

UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA ELÉTRICA  
SETEMBRO DE 1993

Este exemplar é a edição final da tese  
defendida por Evelin Maria Abreu Teixeira  
Teixeira aprovada pela Comissão  
Julgadora em 29 09 93  
Orientador Jamaguchi

PROGRAMAÇÃO HORÁRIA DE PEÇAS EM UMA  
CÉLULA FLEXÍVEL DE MANUFATURA

Por : Evelin Maria Abreu Teixeira *11/02/93*

Orientador : Prof. Dr. Akebo Yamakami *11*

Tese apresentada à Faculdade de Engenharia  
Elétrica, da Universidade Estadual de  
Campinas, como requisito parcial para a  
obtenção do título de Doutor em Engenharia  
Elétrica.

*Aos meus pais, Athos (in memoriam) e Eva.*

*Aos meus filhos, Octávio e Andréa.*

*Aos meus irmãos, Ericina e Paulo.*

## A G R A D E C I M E N T O S

Ao Professor Akebo Yamakami pela orientação e constante estímulo.

À Fundação Centro Tecnológico para Informática pelo apoio institucional.

Ao Professor Plínio Stange da Universidade Federal de Santa Catarina pelas sugestões e incentivo.

Ao Professor Evandro de Oliveira Araújo da Universidade Federal de Minas Gerais pelas contribuições na fase inicial deste trabalho.

Ao Eng<sup>o</sup> Ronaldo Cardoso Lemos, Diretor do Instituto de Automação da FCTI, por permitir o desenvolvimento deste trabalho em paralelo às outras atividades e projetos do Instituto.

Ao Eng<sup>o</sup> Carlos Alberto dos Santos Passos, Chefe do Departamento de Automação da Manufatura do Instituto de Automação da FCTI, pelo estímulo durante a finalização deste trabalho.

A Clarinda Rodrigues Lucas e Cleonice Maruci Correa Teracine pelo valioso auxílio no levantamento bibliográfico.

Ao João Batista Cury Gutierrez e à Vera Lídia Vedovello Machado pelo imprescindível auxílio na utilização dos computadores VAX 11/785 e VAX 8530.

Aos meus colegas da FCTI pelo convívio agradável e, em particular, aos Eng<sup>os</sup> Kiyoshi Asanuma, Március Fábius de Carvalho e Oscar Salviano Silva Filho pelas proveitosas discussões durante o desenvolvimento deste trabalho e ao Eng<sup>o</sup> Carlos Alberto de Oliveira Fernandes pelo constante auxílio em questões de hardware e software.

Ao Mitur Sakoda da EMBRAPA pelo desenvolvimento da interface gráfica.

Aos Eng<sup>os</sup> Antonio de Freitas Rentes e Eng<sup>o</sup> Morio Hirama pelas discussões iniciais.

Ao Roberto de Oliveira pela qualidade dos desenhos.

A Célia Maria Dorázio pelo eficiente trabalho de edição e, especialmente, à minha família pela compreensão, paciência e incentivo.

## G L O S S Á R I O

- AGV - Veículos guiados automaticamente
- BUFFER - Local para armazenamento das peças
- CAD - Projeto assistido por computador
- CAE - Engenharia assistida por computador
- CAM - Manufatura assistida por computador
- CAPP - Planejamento de processo assistido por computador
- CAR - Reparo assistido por computador
- CAT - Teste assistido por computador
- CIM - Manufatura integrada por computador
- CLOSED-SHOP - Produção para reposição do estoque
- CNC - Controle numérico computadorizado
- CRP - Planejamento das necessidades de capacidade
- DNC - Controle numérico direto
- EDD - Data devida mais cedo
- FCFS - Primeiro que chega é o primeiro a ser servido
- FIFO - Primeiro que entra é o primeiro que sai
- FLOW-SHOP - Produção de peças em uma mesma ordem nas máquinas
- FMC - Célula flexível de manufatura
- FMM - Módulo flexível de manufatura
- FMS - Sistema flexível de manufatura
- GRÁFICO DE GANTT - Gráfico com os horários de produção
- JIT - Produção somente quanto e quando necessário
- JMIN - Índice mínimo de operação na máquina gargalo
- JOB - Módulo de tarefas básicas
- JOB-SHOP - Produção de peças em uma ordem qualquer nas máquinas
- LAYOUT - Projeto do espaço físico
- LCFS - Último que chega é o primeiro a ser servido
- LPT - Tempo máximo de processamento
- LWKR - Operação associada ao "job" com menor quantidade de trabalho restante a ser processado
- MAKESPAN - Tempo total para processamento das peças
- MAGAZINE - Local para armazenamento de ferramentas
- MIX - Conjunto de tipos de produtos
- MOP - "Job" com maior número de operações
- MOPR - Operação associada ao "job" com maior número de operações restantes a serem processadas

MOPR-SPT - Usa a regra MOPR e, em caso de empate, a SPT

MPS - Plano mestre de produção

MRP - Planejamento das necessidades de materiais

MRPII - Planejamento dos recursos de manufatura

MWKR - Operação associada ao "job" com maior quantidade de trabalho restante a ser processado

NC - Controle numérico

NO PREEMPTION - Sem interrupção

OPEN-SHOP - Produção para atendimento da demanda

OPT - Tecnologia de produção otimizada

PALLETS - Local para armazenamento de peças

PI - Programação inteira

PL - Programação linear

PLIM - Programação linear inteira mista

PM - Programação matemática

PO - Pesquisa operacional

SET-UP - Preparação da máquina

SHOP-FLOOR - Chão de fábrica

SOFTWARE - Programa de computador

SPT - Tempo mínimo de processamento

SWMQ1 - "Job" que está com sua última operação na máquina

SWMQ2 - "Job" cuja próxima operação é na máquina com menor carga a ser processada

TRADE-OFF - Negociação

WORK-IN-PROCESS (WIP) - Trabalho em processo

## LISTA DE FIGURAS

- FIGURA 1.1 - Um Modelo Global de CIM
- FIGURA 1.2 - O Ciclo de Manufatura no CIM
- FIGURA 1.3 - Níveis de Problemas em um FMS
- FIGURA 2.1 - Uma Célula Flexível de Manufatura
- FIGURA 4.1 - Diagrama de Blocos do "Branch and Bound" implementado
- FIGURA 5.1 - Diagrama de Blocos da Heurística
- FIGURA 5.2 - Intervalos Encontrados na Máquina e no Robô
- FIGURA 5.3 - Relações entre as Novas e Antigas Posições
- FIGURA 6.1 - Gráfico de Gantt Gerado pelo Modelo para 2 Peças em 2 Máquinas
- FIGURA 6.2 - Gráfico de Gantt Gerado pela Heurística para 2 Peças em 2 Máquinas
- FIGURA 6.3 - Gráfico de Gantt Gerado pelo Modelo para 3 Peças em 3 Máquinas
- FIGURA 6.4 - Gráfico de Gantt Gerado pela Heurística para 3 Peças em 3 Máquinas
- FIGURA 6.5 - Gráfico de Gantt Gerado com a Regra LPT para 8 Peças em 2 Máquinas
- FIGURA 6.6 - Gráfico de Gantt Gerado com a Regra SPT para 8 Peças em 2 Máquinas
- FIGURA 6.7 - Gráfico de Gantt Gerado com a Regra JMIN para 8 Peças em 2 Máquinas
- FIGURA 6.8 - Gráfico de Gantt Gerado com a Regra LPT para 10 Peças em 2 Máquinas
- FIGURA 6.9 - Gráfico de Gantt Gerado com a Regra SPT para 10 Peças em 2 Máquinas
- FIGURA 6.10 - Gráfico de Gantt Gerado com a Regra JMIN para 10 Peças em 2 Máquinas
- FIGURA 6.11 - Gráfico de Gantt Sincronizando só Máquinas para 8 Peças em 2 Máquinas
- FIGURA 6.12 - Gráfico de Gantt com 3 Peças em 3 Máquinas sem Máquina Alternativa
- FIGURA 6.13 - Gráfico de Gantt com 3 Peças em 3 Máquinas com Máquina Alternativa
- FIGURA 6.14 - Gráfico de Gantt com 3 Peças em Processamento, 1 Nova Peça e 3 Máquinas
- FIGURA 6.15 - Gráfico de Gantt com 5 Peças em 2 Máquinas
- FIGURA 6.16 - Gráfico de Gantt com 6 Peças em 2 Máquinas
- FIGURA 6.17 - Gráfico de Gantt com 7 Peças em 2 Máquinas
- FIGURA 6.18 - Gráfico de Gantt com 20 Peças em 2 Máquinas
- FIGURA 6.19 - Gráfico de Gantt com 30 Peças em 3 Máquinas

## LISTA DE TABELAS

TABELA 6.1 - Dados de Produção de 2 Peças em 2 Máquinas

TABELA 6.2 - Dados de Produção de 3 Peças em 3 Máquinas

TABELA 6.3 - Dados de Produção de 10 Peças em 2 Máquinas

TABELA 6.4 - Dados de Produção de 3 Peças em 3 Máquinas com Máquinas Alternativas

TABELA 6.5 - Dados de Produção da Nova Peça

TABELA 6.6 - Dados de Produção de 30 Peças em 2 Máquinas

TABELA 6.7 - Dados de Produção de 30 Peças em 3 Máquinas

TABELA 6.8 - Número de Peças x Tempo de CPU

## R E S U M O

O problema da programação horária de peças em uma Célula Flexível de Manufatura com um robô responsável pelo transporte das peças, desde a entrada na célula até cada uma das máquinas que a processarão e depois até a saída, pode ser modelado como um "Programa Linear Inteiro Misto" com o objetivo de minimização do tempo total de utilização da célula.

Para este modelo, supomos, inicialmente, que as peças a serem processadas têm sua rota tecnológica em uma mesma direção, embora com a possibilidade de alguma delas não passar em todas as máquinas da célula. Posteriormente, o modelo foi ampliado para tratar o caso mais geral de rota tecnológica multidirecional.

O problema sendo modelado desta forma, a solução ótima pode ser obtida por um algoritmo "Branch and Bound". Um "software" foi desenvolvido em FORTRAN 77 e implementado em um computador VAX 11/785. O modelo foi validado pelos resultados obtidos em vários exemplos.

Procedimentos de aceleração foram investigados e implementados. Paralelamente, foi implementado um procedimento heurístico para gerar a programação horária em tempo real, considerando a célula em funcionamento. Os resultados foram comparados com os do modelo anterior, validando a heurística.

A heurística aceita rotas tecnológicas alternativas e prioriza as peças de acordo com a regra selecionada entre três disponíveis.

Da mesma forma que o modelo, a heurística localiza o gargalo da célula para o conjunto de peças a ser processado. Entretanto, como a heurística trabalha com rotas tecnológicas alternativas, o gargalo pode ser alterado dinamicamente.



## A B S T R A C T

The problem of the scheduling of a set of workpieces in a Flexible Manufacturing Cell with a robot responsible for the transport of the workpieces from the input of the cell to each one of the machines which will process it and then to the output, can be modeled as a Mixed Integer Linear Program with the objective of minimizing the utilization total time of the cell. For this model we suppose initially that the workpieces to be processed have their technological route in a same direction although with the possibility that some of them do not necessarily pass through all the machines in the cell.

Furthermore the model has been extended to deal with the most general case of multidirecional technological route.

Modeling the problem in this way, the optimum solution was obtained by a Branch and Bound algorithm.

A software was developed in a FORTRAN 77 language and implemented in a VAX 11/785 computer. By the obtained results in many examples we could validate the model.

Acceleration procedures were investigated and implemented. Parallel to this a heuristic procedure was implemented to generate the real-time scheduling considering the cell in working condition. The results were compared with the model validating the heuristic.

The heuristic accepts alternative technological routes and prioritizes the workpieces according with the selected rule among three rules available.

As the model, the heuristic localizes the cell bottleneck for the set of workpieces that will be processed. However the heuristic works with alternative technological routes, the bottleneck can be changed dinamically.

# Í N D I C E

AGRADECIMENTOS

GLOSSÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

ABSTRACT

<b>CAPÍTULO 1 - INTRODUÇÃO</b> .....	1
1.1. Objetivos .....	2
1.2. Visão Atual da Manufatura .....	3
1.3. Problema da Programação Horária .....	15
1.4. Levantamento Bibliográfico em Programação Horária de FMC e FMS .....	22
<b>CAPÍTULO 2 -PROGRAMAÇÃO HORÁRIA DE PEÇAS EM UMA FMC E O CONCEITO DE GARGALO DINÂMICO</b> .....	35
2.1. Problema da Programação Horária de Peças em uma FMC .....	36
2.2. Conceito de Gargalo Dinâmico .....	37
2.3. Sincronização de Máquinas e Robô .....	38
2.4. Aproveitamento das Ociosidades Intermediárias Programadas .....	38
<b>CAPÍTULO 3 -MODELO PARA O PROBLEMA DE PROGRAMAÇÃO HORÁRIA DE PEÇAS EM UMA FMC</b> .....	40
3.1. Modelo com Fluxo Unidirecional .....	41
3.2. Modelo com Fluxo Multidirecional .....	44
3.3. Utilização do Conceito de Gargalo Dinâmico .....	46
3.4. Comparação com Outras Modelagens .....	47
<b>CAPÍTULO 4 - ALGORITMO PROPOSTO</b> .....	48
4.1. Programação Inteira e Programação Linear Inteira Mista .....	49
4.2. Características do "Branch and Bound" Implementado .....	52
4.3. Características das Versões Implementadas .....	54
4.4. Características da Versão Interativa .....	54
4.5. Acelerações Implementadas .....	55

<b>CAPÍTULO 5 - HEURÍSTICA PROPOSTA</b> .....	59
5.1. Descrição da Heurística .....	60
5.2. Descrição do Algoritmo .....	60
5.3. Comparação com Heurísticas Clássicas .....	65
<b>CAPÍTULO 6 - EXEMPLOS E RESULTADOS</b> .....	67
6.1. Comparação do Modelo Multidirecional com a Heurística .....	68
6.2. Comparação das Regras LPT, SPT e JMIN Utilizando a Heurística .....	74
6.3. Comparação da Heurística Sincronizando Máquinas e Robô com a Heurística Sincronizando só Máquinas .....	82
6.4. Comparação da Solução com Máquina Alternativa com a Solução sem Máquina Alternativa .....	84
6.5. Exemplo com a Célula em Funcionamento .....	87
6.6. Índices de Desempenho .....	89
<b>CAPÍTULO 7 - CONCLUSÃO</b> .....	98
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	102
<b>BIBLIOGRAFIA</b> .....	110

## **CAPÍTULO 1**

### **INTRODUÇÃO**

## 1.1. OBJETIVOS

Neste trabalho vamos estudar e solucionar o problema da Programação Horária de Peças em uma Célula Flexível de Manufatura (FMC).

Os principais pontos deste trabalho são:

- Apresentação de características do problema da Programação Horária de Peças em uma FMC.
- Introdução do conceito de gargalo dinâmico e sua relação com a sincronização de máquinas e robô e com o aproveitamento das ociosidades intermediárias.
- Modelagem do problema utilizando Programação Matemática e o conceito de gargalo dinâmico.
- Apresentação de uma heurística com as seguintes características:
  - .. utiliza o conceito do gargalo dinâmico;
  - .. sincroniza máquinas e robô;
  - .. aproveita as ociosidades intermediárias;
  - .. utiliza a flexibilidade de máquinas alternativas para reduzir o atual gargalo;
  - .. permite a escolha de regras, entre as quais estão as clássicas LPT e SPT, a serem adotadas para a seleção da próxima peça que terá seus horários programados;
  - .. introduz a regra JMIN, com base na análise do atual gargalo, versus a rota da peça.
- Apresentação de um sistema híbrido para ser utilizado em uma FMC com o modelo sendo ativado periodicamente e a heurística sendo ativada quando houver necessidade de rápida geração da programação horária ou de reprogramação.

O trabalho está organizado da seguinte forma:

Neste capítulo apresentamos uma visão atual da Manufatura através da Manufatura Integrada por Computador (CIM) e dos Sistemas Flexíveis de Manufatura (FMS), uma visão geral do problema da Programação Horária e um breve levantamento bibliográfico na área de Programação Horária de FMS e de Células Flexíveis de Manufatura (FMC).

No capítulo 2 descrevemos o problema da Programação Horária de Peças em uma FMC e apresentamos o novo conceito de gargalo dinâmico utilizado neste nosso trabalho.

No capítulo 3 modelamos o problema como um problema de Programação Linear Inteira Mista (PLIM) para obter a solução ótima.

No capítulo 4 descrevemos o algoritmo "Branch and Bound" utilizado. Este algoritmo é conhecido tanto na Programação Matemática (PM) como na Inteligência Artificial, pois o algoritmo  $A^*$  é um caso particular do mesmo (Nau, 1984). A variação adotada na implementação do algoritmo é detalhada. As acelerações algebricamente obtidas, que chegaram a fornecer uma redução de 50% no tempo de CPU, são apresentadas também neste capítulo.

Como a velocidade computacional ainda não avançou o suficiente para que um "software" de Programação Matemática rode em tempo real, desenvolvemos uma heurística que é apresentada no capítulo 5, seguida da descrição da implementação computacional.

Exemplos e resultados obtidos são apresentados no capítulo 6 e, finalmente, no capítulo 7 apresentamos nossas conclusões e sugestões para futuros trabalhos.

## 1.2. VISÃO ATUAL DA MANUFATURA

O atual cenário mundial da Manufatura caracteriza-se pela internacionalização da economia e pelas constantes mudanças no perfil da demanda, com conseqüente aumento da diversificação dos produtos e diminuição de seus ciclos de vida.

Para adequar-se a este cenário, as fábricas precisam ter um alto grau de flexibilidade e um tempo de resposta muito curto para serem capazes de produzir a variedade de produtos que o mercado requer. Esta capacidade é melhor descrita, na linguagem de economistas, como "Economia de Escopo" (Goldhar, 1990). Para um gerente de operações, economia de escopo significa que a variedade é efetivamente livre e o pedido econômico é unitário. Para um gerente de marketing, economia de escopo significa que não existe custo associado com qualquer projeto específico de produto.

As empresas de manufatura necessitam de equipamentos que permitam a absorção das alterações de produto sem perda de produtividade nem aumento de custos e com garantia de qualidade. Conseqüentemente, a tendência para a manufatura automatizada flexivelmente deverá continuar indefinidamente (Lima e Agostinho, 1989). Com toda certeza, essa tendência será fortalecida pelo desenvolvimento de equipamentos com bases multioperacionais, ou seja, as operações constantes dos roteiros deverão, cada vez mais, ser agrupadas em máquinas multioperacionais como os centros de usinagem. O objetivo deverá ser a usinagem completa de uma peça com uma única fixação. Os custos de produção são deslocados do pessoal para as máquinas-ferramentas, de forma que o planejamento de utilização das mesmas é muito importante.

Estas máquinas-ferramentas podem ser agrupadas em células e as células em siste-

mas e assim, sucessivamente, até chegar à fábrica do futuro totalmente automatizada. (Ortiz, 1990)

Desde a introdução do controle numérico (NC), no final dos anos 50, muito esforço tem sido feito para computadorizar as diferentes tarefas envolvidas no processo produtivo. Como resultado, temos o desenvolvimento de diferentes sistemas computacionais sob o nome de: Projeto Assistido por Computador (CAD), Planejamento de Processo Assistido por Computador (CAPP), Engenharia Assistida por Computador (CAE), Manufatura Assistida por Computador (CAM), Teste Assistido por Computador (CAT), Reparo Assistido por Computador (CAR), etc. (Crestin, 1990). Depois começaram os esforços para interfacear os diferentes sistemas, pois a troca manual dos dados resulta em perda de tempo e ocorrência de erros. Por exemplo, muitos sistemas CAD incluem um módulo de Controle Numérico Computadorizado (CNC) que calcula os dados diretamente dos dados geométricos resultantes do CAD, podendo também passar antes pelo CAPP. Estes sistemas são usualmente organizados em módulos, cada um dos quais desempenhando uma função específica.

A melhoria dos hardwares computacionais, dos métodos de controle de processo e da tecnologia de software, junto com uma demanda crescente de informações, têm tornado possível a integração de todas as operações de manufatura.

A Manufatura Integrada por Computador (CIM) é uma nova tecnologia adequada à Automação Flexível da Manufatura, assim como as Células Flexíveis de Manufatura (FMC), constituídas por máquinas controladas por computador e equipadas com dispositivos automáticos de troca de ferramenta e robôs, e os Sistemas Flexíveis de Manufatura (FMS), constituídos por um grupo de FMC's, um sistema de manuseio de materiais e um sistema de controle por computador.

A Figura 1.1 mostra um modelo global de CIM (Ranky, 1986) integrando todos processamentos de dados, desde a entrada no sistema até a saída de um produto.

Já na Figura 1.2, a ilustração destaca o ciclo de manufatura em um CIM (Newman, 1988), ressaltando a base de dados de Engenharia e Manufatura como integradora das principais funções de um sistema produtivo.

Os sistemas CIM começam com o projeto de interfaces entre programas dedicados a tarefas especializadas. A organização e a estruturação de dados são consideradas como uma das maiores questões na implantação do CIM. O crescimento do uso de supercomputadores, estações de trabalho e computadores pessoais implica em equipamentos não homogêneos. É claro, então, que a rede CIM será heterogênea e não há razão para esta tendência mudar (Crestin, 1990).

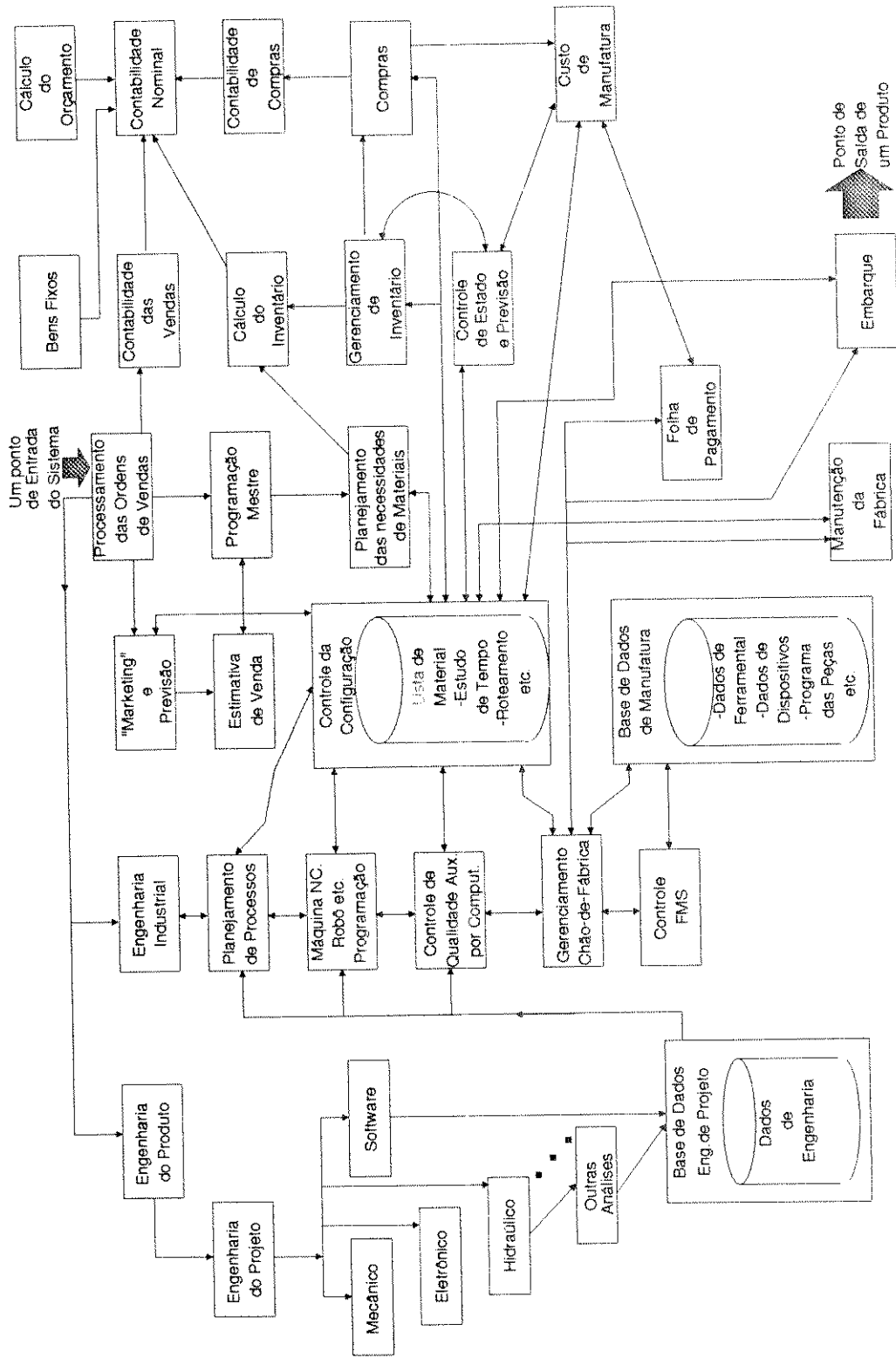


FIGURA 1.1 - Um modelo global de CIM



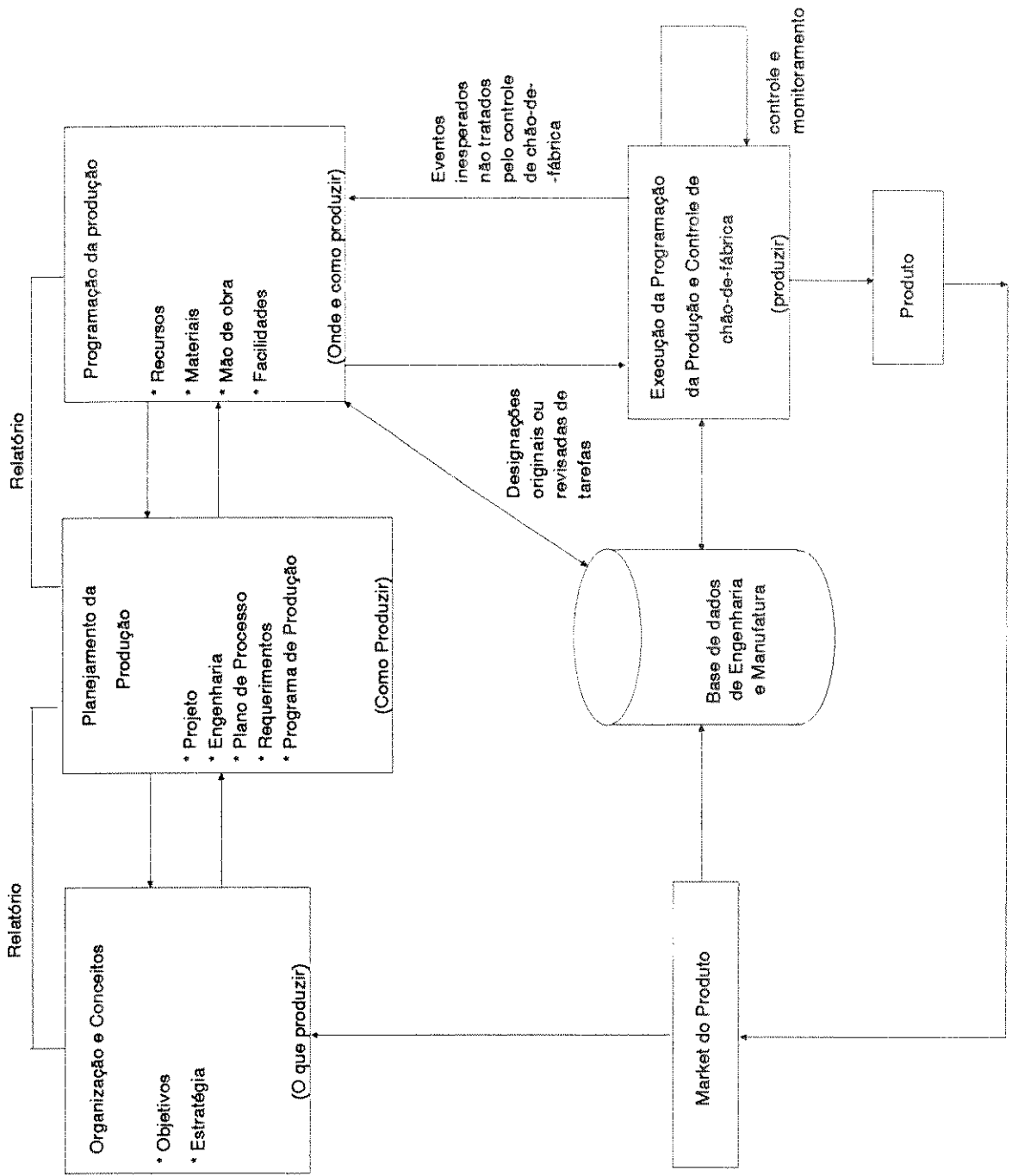


FIGURA 1.2 - O ciclo de manufatura no CIM

A solução para a integração aponta para a adoção de padrões internacionais. Em 1978 a International Organization for Standardization (ISO) apresentou o Modelo de Referência Open Systems Interconnection (OSI) e, em 1984, a GM americana apresentou o Manufacturing Automation Protocol (MAP) para forçar a adoção de um padrão de comunicação.

Os sistemas CIM multiplicam os potenciais de controle e variabilidade em cada estágio do projeto até o final, maximizando a aquisição, o uso e reuso da informação para obter economia de escopo. Nas fábricas tradicionais, uma informação é frequentemente gerada novamente ou colocada novamente, com chances potencialmente catastróficas de erro ou perda. As fábricas CIM simplesmente acessam e reutilizam a informação depois de sua criação, melhorando a precisão e consistência das decisões, compromissos e atividades futuras. Investimentos em CAD, CAM, FMS e outras tecnologias avançadas fornecerão retornos aceitáveis se a flexibilidade estender-se à engenharia, distribuição e marketing. O próximo passo será na direção da "Automação da Customização", ou seja, a manufatura será um negócio de serviços. A abordagem do século 21 da "Manufatura como um negócio de Serviços" usará o CIM como base para manter superioridade sobre o concorrente (Goldhar, 1990).

Os benefícios da tecnologia CIM são muitos. Suh (1990) apresentou os seguintes:

**- Flexibilidade**

O gerenciamento de informação e manufatura, baseada em computador, permitirá que indústrias atendam à demanda do mercado rápida e eficientemente e aumentará a produtividade de pequenos lotes de peças.

**- Resposta rápida às mudanças da demanda do mercado**

O tempo de desenvolvimento do produto é reduzido usando CIM. Além disso, como fábricas "virtuais" podem ser projetadas e simuladas, usando computadores (sem a necessidade de construir protótipos de fábricas e linhas de produção), novos produtos podem ser projetados, testados e produzidos num tempo muito curto.

**- Decisões mais rápidas e custos gerais inferiores**

Uma vez que os computadores podem localizar e integrar as informações, decisões podem ser tomadas rapidamente. Isto ajudará a eliminar a necessidade de muitos gerentes de nível médio, diminuindo assim a taxa de custos gerais, bem como reduzir o tempo para introdução de novos produtos.

**- Inventário reduzido**

CIM permite o controle do fluxo de material e, por isso, reduz o inventário e o

trabalho em processo. Quando a necessidade de um inventário grande é eliminada, os custos são reduzidos.

