615$$

A Tabela A3 mostra as comparações entre pares de elementos (casas) no nível 3 com relação a cada critério no nível 2. Existem quatro matrizes de comparação de dimensão 3×3 , pois tem-se quatro critérios e três casas. O autovetor principal de cada matriz representa o vetor de prioridades ou pesos das casas em relação a cada critério. Por exemplo, a casa A é a maior e a

mais cara dentre as três casas. Sua localização é inferior à casa C (superior à casa B) e suas condições gerais são piores quando comparadas com as casas B e C.

Tabela A3: Matrizes de comparação para o nível 2

1	A	B	C	Vetor de prioridades w^1
A	1	6	8	$w_A^1 = 0.754$
B	1/6	1	4	$w_B^1 = 0.181$
C	1/4	1/4	1	$w_C^1 = 0.065$

$$\lambda_{\max} = 3.136, IC = 0.068, RC = 0.117$$

2	A	B	C	Vetor de prioridades w^2
A	1	7	1/5	$w_A^2 = 0.233$
B	1/7	1	1/8	$w_B^2 = 0.055$
C	1/5	1/8	1	$w_C^2 = 0.713$

$$\lambda_{\max} = 3.247, IC = 0.124, RC = 0.213$$

3	A	B	C	Vetor de prioridades w^3
A	1	1/2	1/2	$w_A^3 = 0.200$
B	2	1	1	$w_B^3 = 0.400$
C	1/2	1	1	$w_C^3 = 0.400$

$$\lambda_{\max} = 3.000, IC = 0.000, RC = 0.000$$

4	A	B	C	Vetor de prioridades w^3
A	1	1/7	1/5	$w_A^3 = 0.072$
B	7	1	3	$w_B^3 = 0.650$
C	5	1/3	1	$w_C^3 = 0.278$

$\lambda_{\max} = 3.065, IC = 0.032, RC = 0.056$

A partir das prioridades nas tabelas A2 e A3 pode-se calcular o vetor de prioridades global w_G das casas. A prioridade global de casa A, w_{GA} , é calculada a partir da sua contribuição a cada critério e da influência de cada critério no objetivo global. Assim,

$$\begin{aligned}
 w_{GA} &= w_A^1 w_1 + w_A^2 w_2 + w_A^3 w_3 + w_A^4 w_4 \\
 &= (0,754)(0,152) + (0,233)(0,051) + (0,200)(0,300) + (0,072)(0,495) \\
 &= 0,223
 \end{aligned}$$

As prioridades globais das casas w_{GB} e w_{GC} são calculadas de forma análoga e resultam em $w_{GB} = 0,473$ e $w_{GC} = 0,305$. Isto implica que a casa B é a de maior prioridade e deve ser a escolhida.

REFERÊNCIAS BIBLIOGRÁFICAS

- Aartz, E. H. L.; P. J. M. van Laarhoven, J. K. Lenstra, N. L. J. Vlied, 1994; "A computational study of local search algorithms for job shop scheduling". *ORSA Journal on Computing*, Vol. 6, pp 118-125.
- Apple, J. M.; 1972; *Material Handling Systems Design*. Ronald.
- Askin, R. G. And C. R. Standridge; 1993; *Modeling and Analysis of Manufacturing Systems*. John Wiley.
- Baker, K. R.; 1984; "Sequencing rules and due-date assignments in a job shop". *Management Science*, Vol. 30; pp 1093-1104.
- Baker, K. R.; 1974; *Introduction to Sequencing and Scheduling*. John Wiley.
- Bartholdi, J. J. , III and L. K. Platzman; 1989; "Decentralized control of automated guided vehicles on a simple loop". *IIE Transactions*, Vol. 21, pp 76-81.
- Bengü, G; 1994; "A simulation-based scheduler for flexible flowlines". *International Journal of Production Research*, Vol. 32, pp 321-344.
- Blazewicz, J.; K. Ecker; G. Schmidt and J. Weglarz; 1993; *Scheduling in Computer and Manufacturing Systems*. Springer-Verlag.
- Bose, P. P. ; 1988; "Simulation models factory floors". *American Machinist*, Vol. 132, pp 49-53.

Brandimarte, P.; 1993; "Routing and scheduling in a flexible job shop by tabu search". *Annals of Operations Research*, Vol. 41, pp 157-183.

Browne, J., D. Dubois, K. Rathmill, S. P. Sethi and K. E. Stecke; 1984; "Classification of flexible manufacturing system". *The FMS Magazine*, April, pp 114-117.

Buzacott, J. A.; 1982; "The fundamental principles of flexibility in manufacturing systems". *Proceedings of the 1st International Conference on flexible manufacturing systems*. Brighton, UK, 20-22 October, 1982, pp 13-22.

