

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

UMA ABORDAGEM DE INTELIGÊNCIA ARTIFICIAL  
PARA O PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA.

VINÍCIUS MEDINA KERN

FLORIANÓPOLIS, SETEMBRO DE 1991.



0.196.382-9

UFSC-BU

UMA ABORDAGEM DE INTELIGÊNCIA ARTIFICIAL  
PARA O PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO

VINÍCIUS MEDINA KERN

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO DE

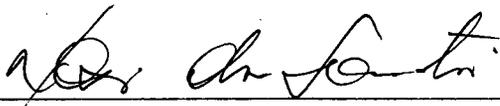
MESTRE EM ENGENHARIA

ESPECIALIDADE ENGENHARIA DE PRODUÇÃO E APROVADA EM SUA FORMA FINAL  
PELO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO



---

RICARDO MIRANDA BARCIA, Ph. D. - ORIENTADOR



---

NERI DOS SANTOS, Dr. Ing. - COORDENADOR DO CURSO

BANCA EXAMINADORA



---

RICARDO MIRANDA BARCIA, Ph. D. - PRESIDENTE



---

SÉRGIO FERNANDO MAYERLE, M. Sc.



---

EDGAR AUGUSTO LANZER, Ph. D.

## AGRADECIMENTOS

Agradeço ao prof. Ricardo Barcia pela orientação, estímulo e disposição, mesmo trabalhando com tempo escasso.

Agradeço a amizade e a estimulante competência e dedicação de professores, colegas e funcionários do Departamento de Engenharia de Produção e Sistemas: ao Roberto Pacheco e ao Paulo Del Grande, pela amizade de sempre, pela programação das rotinas em Turbo C++ (Beto) e incontáveis mãos-na-roda e galhos quebrados. Aos demais colegas do grupo de pesquisa em inteligência artificial agradeço a oportunidade de trabalhar com pessoas competentes, inteligentes e sérias (no bom sentido). Ao professor Sérgio Mayerle, pela capacidade, dedicação e interesse incomuns e à Zelita Chaves, secretária do PPGEF, pelo cuidado e carinho com que trata dos interesses de todos os alunos.

Ao CNPq e à Capes agradeço o auxílio financeiro.



|         |   |    |
|---------|---|----|
| 3.4     | Sistemas Especialistas                                    | 45 |
| 3.4.1   | Estrutura de um Sistema Especialista                      | 48 |
| 3.4.1.1 | Representação de Conhecimento                             | 48 |
| 3.4.1.2 | Motor de Inferência                                       | 48 |
| 3.4.1.3 | Aquisição de Conhecimento                                 | 49 |
| 3.4.1.4 | Esquema de Funcionamento                                  | 49 |
| 3.4.2   | Etapas de Construção                                      | 50 |
| 3.4.2.1 | Definição do Problema                                     | 50 |
| 3.4.2.2 | Aquisição, Representação e<br>Coordenação do Conhecimento | 51 |
| 3.4.2.3 | Mecanismo de Inferência                                   | 53 |
| 3.4.2.4 | Implementação   | 55 |
| 3.4.2.5 | Aprendizagem  | 57 |
| 3.4.3   | Abordagens de Sistemas Especialistas                      | 57 |
| 3.4.3.1 | Abordagem Hierárquica                                     | 57 |
| 3.4.3.2 | Abordagem Não-hierárquica                                 | 58 |
| 3.4.3.3 | Abordagem Baseada em Script                               | 58 |
| 3.4.3.4 | Abordagem Oportunística                                   | 58 |
| 3.4.3.5 | Abordagem Dirigida por Restrições                         | 58 |
| 3.4.3.6 | Abordagem Múltipla ou Tandem                              | 59 |
| 3.4.4   | Sistemas Desenvolvidos                                    | 59 |

Cap. IV - Um Sistema Especialista para Programação da Produção 71

|       |   |    |
|-------|---|----|
| 4.1   | Características de um<br>programa escalonador da produção   | 71 |
| 4.2   | Arquitetura do Sistema                                      | 72 |
| 4.2.1 | Banco de Dados  | 73 |
| 4.2.2 | Banco de Conhecimento                                       | 74 |
| 4.2.3 | Ajuste de Datas   | 76 |
| 4.2.4 | Módulo de Programação por<br>Análise de Restrições de Tempo | 77 |
| 4.2.5 | Módulo Escalonador Difuso                                   | 78 |
| 4.2.6 | Módulo de Controle  | 81 |