**- Qualidade e produtividade superiores através da automação**

A variabilidade na qualidade dos produtos é causada pelas máquinas e operadores humanos durante a produção. A eliminação da variabilidade gerada pelo homem melhorará a qualidade dos produtos e aumentará a produtividade.

**- Manufatura inteligente**

Ultimamente, computadores podem ser utilizados para inovar em produtos, desenvolver projetos práticos para produtos que podem ser manufaturados rapidamente, simular métodos de produção, projetar planos de processo, programar a produção, operar as fábricas, empacotar e distribuir os produtos.

O conceito CIM, como uma filosofia de produção, representa um dos últimos estágios da revolução tecnológica, o mais extensivamente desenvolvido em países industrialmente avançados, dentro do nível de computadorização atingido pelas indústrias dos Estados Unidos, Japão, Alemanha Federal e Inglaterra. Características específicas na implementação deste conceito podem ser distingüidas.

Majstorovic e Milacič (1990) apresentaram as características a seguir:

- a) A automação flexível é amplamente implementada na indústria norte-americana, particularmente, sistemas CAD-3D, conceitos de controle da produção com Planejamento de Recursos da Manufatura (MRP II) e linhas flexíveis de manufatura. A integração do conceito CIM é obtida através de duas abordagens: (i) tecnologia de grupo (TG) e (ii) integração CAD/CAPP/CNC.
- b) O conceito japonês de automação flexível é caracterizado pelo uso de sistemas CAD-2D, inicialmente para desenho, pelo desenvolvimento e implementação do conceito de Produção somente quanto e quando necessário (JIT) para o controle de produção e pelo uso de um grande número de FMS. O conceito de CIM é realizado através de: (i) integração de CAPP-DNC-FMS; (ii) automação de fábrica e (iii) organização e interconexão das ilhas de automação de manufatura, montagem e manuseio.
- c) A automação flexível na indústria da República Federal da Alemanha caracteriza-se pela implementação de um grande número de diferentes sistemas CAD 2-3D, pelo uso de sistemas de Planejamento das Necessidades de Materiais (MRP) para controle da Produção, bem como por diferentes formas e níveis de FMS. A abordagem de integração para introdução do conceito de CIM é obtida no nível de chão de fábrica através da ligação CAD-CAM e do sistema de controle da produção.

d) Conclusões semelhantes podem ser deduzidas acerca da indústria inglesa.

Martin (1990) apresentou alguns dados de um trabalho feito com 1300 firmas representativas, na capital germânica, do setor de bens de manufatura, sobre a intensidade com que tecnologias de informática vêm sendo usadas atualmente. Em mais de 90% destas firmas, pelo menos uma função é auxiliada por computador, sendo a maioria do setor comercial. Somente 15% destas firmas usam a tecnologia para atividades técnicas, tais como projeto de engenharia e programação da produção. No setor produtivo, mais de um terço destas companhias tem máquinas-ferramentas, mas menos de 1% tem sistema de fluxo de material automatizado. Somente 9% destas firmas têm uma ligação interna entre as várias áreas funcionais da companhia. A interconexão implementada mais frequentemente (4%) é a do planejamento de produção assistido por computador e planejamento do trabalho.

Suh (1990) fez uma ampla análise do impacto do CIM no crescimento da economia dos Estados Unidos. CIM já está sendo implantado por algumas indústrias, mas a velocidade de adoção tem sido lenta. Na maioria dos casos, CIM será introduzido pela indústria em pequenos passos, com a automação a nível de célula precedendo a computadorização da indústria toda. A obtenção da integração da informação de projeto para manufatura e outros conceitos relacionados ao CIM pode levar muitos anos devido às dificuldades tecnológicas e aos custos. Aspectos não produtivos do CIM podem ser implementados antes das funções de engenharia e produção, já que estas requerem entendimento e modelagem completa dos processos e sistemas existentes, os quais são frequentemente específicos, não existindo software de propósitos gerais. Por outro lado, funções de escritório são similares em muitas companhias. Portanto, partes não técnicas do CIM podem usar pacotes de software disponíveis.

Um dos pré-requisitos para implementar CIM é a disponibilidade de recursos humanos. Muitas universidades norte-americanas estão expandindo seus programas educacionais em manufaturas, mas a maioria dos programas universitários estão ainda em suas fases iniciais de desenvolvimento. Em futuro próximo, a tarefa de educar as pessoas no CIM pode ser compartilhada pelas indústrias e universidades.

Outro pré-requisito para o desenvolvimento do CIM é uma base teórica forte para projetar sistemas produtivos baseados no computador.

Muitas indústrias norte-americanas, tais como a GM, estão engajadas em Pesquisa e Desenvolvimento para implementar conceitos CIM para encurtar o tempo de desenvolvimento de produto. É reconhecido que, para ser competitiva, uma indústria precisa ser capaz de desenvolver e manufaturar novos produtos rapidamente a um custo inferior.

O uso do CIM não é sinônimo de fábricas sem pessoas. Fábricas totalmente automatizadas podem ser economicamente inviáveis. Nações industrialmente avançadas, como os Estados Unidos e Japão, precisam trabalhar juntas para tornar o CIM realidade (Suh, 1990).

No Brasil, a IBM tem um Centro de Apoio ao CIM, na sua fábrica em Sumaré, com o objetivo de mostrar ao mercado sua experiência na área. Além disso, em 1989 a Secretaria Especial de Informática (SEI) e a FINEP promoveram um encontro nacional reunindo centros de pesquisa, universidades, entidades associativas de fabricantes e usuários, com atividades relacionadas ao desenvolvimento e à implantação de projetos "CIM" no país. Foi emitido um relatório divulgando os trabalhos apresentados pelo CERTI/UFSC, CTI, UNICAMP e USP/São Carlos e as recomendações que resultaram deste encontro.

Para a continuidade das atividades e acompanhamento da execução das recomendações, foram criados quatro grupos técnicos, a saber: a) Estratégias de Automação; b) Planejamento do Produto, Processo e Produção; c) Célula Flexível de Manufatura; e d) Integração. Também foi criado um Comitê de Diretrizes e Avaliação - CDA, com a finalidade de estabelecer diretrizes e recomendações técnicas relacionadas ao CIM.

Crestin (1990) conclui que o caminho para o CIM passa por muitos avanços em diferentes campos técnicos. Avanços independentes em todas as tarefas auxiliadas por computador são necessários. Padrões e protocolos bem aceitos serão necessários para lidar com a complexidade dos sistemas CIM. Parece realístico que, nos próximos dez anos, progressos no hardware irão atender aos requerimentos do CIM. Não é claro que o mesmo acontecerá com o software, pois o desafio da engenharia de software é grande. Finalmente, existe uma necessidade mais teórica de entender e modelar o sistema de manufatura completo com seus complexos interrelacionamentos envolvendo homens, máquinas e computadores.

Como já vimos anteriormente, entre as novas tecnologias de Automação Flexível de Manufatura temos o CIM e o FMS. Tendo já uma visão da primeira, passaremos agora a abordar a segunda. De acordo com Kalkunte e outros (1986), que apresentaram uma revisão bibliográfica ampla abordando projeto, gerenciamento e controle de FMS, a definição fundamental de um FMS é, nas palavras de Buzacott e Shanthikumar (1980): *"um conjunto de máquinas interligadas por um sistema de manuseio de materiais e tudo sob controle de um computador central"*.

Para Stecke e Morin (1985) um FMS consiste de máquinas- -ferramentas a comando numérico interligadas por um sistema automático de manuseio de materiais. O controle, via computador, precisa atuar em tempo real sobre as operações de fabricação, movi-

mentação das partes e troca de ferramentas. Um FMS pode produzir, simultânea e eficientemente, várias peças diferentes. Esta combinação de automação e flexibilidade crescente oferece um amplo potencial para melhorias na produtividade, mas, como observado por Graves (1981), também uma crescente complexidade dos problemas de Gerenciamento de Produção.

Lee e Mirchandani (1986) têm uma visão diferente, explicada a seguir. Em ambientes convencionais de manufatura de volumes médios de produção, máquinas dedicadas são usadas em um ambiente do tipo "job-shop". Cada máquina, em um sistema produtivo deste tipo, é usualmente capaz de executar somente um tipo de operação. Estatísticas revelam que tais sistemas são geralmente ineficientes, tendo uma taxa média de utilização de máquina de somente 6%. Além disso, apresentam excessivo estoque de trabalho em processo ("work-in-process" (WIP)) e grandes tempos de preparação da máquina ("set-up") não produtivos. Uma completa revolução no mundo da manufatura tem mudado os métodos de produção de volumes médios. Computadores vêm sendo usados para controlar e monitorar os processos, máquinas programáveis capazes de executar uma grande variedade de tipos de operação vêm sendo desenvolvidas, tendo "magazines" de ferramentas que podem carregar várias ferramentas. Quando uma máquina deste tipo é integrada com um trocador de ferramentas, temos um módulo flexível de manufatura (FMM). Vários FMM juntos, física ou logicamente, constituem uma FMC. Interligando um grupo de FMC's com um sistema de manuseio de materiais e um sistema de controle por computador, temos um FMS.

Kalkunte e outros (1986) também apresentam um esquema de classificação dos problemas em um FMS, separando-os por níveis de gerenciamento e amplitude do horizonte de planejamento associado. A Figura 1.3, a seguir, ilustra estes níveis.

Abordagens quantitativas recentes têm sido desenvolvidas para ajudar os que tomam decisões em cada um destes níveis e são descritas em detalhes suficientes para fornecer uma visão de suas atuais potencialidades e limitações.

Modelos de programação inteira e simulação têm sido usados para avaliar FMS's. Procedimentos de planejamento estratégico têm sido descritos e sugestões para selecionar e avaliar um FMS têm sido feitas.

Uma dificuldade encontrada no nível 1 é a aplicação da análise de engenharia econômica tradicional. Esta abordagem requer informações detalhadas das peças que serão produzidas (seqüências de operações, por exemplo), bem como as máquinas específicas que serão usadas e os tempos de cada operação. Entretanto, não é possível descrever as peças em detalhes, uma vez que neste nível o horizonte considerado é de 3 a 10 anos.

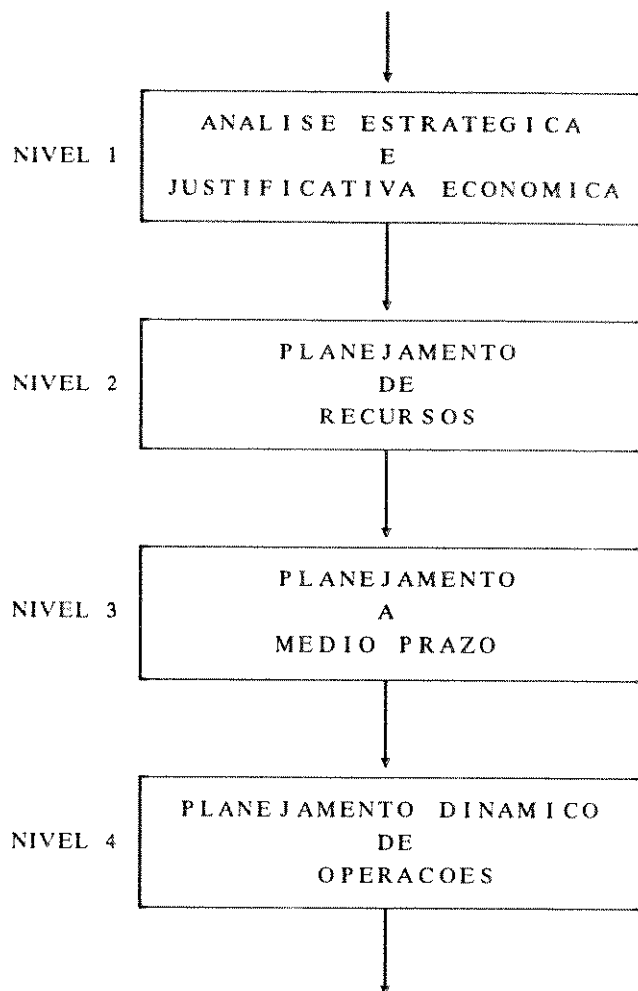


FIGURA 1.3 - Níveis de problemas em um FMS

É importante que abordagens adequadas para análise estratégica e justificativa econômica de FMS sejam desenvolvidas.

No nível 2 os planos estratégicos são unidos ao planejamento de recursos necessários para atendimento dos objetivos de longo prazo. O planejamento de recursos para um FMS requer especificações do conjunto de peças a serem processadas; do número e tipo de máquinas de processamento; do tipo de sistema de transporte e manuseio de material (por exemplo: robôs, veículos guiados automaticamente (AGV), correias aéreas, trilhos, etc.); das interfaces entre processamento e manuseio (robôs, estações de cargas e descarga); da forma de armazenagem do estoque de "work-in-process" (por exemplo: um local para armazenamento das peças ("buffer") para cada máquina ou um "buffer" central ou uma combinação de ambos); do tipo de computador e dos componentes do sistema de controle e do projeto do espaço físico ("layout") do sistema.

Os modelos disponíveis incorporam somente um subconjunto de componentes, mas podem fornecer uma visão útil para desenvolver planejamentos eficientes.

Uma das decisões fundamentais neste nível é determinar o conjunto de peças que pode ser processado vantajosamente em um FMS. O impacto desta decisão é descrito por vários modelos que também refletem a interação desta decisão com outras.

O balanceamento das cargas de trabalho pode não ser uma estratégia ótima quando o número de peças é restrito, por exemplo, por recursos limitados de ferramentas. Buzacott e Shantikumar (1980) aplicaram um modelo de filas de capacidade finita, aberto e indicaram que a capacidade depende da carga das máquinas. Demonstraram a importância do roteamento das peças, usando um "job-shop" balanceado com duas máquinas. O modelo utilizado serviu também para demonstrar que a capacidade é maximizada por selecionar peças que levem a uma estrutura de "job-shop" puro e minimizada por uma estrutura de "flow-shop". É, portanto, melhor permitir flexibilidade para as operações de seqüenciamento durante o despacho do que fixar seqüências no planejamento pré-produção. Isto pode ser obtido selecionando uma variedade de peças e tomando as decisões de roteamento para despacho no nível 2.

A capacidade de armazenamento local nos atuais FMS's é limitada. Bloqueios de máquinas ocorrem quando uma peça processada não pode mover-se para a próxima operação por causa do espaço de armazenamento limitado. Uma prioridade, portanto, precisa ser estabelecida entre fazer uma nova liberação e transferir um "job" de uma máquina bloqueada. Buzacott e Shantikumar (1980) desenvolveram modelos de sistemas somente com armazenagem local; somente com armazenagem central e com alguma combinação destes.

Armazenagem central oferece eficiências inerentes por consolidar filas. Buzacott e Shantikumar (1980) afirmam que armazenagem central é superior à local uma vez que a primeira restringe automaticamente o número de "jobs" no sistema a um valor ótimo. O melhor projeto adiciona armazenagem central com local para uma ou duas peças a cada estação. Este esquema reduz tempo de máquina vazia, mantendo o uso eficiente dos recursos de armazenamento.

A maioria dos modelos que podem ser aplicados no nível 2 é baseada ou em simulação ou em rede de filas fechada. Modelos de simulação são muito detalhados e, geralmente, incorporam controle operacional como discutido no nível 4.

Modelos de rede de filas fechadas são baseados em várias suposições que podem não representar operações reais de um FMS, por exemplo, os tempos de serviço são determinísticos.

Outra abordagem para modelar FMS, usando técnicas matemáticas simples, foi desenvolvida por Suri e Hildebrant (1984). Esta abordagem é baseada na análise do valor



médio de filas, resultando em um algoritmo recursivo simples para determinar medidas de desempenho. Esta abordagem permite também que sejam especificadas classes de peças múltiplas com número fixo de "pallets" e dispositivos de fixação dedicados para cada tipo.

A abordagem de Análise de Perturbação é um método relativamente novo, originalmente usado para estudar a localização de "buffers" em uma linha de produção, mas, recentemente, tem sido aplicada ao estudo de linhas de transferência, linhas de montagens, redes fechadas de filas e FMS. Ela combina a precisão dos métodos experimentais com a potência de análises teóricas. Seus conceitos chaves são geração de perturbação e propagação de perturbação.

No nível 3, decisões relacionadas ao Programa Mestre de Produção e ao Planejamento, com horizonte de 2-3 dias a 3-6 meses, são tratadas com questões específicas como roteamento de peças para maximizar a produção, carregamento das máquinas para determinar alocações alternativas das operações e alocação de ferramentas para máquinas.

Kimemia e Gershwin (1979) formularam o problema de roteamento das peças, o qual pode ser visto alternativamente como um problema de nível 2 ou 3, dependendo da aplicação. Eles supõem que cada peça pode ser produzida, usando estratégias alternativas (i.e., seqüência de operações) para explorar a vantagem da flexibilidade de um FMS. Estratégias alternativas podem ser planejadas para balancear cargas de trabalho e, além disso, podem ser especificadas no nível 4 para responder dinamicamente à quebra de máquina ou às sobrecargas temporárias. O modelo é baseado em fluxo de rede com variáveis de decisão de taxa de fluxo das peças e restrições que asseguram o atendimento de características próprias do FMS, como taxa de produção especificada, capacidade limitada das máquinas, nível médio de limitações do WIP, bem como características de modelagem como conservação do fluxo nos nós e limitação da capacidade dos arcos. Para assegurar que as restrições do WIP são observadas, uma fila com comprimento médio em cada máquina precisa ser estimada. Isto é um problema não linear que os autores resolveram usando o método do Lagrangeano Aumentado com decomposição Dantzig-Wolfe. As soluções dão planos ótimos de roteamento das peças para maximizar as saídas da produção. Se as chegadas e os tempos de processamento são determinísticos, o modelo reduz-se a um Programa Linear.

Kimemia e Gershwin (1985) voltaram a abordar o problema de roteamento ótimo das peças em um FMS modelado como rede de filas, utilizando um método aplicável ao controle, estendendo o estudo apresentado por eles anteriormente para comparar os resultados obtidos com os fornecidos por um simulador no problema exemplo de 6 peças e 4 máquinas com todos os tempos determinísticos.

Gershwin, Akella e Choong (1985) implementaram a política hierárquica Kimemia-Gershwin com simplificações que minimizaram o tempo computacional. São apresentados resultados de uma simulação baseada no modelo detalhado de uma fábrica de montagem de placas de circuito impresso.

Entretanto, resolver o problema, no qual cada tipo de operação pode ser alocado para mais do que uma máquina, é difícil.

Stecke (1983) identificou vários objetivos que podem ser usados neste problema (por exemplo, balancear os tempos de processamento das máquinas alocadas). As restrições, em tal problema, podem incluir a capacidade de ferramentas e alocação das operações e ferramentas para, pelo menos, uma máquina. Stecke (1983) formulou este problema com um modelo de programação inteira mista 0-1. Ela aplicou procedimentos de linearização para facilitar a solução. Posteriormente, Stecke (1986) reformulou este problema, considerando a alocação de máquinas-ferramentas, operações e ferramentas de corte associadas aos tipos de peças que foram selecionadas para serem produzidas simultaneamente, como um problema de programação matemática não-linear inteira e utilizou a técnica "Branch and Bound" para resolvê-lo.

O nível 4 está envolvido com operações dinâmicas, minuto a minuto, de um FMS, de forma que as decisões deste nível precisam ser feitas para liberar peças específicas dentro do FMS para produção e as peças precisam ser despachadas.

Procedimentos detalhados para controlar o transporte de material precisam também ser implementados para despachar efetivamente, mantendo alta utilização de máquina e evitando conflito entre os equipamentos de transporte. Decisões dinâmicas também precisam ser tomadas para usar máquinas alternativas para compensar a quebra ou evitar filas temporárias.

A maior parte dos estudos relacionados às questões deste nível utilizaram a abordagem da simulação porque operações dinâmicas são muito complexas, sendo um grande desafio às técnicas de modelagem analítica.

A seguir vamos abordar o problema de Programação Horária.

### **1.3. PROBLEMA DE PROGRAMAÇÃO HORÁRIA**

Baker (1974) define Programação Horária como sendo a alocação de recursos sobre o tempo para a execução de um conjunto de tarefas e afirma que a teoria existente se

caracteriza predominantemente por uma abordagem quantitativa, obtida através de modelos matemáticos. A maior parte dos desenvolvimentos nesta área foi motivada por problemas de manufatura. Portanto, é natural, ao descrever um problema de programação horária, utilizar o vocabulário da manufatura: recursos são usualmente chamados máquinas e módulos de tarefas básicas são chamados "jobs". Algumas vezes "jobs" podem se constituir de várias tarefas elementares que são inter-relacionadas por restrições de precedências. Estas tarefas elementares são referidas como operações.

Para classificar a maior parte dos modelos de programação horária, é necessário caracterizar a configuração das máquinas e o comportamento das tarefas.

O modelo básico composto por uma única máquina é fundamental no estudo do seqüenciamento e da programação horária, pois neste caso os sub-problemas de alocação de recursos e de seqüenciamento têm a mesma dimensão que o problema de programação horária. Mesmo neste problema, o conjunto de soluções factíveis pode ser muito grande e a determinação de um ótimo pode ser um problema combinatorial de grande dimensão.

No desenvolvimento de modelos de programação horária mais gerais do que o modelo de uma única máquina, o "flow-shop" representa a extensão mais direta para situações de múltiplos recursos. Nesta área, a teoria de programação horária tem sido fortemente influenciada pelo trabalho anterior de Johnson (1954), talvez por fornecer a única regra de programação horária ótima aplicável a uma extensa classe de "flow-shop".

No "flow-shop" a operação  $k$  de qualquer "job" é executada pela máquina  $k$  e não há necessidade de distinguir entre número de operação e número de máquina.

As características dos problemas "flow-shop" são as seguintes:

- 1) Um conjunto de  $n$  "jobs", cada "job" requerendo  $m$  operações e cada operação requerendo uma máquina diferente, está disponível para processamento no tempo zero (as máquinas são numeradas de 1 a  $m$ ).
- 2) Os tempos de "set-up" para as operações são independentes da seqüência e incluídos nos tempos de processamento.
- 3) Os tipos de "jobs" são conhecidos a priori.
- 4) As  $m$  diferentes máquinas estão continuamente disponíveis.
- 5) As operações individuais não podem ser particionadas.

Neste problema existem  $n!$  diferentes seqüências de "jobs" possíveis para cada

máquina e, portanto,  $(n!)^m$  diferentes soluções, para o problema de programação horária, a serem examinadas.

Na busca da solução ótima, é necessário considerar as diferentes seqüências de "jobs" nas diferentes máquinas com duas exceções:

- 1) Para qualquer função objetivo é suficiente a mesma ordem de "jobs" ocorrer nas máquinas 1 e 2, de forma que  $(n!)^{m-1}$  soluções deverão ser consideradas.
- 2) Para a função objetivo minimizar o tempo requerido para completar todos os "jobs" é suficiente a mesma ordem de "jobs" ocorrer nas máquinas  $m-1$  e  $m$ , de forma que  $(n!)^{m-2}$  soluções constituem um conjunto dominante para  $m > 2$ . O melhor resultado conhecido para programação horária de "flow-shop" com 2 máquinas e esta função objetivo ainda é o de Johnson (1954).

No clássico problema de programação horária de "job-shop", o fluxo de trabalho não é unidirecional como no "flow-shop", sendo mais apropriado descrever uma operação com a tripla  $(i,j,k)$  para indicar que a operação  $j$  do "job"  $i$  requer a máquina  $k$ .

Dada uma seqüência de operação para cada máquina, existe somente uma solução de programação horária, na qual nenhum deslocamento local à esquerda no gráfico com os horários de produção ("gráfico de Gantt") pode ser feito. O conjunto de todas as soluções nas quais nenhum deslocamento destes pode ser feito, é chamado o conjunto de soluções semi-ativas e é equivalente ao conjunto de todas as soluções que não contém tempo vazio supérfluo. É suficiente considerar este conjunto para otimizar qualquer medida de desempenho.

Considere o problema "job-shop" clássico, no qual cada "job" tem exatamente uma operação em cada máquina e cada máquina precisa processar  $n$  operações. O número de seqüências possíveis é, portanto,  $n!$  para cada máquina. Se as seqüências em cada máquina são inteiramente independentes, existem  $(n!)^m$  soluções de programação horária semi-ativas. Em uma solução deste tipo, o tempo inicial de uma particular operação é restrito, ou pelo processamento de um "job" diferente na mesma máquina ou pelo processamento da operação imediatamente precedente em uma máquina diferente. O tipo de ajustamento no qual alguma operação é iniciada mais cedo sem atrasar qualquer outra operação é chamado deslocamento global à esquerda. O conjunto de todas as soluções, nas quais nenhum deslocamento deste tipo pode ser feito, é chamado o conjunto de soluções de programação horária ativas, que é claramente um subconjunto das semi-ativas. Geralmente, é o menor conjunto dominante do problema estudado.

Uma importante classe de procedimentos para gerar soluções sub-ótimas para o

problema de programação horária de "job-shop" é a classe de regras de despacho, tais como:

- LPT - (Longest Processing Time): Seleção da operação com o tempo máximo de processamento.
- SPT (Shortest Processing Time): Seleção da operação com o tempo mínimo de processamento.
- FCFS (First Come First Served): Seleção da operação que chegou primeiro.
- MWKR (Most Work Remaining): Seleção da operação associada com o "job" que tem a maior quantidade de trabalho restante a ser processado.
- MOPR (Most Operations Remaining): Seleção da operação associada com o "job" que tem o maior número de operações restantes a serem processadas.
- LWKR (Least Work Remaining): Seleção da operação associada com o "job" que tem a menor quantidade de trabalho restante a ser processado.
- Random: Seleção randômica da operação.

Graves (1981) apresenta uma ampla classificação dos vários problemas de programação horária e procura estabelecer perspectivas sobre o status da pesquisa em programação horária relativamente à prática da mesma.

Os problemas de programação horária podem ser classificados de acordo com:

- Geração da demanda
- Complexidade do processamento
- Critérios de programação horária.

As demandas podem ser geradas diretamente pelos clientes - OPEN SHOP - ou indiretamente pelas decisões de reabastecimento do estoque - CLOSED SHOP.

Para "open-shop", a programação horária da produção é um problema de seqüenciamento no qual ordens abertas são seqüenciadas em cada processador.

Para "closed-shop", a programação horária da produção envolve não somente decisões de seqüenciamento, mas também decisões do tamanho de lote associado com o processo de reabastecimento do estoque.

Um "open-shop" ou "closed-shop" puro é raro.

A complexidade do processamento está relacionada ao número de passos de proces-

samento associado a cada tarefa de produção. Uma decomposição para esta dimensão é como segue:

- Um estágio, uma máquina
- Um estágio, máquinas paralelas
- Multiestágio, "flow-shop"
- Multiestágio, "job-shop".

Comparando com a terminologia adotada por Baker (1974), estágio corresponde a operação.

Crítérios de programação horária podem ser:

- custo da solução da programação horária, incluindo custos fixos, custos variáveis, custos de estoque, multas por atraso e custos de expedição;
- desempenho da solução medido pelo nível de utilização dos recursos de produção, pelo tempo de fluxo médio ou máximo para um conjunto de tarefas, pelo atraso máximo ou médio de um conjunto de tarefas e pelo percentual de tarefas atrasadas.

Na maior parte dos ambientes produtivos, a avaliação da solução da programação horária é baseada em ambos os critérios. Entretanto, a maior parte da literatura apresenta problemas com um único critério.

Além disso, a literatura apresenta o critério de desempenho para "open shop" e o critério de mínimo custo para "closed shop".

Duas outras dimensões, que podem ser incluídas, são: a natureza da especificação da demanda, que pode ser determinística ou estocástica, e o ambiente da programação horária, que pode ser estático ou dinâmico.

A maior parte dos ambientes produtivos para o problema de programação horária são estocásticos e dinâmicos, enquanto na maior parte os modelos para os problemas de programação horária são determinísticos e estáticos.

Numerosos procedimentos de otimização combinatorial têm sido propostos para resolver o problema geral da programação horária de "flow-shop" com o critério de minimização do tempo máximo de fluxo. As abordagens de maior sucesso são procedimentos "branch and bound" que usam limites e métodos para eliminar seqüências dominadas.

Já o problema da programação horária de "job-shop", para o qual também se utilizam procedimentos "branch and bound", requer no caso dinâmico simulações nas quais comparam-se várias regras de despacho.

Todos os problemas de programação horária em "closed-shop" determinísticos podem ser formulados como Programação Linear Intelta Mista e a maioria dos procedimentos de solução são procedimentos enumerativos que tentam explorar alguma estrutura especial. Os problemas de dimensionamento de lotes têm sido freqüentemente modelados como fluxo de mínimo custo em rede, sendo, portanto, eficientemente resolvidos.

Graves (1981), após uma revisão ampla dos trabalhos teóricos em problemas de programação horária da produção, constatou um significativo avanço, mas verificou a existência de muitos problemas ainda não resolvidos. Além disso, ressaltou que não é simples fazer uma revisão bibliográfica e sumariar a prática correspondente, pois o ambiente de programação horária é muito variável e os trabalhos nesta área, geralmente, não são publicados.

De qualquer forma, Graves verificou a predominância da programação horária puramente manual.

Em ambientes mais complexos existem sistemas de controle de chão-de-fábrica ("shop floor") ou de Planejamento de Necessidades de Materiais (MRP) ou ambos. Mas, estes sistemas não têm mecanismos para considerar os "trade-offs" padrões associados com as decisões de programação horária. Estes sistemas não garantem a factibilidade de produção. Há necessidade de completar o desenvolvimento da teoria desta área, visando aplicações.

Seis áreas de pesquisa são sugeridas por Graves (1981):

### **1. Diagnóstico**

Uma linha de pesquisa, nesta área, é a avaliação do sistema de programação horária da produção, para determinar quando o sistema é eficaz e quando ele pode ser melhorado.

### **2. Robustez da Programação Horária**

Uma linha de pesquisa nesta área é a caracterização do horizonte de planejamento para os modelos determinísticos de forma que as decisões atuais da programação horária continuem válidas além do horizonte considerado.

Outra linha é a análise dos efeitos dos horizontes da programação horária.

Uma terceira linha é relativa à sensibilidade do sistema à característica dinâmica do ambiente da programação horária.

### **3. Interação da Programação Horária**

As soluções da programação horária da produção são implementadas não isoladamente, mas como parte de um sistema operacional global. As decisões da programação horária afetam, e são afetadas por decisões de planejamento, mercado, transporte e distribuição. Esta interação não é refletida na maioria dos modelos.

### **4. Valor da Informação**

Dada a predominância dos sistemas de Planejamento de Necessidades de Materiais e de Controle "shop-floor" aparece a questão: *Qual o valor das informações destes sistemas para a geração de boas soluções da programação horária?* Uma vez que ambos os sistemas são sistemas elaborados de informações, é desejável entender o "trade-off" entre os custos crescentes do sistema para obtenção e manutenção das informações acuradas versus a economia de custos obtida com melhores soluções. Além disso, é necessário determinar quão elaborados tais sistemas necessitam ser e a partir de que ponto há retorno líquido negativo por informações mais detalhadas.

### **5. Funções Especializadas de Programação Horária**

É importante analisar como a expedição de tarefas essenciais pode ser feita de forma a garantir o melhor desempenho global da programação horária. Outra função especializada é a liberação das tarefas para o "shop-floor", especialmente para "open-shop".