Carrie, A.; 1988; *Simulation of Manufacturing Systems*. John Wiley & Sons.

Chang, Y.-L.; H. Matsuo and R. S. Sullivan; 1989; "A bottleneck-based beam search for job scheduling in a flexible manufacturing system". *International Journal of Production Research*, Vol. 27, pp 1949-1961.

Conway, R. W.; W.L. Maxwell and L.W. Miller; 1967; *Theory of Scheduling*. Addison-Wesley Publishing Company.

Cook, S. A.; 1971; "The complexity of theorem - proving procedures". *3rd ACM Symposium on Theory of Computing*, pp 151 - 158.

Dell'Amico, M. And M. Trubian; 1993; "Applying search to the job shop scheduling problem". *Annals of Operations Research*, Vol. 41, pp 231-252.

Desrosiers, J.; F. Soumis; M. Desrochers and M. Sauvé; 1986; "Methods for routing with time windows". *European Journal of Operational Research*, Vol. 23, pp 236-245.

- Dubois, D and K. E. Stecke; 1983; "Using Petri net techniques for comprehensive modeling, analysis and simulation of flexible manufacturing systems". *Proceeding of 22nd conference on Decision and Control*, December, 1983; pp 1062-1067.
- Egbelu, P. J. and J. M. A. Tanchoco; 1986; "Potentials for bi-directional guide-path for automated guided vehicle based systems". *International Journal of Production Research*, Vol. 24, pp 1075-1097.
- Egbelu, P. J. and J. M. A. Tanchoco; 1984; "Characterization of automatic guided vehicle dispatching rules". *International Journal of Production Research*, Vol. 22, pp 359-374.
- Ferland, T. A. and L. Fortin; 1989; "Vehicles scheduling with sliding time windows". *European Journal of Operational Research*, Vol. 38, pp 213-226.
- French, S.; 1982; *Sequencing and Scheduling: An introduction to the Mathematics of the Job Shop*. John Wiley.
- Garey, M. R. and D. S. Johnson; 1979; *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.
- Goicoechea, A; D. R. Hansen e L. Duckstein; 1982; *Multiobjetivo Decision Analysis with Engineering and Business Applications*. John Wiley & Sons.
- Greene, T. J. and R. P. Sadowski; 1986; "A mixed integer program for loading and scheduling multiple flexible manufacturing cells". *European Journal of Operational Research*, Vol. 24, pp 379-386.
- Greenwood, N. R.; 1988; *Implementing Flexible Manufacturing Systems*. Joh Wiley & Son.

- Groover, M. P.; 1987; *Automation, Production Systems, and Computer-Integrated Manufacturing*. Printice-Hall, New Jersey.
- Gupta, Y. P. and Goyal, S.; 1989; "Flexibility of manufacturing systems: concepts and measurements". *European Journal of Operational Research*, Vol. 43, pp 119-135.
- Han, M.-H. and L. F. McGinnis; 1989; "Control of material handling transporter in automated manufacturing". *IIE Transactions*, Vol. 21, pp 184-190.
- Hartley, J.; 1984; *FMS at work*. IFS Publications Ltd.
- Hutchison, J.; K. Leong; D. Snyder and P. Ward; 1991; "Scheduling approaches for random job shop flexible manufacturing systems". *International Journal of Production Research*, Vol. 29, pp 1053-1067.
- Jackson, J. R.; 1955; "Scheduling a production line to minimum tardiness". Research Report 43, Management Sciences Research Project, UCLA, January 1955 (ou ver Conway et al. 1967).
- Johnson, S. M.; 1954; "Optimal two and three-stage production schedules with set-up times included". *Naval Research Logistics Quarterly*, Vol. 1, pp 61-68.
- Kalkunte, M. V., S. C. Sarin, and W. E. Wilhelm; 1986; "Flexible manufacturing systems: A review of modeling approaches for design, justification and operation". *Flexible Manufacturing Systems: Method and studies*, Andrew Kusiak (ed.), Elsevier Science Publishers B. V. (North-Holland).
- Karp, R. M.; 1972; "Reducibility among combinatorial problems". em *Complexity of Computer Computation*, R. E. Miller and J. W. Thatcher (eds.), Plenum Press, pp 85-104.

Kim, C. W. and J. M. A. Tanchoco; 1993; "Operational control of a bidirectional automated vehicle system". *International Journal of Production Research*, Vol. 31, pp 2123-2138.

Klahorst, T.; 1981; "FMS". *Commline*, September/October, 1981. Reprinted in *Flexible Manufacturing Systems*. J. R. Holland (ed.), Society of Manufacturing Engineer, 1984.

Kusiak, A.; 1990; *Intelligent Manufacturing Systems*. Prentice Hall.