Cap. V - Aplicação 84

Cap. VI - Conclusões 93

Cap. VII - Bibliografia 98

|     |                            |     |
|-----|----------------------------|-----|
| 7.1 | Referências Bibliográficas | 98  |
| 7.2 | Bibliografia Suplementar   | 102 |

Anexos

|         |  |     |
|---------|--|-----|
| Anexo 1 | - Glossário de Termos usados em Administração<br>da Produção e Inteligência Artificial | 106 |
|---------|--|-----|

|           |  |         |
|-----------|--|---------|
| QUADRO 1  | - Características da fabricação tradicional  | 14      |
| QUADRO 2  | - Comparação entre empresas japonesas e americanas   | 17      |
| QUADRO 3  | - Benefícios alcançáveis usando JIT  | 17      |
| QUADRO 4  | - Filosofias de fabricação - Quadro comparativo  | 18      |
| QUADRO 5  | - Evolução dos sistemas especialistas  | 60      |
| QUADRO 6  | - Quadro-resumo das aplicações de sistemas especialistas no planejamento, programação e controle da produção           | 61      |
| QUADRO 7  | - Processo de cálculo para tomada de decisão de prioridade no módulo de programação por análise de restrições de tempo | 77 e 78 |
| QUADRO 8  | - Exemplo - Resumo das datas-limite alteradas - Ajuste de datas segundo os roteiros de fabricação                      | 85      |
| QUADRO 9  | - Resumo das datas-limite alteradas II   | 89      |
| QUADRO 10 | - Resumo das datas-limite alteradas III  | 91      |
| QUADRO 11 | - Resumo das datas-limite alteradas IV   | 92      |
|           |  |         |
| TABELA 1  | - Avaliação dos aspectos problemáticos da gestão da produção antes e depois da implantação do sistema MRP              | 28      |
| TABELA 2  | - Variáveis e equações envolvidas em uma formulação do problema geral job shop por programação inteira                 | 36      |
| TABELA 3  | - Ferramentas e linguagens utilizadas na construção de sistemas especialistas  | 56      |
| TABELA 4  | - Uso de técnicas de representação de conhecimento   | 56      |
| TABELA 5  | - Hardware utilizado   | 56      |
|           |  |         |
| FIGURA 1  | - Níveis de programação  | 11      |
| FIGURA 2  | - Sistema Kanban simplificado  | 22      |
| FIGURA 3  | - Ilustração do funcionamento do OPT   | 29      |
| FIGURA 4  | - Analogia do sistema Tambor-Pulmão-Corda  | 30      |
| FIGURA 5  | - Sistemas especialistas são sistemas baseados em conhecimento   | 46      |
| FIGURA 6  | - Estrutura de um sistema especialista   | 50      |
| FIGURA 7  | - Rede semântica   | 52      |
| FIGURA 8  | - Encadeamento para frente   | 54      |
| FIGURA 9  | - Encadeamento para trás   | 54 e 55 |
| FIGURA 10 | - Esquema de funcionamento do Sistema Especialista   | 74      |
| FIGURA 11 | - Janelas de tempo de operações não ordenadas (exemplo)  | 75      |
| FIGURA 12 | - Janelas de tempo para as duas situações possíveis de ordenação   | 75      |
| FIGURA 13 | - Representação dos conjuntos difusos Favorável, Indiferente e Desfavorável  | 80      |
| FIGURA 14 | - Exemplo - Roteiros de fabricação   | 84      |
| FIGURA 15 | - Exemplo - Janelas de tempo iniciais  | 85      |
| FIGURA 16 | - Exemplo - Janelas de tempo intermediárias  | 86      |
| FIGURA 17 | - Exemplo - Janelas de tempo intermediárias II   | 86      |
| FIGURA 18 | - Exemplo - Janelas de tempo intermediárias III  | 87      |
| FIGURA 19 | - Exemplo - Janelas de tempo intermediárias IV   | 89      |
| FIGURA 20 | - Exemplo - Janelas de tempo intermediárias V  | 91      |
| FIGURA 21 | - Exemplo - Janelas de tempo finais  | 92      |

## RESUMO

Esta dissertação apresenta a arquitetura modular de um sistema especialista para programação da produção, agregando técnicas de propagação de restrições e decisões de precedência de operações baseadas em regras, utilizando a teoria de conjuntos difusos.