### **6. Programação Horária de Sistemas de Manufatura Computadorizados**

Estes sistemas são altamente automatizados e consistem de máquinas controladas por computadores e manuseio automático de materiais. A programação horária destes sistemas precisa operar em tempo real, refletir a flexibilidade de rotas alternativas e incorporar a restrição de capacidade finita do sistema de transporte e manuseio entre as máquinas e os "buffers" de armazenagem.

Decisões da programação horária final precisam ser feitas dinamicamente com base no estado do sistema e permitindo ajustes a interrupções. Como os componentes de um FMS estão particularmente inter-relacionados, a tarefa de programação horária dos "jobs" determina a quantidade de ineficiência devida à máquina bloqueada e à interferência entre equipamentos de transporte. Por esta razão, é importante modelar, de forma detalhada, o sistema de manuseio de materiais e a lógica de operação do mesmo.

Coordenação das peças, máquinas, ferramentas e dos equipamentos de manuseio de



materiais em um FMS é uma tarefa de programação horária mais complexa do que a encontrada em "job-shop" convencional. Regras de despacho com competência para antever e evitar interferência e/ou acúmulo constituem um método primário para programação horária de um FMS complexo.

Nosso trabalho está inserido nesta última área de pesquisa.

#### 1.4. LEVANTAMENTO BIBLIOGRÁFICO EM PROGRAMAÇÃO HORÁRIA DE FMC E FMS

Neste item vamos apresentar um breve levantamento bibliográfico em ordem cronológica na área de programação horária em FMC e FMS.

Hitz (1979) desenvolveu um procedimento que prescreve soluções ótimas para o problema de programação horária de um FMS, o qual processa peças de acordo com uma estrutura de "flow-shop". Ele estudou um sistema determinístico no qual os tempos de processamento e a seqüência de operações são conhecidos com certeza. O FMS foi caracterizado por transporte das peças totalmente automático e capacidade unitária do "buffer" local de cada máquina. O objetivo desta abordagem foi maximizar a taxa de saída das peças prontas, sujeito ao requisito da produção das peças em taxas especificadas. Soluções ótimas caracterizadas pela liberação periódica das seqüências que minimizam tempos ociosos da máquina gargalo foram desenvolvidas por Hitz (1979). O uso deste objetivo oferece significativa economia computacional se comparado com outros objetivos, tais como minimização do "makespan" ou do tempo médio ponderado de fluxo. Hitz (1979) utilizou um algoritmo de enumeração implícita para resolver o problema. Em contraste com abordagens tradicionais do seqüenciamento de "flow-shop", uma peça pode ultrapassar outra nesta formulação. De fato, Hitz (1979) estendeu o modelo básico para permitir armazenagem local de uma dada capacidade e ultrapassagem de três peças consecutivas.

Buzacott e Shanthikumar (1980) compararam as regras primeiro que chega é o primeiro a ser servido (FCFS) e liberação de máquina vazia. Eles também demonstraram o efeito da quebra na capacidade de produção. Entretanto, nenhuma abordagem prática tem sido vislumbrada para controlar operações dinamicamente segundo suas evoluções ao longo do tempo.

Uma nova abordagem para controle em tempo real de FMS foi apresentada por Hildebrant e Suri [1980]. O problema é: *Como determinar a solução da programação horária de um conjunto de "jobs" em máquinas propensas a falhas para minimização do tempo total de processamento?* Este problema foi dividido em três estágios de decisão.

Os recursos críticos considerados no primeiro estágio são máquinas e dispositivos e supõe-se que a distribuição de tarefas no sistema tem características de perfil constante e que a média de atraso das tarefas esperando por recursos estará abaixo de certos valores. O segundo estágio determina a solução do conflito de recursos a curto prazo para minimizar o atraso médio das tarefas. O terceiro estágio é muito detalhado, difícil de formular, sendo por isso utilizadas regras de despacho interativamente.

Kalkunte e outros (1986) observaram que a maior parte dos modelos disponíveis são descritivos (em vez de prescritivos) e precisam ser aplicados iterativamente para determinar soluções eficientes.

Nos estudos de caso apresentados por Ranky (1986), as melhores soluções para FMS são modulares, flexíveis e expansíveis, contendo não só a programação horária prévia como também programação horária em tempo real do fluxo de trabalho dentro das células, e, além disso, sincronizando atividades de transporte e atividades da célula dentro da área controlada pelo DNC. Em particular, foi apresentado um FMS produzindo 543 tipos diferentes, 11120 itens de peças. Num sistema convencional, seriam necessárias 90 máquinas, em vez das 43 necessárias no FMS. Em lugar de 170 operadores, bastariam somente 36 nas salas do computador, das ferramentas e no "shop-floor" e somente de 3 lidando com a transferência de peças e controle da produção, em lugar das 25. Os tempos de processo foram reduzidos sensivelmente: no FMS são gastos somente 3 dias fabricando, 7 montando unidades e 20 na montagem final, comparado com 35, 14 e 42 dias no sistema convencional respectivamente. Finalmente, o espaço ocupado foi reduzido de 16500m<sup>2</sup> para 6600m<sup>2</sup>.

O método indicado por Ranky (1986) para a programação horária em FMS é dinâmico e multinível, mas considera cada célula como uma única máquina reduzindo o problema de programação horária em FMS ao problema de programação horária de "job-shop" geral.

Já Stecke (1983) afirma que gerenciar a produção em um FMS é mais difícil do que em linhas de produção e em "job-shop", porque cada máquina é versátil e capaz de executar muitas operações diferentes. Assim, o sistema pode fabricar vários tipos de peças simultaneamente e cada peça pode ter rotas alternativas no sistema. Estas aptidões adicionais e opções de planejamento aumentam o número de variáveis de decisão e de restrições associadas à preparação de um FMS.

Maimon (1987) apresenta um método para controlar as atividades de um FMS. A entrada do controlador é dada pela demanda da produção para os próximos dias e pelo estado atual do sistema. A saída fornece os comandos para executarem um programa ou um particular movimento. Uma versão prévia deste controlador está atualmente sendo

testada em uma indústria. O controlador precisa lidar com uma grande quantidade de dados estáticos e dinâmicos e tem uma estrutura hierárquica com três níveis. No primeiro é determinada a taxa de produção de cada tipo de peça. No segundo são determinados os movimentos internos das peças com resolução de alguns conflitos. No terceiro são transmitidas as decisões e é recebida a realimentação dos controladores das máquinas. Um processador de eventos coordena todas as atividades no controlador. Esta estrutura é geral, mas os algoritmos, em cada nível, não são necessariamente gerais. Algoritmos mais gerais são necessários, motivando futuras pesquisas. É importante observar que algumas soluções de conflito no segundo nível descem para o terceiro para simplificar a solução. Por exemplo, no segundo nível, dois robôs foram designados para pegar uma peça cada um. A colisão, que pode acontecer, é resolvida no terceiro nível. No primeiro nível utiliza-se Programação Dinâmica, enquanto que no segundo nível utilizam-se regras e máquina de inferência e no terceiro regras temporais para uma rápida resolução dos conflitos restantes. Um estudo de caso é apresentado: uma FMC com cinco estações de trabalho e dois robôs.

Edghill e Davies (1985) consideram que a programação horária em sistemas de controle de FMS está muito pouco desenvolvida, a maioria usando a regra primeiro que entra é o primeiro que sai (FIFO). Pesquisas acadêmicas têm investigado o efeito de diferentes regras de programação horária aplicadas às peças e ferramentas, mas é necessário reconhecer que cada FMS é uma combinação única de equipamentos complexos.

Tang (1986) utilizou um procedimento "Branch and Bound" para resolver a programação horária em indústrias de metal, quando máquinas numericamente controladas são usadas para fabricar pequenos lotes de peça, utilizando ferramentas de capacidade limitada. O tempo necessário para troca de ferramentas é curto, mas o tempo necessário para transferir ferramentas da área de armazenagem até o "magazine" é longo, de forma que se deve evitar desnecessárias trocas para aumentar a utilização das máquinas e reduzir a intervenção do operador.

Lee e Mirchandani (1986) analisaram o problema do "set-up" em FMS. Geralmente existem dois tipos: "set-up" do sistema e "set-up" das máquinas. O "set-up" do sistema corresponde à preparação do FMS para produção já especificada relativamente aos dispositivos de fixação e manuseio, ferramentas e "software". Já o "set-up" das máquinas se decompõe no "set-up" do local para armazenamento de ferramentas ("magazine"), que corresponde ao carregamento do "magazine" com as ferramentas necessárias, e no "set-up" de troca de ferramentas, que ocorre entre duas tarefas consecutivas da máquina.

O "set-up" do sistema tem sido tratado no problema de carregamento do FMS por

vários pesquisadores, como já vimos anteriormente.

O "set-up" de troca de ferramentas tem sido apropriadamente negligenciado pelos métodos da programação horária de FMS, já que a sofisticação tecnológica dos equipamentos de troca de ferramentas virtualmente elimina este tempo.

Entretanto, o "set-up" do "magazine" tem sido menos entendido em relação à sua influência na programação horária. Sem "set-up" do "magazine" cada FMM está restrito a executar somente aqueles tipos de operações já alocadas ao módulo, restringindo, portanto, o número de alternativas disponíveis no módulo para roteamento de cada peça. Lee e Mirchandani (1986) apresentaram uma heurística para determinar a programação horária de duas máquinas flexíveis para três casos: sem "set-up", com um "set-up" e com dois "set-ups", já que não é fácil determinar quantos "set-ups" a solução ótima da programação horária terá. No caso de não haver "set-up" de "magazine", a solução ótima pode ser obtida com o algoritmo de Johnson e foi comparada com a heurística.

Embora cada FMS seja uma combinação única de complexos equipamentos, como Edghill e Davies (1985) afirmaram, Dub (1984) apresentou um "software" para FMS padronizado com quatro níveis hierárquicos: programação horária, monitoramento, controle lógico e controle físico.

O controle do FMS é assegurado pela execução automática da programação horária, pelo controle do sistema de transporte e pela distribuição automática dos programas das máquinas a comando numérico.

A padronização de FMS é obtida através da modelagem funcional de um sistema de manufatura que permite a construção de um gráfico em duas dimensões, onde no eixo horizontal temos o tempo relativo dos eventos e no eixo vertical os processos sendo executados. Dub (1984) apresentou o módulo DYNAR para a programação horária dinâmica das tarefas de manufatura, juntamente com os demais módulos de software necessários para o controle de FMS.

Morton e Smunt (1986) apresentaram o "software" PATRIARCH com modificações requeridas por um FMS e propõem um sistema de programação horária que incorpora conceitos de suporte à decisão com interação manual para problemas fracamente estruturados e heurísticas para problemas bem estruturados, fornecendo soluções próximas do ótimo.

Kleebank e Walter (1988) apresentaram técnicas de Inteligência Artificial (IA) que têm se mostrado uma alternativa interessante para atender ao problema de planejamento da produção, tratando mais detalhadamente o nível de seqüenciamento de tarefas.

Compararam as características dos métodos algorítmicos e das técnicas de IA e discutiram as principais arquiteturas existentes, ou em desenvolvimento, descritas rapidamente a seguir:

ISIS (Intelligent Scheduling and Information System) é um sistema especialista para seqüenciamento de tarefas que está sendo testado em uma fábrica de componentes para turbinas da Westinghouse. Ele foi desenvolvido pela "University of Carnegie-Mellon". O sistema determina a prioridade dos pedidos e o seqüenciamento global da fábrica é gerado incrementalmente, pedido a pedido. A geração do seqüenciamento de um pedido é feita através de uma busca hierárquica dirigida por restrições no espaço dos seqüenciamentos alternativos.

CROPS (Coherent Rules for On-Line Production Scheduling") foi concebido no "Centre de Recherches Scientifiques et Techniques de l'industrie des fabrications metalliques" de Bruxelas e objetiva realizar o seqüenciamento e o controle de um FMS. Adota a abordagem hierárquica para representar ações e recursos envolvidos no seqüenciamento. Informações de tempos são tratadas, usando intervalos de referência e técnicas de propagação de restrições. A combinação destas características resulta em uma estrutura denominada Rede Temporal Hierárquica, que é um grafo cujos nós representam ações e os arcos indicam as relações entre as ações e é construída a partir do roteiro de peças e dos pedidos.

TANDEM é uma estrutura de sistema proposta por Andrew Kusiak da "University of Manitoba". A arquitetura combina as vantagens obtidas através de técnicas de IA com as vantagens fornecidas pelas técnicas algorítmicas e pode ser aplicada a vários problemas do ambiente de manufatura. Basicamente esta arquitetura é composta por um sistema especialista e um ou mais algoritmos, cada um associado a seu respectivo modelo de fábrica. Com as informações recebidas do ambiente de manufatura, o sistema especialista seleciona um par modelo/algoritmo e fornece ao modelo os dados necessários para o seqüenciamento. O algoritmo selecionado com base no modelo gera uma solução inicial. Se a solução obtida não é satisfatória, o sistema especialista decide se o algoritmo deve ser mantido ou um novo par modelo/algoritmo deve ser selecionado. Depois, um novo conjunto de dados (restrições adicionais) é fornecido ao modelo e, assim, um novo ciclo recomeça, até que um resultado satisfatório é obtido. Devido à complexidade computacional do problema do seqüenciamento não se exige que o algoritmo obtenha resultados ótimos.

O OPIS (Opportunistic Intelligent Scheduler) é um sistema especialista para seqüenciamento da produção que surgiu de uma tentativa de superar as falhas encontradas no ISIS, o qual não conseguia manipular corretamente os conflitos baseados em

recursos. A estrutura do OPIS emprega os mesmos princípios da arquitetura padrão do tipo quadro-negro (um grupo de especialistas com diferentes estratégias de seqüenciamento se reúne diante de um quadro-negro para propor hipóteses que são geradas, combinadas e avaliadas, até que uma solução seja encontrada). Atualmente, existem duas estratégias de seqüenciamento: seqüenciamento de recursos individuais e seqüenciamento de pedidos individuais.

O "Politécnico di Torino" desenvolveu uma ferramenta para seqüenciamento da produção, para ser aplicado em FMS, que dispõe de um módulo sequenciador, que usa técnicas de IA e um módulo avaliador de cargas, que algoritmicamente fornece, ao sequenciador, um conjunto de medidas de desempenho que auxiliam o sequenciador no processo de tomada de decisão. O seqüenciamento é, na realidade, uma simulação baseada em eventos discretos. E o módulo avaliador de cargas implementa um algoritmo para o modelo de redes de filas fechadas, que recebe informação sobre os pedidos, verifica o tamanho das filas de peças que se formam junto às máquinas, e gera estimativa de desempenho (tamanho médio das filas, utilização dos equipamentos, etc.).

Kleebank e Walter (1988) referiram-se também ao SISEP que estão desenvolvendo na UFRGS para micro PC-compatível, o qual procura implementar mecanismos para fazer com que o planejamento da produção, apesar de automatizado, seja um processo bastante flexível e dinâmico. Recursos gargalos da fábrica são tratados de forma especial, para minimizar problemas de alocação e o seqüenciamento poderá ser feito visando o cumprimento dos prazos ou a execução o mais cedo possível dos pedidos.

Os autores constataram que a maioria dos sistemas especialistas existentes para o seqüenciamento da produção ainda estão em fase de testes nas universidades e que alguns destes sistemas conciliam características heurísticas e algorítmicas para aproveitar as vantagens de cada uma.

Chryssolouris e outros (1988) apresentam o MADEMA, um sistema baseado em regras para a programação horária do trabalho considerando múltiplos critérios. Compararam sua performance com quatro regras clássicas: SPT, LPT (Longest Processing Time), FCFS e LCFS (Last Come First Served). A LPT seleciona a operação com maior tempo de processamento e a LCFS a operação que chegou por último.

Foram usados dois critérios: Tempo médio de fluxo e atraso médio.

Entre estas regras a SPT apresentou melhores resultados.

Rodammer e White (1988) procuram fornecer uma visão ampla e estruturada de sete paradigmas da Programação Horária da Produção:

- Prática Industrial corrente;
- Teoria de seqüenciamento e programação horária das máquinas;
- Programação horária de Projetos;
- Teoria de Controle;
- Simulação de eventos discretos;
- Otimização estocástica;
- Inteligência Artificial.

Concluem que nenhum destes paradigmas oferece, sozinho, a base para uma teoria unificada da Programação Horária da Produção.

O MRP, OPT e JIT, da prática industrial corrente, não foram projetados especialmente para o problema da programação horária.

Chang e outros (1989) destacam a flexibilidade de rota como a característica que distingue a programação horária do FMS da programação horária de um "job-shop". Com este grau de liberdade, pode-se eliminar gargalos. A proposta apresentada para a programação horária de FMS foi adaptada da conhecida técnica de Avaliação de Performance e Revisão/Métodos do Caminho Crítico (PERT/CPM), focalizando a utilização das máquinas alternativas para diminuir o gargalo. O método utilizado é um método de busca que gera boas programações horárias, como mostrado para três tipos de conjuntos de dados, pois utiliza informações globais do estado do chão-de-fábrica, diferentemente das regras de despacho que utilizam muito pouco destas informações.

Entretanto, não são utilizadas as informações relativas ao sistema de transporte para a geração de programações horárias realistas.

Perkins e Kumar (1989) apresentam políticas de programação horária implementáveis em tempo real de maneira distribuída para sistemas flexíveis de manufatura, montagem e desmontagem. Estas políticas estabilizam o sistema que produz todas as peças em taxas desejadas. De qualquer forma, é um ponto de partida para obtenção de uma teoria tratável de programação horária dinâmica para os sistemas de manufatura.

Shanker e Srinivasulu (1989) afirmam que os problemas básicos em FMS podem ser agrupados em projeto e operação e suas complexidades dependem do FMS ser do tipo dedicado, projetado para produzir uma pequena família de peças, ou randômico, projetado para produzir uma ampla variedade de peças com um conjunto de tipos de produtos ("mix") de produção em aberto. Eles propõem uma solução para o problema operacional de carregamento de FMS do tipo randômico que se baseia na seleção da máquina que tem a maior capacidade restante para o "job" que tem o maior tempo de processamento.

Bretthauer e Venkataramanan (1990) ressaltaram que o conjunto das rotas alternativas para a produção de peças em FMS desempenha papel chave para a obtenção dos potenciais benefícios. De qualquer forma, a flexibilidade obtida com as rotas alternativas tem um custo do aumento da complexidade do controle em tempo real do FMS.

Solot (1990) propôs a integração de técnicas de PO com técnicas de Sistemas Especialistas para resolver problemas de Planejamento e Programação Horária. Ele indica a necessidade de desenvolvimento de algoritmos e de regras para obter-se um sistema eficiente de planejamento e programação horária apropriado à dinâmica do FMS.

Chen e Askin (1990) comparam o desempenho de seis heurísticas de carregamento em três FMSs existentes que utilizam máquinas versáteis, capazes de executar uma ampla variedade de operações com rápida troca de ferramenta e mudança de instruções, permitindo a produção simultânea de várias peças. A flexibilidade do sistema permite ajustes do planejamento e de roteamento às mudanças da demanda e/ou quebra de máquinas. Cada heurística requer três passos básicos: seleção do tipo de máquina para cada operação; agrupamento de máquinas e alocação de ferramentas e operações aos grupos de máquinas. A alocação de operações baseada no balanceamento da carga, fornece melhores resultados que a alocação à máquina mais eficiente. Cabe lembrar que o balanceamento é estático.

Ronen e Starr (1990) analisam a filosofia, os conceitos e as regras do sistema de Tecnologia de Produção Otimizada (OPT) com base em conceitos e teorias de programação matemática (PM). As regras do OPT dão a programação horária ótima para o caso de  $n$  peças em 2 máquinas. A idéia de recurso gargalo corresponde à de recurso escasso na PM. Uma hora economizada em um recurso não-gargalo não melhora o desempenho do sistema. Esta afirmação pode ser provada pela PM: o custo marginal deste recurso não crítico é zero e um decréscimo em sua quantidade não irá alterar o desempenho do sistema. Já uma hora economizada em um recurso gargalo melhora o desempenho do sistema. Esta afirmação também pode ser provada pela PM: o custo marginal deste recurso é positivo e um decréscimo em sua quantidade irá alterar o desempenho do sistema.

Estas análises deveriam ser mais precisas, pois, dependendo da alteração proposta, a solução pode deixar de ser ótima.

Walter (1990) apresentou um algoritmo de escalonamento finito e reverso que se aplica tanto ao nível de decisão tático, no lugar de algoritmos de Planejamento de Recursos de Manufatura (MRP II) e Planejamento das Necessidades de Capacidade (CRP), como ao nível operacional, automatizando ou apoiando uma tarefa realizada manualmente. Os lotes de produção são dimensionados dinamicamente. Os conflitos entre dois



conjuntos de peças diferentes, que disputam o mesmo centro de trabalho, são resolvidos por uma regra derivada da heurística MWKR.

Um sistema, utilizando o algoritmo e a regra propostos, foi implantado em uma indústria do ramo metal-mecânico. O tempo de processamento em um PC-386 resultou semelhante ao tempo de processamento de um Sistema de Planejamento das Necessidades de Materiais (MRP), apesar das significativas vantagens em termos de qualidade.

Garetti e outros (1990) apresentam uma análise da influência do "mix" de produção no desempenho de regras de carregamento e despacho. Consideraram 14 diferentes mix, sendo 7 do tipo "flow-shop" e 7 do tipo "job-shop"; 6 algoritmos de carregamento, sendo 3 de balanceamento de carga, 1 de natureza randômica e os outros 2 centrados na máquina gargalo e 7 regras de despacho, sendo as 2 clássicas SPT e FCFS e as demais descritas a seguir.

MOP - o "job" que tem o maior número de operações tem precedência.

MOPR - o "job" que tem o maior número de operações restantes tem preferência.

MOPR-SPT - quando o anterior dá empate, desempata com SPT.

SWNQ1 - dá preferência para o "job" que está com sua última operação na máquina em questão.

SWNQ2 - dá preferência para o "job", cuja próxima operação se dá na máquina que tem a menor soma de tempos operacionais a serem processados.

Entre as diversas conclusões apresentadas, temos:

- As regras de despacho têm maior influência no "flow-shop" do que no "job-shop" (o erro na escolha de um "job" em uma das primeiras máquinas reflete-se diretamente nas seguintes no caso da sequência fixa);
- "job-shop" tem melhor desempenho que o "flow-shop";
- a importância das regras de carregamento é maior que as de despacho;
- as melhores regras de despacho são as que se baseiam no balanceamento de carga;
- a regra de despacho mais promissora é a SWNQ2.

A consideração da máquina gargalo, neste trabalho, leva em conta o tempo total de processamento das peças nas máquinas ou o tempo restante.

Ro e Kim (1990) apresentam regras para o controle operacional em um FMS, que foram desenvolvidas para fazer a programação horária simultânea de "jobs" e de equipamentos de manuseio de materiais. A justificativa para este tratamento simultâneo é que os tempos de "set-up" em um FMS são reduzidos, tornando os tempos de processamento comparáveis aos tempos de transporte. As medidas de desempenho consideram simultaneamente o "makespan", o tempo de fluxo médio, o atraso médio, o atraso máximo e a utilização do sistema. Uma consideração importante é de que em um FMS existem várias rotas alternativas e as regras apresentadas se diferenciam na forma como esta consideração é tratada:

ARD (rotas alternativas escolhidas dinamicamente) - Uma peça é liberada para a próxima máquina que tem o menor somatório dos tempos de transporte, fila e operação, sendo o tempo de fila definido como o tempo que falta para processar a peça que está na máquina, acrescido dos tempos operacionais das peças com prioridade mais alta que já estão esperando. A limitação desta regra é a consideração da peça isoladamente.

ARP (rotas alternativas planejadas) - As rotas alternativas são planejadas usando um modelo de Programação Linear que minimiza o "makespan". Entretanto, analisando o modelo, verificamos que este trabalha com os tempos de processamento e transporte sem alocá-los em uma escala de tempo.

ARPD (rotas alternativas planejadas e escolhidas dinamicamente) - A regra anterior, ARP, é estendida para permitir decisões dinâmicas de despacho de forma a adaptar-se às mudanças do estado do FMS. Em um passo inicial, a máquina é escolhida pela regra ARP. Se ela não está vazia, procura-se uma máquina alternativa vazia, usando a regra ARD.

Estas três regras foram comparadas com:

NAR (rotas alternativas não permitidas) - A rota com mínimo tempo total de processamento é selecionada e fixada.

WINQ (mínimo trabalho na fila) - A máquina que tem o mínimo trabalho na fila, em termos de tempo de processamento, é selecionada.

A última foi selecionada, pois parece ser uma das melhores.

Os melhores resultados foram obtidos com a ARD que, juntamente com a ARPD e WINQ, foram superiores a ARP e NAR.

Como já observamos anteriormente, a programação horária que poderia resultar da aplicação das regras não é fornecida.

Ahluwalia e Ji (1991) propõem uma abordagem distribuída para a programação horária de "jobs" em um FMS que considera cada máquina-ferramenta como um nó que pode comunicar-se com outros nós. Todas as decisões de programação horária são feitas de uma maneira distribuída por cada nó. Em um FMS hipotético, considerado para ilustrar a abordagem, os nós estão conectados via rede aérea local. As decisões de roteamento e re-roteamento dos "jobs" são tomados em cada nó da rede com base no estado do sistema. Mas, os resultados da programação horária, propriamente dita, não são apresentados.

Hutchison (1991) destaca a importância do problema da programação horária de FMS e analisa 17 trabalhos representativos da literatura existente, segundo uma classificação proposta. A classificação, quanto ao tipo de FMS, é a seguinte:

- Número de tipos de peças:
  - Dedicado (no máximo 15)
  - Intermediário (entre 16 e 99)
  - Random (mais que 100)
- Padrão de fluxo predominante:
  - Unidirecional
  - Multidirecional
- Padrão de demanda:
  - Periódica
  - Contínua

e a classificação estende-se ao esquema de programação horária:

- Problemas de programação horária tratados:
  - Sequenciamento de entrada
  - Programação horária detalhada
- Número de decisões no tempo
  - Preditivo
  - Tempo real
- Características consideradas:
  - Quebra de máquinas
  - Capacidade de manuseio de materiais
  - Capacidade de magazine de ferramentas
  - Capacidade de dispositivos de fixação

Capacidade de armazenamento no sistema

Flexibilidade de rotas

Ele considera que, no futuro, predominará o FMS multidirecional, intermediário ou randômico com demanda periódica para o qual são relevantes tanto o seqüenciamento de entrada como a programação horária detalhada e onde tanto o esquema prévio quanto o esquema em tempo real são usados. Ele argumenta que o primeiro esquema levará à obtenção de vantagens diretamente proporcionais ao aumento da flexibilidade de rotas.

Wilson (1992) considera que a PI (Programação Inteira) é uma técnica que pode ser usada para resolver problemas de carregamento. A formulação matemática é apresentada e resultados obtidos com o auxílio de um pacote de software comercialmente disponível (SCICONIC/VM) de Programação Matemática são parcialmente apresentados. Uma heurística também é apresentada e avaliada, tendo uma implementação direta. Os resultados basicamente indicam em qual máquina foi alocada uma operação, mas não indicam a seqüência de atendimento das operações alocadas a cada máquina.

Taboun e Ulger (1992) ressaltaram que a tecnologia de FMS requer grande investimento de tempo e dinheiro, enquanto fornece um alto grau de flexibilidade às células de manufatura e propõem um modelo multi-objetivo para o problema de alocação das operações. Entretanto, assim como Bretthauer e Venkataramanam (1990), não trata da programação horária simultaneamente, o que levaria a um melhor aproveitamento da flexibilidade referida.

Liang e Taboun (1992) destacaram que a máquina gargalo em um período pode não ser o gargalo em nenhum outro período. O modelo apresentado para o problema de seleção e alocação de peças em um FMS com "layout" celular (um conjunto de máquinas diferentes alocadas fisicamente juntas para processar um conjunto de peças semelhantes) procura selecionar um conjunto de peças compatíveis entre si e com o "layout" celular e alocar as peças em células adequadas de forma a maximizar o lucro total.

A solução de um problema exemplo mostra que, dentre 10 peças, 5 foram selecionadas e mostra como cada par (peça, operação) foi alocado às 3 células consideradas. No entanto, não é apresentada a programação horária que precisará ser obtida a seguir. Aliás, a tabela de disponibilidade das máquinas em horas não é colocada em uma escala de tempo.

Souza e outros (1992) analisam uma tecnologia de informação para integração entre Planejamento e Controle da Produção conhecida como Leitstand. Seus componentes são:

- Interface gráfica, que possibilita representação visual, do despacho das ordens de fabricação.
- Editor de despacho, que possibilita ao operador humano manipular o despacho dos recursos de produção.
- Módulo de avaliação, que permite a análise das características de desempenho de cada programa de produção.
- Módulo de geração automática, que fornece automaticamente uma possível solução de partida para o usuário.
- Gerenciador de Base de Dados para manipulação da base de dados interna e para comunicação com o ambiente, das informações necessárias do Planejamento e Controle da Produção, dados de engenharia (tempos de processamento, set-up, roteamento) e dados de chão-de-fábrica (calendário, estado dos recursos).

Em uma primeira análise, o Leitstand assemelha-se a um gráfico de Gantt eletrônico, porém suas possibilidades são acrescidas da capacidade de manipulação e avaliação dos programas, sendo mais adequado conceituá-lo como um sistema de suporte à decisão no controle da produção.

Os programas de produção podem ser seqüenciados de forma manual ou automática. A manipulação manual de um programa é executada com o uso do mouse, ativando uma determinada tarefa e deslocando-a no tempo e/ou máquina. A forma automática consiste da escolha de uma regra de seqüenciamento.

Feito este levantamento bibliográfico, podemos observar que o problema da Programação Horária de Peças em uma FMC tem sido muito pouco estudado.

## **CAPÍTULO 2**

### **PROGRAMAÇÃO HORÁRIA DE PEÇAS EM UMA FMC E O CONCEITO DE GARGALO DINÂMICO**

## 2.1. PROBLEMA DA PROGRAMAÇÃO HORÁRIA DE PEÇAS EM UMA FMC

O problema da Programação Horária de peças em uma FMC com um robô responsável pelo transporte das peças desde o "buffer" de entrada até o "buffer" de saída, passando pelas máquinas que as processarão, de acordo com suas rotas, e pelo "buffer" da célula onde aguardam pela liberação das máquinas, consiste em determinar os horários de início e fim de cada transporte, pelo robô, e de cada processamento, pelas máquinas, de forma sincronizada.

Em uma FMC, devido à troca automática de ferramentas, é desnecessário considerar o tempo de "set-up" entre operações consecutivas, como já colocado por Stecke (1986). Desta forma, os tempos de transporte são comparáveis aos tempos de processamento, como já observado por Ro e outros (1990), tornando necessária a referida sincronização.

Uma boa solução para este problema pode ser a que minimiza o tempo total para processamento das peças ("makespan") acrescido do somatório dos horários de início dos transportes, pois assim os recursos da FMC estarão liberados mais cedo.

Este problema localiza-se no nível 3 da hierarquia apresentada por Kalkunte e outros (1986), podendo estender-se ao nível 4, já que o mesmo problema se repete no nível 4 com horizontes menores. As peças que serão processadas na FMC são determinadas no nível 2, da mesma forma que a alocação de ferramentas necessárias.