Kusiak, A.; 1986; "Application of operation research models and techniques in flexible manufacturing systems". *European Journal of Operational Research*, Vol. 24, pp 336-345.

Kusiak, A.; 1985; "Flexible manufacturing systems:A structural approach". *International Journal of Production Research*, Vol. 23, pp 1057-1073.

Kusiak, A. and M. Y. Chen; 1988; "Expert systems for planning and scheduling manufacturing systems". *European Journal of Operational Research*, Vol. 34, pp 113-130.

Law, A. M. and S. W. Haider; 1989; "Selection simulation software for manufacturing applications: practical guidelines & software survey". *Industrial Engineering*, Vol. 31, No. 5.

Lawler, E. L.; J. K. Lenstra; A. H. G. Rinnooy Kan and D. B. Shmoys; 1993; "Sequencing and scheduling: Algorithm and complexity". em *Logistics of Production and Inventory*, Graves, S. C.; A. H. G. Rinnooy Kan and P. H. Zipkin (eds.), North-Holland. (Handbook in Operations Research and Management Science, Vol. 4).

Lee, Doo Yong and Frank DiCesare; 1994; "Scheduling flexible manufacturing systems using Petri nets and heuristic search". *IEEE Transactions on Robotics and Automation*, Vol. 10, April, pp 123-132.

Lenz, J. E.; 1989; *Flexible Manufacturing*. Marcel Dekker, Inc.

Luggen, W. W.; 1991; *Flexible Manufacturing Cells and Systemas*. Prentice Hall.

Mahadevan, B. and T. T. Narendran; 1990; "Design of an Automated guided vehicle-based material handling system for a flexible manufacturing system". *International Journal of Production Research*, Vol. 28, pp 1611-1622.

Maleki, R. A.; 1991; *Flexible Manufacturing Systems - The Technology and Management*. Prentice Hall.

Malmborg, C. T.; 1990; "A model for the design of zone control automated guided vehicle systems". *International Journal of Production Research*, Vol. 28, pp 1741-1758.

Maxwell, W. I. and J. A. Muckstadt; 1982; "Design of automatic guided vehicle systems". *IIE Transactions*, Vol. 14, pp 114-124.

Merchant, E. M.; 1983; "Production: a dynamic challenge". *IEEE Spectrum*, Vol. 25, No. 5, 1983, pp36-39. Reprinted in *Models and Control of Automated Manufacturing Systems*. A. A. Desrochers (Ed.), 1990.

Montazeri, M. and L. N. Van Wassenhove; 1990; "Analysis of scheduling rules for an FMS". *International Journal of Production Research*, Vol. 28, pp 785-802.

Morton, T. E. And D. W. Pentico; 1993; *Heuristic scheduling systems*. Jonh Wiley.

Nasr, N. and E. A. Elsayed; 1990; "Job shop scheduling with alternative machines". *International Journal of Production Research*, Vol. 28, pp 1595-1609.

O'Gorman, P.; J. Gibbons and J. Browne; 1986; "Evaluation of scheduling systems for a flexible transfer line using a simulation model". em *Flexible Manufacturing Systems: Methods and Studies*; A. Kusiak (ed.), Elsevier Science Publishers B. V. (North-Holland).

Ozden, M.; 1988; "A simulation study of multiple-load-carrying automated guided vehicles in a flexible manufacturing system". *International Journal of Production Research*, Vol. 26, pp 1353-1366.

Panwalkar, S. S. and W. Iskander; 1977; "A survey of scheduling rules". *Operations Research*, Vol. 25, pp 45-61.

Peterson, J. L.; 1981; *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, Inc.

Pinedo, M.; 1995; *Scheduling: Theory, Algorithm and Systems*. Prentice Hall.

Raman, N.; F. B. Talbot and R. V. Rachamadgu; 1986; "Simultaneous scheduling of machines and material handling system devices in automated manufacturing". *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems*, K. E. Stecke and R. Suri (eds.), Elsevier Science Publishers B. V. (North-Holland), pp. 455-466.

Ráanky, P.; 1983; *The Design and Operation of FMS*. IFS Publication Ltd.

Saaty, T. L.; 1991; *Método de Análise Hierárquica*, tradução do original *The Analytic Hierarchy process*. Makron do Brasil.

Saaty, T. L.; 1990; "How to make a decision: the analytic hierarchy process". *European Journal of Operational Research*, Vol. 48, pp 9-26.

Saaty, T. L.; 1986; "Axiomatic foundation of the analytic hierarchy process". *Management Science*, Vol. 32, pp 841-855.

Sabuncuoglu, I. and D. L. Hommertzheim; 1992a; "Dynamic dispatching algorithm for scheduling machine and automated guided vehicles in a flexible manufacturing system". *International Journal of Production Research*, Vol. 30, pp 1059-1079.