Fazem parte da pesquisa a descrição das técnicas usuais de administração da produção e estoques e um levantamento dos estudos feitos na área de sistemas especialistas para programação da produção.

O modelo apresentado permite o uso complementar de várias formas de conhecimento. A abordagem de propagação de restrições de tempo, principal estratégia de inferência do sistema, preserva a robustez do programa, mas carece de restrições suficientes para chegar a um programa completo. As regras de liberação e regras e heurísticas de programação tratadas por lógica difusa criam novas restrições de tempo e precedência, podendo ser enunciadas de forma imprecisa.

A programação é feita a nível de operações elementares e o programa resultante é expresso segundo intervalos de tempo para cada operação, suportando pequenos desvios até o limite das folgas.

Um programa de produção com três tarefas e três máquinas é apresentado como exemplo.

ABSTRACT

This dissertation presents an expert system modular architecture for scheduling, matching constraint propagation techniques with priority and heuristic rules handled by fuzzy logic.

This work includes a description of traditional production management techniques and a survey on studies and expert systems developed for scheduling and production management.

In the architecture described, two strategies of inference are used to represent the domain knowledge: the main strategie in the inference engine is time constraints propagation. Using this approach, good results in terms of maintaining the flexibility and robustness of the schedule in course were achieved. Dispatching and heuristic rules weighed by fuzzy logic play the role of creating some new constraints and allowing the process to go on.

Scheduling is done in terms of elementary operations and their 'time windows'. The resulting schedule is presented as critic time intervals for each operation, with possible time slacks. An example is presented.

## I - INTRODUÇÃO

### 1.1 ORIGEM DO TRABALHO

As empresas industriais buscam sobreviver no mercado competitivo produzindo lucros, reagindo às imposições de mercado e procurando nele influir através de estratégias deliberadas. Sua principal atividade é a transformação, através da qual agregam valor aos produtos que vendem.

Dentre as várias funções desempenhadas pelas empresas para atingirem seus objetivos está a programação da produção, que trata da alocação no tempo dos recursos de produção. Tem uma estreita ligação com outras funções: É executada com os dados fornecidos por departamentos de marketing e vendas, está sujeita às limitações físicas das instalações e à disponibilidade de mão-de-obra. Seu efeito determinará decisões sobre gerenciamento de estoques e finanças.

De acordo com Allora [ALL88], "há inúmeros 'pacotes' de 'software' para contabilidade, folha de pagamento, ativo fixo, e assim por diante. Mas há bem poucos de custos e controle da produção".

Desde a década de 50 a pesquisa operacional tem buscado construir algoritmos e métodos que possam representar o ambiente fabril e fornecer uma solução ótima para o problema. Porém, "as técnicas correntes de pesquisa operacional podem não estar à altura, ou ser inadequadas, ou ainda não serem necessárias para muitos ambientes de produção" [GRAB1]. Ainda assim, durante as décadas de 60, 70 e ainda nos anos 90, a literatura continua a registrar publicações propondo soluções de problemas via pesquisa operacional sem, no entanto, obter avanços significativos para a solução dos problemas reais, restringindo-se a casos especiais e aplicações acadêmicas. Com efeito, segundo Graves [GRAB1], "temos visto que há uma lacuna entre a teoria e a prática da programação da produção. Os futuros esforços de pesquisa devem ser dirigidos à redução destas diferenças".

O problema de programação da produção é resolvido na prática por especialistas, pessoas que usam técnicas pouco estruturadas, como: "experiência", "intuição", "bom senso", etc.

O surgimento da inteligência artificial, que tem nos sistemas especialistas o seu "primeiro e mais avançado produto comercialmente viável" [SCH87], tem levado pesquisadores a projetar e construir sistemas baseados no conhecimento desses peritos, em conjunto com técnicas bem estruturadas [BEL88] [BEN88] [BRU86] [FOX84] [KAN87] [KLE89] [KUS87] [KUS88] [LEC87] [PAC90] [SMI86].