O robô aqui considerado movimenta-se na célula com uma velocidade média conhecida, tendo graus de liberdade suficientes para carregar e descarregar peças em uma máquina.

As máquinas aqui consideradas não são paralelas e o processamento de uma peça em uma máquina não pode ser interrompido. Ou seja, cada máquina, embora capaz de executar uma variedade de processamentos, pode executar somente um de cada vez.

O fluxo de processamento na FMC é, inicialmente, considerado unidirecional, como em um "flow-shop", depois é considerado multidirecional, como em um "job-shop", para tornar o problema mais geral e, finalmente, são consideradas as máquinas alternativas. Esta última consideração é que diferencia um FMS de um "job-shop", como observado por Chang e outros (1989).

Os "buffers" considerados são os seguintes: de entrada, onde ficam as peças esperando pelo processamento inicial, o da FMC, onde ficam as peças esperando para serem atendidas pelas máquinas e/ou pelo robô, e o da saída, onde ficam as peças já processadas.

Os lotes de peças são unitários para maior flexibilidade. Idealmente o tamanho de lote em FMS é um (Stecke, 1986). Não é adequado, na nossa opinião, trabalhar com taxa de produção, ou seja, com um número de peças de um tipo a serem produzidas por unidade de tempo, em um ambiente produtivo que se caracteriza pela flexibilidade, ou seja, capaz de produzir uma grande variedade de tipos de peças simultaneamente.

A seqüência de operações para processamento de cada peça é conhecida, assim como as máquinas que podem processar cada operação. É preciso ressaltar que, ao resolver a programação horária usando máquinas alternativas, resolve-se também o carregamento. Esta abordagem é mais flexível que a de Stecke (1986).

Os tempos de processamento de cada operação de cada peça em cada máquina são conhecidos, assim como os tempos de transporte entre máquinas e "buffers" para cada peça. Na era dos CNC, que executarão programas de controle numérico e, portanto, com tempos operacionais previamente determinados, é importante utilizar as informações disponíveis.

Não consideramos limite de capacidade para os "buffers".

A Figura 2.1, a seguir, ilustra a célula considerada neste trabalho.

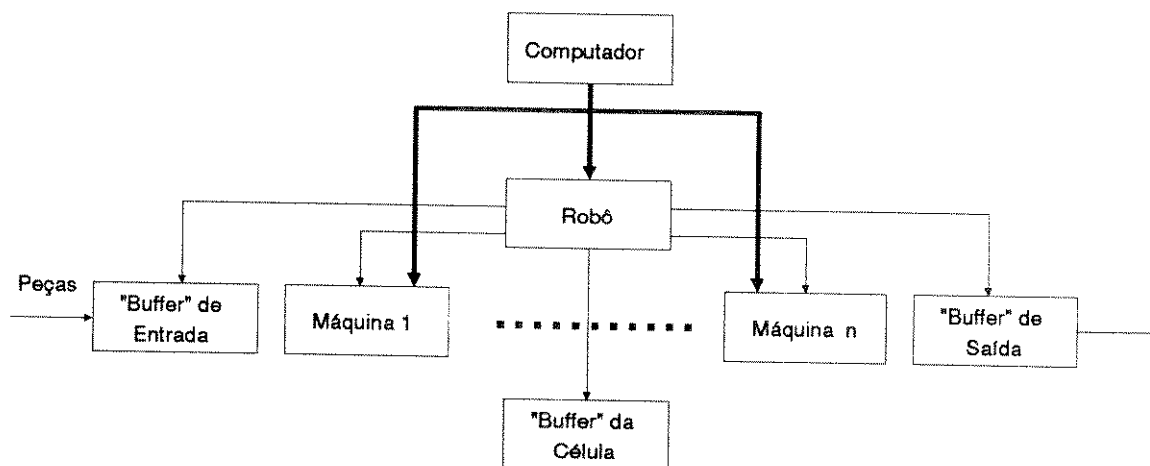


FIGURA 2.1 - Uma Célula Flexível de Manufatura

## 2.2. CONCEITO DE GARGALO DINÂMICO

O conceito clássico de máquina gargalo é o da máquina que tem a maior carga de processamento do chão-de-fábrica. Só que este conceito é adequado a um "mix" de produção fixo. Como foi exaustivamente abordado no capítulo anterior, a flexibilidade da



manufatura é uma característica chave para a competitividade e o "mix" de produção, na maioria dos casos, é variável. Desta forma um novo conceito de gargalo vem substituir o clássico, não existindo mais "a máquina gargalo", pois o gargalo vai variar de acordo com a variação do "mix", como já foi abordado por Liang e Taboun (1992).

Além disso, a máquina gargalo deve ser identificada, não por sua carga absoluta, mas sim por sua carga relativa, pois no caso mais geral de fluxo multidirecional, a máquina gargalo pode não ser a que tem a maior carga absoluta, mas sim uma que, mesmo tendo carga menor, é requisitada por mais peças em intervalos de tempo sobrepostos.

A identificação deste gargalo dinâmico se dá através do cálculo das sobreposições das tarefas, do robô e das máquinas, de atendimento às peças.

Bretthauer e Venkataramanan (1990), como já colocado no capítulo anterior, ressaltaram que o conjunto de rotas alternativas para as peças desempenha um papel chave no gerenciamento de FMS.

Este novo conceito de gargalo ressalta a importância de resolver simultaneamente o problema de alocação e de programação horária. Deste modo, consegue-se explorar melhor a potencialidade do uso de máquinas alternativas para diminuir o gargalo. Esta abordagem já é sugerida por Greene e outros (1986), diferindo da de Kusiak (1986) e de Stecke (1986) que propõe a alocação sem analisar as implicações desta alocação na programação horária.

### **2.3. SINCRONIZAÇÃO DE MÁQUINAS E ROBÔ**

Como já foi visto no capítulo 1, em um FMS e, portanto, em uma FMC, os tempos de processamento tornam-se comparáveis aos de transporte e, desta forma, ao gerarmos a programação horária que sincronize máquinas e robô, no caso de ser este o responsável pelo transporte, estamos fornecendo uma programação realizável.

### **2.4. APROVEITAMENTO DAS OCIOSIDADES INTERMEDIÁRIAS PROGRAMADAS**

Na geração da Programação Horária das peças em uma FMC, muitas ociosidades intermediárias são geradas pelo desbalanceamento naturalmente associado ao "mix" considerado.

A análise desta programação pode levar a um aproveitamento destas ociosidades pela programação horária seguinte a ser iniciada durante o período da programação

anterior.

Neste nosso trabalho, vamos utilizar estes procedimentos. Os exemplos de aplicações, que serão apresentados no capítulo 6, mostram a eficiência da nossa decisão.

Esta abordagem não foi encontrada na literatura especializada até o momento.

## **CAPÍTULO 3**

### **MODELO PARA O PROBLEMA DA PROGRAMAÇÃO HORÁRIA DE PEÇAS EM UMA FMC**

### 3.1. MODELO COM FLUXO UNIDIRECIONAL

O problema descrito no capítulo 1, item 1.2, pode ser modelado como um problema de Programação Linear Inteira Mista de forma que a solução ótima, relativamente ao objetivo escolhido, possa ser encontrada.

Neste modelo, consideramos o fluxo como unidirecional e o objetivo de minimizar apenas o somatório dos horários.

Os parâmetros de entrada são os tempos de processamento das peças nas máquinas e os tempos de transporte pelo robô das peças para as máquinas. As máquinas estão ordenadas de acordo com a ordem de processamento das peças nas mesmas. Como já indicamos na descrição do problema, a rota de cada peça pode não conter todas as máquinas que constituem a célula.

As variáveis do problema são os horários de início do transporte pelo robô de cada peça para cada máquina que a processará, sendo, portanto, necessário considerar o "buffer" de saída como uma última máquina. Como o objetivo escolhido é a minimização do somatório destes horários, consideraremos também variáveis binárias de precedência entre as peças, quanto ao atendimento de seus requisitos de transporte pelo robô e quanto ao processamento pelas máquinas. O número de variáveis binárias é o número de quádruplas ((peça, máquina), (peça, máquina)), com peças diferentes entre si e máquinas quaisquer de suas respectivas rotas.

Os horários de início do processamento de cada peça em cada máquina que a processará são obtidos diretamente dos horários de início do transporte pelo robô da referida peça para a referida máquina, adicionando o tempo de transporte considerado.

As restrições do problema estão classificadas em quatro grupos:

1. Restrições que asseguram que o processamento de uma peça em uma máquina não pode iniciar sem ter terminado o processamento na máquina imediatamente anterior a esta na sua rota, e ser transportada desta pelo robô. Existe uma restrição para cada par de máquinas consecutivas da rota de uma peça e para cada peça.
2. Restrições que asseguram que o robô só pode iniciar um transporte depois de liberado do anterior. Existem duas restrições para cada quádrupla ((peça, máquina), (peça, máquina)) com peças diferentes entre si e máquinas, diferentes entre si, da rota das peças.
3. Restrições que asseguram que o robô só pode iniciar o transporte de uma peça para uma máquina depois que esta máquina já terminou de processar a peça que a precede

nesta máquina. Existem duas restrições para cada quádrupla, como no grupo anterior, só que com a mesma máquina.

4. Restrições que asseguram que o robô só pode iniciar o transporte de uma peça para uma máquina depois de transportar a peça que a precede nesta máquina para a máquina seguinte da sua rota. Existem duas restrições para cada quádrupla do grupo anterior.

As restrições dos tipos 2, 3 e 4 são dicotômicas, pois a precedência será determinada na solução do problema.

A função objetivo do problema é minimizar o somatório dos horários de início do transporte de cada peça para cada máquina que a processará.

A notação adotada é a seguinte:

#### Subscritos

$I = \{i \mid i = 1, \dots, n\}$  - conjunto de  $n$  peças consideradas;

$J = \{j \mid j = 1, \dots, m, m+1\}$  - conjunto ordenado de  $m$  máquinas existentes na célula, acrescido de uma última máquina, fictícia, para considerar a saída.

#### Parâmetros de Entrada

$P(i)$  - conjunto ordenado de máquinas nas quais a peça  $i$  é processada;

$P = \{(i, j), i \in I, j \in P(i)\}$  - conjunto dos pares (peça, máquina);

$tp(i, j)$  - tempo de processamento da peça  $i$  na máquina  $j$ ;

$tt(i, j)$  - tempo de transporte, pelo robô, da peça  $i$  para a máquina  $j$ ;

$G$  - constante arbitrariamente grande.

#### Variáveis

$t(i, j)$  - horário de início do transporte, pelo robô, da peça  $i$  para a máquina  $j$ ;

$$\delta((q, s), (i, j)) = \begin{cases} 1 & \text{se o robô transporta a peça } q \text{ para a máquina } s \\ & \text{antes de transportar a peça } i \text{ para a máquina } j; \\ 0 & \text{em caso contrário.} \end{cases}$$

## Função Objetivo

$$F = \sum_{(i,j) \in P} t(i,j)$$

O modelo proposto por Teixeira e outros (1987), é o seguinte:

$$\text{Min } \sum_{(i,j) \in P} t(i,j)$$

$$\text{s.a. } t(i,j) - t(i,s) \geq tt(i,s) + tp(i,s), \\ \text{com } s < j, s, j \in P(i), \forall i \in I$$

$$\begin{cases} t(i,j) - t(q,s) \geq tt(q,s) - [1 - \delta((q,s), (i,j))] \cdot G \\ t(q,s) - t(i,j) \geq tt(i,j) - G \cdot \delta((q,s), (i,j)) \end{cases} \quad (2)$$

para  $i \neq q, s \neq j, s \in P(q)$  e  $j \in P(i), \forall i, q \in I$

$$\begin{cases} t(i,j) - t(q,j) \geq tt(q,j) + tp(q,j) - [1 - \delta((q,j), (i,j))] \cdot G \\ t(q,j) - t(i,j) \geq tt(i,j) + tp(i,j) - G \cdot \delta((q,j), (i,j)) \end{cases} \quad (3)$$

para  $i \neq q, j \in P(i) \cap P(q), \forall i, q \in I$

$$\begin{cases} t(i,j) - t(q,s) \geq tt(q,s) - [1 - \delta((q,j), (i,j))] \cdot G \\ t(q,j) - t(i,p) \geq tt(i,p) - G \cdot \delta((q,j), (i,j)) \end{cases} \quad (4)$$

com  $j < p, j, p \in P(i)$  e  $j < s, j, s \in P(q), \forall i, q \in I$

A dimensão máxima deste modelo é dada a seguir:

- Número de variáveis reais:  $NVR = n \cdot m$
- Número de variáveis inteiras:  $NVI = C_{NVR}^2 - n C_{m-1}^2$
- Número de restrições do tipo (1) é  $n \cdot (m-1)$
- Número de restrições dos tipos (2) e (3) é  $2 \cdot NVI$ .
- Número de restrições do tipo (4) é  $2 \cdot NVI$ .

### 3.2. MODELO COM FLUXO MULTIDIRECIONAL

Neste modelo consideraremos o fluxo como multidirecional e o objetivo de minimizar o clássico "makespan" acrescido do somatório dos horários de início do transporte pelo robô de cada peça para cada operação para que, além dos recursos serem utilizados o mais cedo possível, a própria FMC seja liberada o mais cedo possível.

Pela característica multidirecional consideraremos pares de (peça, operação), em lugar de pares de (peça, máquina) e associaremos a cada par (peça, operação) a máquina que o processará. Além disso, consideraremos um buffer dentro da célula para onde as peças são transportadas para liberarem as máquinas.

As restrições do problema estão classificadas em quatro grupos:

1. Restrições que definem o "makespan". Existe uma restrição para cada peça.
2. Restrições que asseguram que a peça não pode ter o processamento da operação seguinte iniciado sem ter terminado o processamento da operação imediatamente anterior a esta na sua rota e ser transportada desta pelo robô. Existe uma restrição para cada par de operações consecutivas para cada peça.
3. Restrições que asseguram que o robô só pode iniciar um transporte depois de liberado do anterior. Existem duas restrições para cada quádrupla ((peça, operação), (peça, operação)) com peças diferentes entre si e operações das peças processadas em máquinas diferentes entre si.
4. Restrições que asseguram que o robô só pode iniciar um transporte de uma peça para a máquina que irá processar a operação depois de ter transportado a peça precedente nesta máquina para o buffer da célula. Existem duas restrições para cada quádrupla ((peça, operação), (peça, operação)) com peças diferentes entre si e operações das peças processadas na mesma máquina.

A notação adotada é a seguinte:

#### Subscritos

$I = \{i \mid i = 1, \dots, n\}$  - conjunto de  $n$  peças consideradas;

$J = \{j \mid j = 1, \dots, l\}$  - conjunto de operações;

$K = \{k \mid k = 1, \dots, m, m+1\}$  - conjunto de máquinas da célula, acrescido de uma fictícia representando o buffer.

### Parâmetros de entrada

$P = \{(i,j), i \in I, j \in J\}$  - conjunto dos pares (peça  $i$ , operação  $j$ );

$NO(i)$  - número de operações da peça  $i$ , duplicado para atender os transportes para o buffer;

$M(i,j)$  - máquina que processa a operação  $j$  da peça  $i$ ;

$tp(i,j)$  - tempo de processamento da operação  $j$  da peça  $i$ ;

$tt(i,j)$  - tempo de transporte, pelo robô, da peça  $i$  para a operação  $j$ ;

$G$  - constante arbitrariamente grande.

### Variáveis

$E$  - makespan;

$t(i,j)$  - horário de início do transporte, pelo robô, da peça  $i$  para a operação  $j$ ;

$$\delta((i,j), (r,s)) = \begin{cases} 1 & \text{se o robô transporta a peça } i \text{ para operação } j \text{ antes} \\ & \text{de transportar a peça } r \text{ para a operação } s; \\ 0 & \text{em caso contrário.} \end{cases}$$

### Função Objetivo

$$F = E + \sum_{(i,j) \in P} t(i,j)$$

O modelo proposto por Teixeira e outros (1992), é o seguinte:

$$\text{Min } E + \sum_{(i,j) \in P} t(i,j)$$

$$\text{s.a. } E - t(i, NO(i)) \geq tt(i, NO(i)), \forall i \in I \quad (1)$$

$$t(i, j+1) - t(i, j) \geq tt(i, j) + tp(i, j), \quad (2)$$

$$\forall i \in I \text{ e } j = 1, \dots, NO(i)-1$$



$$\begin{cases} t(r,s)-t(i,j) \geq tt(i,j)-[1-\delta((i,j),(r,s))].G \\ t(i,j)-t(r,s) \geq tt(r,s)-G.\delta((i,j),(r,s)) \end{cases} \quad (3)$$

para  $i \neq r$  e  $M(i,j) \neq M(r,s)$ ,  $\forall (i,j) \in P$

$$\begin{cases} t(r,s)-t(i,j+1) \geq tt(i,j+1)-[1-\delta((i,j),(r,s))].G \\ t(i,j)-t(r,s+1) \geq tt(r,s+1)-G.\delta((i,j),(r,s)) \end{cases} \quad (4)$$

para  $i \neq r$  e  $M(i,j) = M(r,s)$ ,  $\forall (i,j) \in P$

A dimensão máxima deste modelo é dada a seguir:

- Número de variáveis reais:  $NVR = \sum_{i=1}^n NO(i) + 1$
- Número de variáveis inteiras:  $NVI = C_{NVR}^2 - \sum_{i=1}^n C_{NO(i)}^2$
- Número de restrições do tipo (1) é  $n$
- Número de restrições do tipo (2) é  $\sum_{i=1}^n [NO(i)-1]$
- Número de restrições dos tipos (3) e (4) é  $2NVI$ .

### 3.3. UTILIZAÇÃO DO CONCEITO DE GARGALO DINÂMICO

Ambos modelos têm uma característica interessante: quando relaxamos as restrições que contêm as variáveis binárias, que definem as relações de precedência entre as peças relativamente ao robô e às máquinas, obtemos uma solução que identifica o gargalo dinâmico definido no capítulo 2.

Como as sobreposições que identificam o gargalo dinâmico precisam ser resolvidas para obter-se uma programação horária factível, ou seja, realizável, as restrições relaxadas precisam ser ativadas. De qualquer forma, esta característica dos modelos deve ser sempre aproveitada para uma análise dos gargalos de uma FMC.

Os modelos apresentados não permitem uma avaliação global da utilização de máquinas alternativas para diminuição do gargalo dinâmico, a qual implicaria em uma maior complexidade dos mesmos, mas permitem uma avaliação local da utilização de uma máquina alternativa comparando os resultados obtidos com o modelo (3.1, no caso unidirecional, ou 3.2, no caso multidirecional), inicialmente sem considerar a máquina

alternativa e posteriormente considerando a substituição da máquina gargalo pela máquina alternativa para uma dada operação. Se o gargalo diminuir, adota-se a máquina alternativa para a referida operação.

#### 3.4. COMPARAÇÃO COM OUTRAS MODELAGENS

Como já vimos no capítulo 1, nenhum paradigma de modelagem oferece atualmente, sozinho, uma base para uma teoria unificada de Programação Horária ou para os cálculos dos horários propriamente ditos (Rodammer e White, 1988).

Os dois modelos apresentados são modelos de Programação Matemática (PM) que fornecem a programação horária ótima para um dado conjunto de peças a serem produzidas em uma FMC considerando o robô, responsável pelo transporte das mesmas, e as máquinas simultaneamente. Nenhum outro modelo de PM, com esta consideração, é encontrado na literatura até o momento. Como vimos no capítulo 2, esta consideração leva a uma programação horária realizável.

A simulação, como uma técnica descritiva, pode ser usada, mas não substitui uma técnica prescritiva como a PM que fornece a solução ótima.

## **CAPÍTULO 4**

### **ALGORITMO UTILIZADO**

#### 4.1. PROGRAMAÇÃO INTEIRA E PROGRAMAÇÃO LINEAR INTEIRA MISTA

Geoffrion e Marsten (1972) apresentaram os conceitos básicos dos algoritmos de Programação Inteira: Separação, Relaxação e Sondagem.

##### Separação

Dizemos que o problema (P) é separável em subproblemas  $(P_1), (P_2), \dots, (P_q)$  se as seguintes condições valem:

- (S1) Toda solução factível de (P) é uma solução factível de exatamente um dos subproblemas  $(P_1), (P_2), \dots, (P_q)$ .
- (S2) Uma solução factível de qualquer dos subproblemas  $(P_1), (P_2), \dots, (P_q)$  é uma solução factível de (P).

Sob estas condições, sendo  $F(P)$  a região factível de (P),  $F(P_1), F(P_2), \dots, F(P_q)$  é uma partição de  $F(P)$ . Os subproblemas  $(P_1), (P_2), \dots, (P_q)$  são chamados descendentes de (P). Criar descendentes dos descendentes de (P) equivale a refinar a partição de  $F(P)$ .

Uma estratégia rudimentar de separação é descrita a seguir. Primeiro faça um esforço razoável para resolver (P). Se este esforço não tiver sucesso, separe (P) em dois ou mais subproblemas. Extraia um subproblema desta lista, problema candidato atual (CP), e tente resolvê-lo. Se ele puder ser resolvido com uma quantidade razoável de esforço, volte à lista de candidato e extraia um novo problema candidato. Em caso contrário, separe (CP) e adicione seus descendentes à lista de candidatos. Continue desta forma até que a lista de candidato seja exaurida. Sendo solução incumbente atual a melhor solução encontrada, até o momento, ao resolver um problema candidato qualquer, então a solução incumbente final é obviamente a solução ótima de (P). Se todos problemas candidatos são inactíveis, então assim o é (P). O término de tal abordagem é assegurado se  $F(P)$  é finito.

A mais popular forma de separar um problema de Programação Inteira é por meio de restrições opostas sobre uma variável inteira. Por exemplo se  $x_4$  é uma variável binária então (P) pode ser separado em dois subproblemas pelas restrições mutuamente exclusivas  $x_4 = 0$  e  $x_4 = 1$ .

Algumas estruturas especiais do problema podem ser exploradas para obter técnicas de separação mais eficientes.

## Relaxação

Qualquer problema de otimização (P) pode ser relaxado, por omitir restrições ou condições de integralidade de variáveis, resultando em um novo problema (Pn). O único requerimento para (Pn) ser uma relaxação válida de (P) é  $F(P) \subseteq F(Pn)$  o que, no problema de minimização, implica em:

- (R1) Se (Pn) não tem solução factível, então (P) também não tem.
- (R2) O valor mínimo de (P) não é menor que o de (Pn).
- (R3) Se uma solução ótima de (Pn) é factível em (P), então é a solução ótima de (P).

A relaxação mais polular para Programação Linear Inteira é relaxar a restrição de integralidade.

## Sondagem

Seja (CP) um problema candidato qualquer que surge ao se tentar resolver (P). Ao lidar com (CP) procura-se determinar se a região factível  $F(CP)$  pode conter a solução ótima de (P) e, neste caso, encontrá-la. Se for possível deduzir que  $F(CP)$  não contém uma solução factível melhor que a incumbente, então pode-se retirar (CP) de considerações futuras. Neste caso, diz-se que (CP) foi sondado. Ou, se a solução ótima de (CP) for encontrada, também diz-se que (CP) foi sondado.

É útil distinguir três tipos gerais de sondagem, todas baseadas em relaxação. Seja (CPn) uma particular relaxação de (CP),  $v(CPn)$  o valor ótimo do (CPn) e  $z^*$  o valor da incumbente. Se estivermos tratando um problema de minimização,  $v(P) \geq z^*$ . Então:

- 1) Se (CPn) não tem solução factível, então  $F(CP) = \emptyset$ , e não contém a solução ótima de (P), de forma que (CP) está sondado.
- 2) Se  $v(CP_n) \geq z^*$ , então (CP) foi sondado.
- 3) Se a solução ótima de (CPn) tiver sido encontrada e for factível para (CP), então é a solução ótima de (CP) e (CP) foi sondado. Sendo factível em (P), então se  $v(CP) \leq z^*$ ,  $v(CP)$  é a nova incumbente.

Entre os vários algoritmos existentes, há uma considerável variação nos tipos de análises utilizadas para implementar estes critérios.

Geoffrion e Marsten (1972) também selecionaram algoritmos agrupando-os em:

- Enumeração
- Decomposição de Benders
- Planos de Corte
- Teoria de Grupo.

Além disso, apresentaram um fluxograma geral dos algoritmos de Programação Linear Inteira Mista e observaram que todos os códigos de propósitos gerais usam, com sucesso, Programação Linear para resolver (CPn) e permitem várias opções para selecionar uma variável de separação.

Eles consideram ainda que o advento dos computadores altamente paralelos como o ILLIAC IV, com suas habilidades de processar vários problemas candidatos simultaneamente, deverá ser acompanhada de novas estratégias de enumeração.

O algoritmo "Branch and Bound", que Geoffrion e Marsten (1972) classificam como enumerativo e analisam suas características específicas em relação ao fluxograma geral dos algoritmos de Programação Linear Inteira Mista PLIM, é o que utilizamos para resolver o problema estudado. A principal razão desta escolha é que, sendo o problema modelado como um problema de PLIM com variáveis binárias de decisão de precedência entre as peças, em relação ao robô e às máquinas, este algoritmo pode ser acelerado pela fixação de várias variáveis binárias simultaneamente.

Além disso, o sucesso obtido por Stecke e Berrada (1986) na solução do problema de balanceamento de cargas nas máquinas em um FMS e o trabalho de Kumar e Kamal (1984), mostrando que diversos procedimentos de busca de Inteligência Artificial como  $A^*$ ,  $AO^*$ ,  $SSS^*$ ,  $B^*$  e alpha-beta são casos especiais da formulação do "Branch and Bound", pesaram na nossa escolha.

Cabe ressaltar ainda que a eficiência do algoritmo "Branch and Bound" depende bastante da utilização da estrutura do problema (Hughes e Grawiog, 1973), assim como da ordem de fixação das variáveis (Minoux, 1983) e da ordenação das soluções a serem examinadas (Ichino e Sklansky, 1984). Dentro deste contexto, vamos apresentar no item a seguir, as características do algoritmo implementado para após discutirmos as melhorias introduzidas visando aceleração.

## 4.2. CARACTERÍSTICAS DO "BRANCH AND BOUND" IMPLEMENTADO

O "Branch and Bound" implementado segue os seguintes passos:

**PASSO 1:** Resolva o problema relaxando a restrição de integralidade, utilizando Programação Linear (PL). Se na solução ótima a restrição de integralidade relaxada está satisfeita, então esta é a solução ótima do problema e o algoritmo termina. Caso contrário inicialize a matriz de armazenamento de soluções com esta solução na posição da incumbente (melhor valor da função objetivo).

**PASSO 2:** Escolha uma variável binária na incumbente entre as variáveis que violam a restrição de integralidade.

**PASSO 3:** Fixe a variável binária em 0 e resolva o problema, utilizando o algoritmo simplex de PL.

Se esta solução satisfizer a restrição de integralidade das variáveis binárias, então abandone todas soluções armazenadas cujos valores da função objetivo sejam maiores do que o desta solução.

**PASSO 4:** Repita o passo 3, fixando agora a variável em 1.

**PASSO 5:** Retire da matriz a solução incumbente já explorada. Analise a nova solução incumbente. Se esta solução satisfizer a restrição de integralidade das variáveis binárias, então esta é a solução ótima do problema e o algoritmo termina. Caso contrário volte ao passo 2.

No passo 2 vários critérios de escolha da variável binária podem ser usados. Adotamos dois critérios diferentes que serão detalhados na descrição das versões implementadas.

Rotinas de pré-processamento foram implementadas, de forma que, a partir dos dados de entrada, sejam criadas, automaticamente, as variáveis e restrições do problema, e formatadas para o pacote de PL utilizado.

Uma rotina de pós-processamento também foi implementada para decodificar a solução do PLIM e gerar o gráfico de Gantt correspondente.

A seguir temos a figura 4.1 com o diagrama de blocos do "Branch and Bound" implementado.

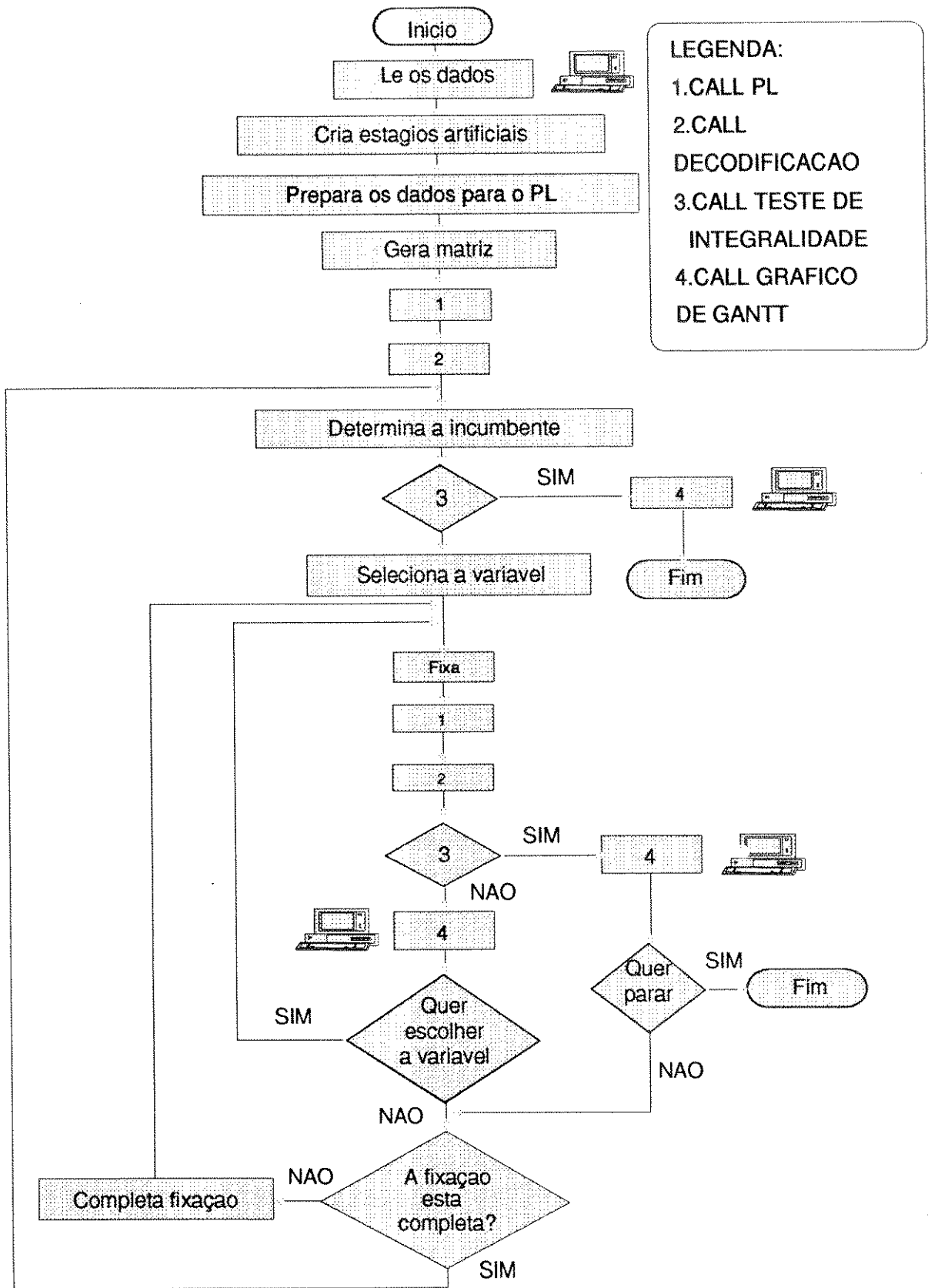


FIGURA 4.1 - Diagrama de Blocos do "Branch and Bound" implementado



### 4.3. CARACTERÍSTICAS DAS VERSÕES IMPLEMENTADAS

Na primeira versão implementada é escolhida, no passo 2, a variável binária que estiver mais infactível, ou seja, com um valor mais próximo de 0.5.