Sabuncuoglu, I. and D. L. Hommertzheim; 1992b; "Experimental investigation of FMS machine and AGV scheduling rules against the mean flow-time criterion". *International Journal of Production Research*, Vol. 30, pp 1617-1635.

Sabuncuoglu, I. and D. L. Hommertzheim; 1989; "An investigation of machine and AGV scheduling rules in an FMS". *Proceedings of the third ORSA/TIMS conference on Flexible Manufacturing Systems: Operations Research Models and Applications*. K. E. Stecke and R. Suri (eds.), Elsevier Science Publications B. V. Amsterdam.

Shaw, M. J.; 1988; "Knowledge-based scheduling in flexible manufacturing sistemas: An integration of pattern-directed inference and heuristic search". *International Journal of Production Research*, Vol. 26, pp 821-844.

Shaw, M. J. and A. B. Whinston; 1986; "Applications of artificial intelligence to planning and scheduling in flexible manufacturing". em *Flexible Manufacturing Systems: Methods and Studies*; A. Kusiak (ed.), Elsevier Science Publishers B. V. (North-Holland).

Smith, W. E.; 1956; "Various optimizers for single-state production". *Naval Research Logistics Quarterly*, Vol. 3, march, 1956 (ou ver Conway et al. 1967).

Stecke, K. E.; 1985; "Design, planning, scheduling and control problems of flexible manufacturing systems". *Annals of Operations Research*, Vol. 3, pp 3-12.

Stecke, K. E.; 1983; "Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems". *Management Science*, Vol. 29, pp273-288.

Stecke, K. E. and J. Browne; 1985; "Variations in Flexible manufacturing systems according to the relevant types of automated materials handling". *Material Flows*, Vol. 2 pp 179-185.

Stecke, K. E. and J. J. Solberg; 1981; "Loading and control policies for a flexible manufacturing system". *International Journal of Production Research* , Vol. 19, pp 481-490.

Subramanyam, S. and R. G. Askin; 1986; "An expert systems approach to scheduling in Flexible manufacturing systems". em *Flexible Manufacturing Systems: Methods and Studies*; A. Kusiak (ed.), Elsevier Science Publishers B. V. (North-Holland).

Ulusoy, G. and Ü. Bilge; 1993; "Simultaneous scheduling of machines and automated guided vehicles". *International Journal of Production Research*, Vol. 31, pp. 2857-2873.

Valavanis, K. P.; 1990; "On the hierarchical modeling analysis and simulation of flexible manufacturing systems with extended Petri Nets". *IEEE trans. on Systems, Man and Cybernetics*, Vol. 20, January/February, pp 94-110.

Van Laarhover, P.J.M.; E.H.L.Aartz, J.K.Lenstra; 1992; "Job shop scheduling by simulated annealing". *Operations Research*, Vol. 40, pp 113-126.

Vargas, L. G.; 1990; "An overview of the analytic hierarchy process and its applications". *European Journal of Operational Research*, Vol. 48, pp 2-8.

Vargas, L. G.; 1982; "Reciprocal matrices with random coefficients". *Mathematical Modelling*, Vol. 3, pp 69-81.

Wu, S. D and R. A. Wysk; 1990; "An inference structure for the control and scheduling of manufacturing systems". *Computer and Industrial Engineering*, Vol. 18, pp 247-262.

Wu, S. D and R. A. Wysk; 1989; "An application of discrete-event simulation to on-line control and scheduling in flexible manufacturing". *International Journal of Production Research*, Vol. 27, pp 1603-1623.

Wu, X.-B.; 1987; "Multiobjective job shop scheduling with finite capacity". em *Modern Production Management System*, A. Kusiak (ed.), Elsevier Science Publishers B. V. (North-Holland), pp 147-157.

Wu, X.-B. and V. A. Armentano; 1994; "Analytic process of state and its application in FMS scheduling". em *Production Management Methods*, C. Walter, F. J. Kliemann and J. P. M. de Oliveira (eds), Elsevier Science Publishers B. V. (North-Holland), pp 271-278.

Wu, X.-B. and V. A. Armentano; 1992a; "Distributive scheduling of flexible manufacturing systems". *7th IFAC/IFIP/IFORS Symposium on Information Control Problems in Manufacturing Technology*, Toronto, Canada, May 1992.

Wu, X.-B. and V. A. Armentano; 1992b; "Scheduling and control of AGV based material handling systems". em *Human Aspects in Computer Integrated Manufacturing*, G.J.Olling and F.Kimura (eds.), Elsevier Science Publishers B. V. (North-Holland), pp 729-736.

Wu, X.-B. and Jiang, M.-W.; 1987; "Job shop scheduling: theory and practice". *Annual Conference of Chinese Operations Research Society*, Hangzhou, China.