A falta de ferramentas efetivas para programação da produção que se utilizem do conhecimento prático, empírico e teórico inerentes ao problema tem restringido o progresso das empresas industriais, sobretudo no Brasil, onde as pequenas e médias empresas, embora responsáveis por significativa parcela da economia nacional, têm deficiências organizacionais e são pouco capazes de absorver mão-de-obra especializada devido ao alto custo e possibilidade de evasão, com a conseqüente perda do investimento para formação do especialista [BAT90] [PAC90].

O desenvolvimento de sistemas especialistas que contemplem a solução de problemas reais de programação da produção, utilizando as diversas formas de conhecimento envolvido e preservando o investimento na formação deste conhecimento constitui um passo importante rumo à capacitação das empresas para a competição.

## 1.2 OBJETIVOS

O objetivo principal do presente trabalho é estudar as técnicas de programação da produção e propor uma arquitetura de sistema baseado em conhecimento para implementar estas técnicas.

São objetivos complementares o estudo das filosofias de fabricação e a implicação da escolha das técnicas de programação da produção frente à filosofia adotada e a estruturação do conhecimento sobre programação da produção compatibilizando-o com a linguagem empregada pela administração da produção.

### 1.3 IMPORTÂNCIA DO TRABALHO

A programação e o controle da produção afetam o desempenho da indústria tanto em relação a grandezas quantificáveis, como volume de perdas, estoques de matéria-prima, trabalho em processo e produtos acabados, retrabalho, etc., quanto a variáveis qualitativas, cuja importância é difícil de precisar, como o índice de atrasos e sua relação com a satisfação do cliente.

No Brasil, mesmo as ferramentas mais difundidas para planejamento, programação e controle são pouco conhecidas e utilizadas pelas empresas.

Em trabalho realizado por Fensterseifer & Bastos [FEN89] sobre a implantação de sistemas MRP/MRP II (Planejamento de Necessidades de Produção/Planejamento de Recursos de Manufatura) nas grandes empresas industriais do Brasil, 32,3% dos responsáveis pelo setor de produção das empresas (de uma amostra de 164 questionários respondidos) declararam não conhecer MRP. Segundo os autores:

*"Este fato surpreende, por tratar-se de grandes empresas industriais e, precisamente, daqueles setores que potencialmente mais se beneficiariam com a utilização de MRP. (...) Com o surgimento de sistemas MRP para microcomputadores, pode-se também prever uma crescente difusão de MRP nas empresas de médio e pequeno porte."*

Em relação às empresas pequenas e médias, um trabalho recente de Batalha & Demori [BAT90] sobre este setor em Santa Catarina conclui que "este controle (da produção) ocorre em grande parte das indústrias, mas quase sempre de maneira parcial e informal".

No entanto, programadores profissionais dão conta da programação da produção, freqüentemente sem uma avaliação da qualidade dos programas ou verificação de alternativas ao programa estabelecido.

O desenvolvimento de sistemas baseados em conhecimento para programação da produção constitui uma estratégia efetiva na busca da melhoria da produtividade. O parágrafo a seguir sintetiza a dificuldade das empresas de ajustarem um sistema e uma equipe de trabalho para a programação da produção:

"Enquanto a busca por alternativas aos sistemas tradicionais de programação da produção é uma prioridade em outras nações, no Brasil a realidade revela que esta etapa ainda está por vir. Mesmo os sistemas tradicionais de programação da produção não possuem a mesma popularidade de que gozam em países desenvolvidos. As empresas enfrentam a questão basicamente de duas formas: formando estes profissionais em seus próprios quadros de funcionários, ou contratando pessoas experientes na utilização de todos os recursos disponíveis (desde princípios básicos de apropriação de custos até sofisticados métodos de pesquisa operacional) para uma adequada programação da produção. No primeiro caso, os custos envolvidos na formação são significativos. Quando optam pela contratação, as empresas enfrentam problemas com a escassez e a conseqüente valorização deste tipo de profissional. Além destes fatos, em ambas alternativas de resolução da questão as empresas estão suscetíveis a mudanças por uma eventual saída do perito" [PAC90].