A versão II caracteriza-se por, no passo 2, escolher a variável binária que é responsável pela maior infactibilidade nas restrições. A interpretação desta infactibilidade em termos da FMC é de que esta é a variável binária responsável pelo maior conflito entre as peças. Se o gráfico de Gantt desta solução fosse gerado, este conflito chamaria a atenção, mostrando máquinas e robô realizando várias tarefas simultaneamente e, obviamente, infactivelmente.

### 4.4. CARACTERÍSTICAS DA VERSÃO INTERATIVA

Uma versão interativa, com base nas versões I ou II, foi desenvolvida para permitir as seguintes formas de utilização:

- Editar um arquivo de dados da FMC;
- Editar o Gráfico de Gantt de soluções intermediárias;
- Editar o Gráfico de Gantt de soluções factíveis;
- Interromper o algoritmo após a obtenção de uma solução factível;
- Escolher uma variável binária a ser fixada com base no Gráfico de Gantt da solução incumbente.

### 4.5. ACELERAÇÕES IMPLEMENTADAS

As acelerações, implementadas em ambas as versões apresentadas no capítulo anterior, são obtidas pela fixação de valores para várias variáveis binárias simultaneamente, explorando as características específicas de modelagem que o algoritmo escolhido permite.

Os resultados obtidos com estas acelerações foram muito significativos, reduzindo o tempo de CPU em cerca de 50% sem perder a solução ótima.

Estas acelerações são de dois tipos. O 1º tipo amplia a fixação ao longo das rotas das peças, enquanto que a do 2º tipo amplia a fixação pela aplicação da propriedade de transitividade das desigualdades das variáveis binárias.

As do 1º tipo só não se aplicam no caso de uma única máquina e as do 2º tipo só não se aplicam no caso de somente duas peças, sendo justamente estes os casos resolvidos pela versão sem estas acelerações, de forma plenamente satisfatória.

A seguir, faremos uma descrição detalhada dos dois tipos de acelerações para o modelo unidirecional. As modificações para o modelo multidirecional são mínimas.

### ACELERAÇÕES DO 1º TIPO

Quando fixamos  $\delta((q,s),(i,j))$  em 1, o robô transporta a peça  $q$  para a máquina  $s$  antes de transportar a peça  $i$  para a máquina  $j$  e, portanto, transporta a peça  $q$  para as máquinas  $k$  anteriores a  $s$ , também antes de transportar a peça  $i$  para a máquina  $j$ . Da mesma forma transporta a peça  $q$  para a máquina  $s$  também antes de transportar a peça  $i$  para as máquinas  $k$  posteriores a  $j$ .

Ou seja,

$$\delta((q,s),(i,j)) = 1 \Rightarrow \begin{cases} \delta((q,k),(i,j)) = 1 & \text{para } k \in P(q) \text{ com } k < s \\ \delta((q,s),(i,k)) = 1 & \text{para } k \in P(i) \text{ com } k > j \end{cases}$$

Quando fixamos  $\delta((q,s),(i,j))$  em 0, o robô transporta a peça  $q$  para a máquina  $s$  depois de transportar a peça  $i$  para a máquina  $j$  e, portanto, transporta a peça  $q$  para as máquinas  $k$  posteriores a  $s$ , também depois de transportar a peça  $i$  para a máquina  $j$ . Da mesma forma, transporta a peça  $q$  para a máquina  $s$ , também depois de transportar a peça  $i$  para as máquinas  $k$  anteriores a  $j$ .

Ou seja,

$$\delta((q,s),(i,j)) = 0 \Rightarrow \begin{cases} \delta((q,k),(i,j)) = 0 & \text{para } k \in P(q) \text{ com } k > s \\ \delta((q,s),(i,k)) = 0 & \text{para } k \in P(i) \text{ com } k < j \end{cases}$$

Desta forma, após a escolha da variável a ser fixada, é feita esta análise ao longo das rotas das duas peças que a variável binária relaciona, a partir das máquinas relacionadas.

### TRANSITIVIDADE DAS VARIÁVEIS BINÁRIAS

Aplicando a propriedade da transitividade das variáveis binárias, que só tem sentido para mais de duas peças, obtemos as oito propriedades a seguir:

**P1.** Se  $\delta((q,s),(i,j)) = 0$  e  $\delta((p,k),(q,s)) = 0$   
então  $\delta((p,k),(i,j)) = 0$ .

**Prova:** Sendo  $\delta((q,s),(i,j)) = 0$  então  $t(q,s) \geq t(i,j) + tt(i,j)$   
Sendo  $\delta((p,k),(q,s)) = 0$  então  $t(p,k) \geq t(q,s) + tt(q,s)$   
Logo  $t(p,k) \geq t(i,j) + tt(i,j)$   
e  $\delta((p,k),(i,j)) = 0$  c.q.d.

**P2.** Se  $\delta((q,s),(i,j)) = 0$  e  $\delta((q,s),(p,k)) = 1$   
então  $\delta((i,j),(p,k)) = 1$

**Prova:** Sendo  $\delta((q,s),(i,j)) = 0$  então  $t(q,s) \geq t(i,j) + tt(i,j)$   
Sendo  $\delta((q,s),(p,k)) = 1$  então  $t(p,k) \geq t(q,s) + tt(q,s)$   
Logo  $t(p,k) \geq t(i,j) + tt(i,j)$   
e  $\delta((i,j),(p,k)) = 1$  c.q.d.

**P3.** Se  $\delta((q,s),(i,j)) = 0$  e  $\delta((i,j),(p,k)) = 0$   
então  $\delta((q,s),(p,k)) = 0$

**Prova:** Sendo  $\delta((q,s),(i,j)) = 0$  então  $t(q,s) \geq t(i,j) + tt(i,j)$   
Sendo  $\delta((i,j),(p,k)) = 0$  então  $t(i,j) \geq t(p,k) + tt(p,k)$   
Logo  $t(q,s) \geq t(p,k) + tt(p,k)$   
e  $\delta((q,s),(p,k)) = 0$  c.q.d.

**P4.** Se  $\delta((q,s),(i,j)) = 0$  e  $\delta((p,k),(i,j)) = 1$   
então  $\delta((p,k),(q,s)) = 1$ .

**Prova:** Sendo  $\delta((q,s),(i,j)) = 0$  então  $t(q,s) \geq t(i,j) + tt(i,j)$   
Sendo  $\delta((p,k),(i,j)) = 1$  então  $t(i,j) \geq t(p,k) + tt(p,k)$   
Logo  $t(q,s) \geq t(p,k) + tt(p,k)$   
e  $\delta((q,s),(p,k)) = 0$  c.q.d.

**P5.** Se  $\delta((q,s),(i,j)) = 1$  e  $\delta((p,k),(i,j)) = 0$   
então  $\delta((p,k),(q,s)) = 0$

**Prova:** Sendo  $\delta((q,s),(i,j)) = 1$  então  $t(i,j) \geq t(q,s) + tt(q,s)$   
Sendo  $\delta((p,k),(i,j)) = 0$  então  $t(p,k) \geq t(i,j) + tt(i,j)$   
Logo  $t(p,k) \geq t(q,s) + tt(q,s)$   
e  $\delta((p,k),(q,s)) = 0$  c.q.d.

**P6.** Se  $\delta((q,s),(i,j)) = 1$  e  $\delta((q,s),(p,k)) = 0$   
então  $\delta((i,j),(p,k)) = 0$

**Prova:** Sendo  $\delta((q,s),(i,j)) = 1$  então  $t(i,j) \geq t(q,s) + tt(q,s)$   
Sendo  $\delta((q,s),(p,k)) = 0$  então  $t(q,s) \geq t(p,k) + tt(p,k)$   
Logo  $t(i,j) \geq t(p,k) + tt(p,k)$   
e  $\delta((i,j),(p,k)) = 0$  c.q.d.

**P7.** Se  $\delta((q,s),(i,j)) = 1$  e  $\delta((p,k),(q,s)) = 1$   
então  $\delta((p,k),(i,j)) = 1$

**Prova:** Sendo  $\delta((q,s),(i,j)) = 1$  então  $t(i,j) \geq t(q,s) + tt(q,s)$   
Sendo  $\delta((p,k),(q,s)) = 1$  então  $t(q,s) \geq t(p,k) + tt(p,k)$   
Logo  $t(i,j) \geq t(p,k) + tt(p,k)$   
e  $\delta((p,k),(i,j)) = 1$  c.q.d.

**P8.** Se  $\delta((q,s),(i,j)) = 1$  e  $\delta((i,j),(p,k)) = 1$   
então  $\delta((q,s),(p,k)) = 1$

**Prova:** Sendo  $\delta((q,s),(i,j)) = 1$  então  $t(i,j) \geq t(q,s) + tt(q,s)$   
Sendo  $\delta((i,j),(p,k)) = 1$  então  $t(p,k) \geq t(i,j) + tt(i,j)$   
Logo  $t(p,k) \geq t(q,s) + tt(q,s)$   
e  $\delta((q,s),(p,k)) = 1$  c.q.d.

## ACELERAÇÕES DO 2º TIPO

Quando fixamos  $\delta((q,s),(i,j))$  em 0, então fazemos as seguintes verificações:

1. Se  $\delta((p,k),(q,s))$  está fixada em 0, então aplicamos a propriedade P1 fixando  $\delta((p,k),(i,j))$  em 0.
2. Se  $\delta((q,s),(p,k))$  está fixada em 1, então aplicamos a propriedade P2 fixando  $\delta((i,j),(p,k))$  em 1.
3. Se  $\delta((i,j),(p,k))$  está fixada em 0, então aplicamos a propriedade P3 fixando  $\delta((q,s),(p,k))$  em 0.
4. Se  $\delta((p,k),(i,j))$  está fixada em 1, então aplicamos a propriedade P4 fixando  $\delta((p,k),(q,s))$  em 1.

Quando fixamos  $\delta((q,s),(i,j))$  em 1, então fazemos as seguintes verificações:

1. Se  $\delta((p,k),(i,j))$  está fixada em 0, então aplicamos a propriedade P5 fixando  $\delta((p,k),(q,s))$  em 0.
2. Se  $\delta(q,s),(p,k)$  está fixada em 0, então aplicamos a propriedade P6 fixando  $\delta((i,j),(p,k))$  em 0.
3. Se  $\delta((p,k),(q,s))$  está fixada em 1, então aplicamos a propriedade P7 fixando  $\delta((p,k),(i,j))$  em 1.
4. Se  $\delta((i,j),(p,k))$  está fixada em 1, então aplicamos a propriedade P8, fixando  $\delta((q,s),(p,k))$  em 1.

Desta forma, após a escolha da variável a ser fixada, são feitas estas verificações para as variáveis binárias que tenham um dos pares (peça, máquina) em comum com a fixada.

## **CAPÍTULO 5**

### **HEURÍSTICA PROPOSTA**

O bloco "determina programação horária da peça" é o mais complexo, pois irá gerar os horários para todas operações da peça selecionada, sincronizando robô e máquinas e aproveitando as ociosidades intermediárias programadas, como referido no capítulo 2.

O aproveitamento das ociosidades intermediárias programadas bem como a sincronização do robô e máquinas são feitos com o auxílio dos vetores de ordenação de atendimento, pelo robô e pelas máquinas, de cada par (peça, operação). Estes vetores são duplicados para conter não só os horários definitivos, que são obtidos a cada iteração para cada peça selecionada, completando-se só na última iteração, como também para conter os horários provisórios que são gerados inicialmente para todas as peças, sendo adotada a ordem crescente destes horários.

Cada vez que o bloco é ativado são gerados os horários da peça, desde a primeira operação até a última, de duas em duas, para considerar simultaneamente o processamento em uma máquina e o transporte para o buffer. Para gerar estes horários, procura-se um intervalo na programação atual da máquina onde será processada a operação e dois intervalos na programação atual do robô para transportar a peça para a máquina e a seguir para o buffer. Com o auxílio do vetor de ordenação do atendimento da máquina referida, compara-se o intervalo entre cada dois atendimentos consecutivos com o intervalo que serve para transportar a peça para a máquina, processar a referida operação e transportar a peça para o buffer. Simultaneamente, com o auxílio do vetor de ordenação do atendimento do robô, compara-se o intervalo entre cada dois atendimentos consecutivos com os intervalos que servem para transportar a peça para a máquina e transportar a peça para o buffer.

Para ilustrar, consideraremos os parâmetros a seguir:

- TFLIM - horário de término do transporte da peça para o buffer;
- TFUL - horário de término da utilização da máquina com uma peça;
- TIUL - horário de início da utilização da máquina com a peça seguinte;
- TFR1 - horário de fim do transporte, pelo robô, de uma peça;
- TIR1 - horário de início do transporte, pelo robô, da peça seguinte;
- TFR2 - horário de fim do transporte, pelo robô, de uma peça;
- TIR2 - horário de início do transporte, pelo robô, da peça seguinte;
- TT1 - tempo necessário para transportar a peça referida para a máquina;
- TT2 - tempo necessário para transportar a peça referida para o buffer;
- TPNEC - tempo necessário para transportar e processar a peça referida.

Seja  $TDI = \max \{ TFUL, TFLIM, TFR1 \}$  e supondo que os referidos parâmetros satisfaçam as desigualdades abaixo, então já encontramos os intervalos.

$$\begin{cases} TDI + TT1 \leq TIR1 \\ TDI + TPNEC + TT2 \leq TIUL \\ TFR2 \leq TDI + TPNEC \\ TDI + TPNEC + TT2 \leq TIR2 \end{cases}$$

Na Figura 5.2 vemos os intervalos [TFR1,TIR1] e [TFR2,TIR2] encontrados na programação horária definitiva do robô e [TFUL,TIUL] encontrado na programação horária definitiva da máquina.

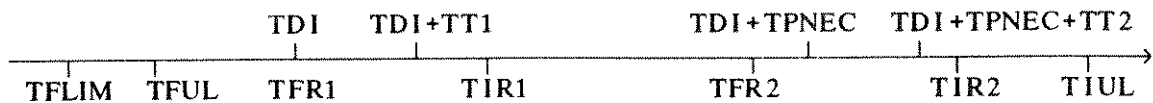


FIGURA 5.2 - Intervalos Encontrados na Máquina e no Robô

Após encontrados os intervalos, os horários da peça referida para a operação na máquina analisada, bem como os horários de transporte desta peça para esta máquina e desta para o buffer, já estão determinados. A seguir, atualizam-se as ordens de atendimento.

A atualização da ordem de atendimento pela máquina é mais simples que a ordem de atendimento pelo robô. Em relação aos horários definitivos, a ordem de atendimento da peça referida na máquina analisada é a ordem seguinte à da peça que terminou de ser processada no horário TFUL. As peças que utilizam a referida máquina a partir de TIUL terão sua ordem de atendimento acrescida de uma unidade. Em relação aos horários provisórios, compara-se o horário de início de processamento desta peça nesta máquina com os demais horários da programação horária provisória desta máquina, encontrando a nova posição na ordem crescente e retirando a antiga posição.

Em relação aos horários definitivos, a ordem de atendimento, pelo robô, para o transporte da peça para a máquina é a ordem seguinte a da peça que terminou de ser transportada no horário TFR1. As peças que são atendidas a partir de TIR1 até TFR2 terão sua ordem de atendimento acrescida de uma unidade. A ordem de atendimento, pelo robô, para o transporte da peça para o buffer é a ordem seguinte a da peça que terminou de ser transportada no horário, TFR2, já acrescida de uma unidade, de forma que as peças que são atendidas a partir de TIR2 terão sua ordem de atendimento acrescidas de duas unidades.

Já em relação aos horários provisórios, comparam-se os horários de início do



transporte desta peça para a máquina referida e para o buffer com os demais horários da programação horária provisória do robô, encontrando as novas posições na ordem crescente e retirando as antigas posições.

Para ilustrar, consideraremos os parâmetros a seguir:

- NP1 - nova posição do primeiro transporte;
- NP2 - nova posição do segundo transporte;
- AP1 - posição antiga do primeiro transporte;
- AP2 - posição antiga do segundo transporte.

E, na Figura 5.3, podemos analisar as seis diferentes relações entre estas posições, para as quais diferentes procedimentos de atualização de ordenação foram criados.

NP1	NP2	AP1	AP2
NP1	AP1	NP2	AP2
NP1	AP1	AP2	NP2
AP1	AP2	NP1	NP2
AP1	NP1	AP2	NP2
AP1	NP1	NP2	AP2

FIGURA 5.3 - Relações entre as Novas e Antigas Posições

Este bloco "determina programação horária da peça" é, como os demais, totalmente modular, podendo utilizar outras heurísticas.

### 5.3. COMPARAÇÃO COM HEURÍSTICAS CLÁSSICAS

Nenhuma das heurísticas apresentadas no capítulo 1 gera a programação horária com sincronização das máquinas e robô e aproveitamento das ociosidades intermediárias, geradas devido a característica multidirecional do fluxo produtivo das peças, bem como com a utilização de máquina alternativa para diminuição do gargalo e, conseqüentemente, do "makespan".

Por exemplo, o método, indicado por Ranky (1986) para a programação horária em FMS, considera cada célula como uma única máquina.

Chang e outros (1989) não utilizam as informações relativas ao sistema de transporte para a geração de programação horária.

Perkins e Kumar (1989) apresentam políticas de programação horária que trabalham com taxas de produção de peças, mas não consideram a sincronização da produção de peças diferentes.

Walter (1990) utiliza uma regra derivada da MWKR para resolver o conflito entre dois conjuntos de peças diferentes que disputam o mesmo centro de trabalho, mas não resolve simultaneamente o conflito pelo recurso de transporte.

Ro e Kim (1990) consideram a programação horária simultânea de jobs e de equipamentos de manuseio de materiais, mas não apresentam as programações horárias geradas.

Sauter (1991) apresenta algoritmos para os problemas de lançamento, roteirização e despacho, mas indica que será necessário reformulá-los para incluir a sincronização de processos.

Ahlwalia e Ji (1991) ilustram sua abordagem em um FMS hipotético, mas não apresentam a programação horária resultante.

Taboun e Ulgar (1992) e Liang e Taboun (1992) tratam da alocação de peças e operações, mas não tratam da programação horária simultaneamente que poderia levar a um aproveitamento da flexibilidade existente em um FMS para alocação às máquinas mais adequadas.

Já a Leitstand, como descrito por Souza e outros (1992), é um sistema de suporte à decisão no controle da produção.

Wilson (1992) apresenta uma heurística cujos resultados basicamente, indicam em qual máquina foi alocada uma operação, mas não indicam a seqüência de atendimento das operações alocadas a cada máquina e muito menos a programação horária.

Desta forma, uma comparação dos trabalhos citados com os resultados que serão apresentados no capítulo seguinte torna-se pouco viável. A comparação que faremos é entre as clássicas regras SPT e LPT e a regra JMIN que apresentamos neste capítulo. Outras regras podem ser colocadas na heurística dentro do bloco "determina peça prioritária" e comparadas entre elas.

## **CAPÍTULO 6**

### **EXEMPLOS E RESULTADOS**

Dos vários exemplos de Programação Horária de Peças em uma FMC, executados no VAX 8530, apresentaremos, com detalhes, aqueles que ilustram:

- a comparação do modelo multidirecional com a heurística;
- a comparação das regras LPT, SPT e JMIN para seleção de peças na heurística;
- a comparação da heurística considerando o robô com a heurística sem considerá-lo;
- a heurística com máquinas alternativas; e
- a heurística com a célula em movimento.

Outros resultados obtidos serão usados para uma análise global com base em índices de desempenho.

Tanto o modelo quanto a heurística geram estágios intermediários após a leitura dos dados de entrada, sendo os tempos de transporte divididos em duas parcelas iguais para atender o transporte para o estágio intermediário e deste para o próximo estágio.

Os resultados são apresentados em gráfico de Gantt onde, no eixo horizontal, temos o tempo em minutos e, no vertical, os recursos utilizados. Estes gráficos foram impressos na Tektronix 4692 Color Graphics Copier.

### 6.1. COMPARAÇÃO DO MODELO MULTIDIRECIONAL COM A HEURÍSTICA

Os dados de um conjunto de 2 peças em 2 máquinas estão na Tabela 6.1.

P E Ç A S	1	2
NÚMERO DE ESTÁGIOS	2	2
MÁQUINA 1 (ESTÁGIO)	1º	2º
TEMPO TRANSPORTE	1,0	1,0
TEMPO DE PROCESSAMENTO	4,0	3,0
MÁQUINA 2 (ESTÁGIO)	2º	1º
TEMPO TRANSPORTE	1,0	1,0
TEMPO PROCESSAMENTO	5,0	6,0

TABELA 6.1 - Dados de Produção de 2 Peças em 2 Máquinas

A solução ótima gerada pelo modelo em 8,06s de CPU após 8 iterações e com "makespan" de 13,5 min é mostrada no gráfico de Gantt, gerado automaticamente, apresentado na Figura 6.1.

A solução gerada pela heurística em 0,50s de CPU após 2 iterações e com "makespan" de 14,0 min é apresentada na Figura 6.2.

Neste caso, portanto, a heurística teve um erro inferior a 4% e o tempo de CPU foi quase 1512% inferior.

O gargalo neste exemplo é a máquina 2. De acordo com a regra JMIN, a peça selecionada é a que começa pelo gargalo e no modelo é este par (peça, operação) que inicia a solução ótima.

Os dados de um exemplo de 3 peças em 3 máquinas estão na Tabela 6.2.

P E Ç A S	1	2	3
NÚMERO DE ESTÁGIOS	2	3	2
MÁQUINA 1 (ESTÁGIO)		2 <sup>o</sup>	1 <sup>o</sup>
TEMPO TRANSPORTE		1,0	1,0
TEMPO DE PROCESSAMENTO		3,0	3,0
MÁQUINA 2 (ESTÁGIO)	1 <sup>o</sup>	3 <sup>o</sup>	
TEMPO TRANSPORTE	1,0	1,0	
TEMPO PROCESSAMENTO	2,0	2,0	
MÁQUINA 3 (ESTÁGIO)	2 <sup>o</sup>	1 <sup>o</sup>	2 <sup>o</sup>
TEMPO TRANSPORTE	1,0	1,0	1,0
TEMPO PROCESSAMENTO	1,0	1,0	4,0

TABELA 6.2 - Dados de Produção de 3 Peças em 3 Máquinas

A solução ótima, gerada pelo modelo em 284,33s de CPU após 67 iterações e com "makespan" de 12,5 min, é mostrada no gráfico de Gantt, apresentado na Figura 6.3.

A solução gerada pela heurística, em 1,04s de CPU após 3 iterações e com "makespan" de 12,5 min, é mostrada no gráfico de Gantt, apresentado na Figura 6.4.

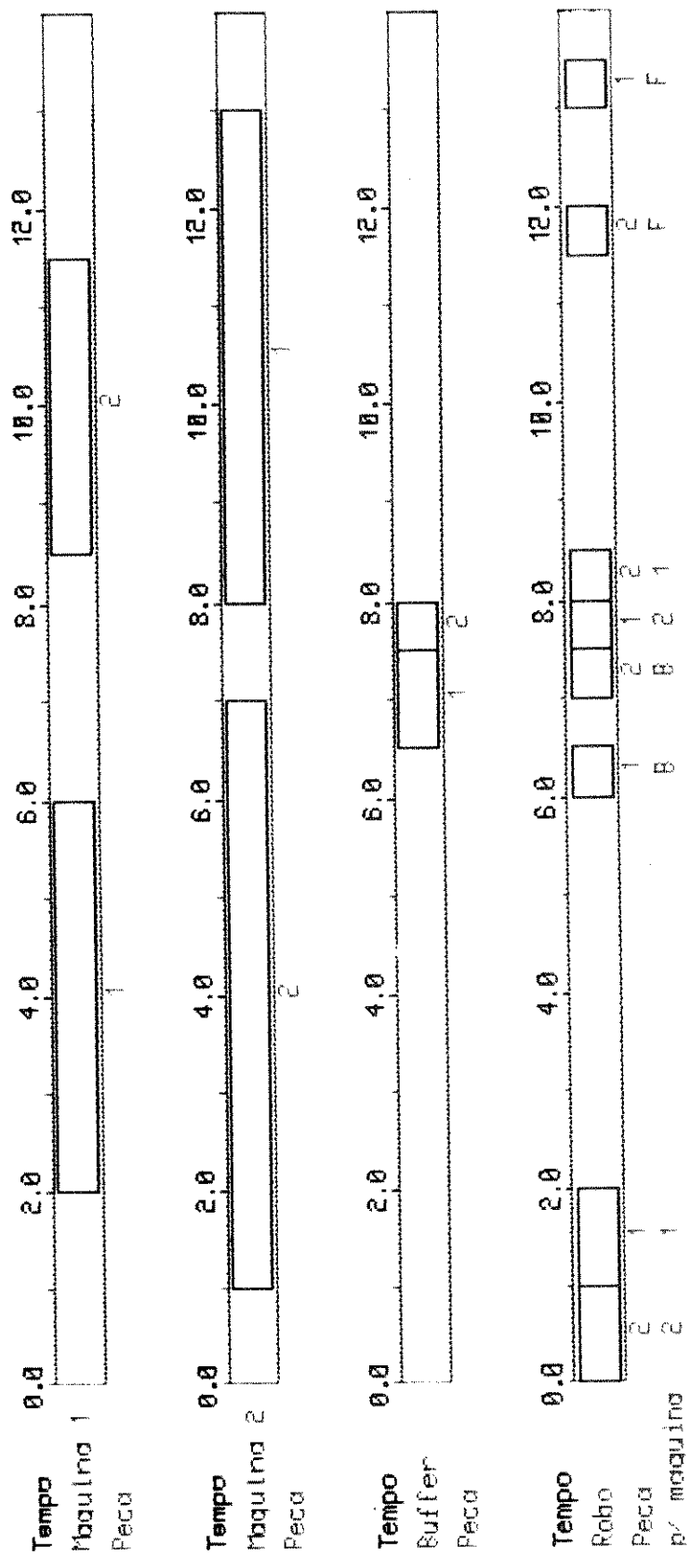


FIGURA 6.1 - Gráfico de Gantt Gerado pelo Modelo para 2 Peças em 2 Máquinas

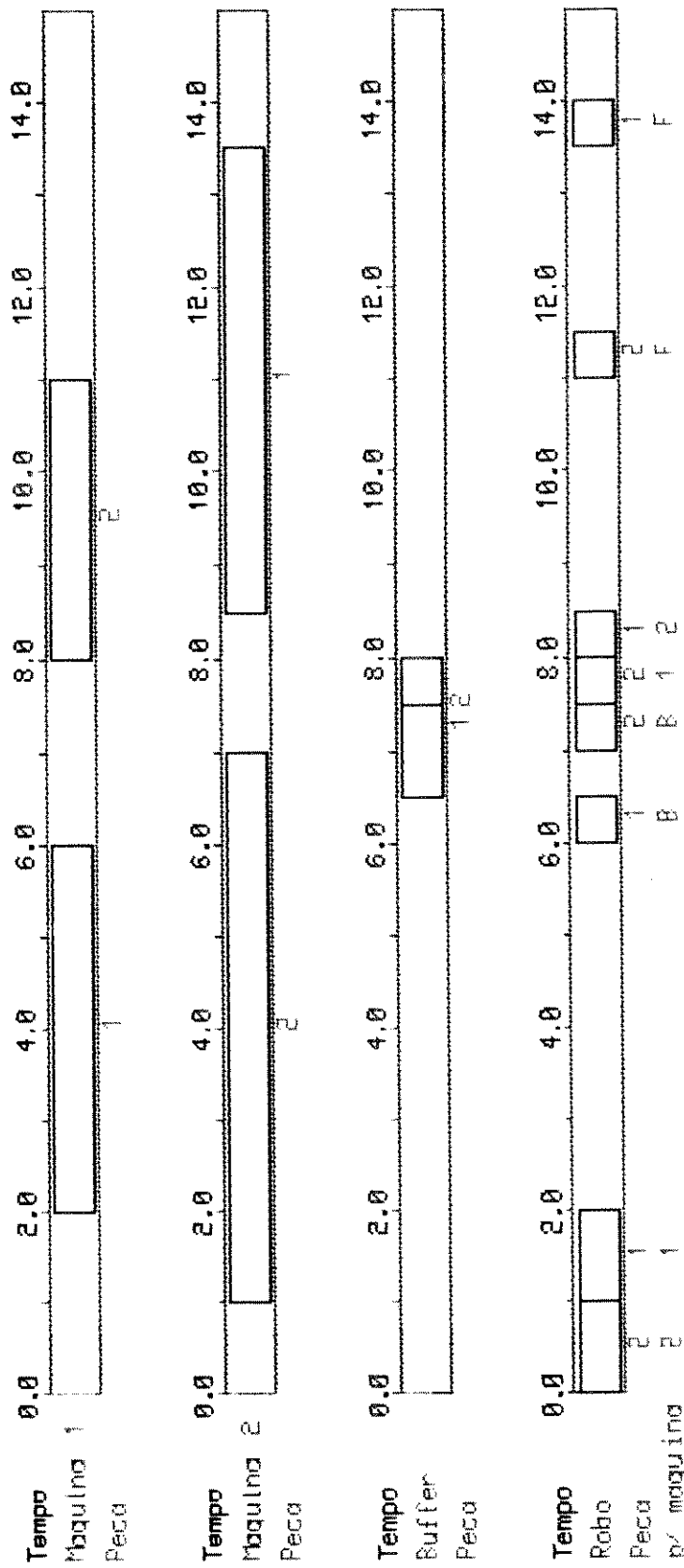


FIGURA 6.2 - Gráfico de Gantt Gerado pela Heurística para 2 Peças em 2 Máquinas

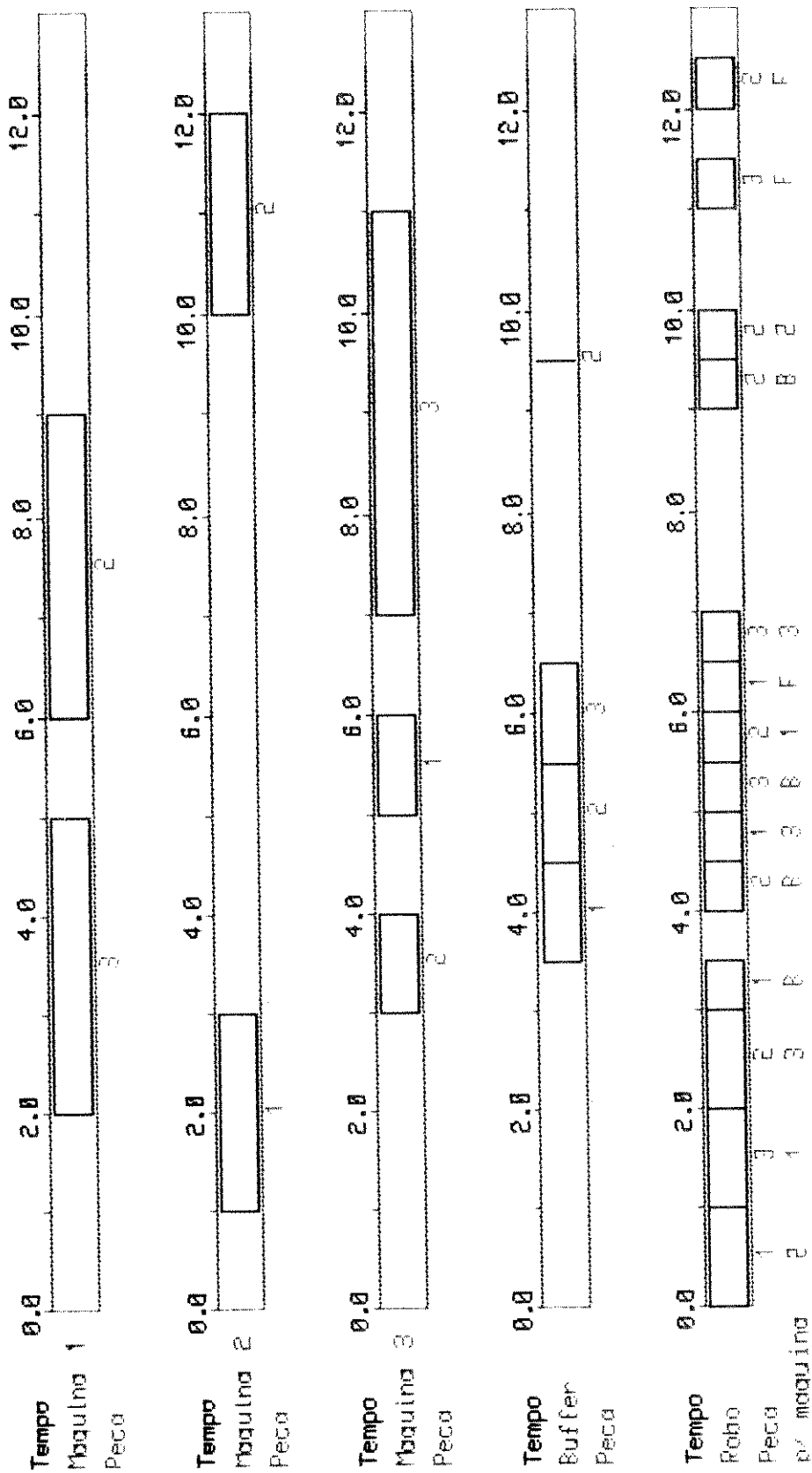


FIGURA 6.3 - Gráfico de Gantt Gerado pelo Modelo para 3 Peças em 3 Máquinas



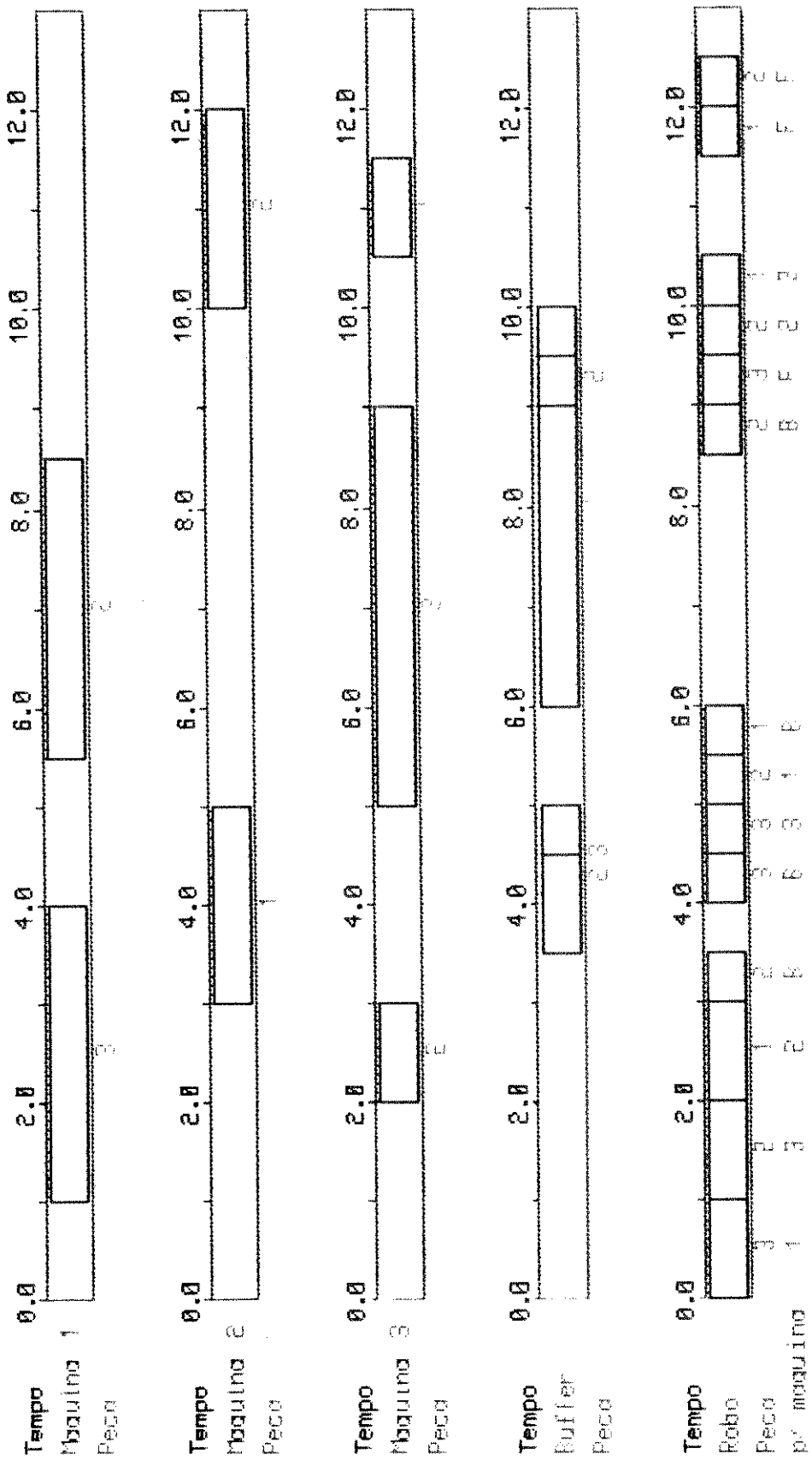


FIGURA 6.4 - Gráfico de Gantt Gerado pela Heurística para 3 Peças em 3 Máquinas

Neste caso, portanto, a heurística forneceu o mesmo valor de "makespan" com um tempo de CPU quase 27.239% inferior.

É interessante observar que as duas soluções ótimas são diferentes.

Devido às limitações de dimensão do PLIM, os exemplos a seguir encontram-se na heurística.

## 6.2. COMPARAÇÃO DAS REGRAS LPT, SPT E JMIN UTILIZANDO A HEURÍSTICA

Os dados que utilizaremos nesta comparação estão na Tabela 6.3.

P E Ç A S	1	2	3	4	5	6	7	8	9	10
NÚMERO DE ESTÁGIOS	1	2	2	1	2	1	2	1	1	1
MÁQUINA 1 (ESTÁGIO)	1º	2º	1º		1º		2º	1º		1º
TEMPO DE TRANSPORTE	1,0	1,0	1,0		1,0		1,0	1,0		1,0
TEMPO PROCESSAMENTO	2,0	3,0	2,0		3,0		3,0	2,0		3,0
MÁQUINA 2 (ESTÁGIO)		1º	2º	1º	2º	1º	1º		1º	
TEMPO TRANSPORTE		1,0	1,0	1,0	1,0	1,0	1,0		1,0	
TEMPO PROCESSAMENTO		3,0	2,0	1,0	1,0	4,0	1,0		5,0	

TABELA 6.3 - Dados de Produção de 10 Peças em 2 Máquinas

Inicialmente, consideraremos as oito primeiras peças da Tabela 6.3.

A solução gerada, utilizando a regra LPT em 3,37s de CPU e com "makespan" 30,5 min, é apresentada na Figura 6.5. A ordem gerada para as peças foi 2, 3, 5, 7, 6, 4, 1, 8.

A solução gerada, utilizando a regra SPT em 3,73s de CPU e com "makespan" 30,5 min, é apresentada na Figura 6.6. Apesar do valor do "makespan" ser o mesmo, a solução é diferente. A ordem gerada foi 4, 6, 1, 8, 3, 5, 7, 2.

A solução gerada, utilizando a regra JMIN em 3,53s de CPU e com "makespan" 29,0 min, é apresentada na Figura 6.7. A ordem gerada foi 2, 7, 6, 3, 5, 4, 1, 8.

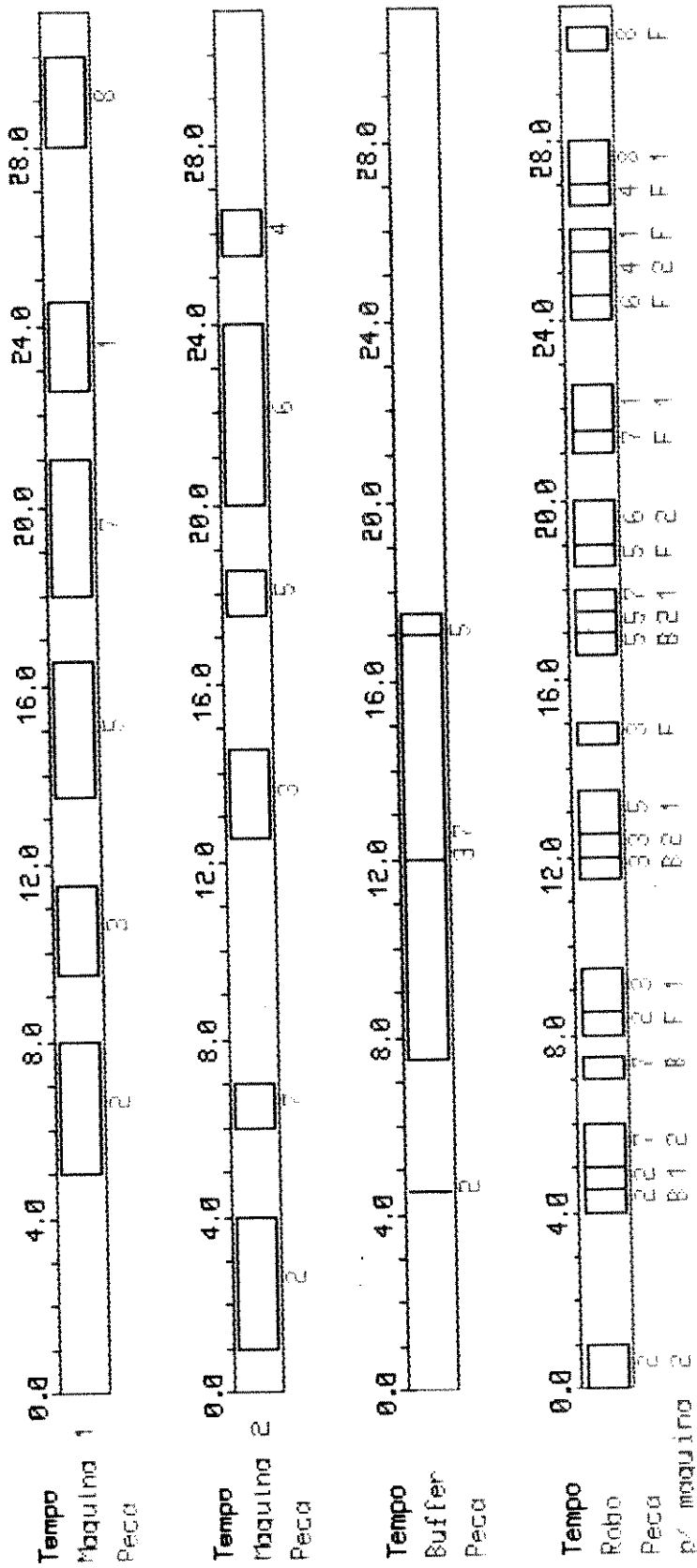


FIGURA 6.5 - Gráfico de Gantt Gerado com a Regra LPT para 8 Peças em 2 Máquinas



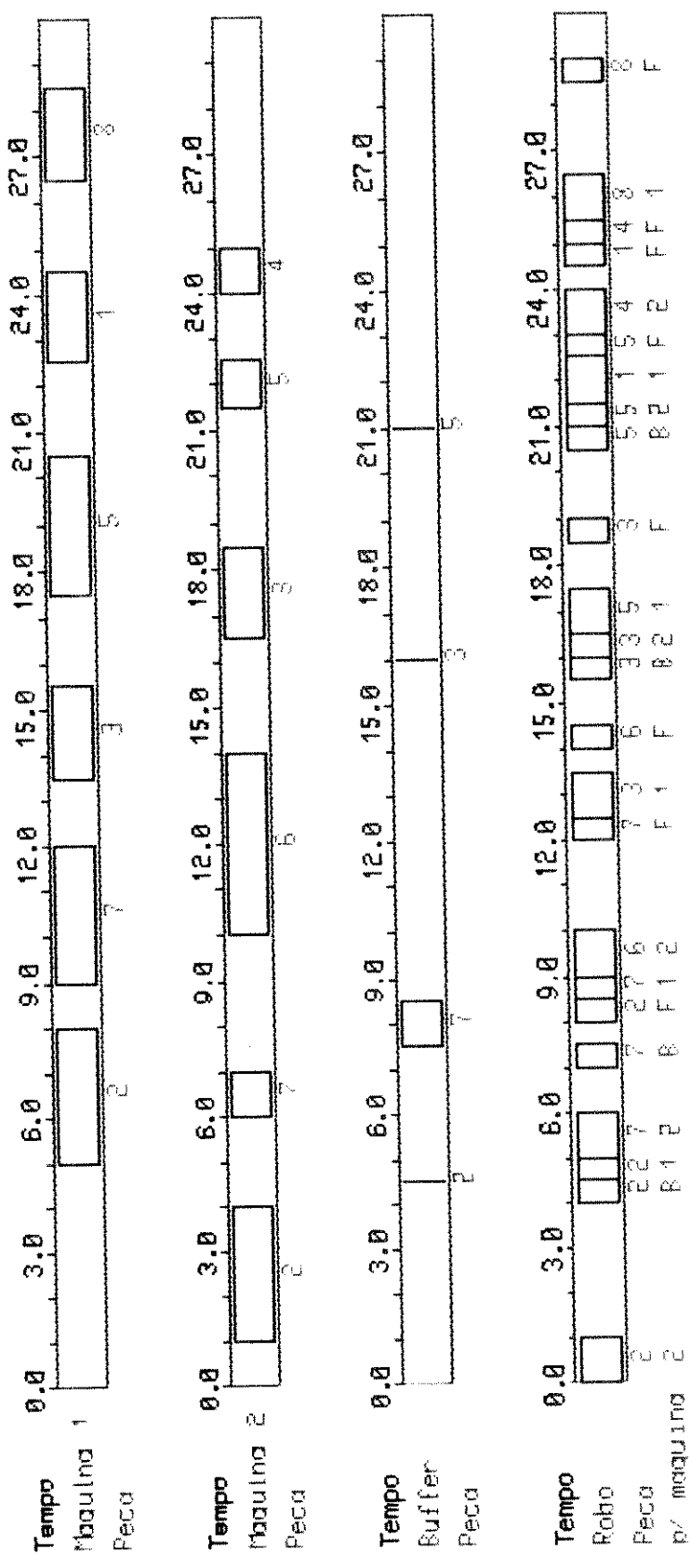


FIGURA 6.7 - Gráfico de Gantt Gerado com a Regra JMIN para 8 Peças em 2 Máquinas

A seguir, consideraremos todas as dez peças da Tabela 6.3. A solução gerada, utilizando a regra LPT em 4,51s de CPU e com "makespan" 35,5 min, é apresentada na Figura 6.8. A ordem gerada para as peças foi 2, 3, 5, 7, 9, 6, 10, 4, 1, 8.

A solução gerada, utilizando a regra SPT em 4,65s de CPU e com "makespan" 34,5 min, é apresentada na Figura 6.9. A ordem gerada para as peças foi 4, 6, 3, 1, 8, 5, 7, 9, 10, 2.

A solução gerada, utilizando a regra JMIN em 4,61s de CPU e com "makespan" 34,5 min, é apresentada na Figura 6.10. A ordem gerada para as peças foi 2, 7, 9, 3, 5, 6, 10, 4, 1, 8.

Com estes exemplos, podemos concluir que o custo computacional das três regras é praticamente o mesmo.

Já, quanto ao valor do "makespan", a regra JMIN mostrou-se superior.

No exemplo de oito peças, a regra JMIN forneceu a melhor solução, enquanto as regras LPT e SPT, mesmo fornecendo soluções diferentes, chegaram ao mesmo valor para o "makespan". No exemplo de dez peças, as regras JMIN e SPT, forneceram o melhor "makespan" com soluções diferentes.

É interessante observar como o gargalo se comporta nestes exemplos. Com as quatro primeiras peças, o gargalo é a máquina 1 que é, também, a máquina com carga absoluta maior.

Já, com as oito peças, o gargalo é a máquina 2, enquanto a máquina 1 é a que tem carga absoluta maior.

E, finalmente, com todas as dez peças, o gargalo é a máquina 2, enquanto a máquina 1 é a que tem carga absoluta maior.

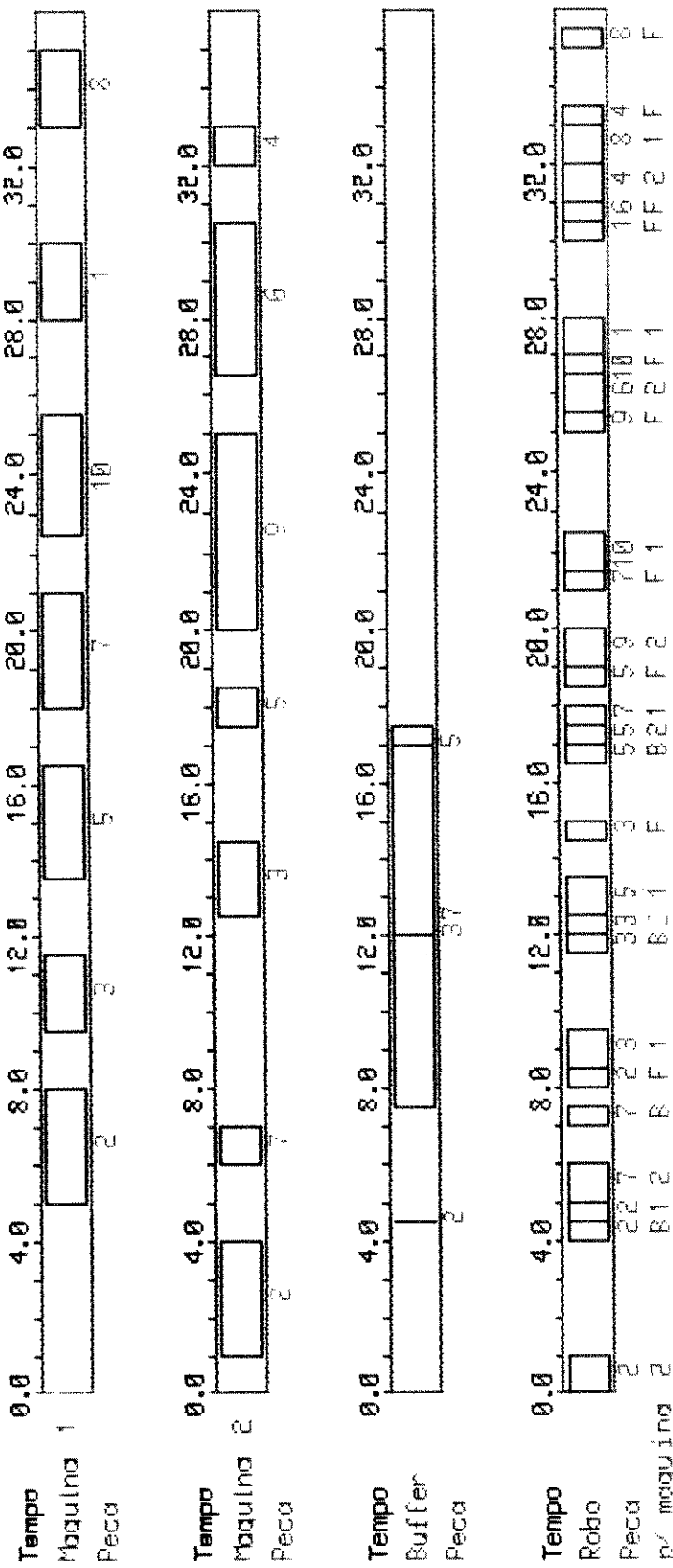


FIGURA 6.8 - Gráfico de Gantt Gerado com a Regra LPT para 10 Peças em 2 Máquinas

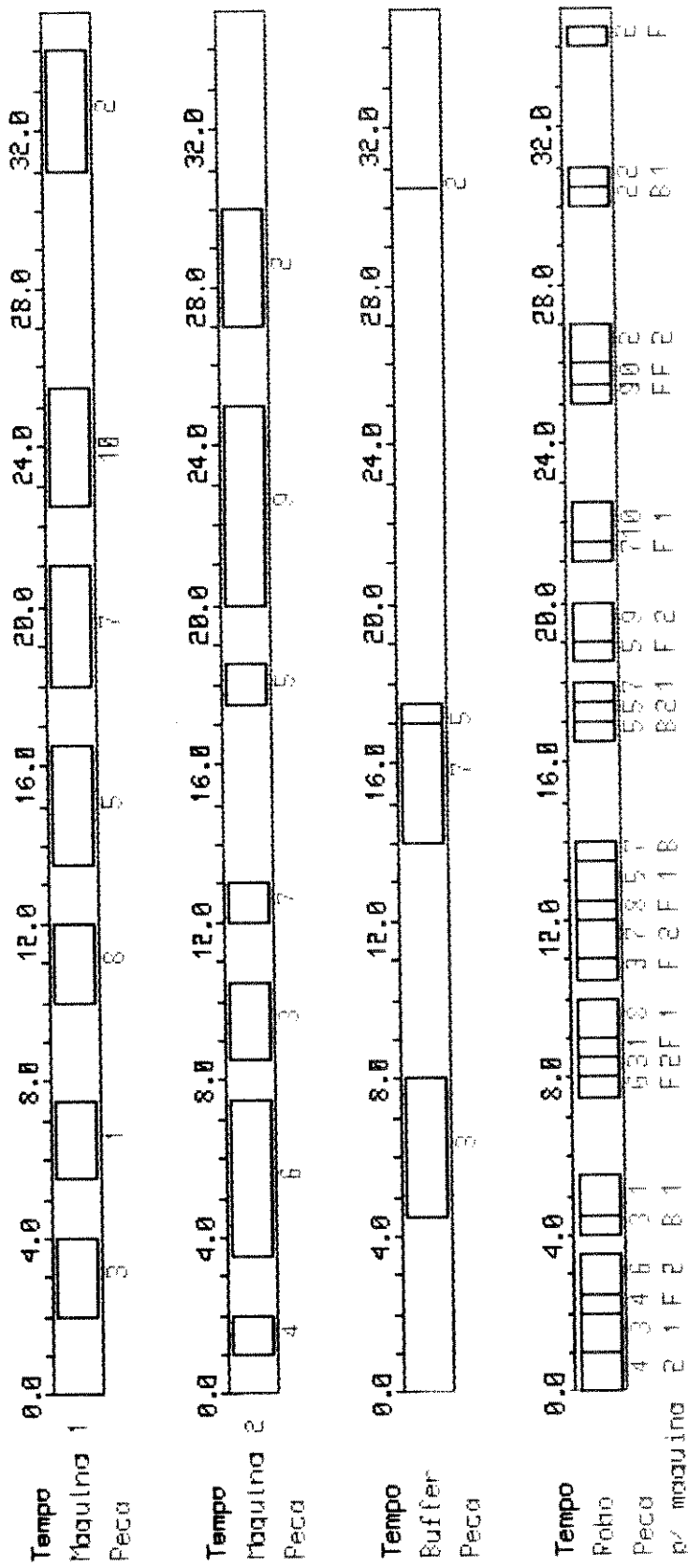


FIGURA 6.9 - Gráfico de Gantt Gerado com a Regra SPT para 10 Peças em 2 Máquinas





### 6.3. COMPARAÇÃO DA HEURÍSTICA SINCRONIZANDO MÁQUINAS E ROBÔ COM A HEURÍSTICA SINCRONIZANDO SÓ MÁQUINAS

A sincronização de máquinas e robô é necessária, como vimos no capítulo 2.

Podemos ilustrar este aspecto através de um exemplo com as oito primeiras peças da Tabela 6.3 e com duas Programações Horárias: a primeira sincronizando só as máquinas, apresentada na Figura 6.11, e a segunda sincronizando máquinas e robô, já apresentada na Figura 6.7.

A primeira não pode ser passada diretamente para execução, pois é preciso decidir qual peça o robô irá transportar primeiro nas várias situações onde mais de uma peça espera por ele no mesmo instante, como pode-se ver na Figura 6.11.

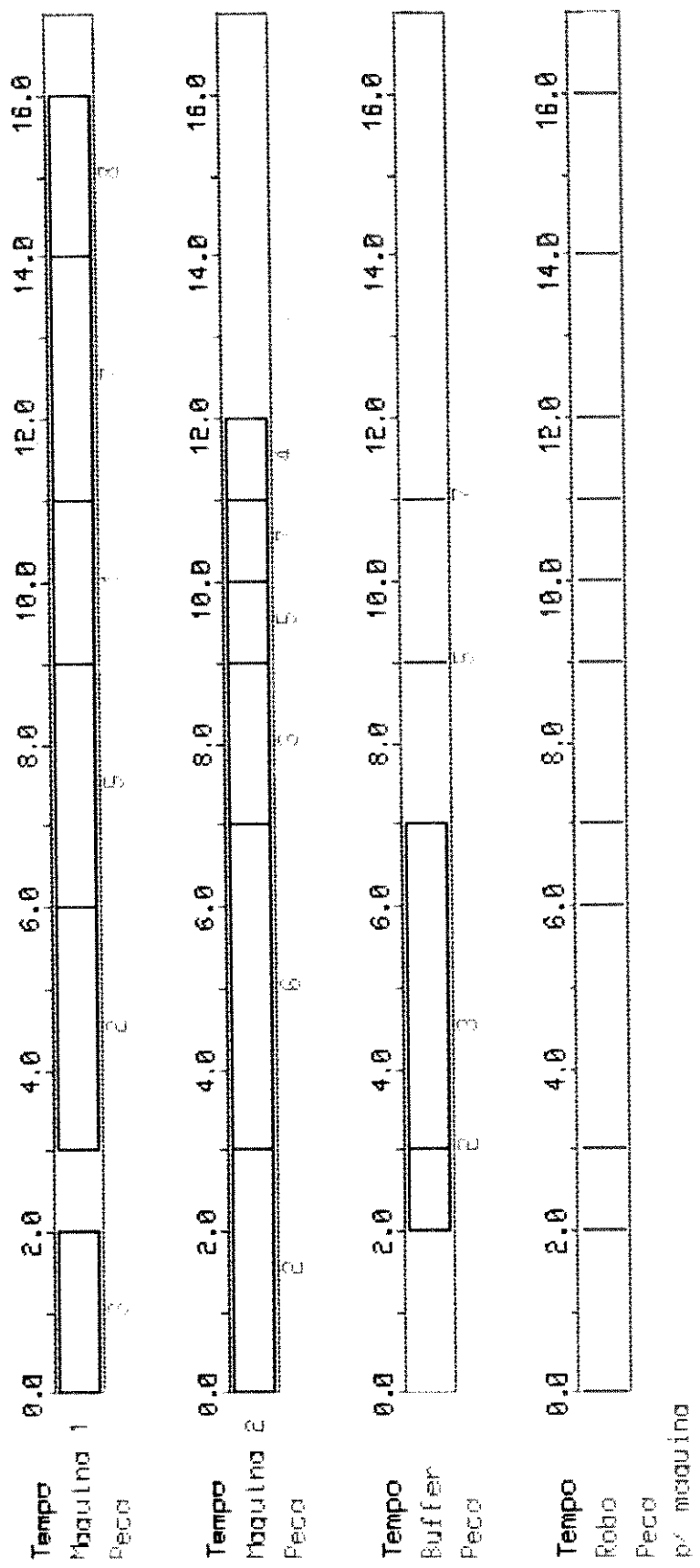


FIGURA 6.11 - Gráfico de Gantt Sincronizando só Máquinas para 8 Peças em 2 Máquinas

#### 6.4. COMPARAÇÃO DA SOLUÇÃO COM MÁQUINA ALTERNATIVA COM A SOLUÇÃO SEM MÁQUINA ALTERNATIVA

Como vimos no capítulo 2, a utilização de máquinas alternativas na resolução simultânea da alocação e da programação horária pode levar a uma diminuição do gargalo e, conseqüentemente, do "makespan". Para ilustrar, utilizaremos os dados que estão na Tabela 6.4.

P E Ç A S		1	2	3
ESTÁGIO 1	MÁQUINA	2	3	1
	Tempo de Transporte	1,0	1,0	1,0
	Tempo de Processamento	8,0	5,0	9,0
ESTÁGIO 2	MÁQUINA	3	1	3
	Tempo de Transporte	1,0	1,0	1,0
	Tempo de Processamento	7,0	3,0	4,0
	Máquina Alternativa			2
	Tempo de Transporte			1,0
	Tempo de Processamento			3,0
ESTÁGIO 3	MÁQUINA		2	
	Tempo de Transporte		1,0	
	Tempo de Processamento		6,0	

TABELA 6.4 - Dados de Produção de 3 Peças em 3 Máquinas com Máquinas Alternativas

A solução gerada, sem considerar máquina alternativa em 1,2s de CPU e com "makespan" 35,0 min, é apresentada na Figura 6.12. O gargalo foi de 6,5 min na máquina 3.

A solução gerada, considerando máquina alternativa em 2,5s de CPU e com "makespan" 23,0 min, é apresentada na Figura 6.13. O gargalo diminuiu com a utilização da máquina 2 no lugar da máquina 3, para o segundo estágio da peça 3, de 6,5 min para 3,5 min e passou a ser a máquina 2 e o "makespan" diminuiu em 12,0 min.

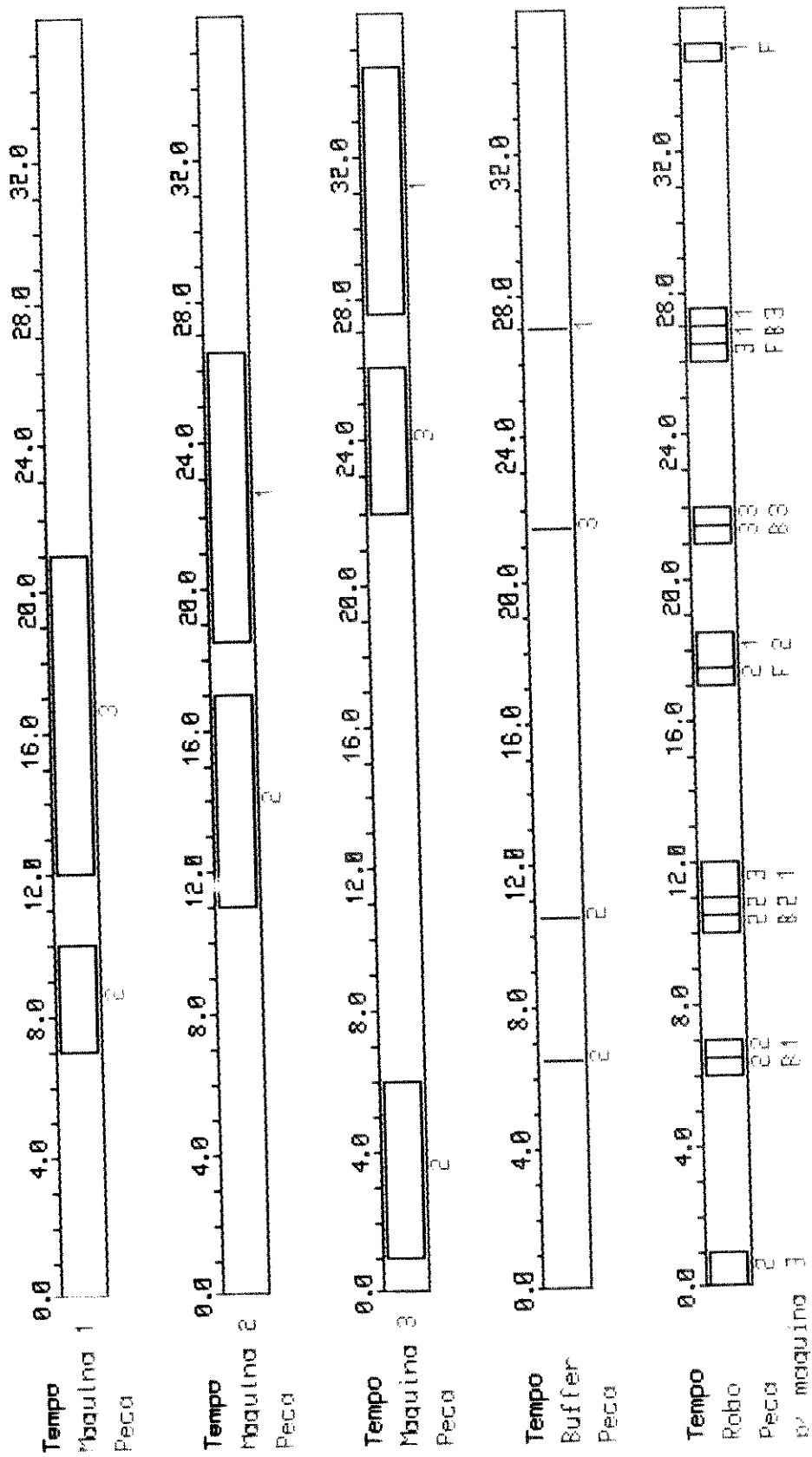


FIGURA 6.12 - Gráfico de Gantt com 3 Peças em 3 Máquinas sem Máquina Alternativa

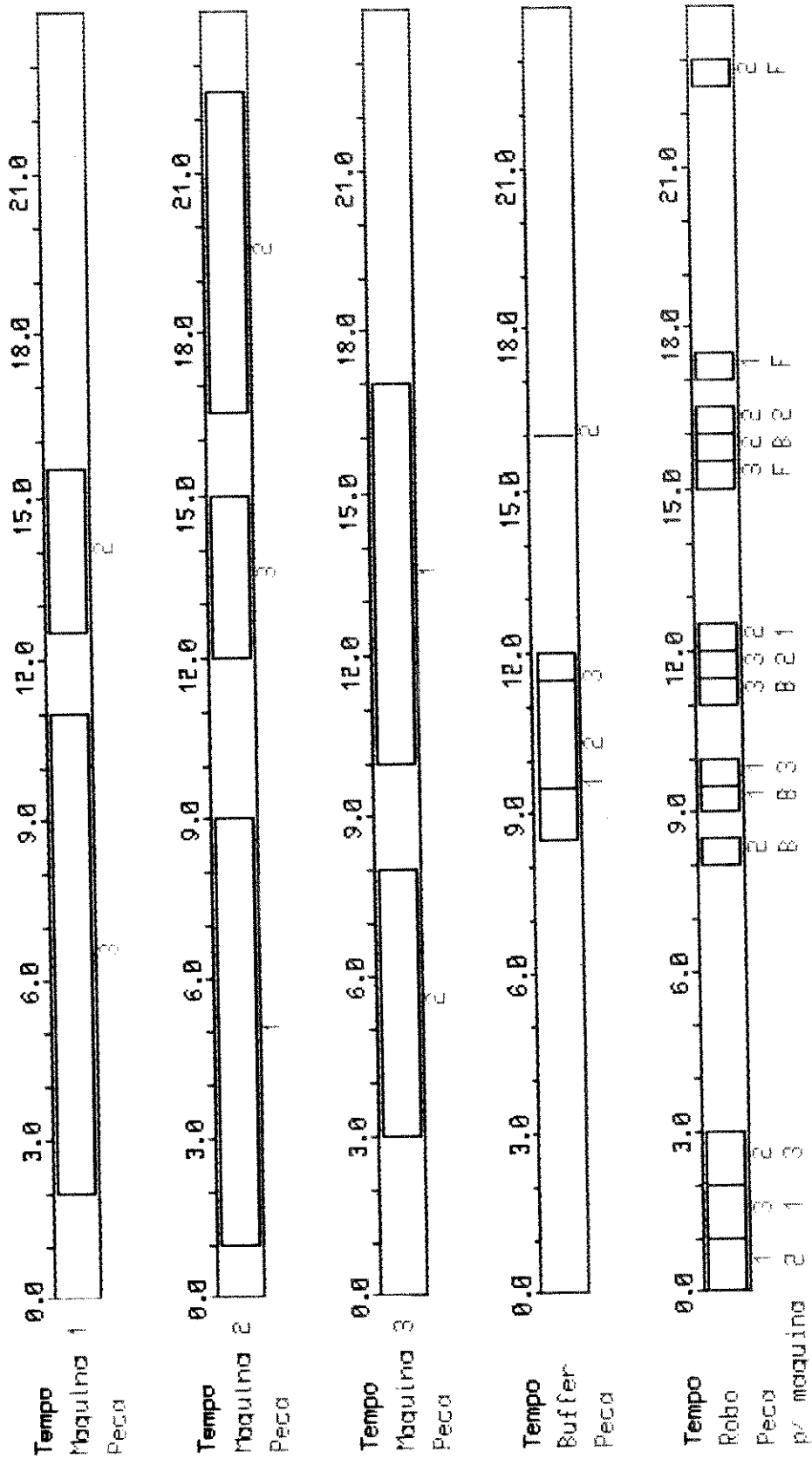


FIGURA 6.13 - Gráfico de Gantt com 3 peças em 3 Máquinas com Máquina Alternativa

## 6.5. EXEMPLO COM A CÉLULA EM FUNCIONAMENTO

Uma das vantagens da heurística em relação ao modelo é que ela pode ser utilizada em tempo real, ou seja, com a célula em funcionamento. Atendendo uma programação horária gerada anteriormente, pode-se gerar rapidamente uma programação para novas peças que entrarão na célula sem interromper seu funcionamento e, inclusive, aproveitando as atuais ociosidades.

Para ilustrar, utilizaremos os dados que estão na Tabela 6.2, apresentada anteriormente, e na Tabela 6.5.

HORÁRIO PARA INÍCIO	4,0
NÚMERO DE ESTÁGIOS	2
MÁQUINA DO ESTÁGIO 1	2
TEMPO TRANSPORTE	1,0
TEMPO DE PROCESSAMENTO	1,0
MÁQUINA DO ESTÁGIO 2	1
TEMPO TRANSPORTE	1,0
TEMPO PROCESSAMENTO	3,0

TABELA 6.5 - Dados de Produção da Nova Peça

A solução gerada pela heurística, com os dados da Tabela 6.2, em 1,04s de CPU com "makespan" 12,5 min, foi apresentada na Figura 6.4.

A solução gerada pela heurística a partir da programação horária apresentada na Figura 6.4, como entrada, juntamente com os dados da Tabela 6.5, em 0,44s de CPU, com "makespan" 14,5 min, é apresentada na Figura 6.14.

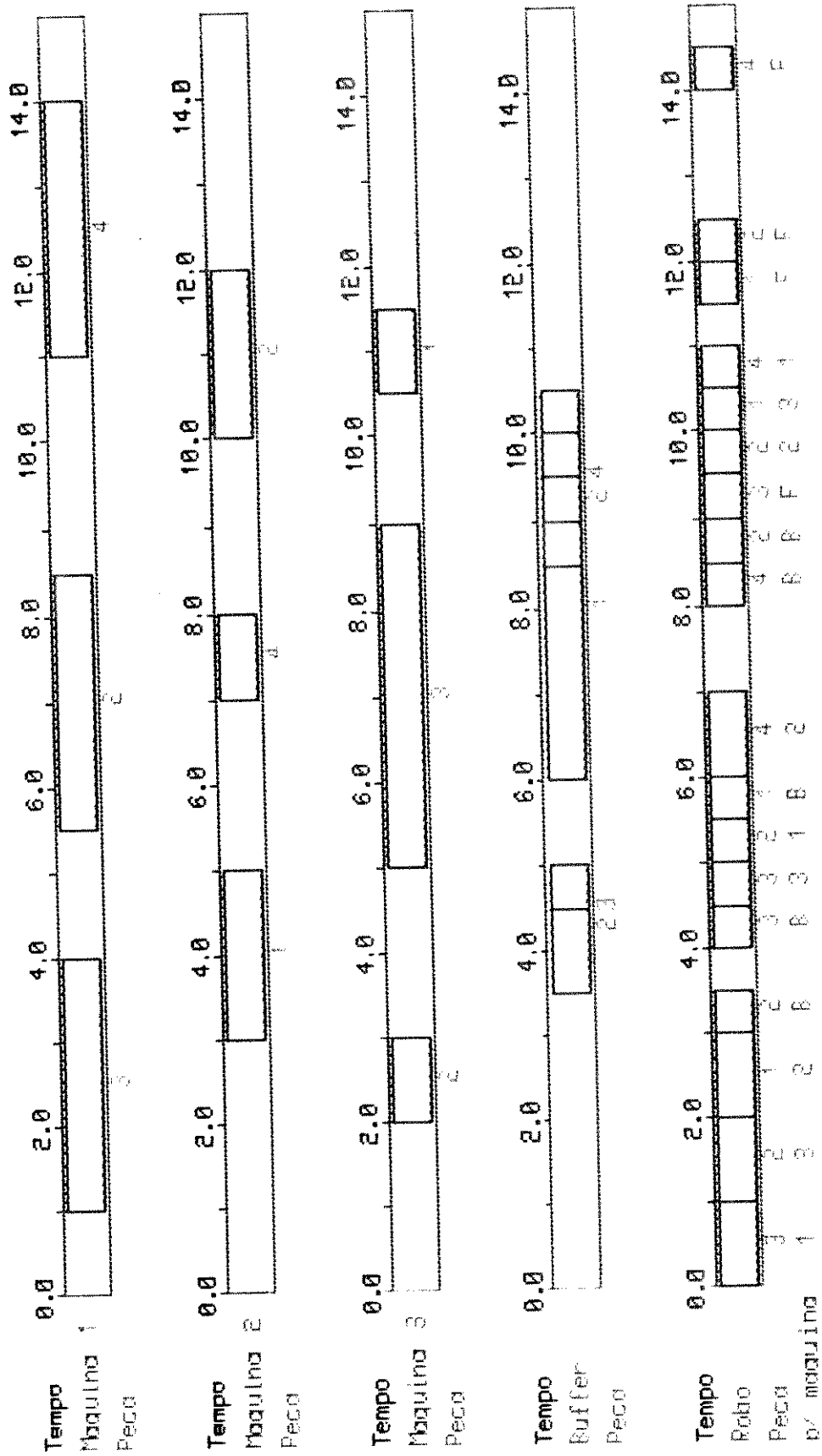


FIGURA 6.14 - Gráfico de Gantt com 3 Peças em Processamento, 1 Nova Peça e 3 Máquinas



Observe que, no Gráfico de Gantt, apresentado na Figura 6.4, existe uma ociosidade intermediária programada para a máquina 2 e no Gráfico de Gantt, apresentado na Figura 6.14, a peça que chegou quando a célula já estava atendendo a programação horária anterior justamente ocupou tal ociosidade intermediária, como sugerido no capítulo 2.

## 6.6. ÍNDICES DE DESEMPENHO

Após as análises mais específicas feitas anteriormente faremos aqui uma análise mais global do desempenho da heurística através da relação do número de peças versus tempo de CPU.

Consideraremos os dados da Tabela 6.3, apresentada anteriormente, e das Tabelas 6.6 e 6.7, apresentadas a seguir.

<b>P E Ç A S</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>
Nº DE ESTÁGIOS	2	1	2	1	1	2	1	1	1	2
MÁQ. 1 (ESTÁGIO)	2º		2º			1º	1º		1º	2º
TEMPO TRANSPORTE	1,0		1,0			1,0	1,0		1,0	1,0
TEMPO PROCESSAM.	3,0		2,0			6,0	5,0		4,0	1,0
MÁQ. 2 (ESTÁGIO)	1º	1º	1º	1º	1º	2º		1º		1º
TEMPO TRANSPORTE	1,0	1,0	1,0	1,0	1,0	1,0		1,0		1,0
TEMPO PROCESSAM.	2,0	4,0	3,0	5,0	4,0	2,0		3,0		2,0
<b>P E Ç A S</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>30</b>
Nº DE ESTÁGIOS	2	1	2	1	1	2	1	1	1	2
MÁQ. 1 (ESTÁGIO)	2º		2º			1º	1º		1º	2º
TEMPO TRANSPORTE	1,0		1,0			1,0	1,0		1,0	1,0
TEMPO PROCESSAM.	3,0		2,0			6,0	5,0		4,0	3,0
MÁQ. 2 (ESTÁGIO)	1º	1º	1º	1º	1º	2º		1º		1º
TEMPO TRANSPORTE	1,0	1,0	1,0	1,0	1,0	1,0		1,0		1,0
TEMPO PROCESSAM.	2,0	4,0	3,0	5,0	4,0	2,0		3,0		2,0

TABELA 6.6 -Dados de Produção de 30 Peças em 2 Máquinas  
(Continuação da Tabela 6.3)

<b>P E Ç A S</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
Nº DE ESTÁGIOS	1	2	2	1	2	1	2	1	1	1
MÁQ. 1 (ESTÁGIO)		2º	1º		1º		2º			1º
TEMPO TRANSPORTE		1,0	1,0		1,0		1,0			1,0
TEMPO PROCESSAM.		3,0	2,0		3,0		3,0			3,0
MÁQ. 2 (ESTÁGIO)		1º	2º	1º		1º	1º		1º	
TEMPO TRANSPORTE		1,0	1,0	1,0		1,0	1,0		1,0	
TEMPO PROCESSAM.		3,0	2,0	1,0		4,0	1,0		5,0	
MÁQ. 3 (ESTÁGIO)	1º				2º			1º		
TEMPO TRANSPORTE	1,0				1,0			1,0		
TEMPO PROCESSAM.	2,0				1,0			2,0		
<b>P E Ç A S</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>
Nº DE ESTÁGIOS	2	1	2	1	1	2	1	1	1	2
MÁQ. 1 (ESTÁGIO)			2º			1º			1º	2º
TEMPO TRANSPORTE			1,0			1,0			1,0	1,0
TEMPO PROCESSAM.			2,0			6,0			4,0	1,0
MÁQ. 2 (ESTÁGIO)	1º	1º	1º		1º	2º		1º		
TEMPO TRANSPORTE	1,0	1,0	1,0		1,0	1,0		1,0		
TEMPO PROCESSAM.	2,0	4,0	3,0		4,0	2,0		3,0		
MÁQ. 3 (ESTÁGIO)	2º			1º			1º			1º
TEMPO TRANSPORTE	1,0			1,0			1,0			1,0
TEMPO PROCESSAM.	3,0			5,0			5,0			2,0

TABELA 6.7 - Dados de Produção de 30 Peças em 3 Máquinas

(Tabela 6.7 - continuação)

P E Ç A S	21	22	23	24	25	26	27	28	29	30
Nº DE ESTÁGIOS	2	1	2	1	1	2	1	1	1	2
MÁQ. 1 (ESTÁGIO)	2º		2º						1º	2º
TEMPO TRANSPORTE	1,0		1,0						1,0	1,0
TEMPO PROCESSAM.	3,0		2,0						4,0	3,0
MÁQ. 2 (ESTÁGIO)	1º		1º		1º	2º		1º		1º
TEMPO TRANSPORTE	1,0		1,0		1,0	1,0		1,0		1,0
TEMPO PROCESSAM.	2,0		3,0		4,0	2,0		3,0		2,0
MÁQ. 3 (ESTÁGIO)		1º		1º		1º	1º			
TEMPO TRANSPORTE		1,0		1,0		1,0	1,0			
TEMPO PROCESSAM.		4,0		5,0		6,0	5,0			

TABELA 6.7 -Dados de Produção de 30 Peças em 3 Máquinas  
(Continuação)

A solução gerada para as cinco primeiras peças da Tabela 6.3, em 1,52s de CPU e "makespan" 19,0 min, é apresentada na Figura 6.15.

A solução gerada para as seis primeiras peças da Tabela 6.3, em 2,02s de CPU e "makespan" 22,0 min, é apresentada na Figura 6.16.

A solução gerada para as sete primeiras peças da Tabela 6.3, em 2,94s de CPU e "makespan" 26,0 min, é apresentada na Figura 6.17.

A solução gerada para as vinte primeiras peças da Tabela 6.6, em 21,20s de CPU e "makespan" 70,5 min, é apresentada na Figura 6.18.

A solução gerada para todas as peças da Tabela 6.6 com "makespan" 105,0 min, foi obtida em 41,41s de CPU.

Finalmente a solução gerada para todas as peças da Tabela 6.7, em 36,67s de CPU e "makespan" 80,50 min, é apresentada na Figura 6.19.

Para melhor analisarmos a relação entre o número de peças e o tempo de CPU, colocaremos os dados obtidos nos exemplos com 2 máquinas, incluindo os relativos às

Figuras 6.10 e 6.13, na Tabela 6.8.

Nº DE PEÇAS	5	6	7	8	10	20	30
TEMPO DE CPU	1,52	2,02	2,94	3,53	4,67	21,20	41,41

TABELA 6.8 - Número de Peças x Tempo de CPU

Nesta tabela, podemos observar que, para um dado número de máquinas, a relação entre o número de peças e o tempo de CPU é aproximadamente linear.

Já nos últimos exemplos com 30 peças, mas com 2 e 3 máquinas, o tempo de CPU diminui com o aumento do número de máquinas.

Já, quanto à relação "makespan" x ociosidades intermediárias programadas, observa-se, que se não houvesse o aproveitamento de algumas destas ociosidades, o "makespan" seria superior ao que é. Para ilustrar este importante aspecto da heurística, aproveitamos os dois primeiros exemplos. Observamos que, quando passamos de cinco para seis peças, com o tempo total de processamento desta sexta peça de 4,0 min, o "makespan" passa de 19,0 min para 22,0 min e não 23,0 min, porque já há um aproveitamento de parte de alguma destas ociosidades.

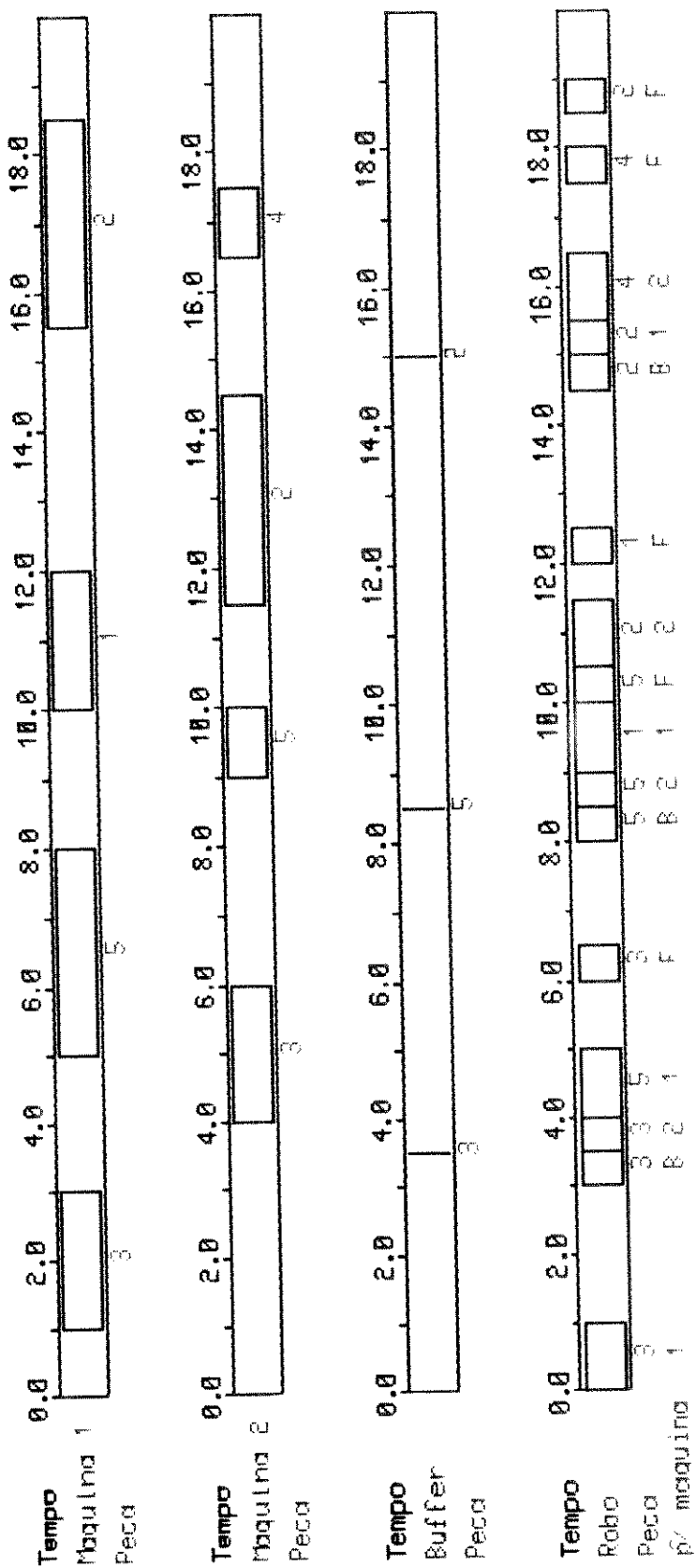


FIGURA 6.15 - Gráfico de Gantt com 5 Peças em 2 Máquinas

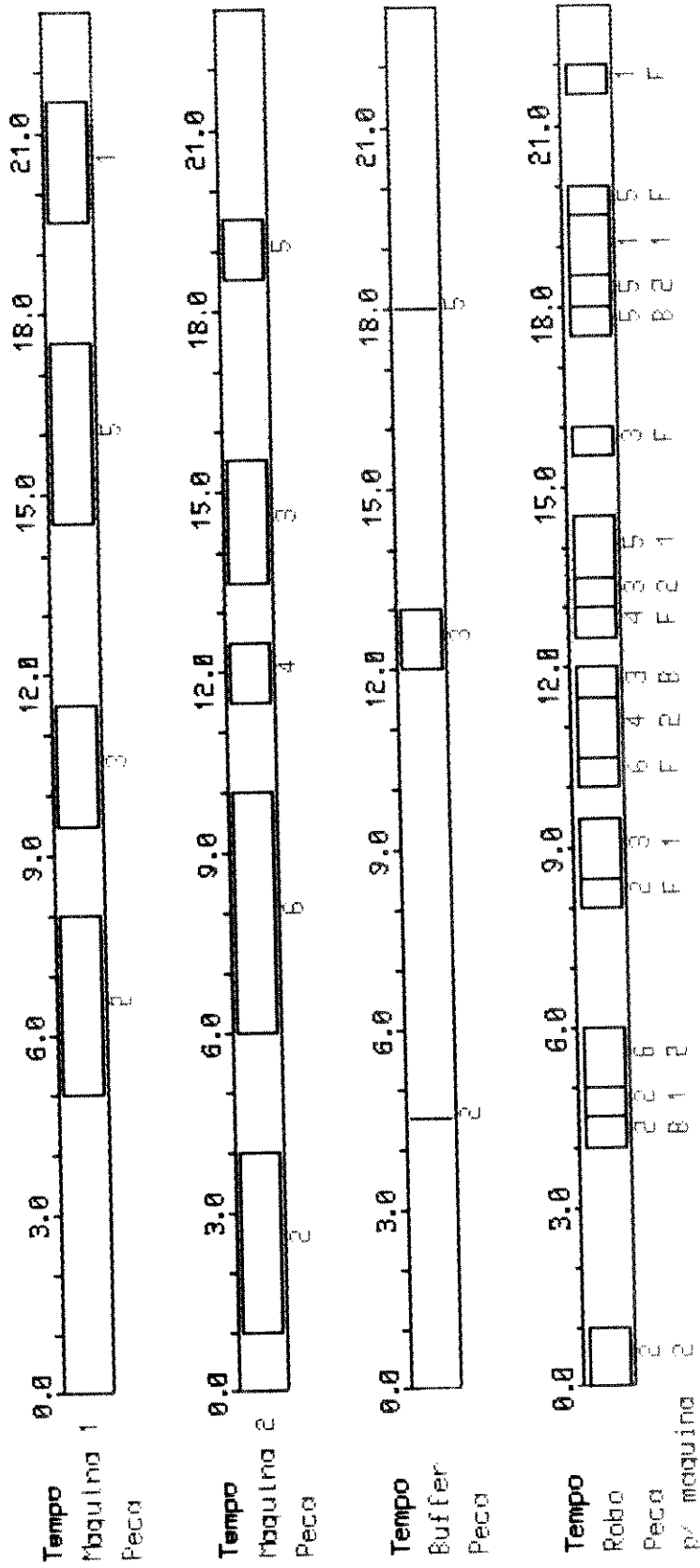


FIGURA 6.16 - Gráfico de Gantt com 6 Peças em 2 Máquinas

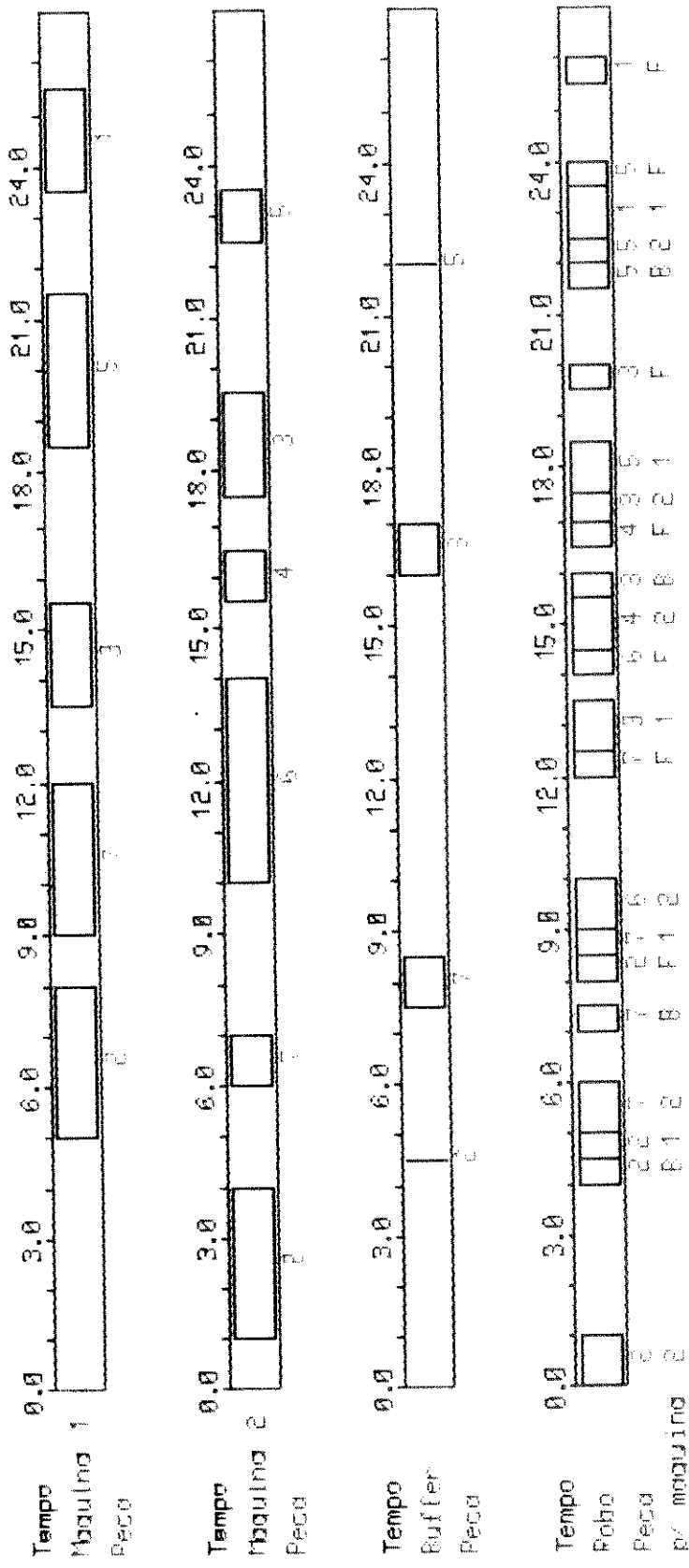


FIGURA 6.17 - Gráfico de Gantt com 7 Peças em 2 Máquinas

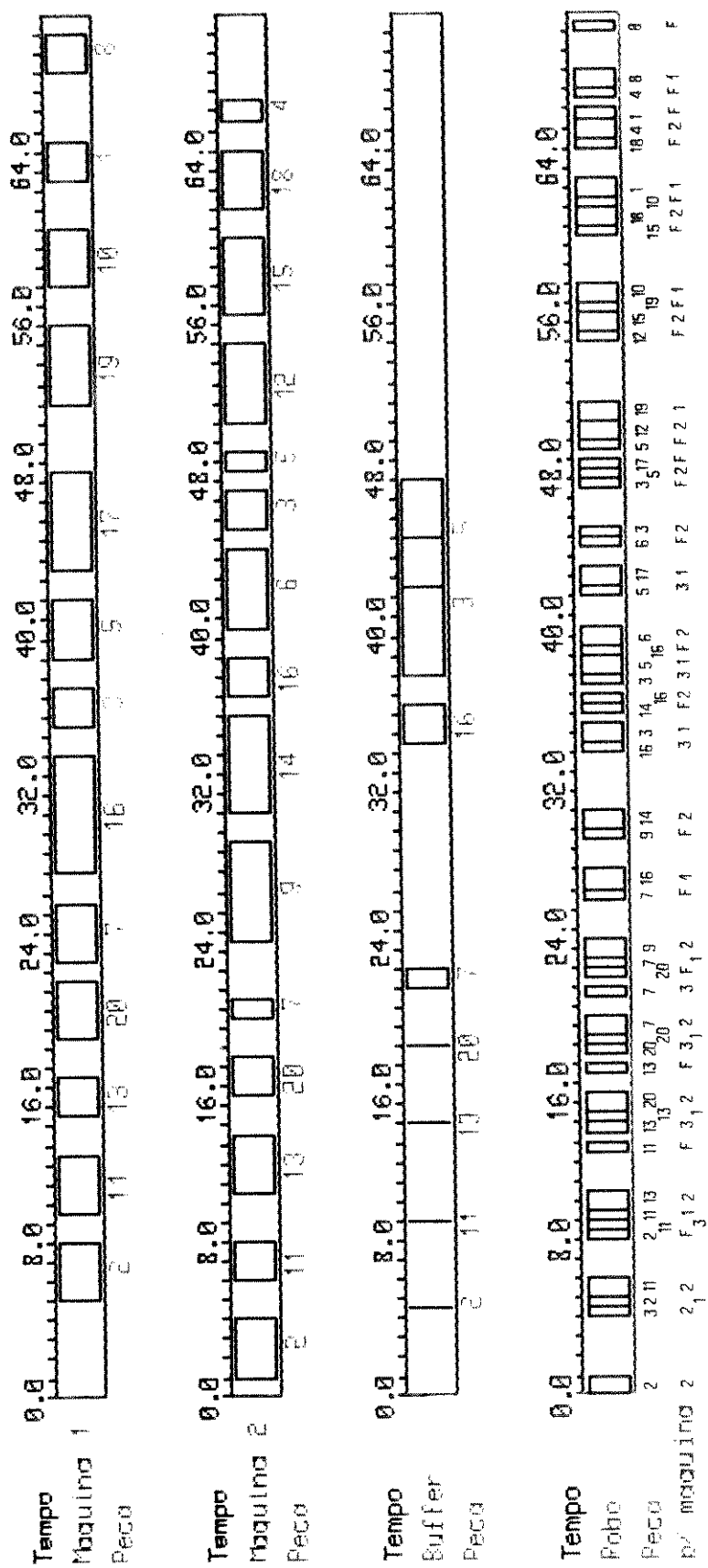


FIGURA 6.18 - Gráfico de Gantt com 20 Peças em 2 Máquinas





**CAPÍTULO 7**

**CONCLUSÃO**

Neste trabalho apresentamos algumas características do problema da Programação Horária de Peças em uma FMC e um levantamento bibliográfico desta área.

Introduzimos o conceito de gargalo dinâmico e sua relação com a sincronização de máquinas e robô e com o aproveitamento das ociosidades intermediárias.

Apresentamos dois modelos de PLIM para o problema, utilizando o referido conceito. O primeiro modelo considera o fluxo unidirecional e o segundo, mais geral, considera o fluxo multidirecional.

Desenvolvemos um "software" baseado nos dois modelos propostos e os resultados obtidos validaram os modelos, mas o tempo de CPU gasto na obtenção da solução ótima, com as versões implementadas, não é satisfatório para os exemplos com muitas peças. Esta medida de desempenho computacional depende de vários fatores: modelo escolhido, algoritmo adotado, tipo de implementação, linguagem adotada, tipo de computador, compilador e sistema operacional. De qualquer forma desenvolvemos procedimentos de aceleração que reduziram o tempo de CPU em cerca de 50% sem perder a solução ótima.

Meurant (1992) relata uma melhoria de 30% ao passar de um compilador FORTRAN para um compilador FORTRAN 77.

Carvalho (1992) observa que a quantidade de transistores encapsulados num chip, no fim da década, vai fazer os atuais microprocessadores parecerem artefatos de uma remota idade do silício lascado.

Desta forma, consideramos que novas versões devem explorar as características do ambiente computacional específico no qual estarão inseridas.

Com os resultados obtidos é possível considerar este "software" como um dos módulos de um sistema de gerenciamento de uma FMC. Este módulo gera a programação horária das peças, o qual é enviado ao módulo responsável pelo controle da FMC, como apresentado por Hirama e outros (1987).

Este módulo também pode ser usado como validador de regras heurísticas para a programação horária de peças na FMC.

Além disso, este "software" pode estar em uma Base de Algoritmos de um Sistema de Gerenciamento de um FMS, gerando as variáveis, as restrições e a função objetivo para o modelo associado à programação horária de cada célula e para os modelos de PLIM para outros problemas de Planejamento e Programação da Produção, como sugerido também por Fernandes (1985).

Propomos uma heurística com as seguintes características:

- Utiliza o conceito do gargalo dinâmico;
- Sincroniza máquinas e robô;
- Aproveita as ociosidades intermediárias;
- Utiliza a flexibilidade de máquinas alternativas para reduzir o atual gargalo;
- Permite a escolha de regras, por exemplo, as clássicas LPT e SPT, a serem adotadas para a seleção da próxima peça que terá seus horários programados;
- Introduce a regra JMIN com base na análise do atual gargalo versus a rota da peça;

Desenvolvemos um "software" baseado na heurística proposta que foi validado pelo segundo modelo, apresentando um resultado ótimo em um dos exemplos considerados e um erro inferior a 4% no outro. O tempo de CPU é a grande vantagem da heurística sobre o modelo.

Os vários exemplos ilustraram as características da heurística que, por sua modularidade, permite que novas heurísticas sejam rapidamente geradas.

As características destes "softwares" permitem a operação automática da FMC com a transmissão dos dados para cada máquina e para o robô, responsável pelo transporte, carga e descarga das peças, da programação horária das peças.

É interessante colocar que o procedimento aqui desenvolvido pode perfeitamente ser utilizado na operação automática de FMS, como aquele que foi observado por Calarge e outros (1993), no Japão.

Também permite a produção de múltiplos produtos em pouca quantidade atingindo, assim, a economia de escopo que é a tendência atual, como observado por Nahmias (1989), entre outros.

Propomos um sistema híbrido constituído por ambos os "softwares" para ser utilizado em uma FMC, ou em um outro ambiente de manufatura com características semelhantes, para a programação horária das peças, com o primeiro sendo ativado periodicamente e o segundo sendo ativado quando houver necessidade de reprogramação, atendendo o requisito de tempo real, como colocado por Ahluwalia (1991).

Futuras direções de pesquisa podem incluir as seguintes:

- Estudo de técnicas recentes de PI para melhoria do desempenho computacional dos modelos, como os apresentados por Schrijver (1987): método de redução de bases de Lovász, método elipsoidal de Khachiyan e o método de Karmakar, os algoritmos de Tardos e de Megiddo, o algoritmo de Lenstra e outras técnicas.
- Estudo de técnicas de processamento paralelo e de tendências de "software", como apresentado por Meurant (1992).
- Análise do Problema de Programação Horária de Peças em uma FMC sob o ponto de vista de Engenharia Concorrente, como proposto por Kusiak (1993).
- Evolução da heurística para incluir, além da análise comparativa das ociosidades intermediárias com os intervalos necessários, uma análise de ampliação de uma determinada ociosidade com a conseqüente alteração da programação horária a partir da referida ampliação.

## **REFERÊNCIAS BIBLIOGRÁFICAS**

- AHLUWALIA, RASHPAL and JI, PING (1991). *A Distributed Approach to Job Scheduling in a Flexible Manufacturing System*. Computers Ind. Eng., Vol. 20, Nº 1.
- BAKER, KENNETH P. (1974). *Introduction to Sequencing and Scheduling*. John Wiley & Sons, Inc., N.Y.
- BRETTHAUER, KURT M. and VENKATARAMANAN, M.A. (1990). *Machine Loading and Alternate Routing in a Flexible Manufacturing System*. Computers Ind. Eng., Vol. 18, Nº 3.
- BUZACOTT, J.A.; SHANTHIKUMAR, J.G. (1980). *Models for Understanding Flexible Manufacturing Systems*. AIIE Transactions, 12, 4.
- CALARGE, FELIPE A.; BATOCHIO, ANTONIO e AGOSTINHO, OSWALDO, L. (1993). *Conceituação, Estudo e Análise de um Sistema Flexível de Manufatura*. Máquinas & Metais, XXXVII, Nº 324, Janeiro.
- CARVALHO, FLÁVIO (1992). *Eletrônica Escolhe e Ganha Poder*. Informática Exame, Ano 7, Março.
- CHANG, YIH-LONG; MATSUO, HIROFUMI and SULLIVAN, ROBERT (1989). *A Bottleneck-based beam Search for Job Scheduling in a Flexible Manufacturing System*. Int. J. Prod. Res., Vol. 27, Nº 11.
- CHEN, YUNG-JUNG and ASKIN, RONALD G. (1990). *A Multiobjective Evaluation of Flexible Manufacturing System Loading Heuristics*. Int. J. Prod. Res., Vol. 28, Nº 5.
- CHRYSSOLOURIS, G.; WRIGHT, K.; PIERCE, J. and COBB, W. (1988). *Manufacturing Systems Operation: Dispatch Rules Versus Intelligent Control*. Robotics & Computer Integrated Manufacturing, Vol. 4, Nº 3/4.
- CRESTIN, J.P. (1990). *Hints Towards Integration*. Computers in Industry, 14.
- DUB, M. (1984). *Standardized FMS Application Software: A Case Study*. Proceedings of 3<sup>rd</sup> I.C. on FMS.
- EDGHILL, J.S.; DAVIES, A. (1985). *FMS - The Myth and Reality*. International Journal of Advanced Manufacturing Technology, Vol. 1, Nº 1, September.

- FERNANDES, C.A. DE OLIVEIRA (1985). *Análise e Implementação de Métodos de Geração de Programas de Produção para Sistemas Modernos de Manufatura*. Tese de Mestrado, Engenharia Elétrica, UNICAMP.
- GARETTI, M.; POZZETTI, A. and BAREGGI, A. (1990). *On-line Loading and Dispatching in Flexible Manufacturing Systems*. Int. J. Prod. Res., Vol. 28, Nº 7.
- GEOFFRION, A.M.; MARSTEN, R. (1972). *Integer Programming Algorithms: A Framework and State of the Art Survey*. Management Science, Vol. 18.
- GERSHWIN, S.B.; AKELLA, R.; CHOONG, Y.F. (1985). *Short-term Production Scheduling of an Automated Manufacturing Facility*. IBM J. Res. Develop., Vol. 29, Nº 4, July.
- GOLDHAR, JOEL and JELINEK, MARIAN (1990). *Manufacturing as a Service Business: CIM in the 21st Century*. Computers in Industry, 14.
- GRAVES, STEPHEN (1981). *A Review of Production Scheduling*. Operations Research, Vol. 29, Nº 4.
- GREENE, TIMOTY and SADOWSKI, RANDALL (1986). *A Mixed Integer Program for Loading and Scheduling Multiple Flexible Manufacturing Cells*. European Journal of Operational Research, 24.
- HAM, I.; HITOMI, K. and YOSHIDA, T. (1985). *Group Technology Applications to Production Management*. Kluwa-Nighoff Publishing.
- HILDEBRANT, R.R.; SURI, R. (1980). *Methodology and Multilevel Algorithm Structure for Scheduling and Real-Time Control of Flexible Manufacturing Systems*. Proceedings 3<sup>rd</sup> International Symposium on Large Engineering Systems, Canada.
- HIRAMA, M.; PRESCINOTTI, E.; SAWADA, F.; OLIVEIRA, M.A. (1987). *Desenvolvimento de Software Aplicativo para Ambiente de Célula Flexível de Manufatura*. Anais do 7º Seminário de Comando Numérico no Brasil.
- HITZ, K.L. (1979). *Scheduling of Flexible Flow-Shop*. Research Report, Lab. of Information and Decision Systems, MIT, March.



- HUGHES, ANN J. and GRAWILOG, DENNIS E. (1973). *Linear Programming Emphasis on Decision Making*. Addison-Wesley Publishing Company.
- HUTCHISON, J. (1991). *Current and Future Issues Concerning FMS Scheduling*. OMEGA Int. J. of Magmt. Sci., Vol. 19, Nº 6.
- ICHINO, MANABU; SKLANSKY, J. (1984). *Optimum Feature Selection by Zero-One Integer Programming*. IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-14, Nº 5, September/October.
- JOHNSON, S.M. (1954). *Optimal Two and Three-Stage Production Schedules with Set-up Times Included*. Nav. Res. Logist. Q., 1.
- KAKU, B.; THOMPSON, GERARD (1986). *An Exact Algorithm for the General Quadratic Assignment Problem*. European Journal of Operational Research 23.
- KALKUNTE, M.V.; SARINI, S.C.; WILHELM, W.E. (1986). *Flexible Manufacturing Systems: A Review of Modelling Approaches for Design, Justification and Operation*. Flexible Manufacturing Systems: Methods and Studies. A. Kusiak (Editor) Elsevier Science Publishers B.V., North-Holland.
- KIMEMIA, J.; GERSHWIN, S.B. (1979). *Network Flow Optimization in Flexible Manufacturing Systems*. Proceedings of 1978 IEEE Conference on Decision and Control.
- KIMEMIA, J.; GERSHWIN, S.B. (1985). *Flow Optimization in Flexible Manufacturing Systems*. Int. J. Prod. Res., Vol. 23, Nº 1.
- KLEEBANK, V.; WALTER, C. (1988). *Técnicas de Inteligência Artificial Aplicadas ao Planejamento*. Anais do 3º CONAI, São Paulo, SP.
- KUMAR, V. and KANAL, L. (1983). *A General Branch-and-Bound Formulation for Understanding and Synthesizing and/or Tree Search Procedures*. Artificial Intell., Vol. 21, Nº 1.
- KUMAR, V. and KANAL, L. (1984). *Parallel Branch-and-Bound Formulations for AND/OR Tree Search*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, Nº 6.

- KUSIAK, ANDREW (1985). *Flexible Manufacturing Systems: A Structural Approach*. International Journal of Production Research, 23.
- KUSIAK, ANDREW (1986). *Application of Operational Research Models and Techniques in Flexible Manufacturing Systems*. European Journal of Operational Research, 24.
- KUSIAK, ANDREW (1993). *Manufacturing Systems: A Concurrent Engineering Perspective*. Conference Proceedings of the International LAAS Seminar.
- LEE, J.; MIRCHANDANI, PITU B. (1986). *Scheduling with Set-ups on a Two Machine FMS*. Proceedings 1986 IEEE International Conference on Robotics and Automation, Vol. 3/1, April.
- LIANG, MING and TABOUN, S. (1992). *Part Selection and Part Assignment in Flexible Manufacturing Systems with Cellular Layout*. Computers and Industrial Engineering, Vol. 23, Nº 1-4.
- LIMA, PAULO e AGOSTINHO, OSWALDO (1989). *Tendências da Evolução dos Sistemas de Planejamento de Recursos de Manufatura Integrada*. Anais do 9º Seminário de Comando Numérico no Brasil.
- MAIMON, ODED Z. (1987). *Real-time Operational Control of FMS*. Journal of Manufacturing Systems, Vol. 6, Nº 2.
- MAJSTOROVIC, V.D.; MILACIC, V.R. (1990). *Expert Systems for Maintenance in the CIM Concept*. Computers in Industry, 15.
- MARTIN, T. (1990). *The Need for Human Skills in Production. The Case of CIM*. Computers in Industry, 14.
- MEURANT, GÉRARD (1992). *Os Computadores Paralelos e o Futuro da Computação Científica*. Boletim da SBMAC, Outubro, Vol. 3, Nº 1, Série II.
- MINOUX, M. (1983). *Programmation Mathématique: Théorie et Algorithmes*. DUNOD.
- MORTON, T.E.; SMUNT, T.L. (1986). *A Planning and Scheduling System for Flexible Manufacturing*. FM Systems: Methods and Studies. A. Kusiak (Editor) Elsevier Science Publishers B.V., North-Holland.

- NAHMIAS, STEVEN (1989). *Production and Operations Analysis*. IRWIN, Boston.
- NAU, D.S.; KUMAR, V. and KANAL, L.N. (1984). *General Branch-and-Bound and its Relation to  $A^*$  and  $AO^*$* . *Artificial Intelligence*, 23.
- NEWMAN, P.A. (1988). *Scheduling in CIM Systems. Artificial Intelligence Implications for CIM*. Edited by Prof. A. Kusiak. IFS (Publications) Ltda., UK, Springer-Verlag.
- ORTIZ, LUIS E.C.; LIMA, ANTONIO MILITÃO DE A.; MUZZI JR., JOÃO JUAREZ (1990). *Alternativas Brasileiras para Viabilização de Sistemas de Manufatura Flexíveis*. *Automação e Indústria*, Ano III, Nº 27.
- PALMER, K.H. et all (1984). *A Model-Management Framework for Mathematical Programming*. John Wiley & Sons, New York.
- PERKINS, JAMES R. and KUMAR, P.R. (1989). *Stable Distributed, Real-Time Scheduling of Flexible Manufacturing/Assembly/ Disassembly Systems*. *IEEE Transactions on Automatic Control*, Vol. 34, Nº 2.
- RANKY, PAUL G. (1986). *Computer Integrated Manufacturing*. Prentice Hall International, U.K.
- REISCH, D. (1987). *Total CIM Concept Embracing Logistics*. *Robotics & Computer-Integrated Manufacturing*, Vol. 3, no 1.
- RO, IN-KYO and KIM, JOANG-IN (1990). *Multi-Criteria Operational Control Rules in FMS*. *Int. J. Prod. Res.*, Vol. 28, Nº 1.
- RODAMMER, FREDERICK and WHITE JR., K. PRESTON (1988). *A Recent Survey of Production Scheduling*. *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 18, Nº 6, November/December.
- RONEN, B. and STARR, M.K. (1990). *Synchronized Manufacturing as in OPT: From Practice to Theory*. *Computers and Industrial Engineering*, Vol. 18, Nº 4.
- SALKIN, H.M. and K. MATHUR (1989). *Foundations of Integer Programming*. Cleveland, Ohio. New York, North Holland.

- SCHRIJVER, ALEXANDER (1986). *Theory of Linear and Integer Programming*. John Wiley & Sons, Ltd.
- SHANKER, KRIPA and SRINIVASULU (1989). *Some Solution Methodologies for Loading Problems in a Flexible Manufacturing System*. Int. J. Prod. Res., Vol. 27, Nº 6.
- SOLOT, PHILIPPE (1990). *A Concept for Planning and Scheduling in an FMS*. European Journal of Operational Research, 45.
- SOUZA, ANTONIO SÉRGIO DE; ALBUQUERQUE, ANTONIO R.P.L.; PICINATO, LUIS CARLOS MOLINA; OTSUKA, TOMIO (1992). *Leitstand: Integração entre Planejamento e Controle da Produção*. Anais do 5º CONAI, São Paulo, SP.
- STECKE, KATHRYN (1983). *Formulation and Solution of Non-Linear Integer Production Planning Problems for Flexible Manufacturing Systems*. Management Science, Vol. 29, Nº 3, March.
- STECKE, K., MORIN, T.L. (1985). *The Optimality of Balancing Work Loads in Certain Types of Flexible Manufacturing Systems*. European Journal of Operations Research, 20.
- STECKE, K.; BERRADA, M. (1986). *A Branch and Bound Approach for Machine Load Balancing in Flexible Manufacturing Systems*. Management Science, Vol. 32, Nº 10, October.
- STECKE, KATHRYN (1986). *A Hierarchical Approach to Solving Machine Grouping and Loading Problems of Flexible Manufacturing Systems*. European Journal of Operational Research, 24.
- SUH, NAM P. (1990). *Strengthening the United States Manufacturing Industry and CIM*. 4th IMEC.
- SURI, R.; HILDEBRANT, R.R. (1984). *Modeling Flexible Manufacturing Systems Using Mean-Value Analysis*. J. of Manufacturing Systems, Vol. 3, Nº 1.
- TABOUN, S.M. and ULGER, T. (1992). *Multi-objective Modelling of Operation-allocation Problem in Flexible Manufacturing Systems*. Computers and Industrial Engineering, Vol. 23, Nºs 1-4.

- TANG, CHRISTHOPER 1986). *A Job Scheduling Model for a Flexible Manufacturing Machine*. Proceedings 1986 IEE International Conference on Robotics and Automation, Vol. 1/1, April.
- TEIXEIRA, E.; STANGE, P.; FERNANDES, C.A.; HIRAMA, M.; ASANUMA, K. (1987). *A Model for the Problem of Workpieces Sequencing in a Flexible Manufacturing Cell*. Proceedings of the Intern. AMSE Conference "Modelling & Simulation", Pomona, California (USA), Vol. 4.
- TEIXEIRA, E.; YAMAKAMI, A. e STANGE, PLÍNIO (1992). *A General Model for the Problem of Workpieces Scheduling in a Flexible Manufacturing Cell*. IFAC Workshop on CIM in Process and Manufacturing Industries. Espoo, Finlândia.
- TEIXEIRA, E. e YAMAKAMI, A. (1993). *Workpieces Scheduling in a Flexible Manufacturing Cell*. Submetido ao IFIP WG 5.7. Working Conference on Evaluation of Production Management Methods.
- WALTER, CLÁUDIO (1990). *UM Método para o Escalonamento Finito de Sistemas Fabris*. Anais do 4º CONAI, São Paulo.
- WILSON, J.M. (1992). *Approaches to Machine Load Balancing in FMS*. J. Opl. Res. Soc., Vol. 43, Nº 5.
- VÁRIOS AUTORES (1989). *Relatório do 1º Encontro Nacional Sobre Projetos CIM ("Computer Integrated Manufacturing")*. Secretaria Especial de Informática, Brasília.

## **BIBLIOGRAFIA**

- AICARDI, M.; DI FEBBRARO, A. and MINCIARDI, R. (1989). *Combined Scheduling and Routing in Discrete Manufacturing Systems*. IFAC Information Control Problems in Manufacturing Technology, Madrid, Spain.
- AKELLA, R.; CHOONG, Y. and GERSHURIN, S.B. (1984). *Performance of Hierarchical Production Scheduling Policy*. IEEE Transactions on Components Hybrids and Manufacturing Technology, 7(3).
- ALTER, S. (1986). *Decision Support Systems for Manufacturing*. Data Processing, 25 (10).
- BLACKSTONE, J.H.; PHILLIPS, D.T. and HOGG, G.L. (1982). *A State-of-the-art Survey of Dispatching Rules for Manufacturing Job Shop Operations*. International Journal of Production Research, 20(1).
- BLAZEURICZ, JACEK; DROR, MOSHE and WEGLARZ, JAN (1991). *Mathematical Programming Formulations for Machine Scheduling: A Survey*. European Journal of Operational Research, 51.
- BOULET, B.; CHHABRA, B.; HARHALAKIS, G.; MINIS, I. and PROTH, J.M. (1991). *Cell Controllers: Analysis and Comparison of Three Major Projects*. Computers in Industry, 16, N° 3.
- BOURNE, D.A. and FOX, M. (1984). *Autonomous Manufacturing: Automating the Job Shop*. IEEE Computer, September.
- BROWN, RICHARD W.; NORTHUP, WILLIAM D. and SHAPIRO, JEREMY F. (1986). *Logs: A Modeling and Optimization System for Business Planning*. Computer Decision Making. G. Mitra (Editor) Elsevier Science Publishers B.V. (North-Holland).
- BUZACOTT, J.A. and YAO, D.D. (1986). *FMS. A Review of Analytical Models*. Manag. Science, 32.
- CHAN, DIND-YU and BEDWORTH, DAND D. (1990). *Design of a Scheduling System for FMC*. Int. J. Prod. Res., Vol. 28, N° 11.
- DAL-EL, E.M. and RABINEVITCH, M. (1988). *Optimal Planning and Scheduling of Assembly Lines*. Int. J. Prod. Res., Vol. 26, N° 9.

- DOUKIDIR, G.I. and PAUL, R.J. (1985). *Research Into Expert Systems to Aid Simulation Model Formulation*. Journal of the O.R. Society, 36.
- FABRYCKY, WALTER J. (1972). *Industrial Operations Research*. Prentice-Hall, Inc. Englewood Cliffs, New Jersey.
- FRENCH, SIMON, (1982). *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*. Ellis Horwood Limited.
- GANT, T.J. (1986). *Lessons for O.R. from A.I.: A Scheduling Case Study*. J. Oper. Res. Soc., Vol. 37, Nº 1.
- GUPTA, YASH P.; GOYAL, SAMEER (1989). *Flexibility of Manufacturing Systems: Concepts and Measurements*. European Journal of Operational Research, 43.
- HAN, MIN-HONG; NA, YOON K. and HOGG, GARY, L. (1989). *Real-time Tool Control and Job Dispatching in Flexible Manufacturing Systems*. Int.J. Prod. Res., Vol. 27, Nº 8.
- HATZIKONSTANTIS, L.; SAHIRAD, M.; RISTIC, M. and BESANT, C.B. (1989). *Interactive Scheduling for a Human-Operated Flexible Machining Cell*. IFAC Information Control Problems in Manufacturing Technology, Madri, Spain.
- HEHL, MAXIMILIAN EMIL (1986). *Linguagem de Programação Estruturada, FORTRAN 77*. McGraw-Hill, SP.
- HOFFMANN, T.R. and SCUDDER, G.D. (1983). *Priority Scheduling with Cost Considerations*. Int. Journal of Prod. Research, 21, 6.
- HOOPER, J.W. and REILLY, K.D. (1982). *An Algorithmic Analysis of Simulation Strategies*. Inter. Journal of Computer and Information Sciences, 11(2).
- JACKSON, R.H.F. and JONES, A.T. (1987). *An Architecture for Decision Making in the Factory of the Future*. Interfaces, Vol. 17, Nº 6.
- JACOBS, F.R. (1983). *The OPT Scheduling System: A Review of a New Production Scheduling System*. Product. Inventory Manag., Vol. 24.



- JACOBS, F.R. (1984). *OPT Uncovered: Many Production Planning and Scheduling Concepts can be Applied with or without the Software*. Ind. Eng., October.
- JACOBS, H.J.; HENTSCHEL, B. and BRÖNNER, B. (1990). *New Logistics and Software for Job Shop Control*. Computers in Industry, 15, Nº 172, October.
- JAFFE, M. (1983). *Decision Support System for Manufacturing*. Info. Systems, 30 (7).
- KARMAKAR, V.S.; KEKRE, S. and FREEMAN, S. (1985). *Lot-sizing and Lead-time Performance in a Manufacturing Cell*. Interfaces, Vol. 15, Nº 2.
- KIM, YEONG-DAE (1990). *A Comparison of Dispatching Rules for Job Shops with Multiple Identical Jobs and Alternative Routeings*. Int. J. Prod. Res., Vol. 28, Nº 5.
- KIMEMIA, J. and GERSHWIN, S.B. (1983). *An Algorithm for Computer Control of a Flexible Manufacturing System*. IIE Transactions, 15 (4).
- KLEEBANK, VIVIAN (1989). *SISEP: Uma Proposta de Sistema Baseado em Conhecimento para Escalonamento da Produção*. Tese de Mestrado, UFRS.
- KUSIAK, ANDREW (1987). *Artificial Intelligence and Operations Research in Flexible Manufacturing Systems*. INFOR, Vol. 25, Nº 1.
- LASSERRE, J.B.; BES, C. and ROUBELLAT, F. (1985). *The Stochastic Discrete Dynamic Lot, Size Problem: An Open-loop Solution*. Oper. Res., Vol. 33, Nº 3.
- LENSTRA, J.K.; KAN, RINNOOY, A.H.G.; BRUCKER, P. (1977). *Complexity of Machine Scheduling Problems*. Annals of Discrete Mathematics, 1.
- LILES, DONALD H. and HUFF, BRIAN L. (1990). *A Computer Based Production Scheduling Architecture Suitable for Driving a Reconfigurable Manufacturing System*. Computers and Industrial Engineering, Vol. 19, Nº 1-4.
- LOW, A.H.; McLEAN, J.A.L.; MILES, P.R.; GORDON, J.C.R. (1986). *MOSIS - A Step Towards the Promised Land of Production Control*. 2º Int. Conf. CAPE.
- MOODY, PATRICIA E. (1990). *Strategic Manufacturing - Dynamic New Directions for the 1990s*. Editor Dow-Jones Irwin. Homewood, Illinois.

- NIEMI, E. and DAVIES, B.J. (1989). *Simulation of an Optimizing FMS-Cell Control System*. Robotics & Computer Integrated Manufacturing, Vol. 5, Nº 2/3.
- NORDSTEN, GUNNAR (1991). *A Volvo Implanta FMS e Obtém Menos Custos e Mais Qualidades*. Máquinas e Metais, Março.
- O'KEEFE, ROBERT M. (1985). *Expert Systems and Operational Research-Mutual Benefits*. J. Apl. Res. Soc., Vol. 36, Nº 2.
- PARADICE, D. and COURTNEY, J.F. (1987). *Casual and non-casual Relationship and Dynamic Model Construction in a Managerial Advisory System*. Journal of Management Information Systems, 3 (4).
- PIRES, SÍLVIO R.T. e AGOSTINHO, OSWALDO L. (1991). *Um Modelo de Seqüenciamento da Produção Celular*. Máquinas & Metais, Outubro.
- PRACT, W.E. (1986). *A Graphical Interactive Structural Modeling Aid for Decision Support Systems*. IEEE Transactions on Systems, Man and Cybernetics, SMC-14.
- RANKY, P.G. (1983). *The Design and Operation of Flexible Manufacturing Systems*. IFS Publications, UK.
- ROWLAND, A.J. and BOUDWIN, N.K. (1986). *A Data and Model Management System in EXXON*. Computer Assisted Decision Making. G. Mitra (Editor). Elsevier Science Publishers B.V. (North-Holland).
- SALKIN, H.M. (1975). *Integer Programming*. Addison-Wesley.
- SCHONBERGER, R.J. (1984). *Just-in-time Production Systems: Replacing Complexity with Simplicity in Manufacturing Management*. Ind. Eng. October.
- SRIKAR, B.; SOUMEN, G. (1986). *A MILP Model for the n-Job, M-Stage Flow-shop with Sequence Dependent Set-up Times*. Int. J. Prod. Res., Vol. 24, Nº 6.
- STEFFEN, M.S. and GREENE, T.J. (1986). *An Application of Hierarchical Planning and Constraint-directed Search to Scheduling Parallel Processors*. In. Proc. IEEE Robotics and Automation Conference.

- STOEVA, STEFKA P. (1990). *A Due Date-based Dispatching Rule for FMS*. Int. J. Prod. Res., Vol. 28, Nº 11.
- TARJAN, R.E. (1978). *Complexity of Combinatorial Algorithms*. SIAM Review, 20.
- TU, Y. and SORGEN, A. (1991). *Real-time Scheduling and Control of Transportation in FMC*. Computers in Industry, 16, Nº 4, August.
- VAITHIANATHAN, R. and MCROBERTS, K. (1982). *On Scheduling in a GT Environment*. Journal of Manufacturing Systems, 1 (2).
- VELAGAPUDI, NAVEEN K. (1992). *Robust Scheduling for Manufacturing Systems*. Computers and Industrial Engineering, Vol. 23, Nº 1-4.
- VERO, S.A. (1983). *Planning in Time: Windows and Durations for Activities and Goals*. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 5 (3).
- WAGNER, H. (1986). *Pesquisa Operacional*. Prentice Hall do Brasil.
- WITTROCH, R.J. (1985). *Scheduling Algorithms for Flexible Flow Lines*. IBM Journal of Research and Development, Vol. 29, Nº 4.
- ZHOU, CHEN and EGBELU, PIUS J. (1989). *Scheduling in a Manufacturing Shop with Sequence-dependent Setups*. Robotics & Computer Integrated Manufacturing, Vol. 5, Nº 1.