#### 1.4 METODOLOGIA

Esta dissertação integra-se ao trabalho do grupo de pesquisa em Inteligência Artificial do Programa de Pós-Graduação em Engenharia de Produção da UFSC. Foram defendidas dissertações sobre áreas básicas do conhecimento em I.A., como: "Sistemas de Apoio à Decisão: Uma Aplicação na Administração do Capital de Giro sob Inflação" [LUN91], "Uma Interface em Linguagem Natural para um Sistema Especialista de Administração de Capital de Giro" [MON91] e "Tratamento de Imprecisão em Sistemas Especialistas" [PAC91]. Dissertações sobre aplicações de I.A.: "Um sistema de Treinamento para Operadores de Reservatórios Baseado em Técnicas de Inteligência Artificial" [THE88], "Inteligência Artificial aplicada na Engenharia Rodoviária" [ROD90] e "Um Sistema Especialista para auxiliar na elaboração de recomendações para o controle da sarna da macieira" [TOD91].

Este trabalho primeiramente descreve o problema de programação da produção e seu contexto, a seguir aborda as técnicas existentes para a solução do problema e propõe uma arquitetura de um sistema especialista, conjugando técnicas de pesquisa operacional com regras definíveis pelo usuário.

### 1.5 ESTRUTURA

O trabalho foi escrito em sete capítulos: Introdução, O problema de programação da produção, Abordagens utilizadas na solução do problema, Um sistema especialista para programação da produção, Aplicação, Conclusões e Bibliografia.

A introdução, feita no capítulo, 1 está dividida em: Origem, Objetivos, Importância, Metodologia, Estrutura e Limitações do trabalho.

O capítulo 2 busca descrever o problema de programação da produção e o contexto no qual está inserido. O capítulo divide-se em: Definição, Classificação do problema de programação da produção, Níveis hierárquicos de programação, Filosofias de fabricação e Técnicas e métodos usuais.

As abordagens ao problema de programação da produção, no capítulo 3, são arroladas e descritas brevemente. É introduzida a abordagem de Sistemas Especialistas. O capítulo divide-se em: Otimização, Propagação de restrições, Técnicas heurísticas e Sistemas Especialistas.

O capítulo 4 contém o detalhamento do sistema proposto, com os módulos de Propagação de restrições e de Regras de prioridade e o Módulo de controle compondo o Motor de inferência.

O capítulo 5 apresenta um exemplo de programação de tarefas utilizando a arquitetura detalhada no capítulo anterior.

No capítulo 6 são colocadas as conclusões e recomendações para futuros trabalhos de pesquisa nesta área.

A bibliografia, no capítulo 7, registra as referências bibliográficas feitas no decorrer do trabalho e indica também uma bibliografia suplementar.

### 1.6 LIMITAÇÕES

A prática da programação da produção apresenta métodos e técnicas muito variados, especialmente levando em conta a maneira informal como é feita na maioria das indústrias brasileiras. Neste trabalho são descritas as técnicas mais difundidas mundialmente.

Naturalmente, não abrangem todas as possibilidades de gestão da produção.

O sistema apresentado aqui deve ser considerado antes como uma arquitetura onde serão implementadas técnicas de programação da produção. Portanto, não estão definidos ambiente específico de fabricação ou outras características específicas de cada ambiente produtivo, o que deverá ser feito para cada aplicação real.

O sistema especialista detalhado no capítulo 4 contempla ambientes de fabricação discreta. A fabricação contínua não pode ser tratada por este sistema.

## II - O PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO

### 2.1 DEFINIÇÃO

A programação da produção está relacionada à alocação eficiente de recursos no tempo para a manufatura de bens. Bensana, Bell e Dubois [BEN88] definem a tarefa como: "dado um conjunto de equipamentos e restrições tecnológicas e dadas as necessidades de produção em termos de quantidade e qualidade do produto e restrições de tempo, encontrar uma seqüência viável de operações nos vários equipamentos, que satisfaça as necessidades de produção".

Para Graves [GRAB1], Programação da Produção pode ser definida como a alocação no tempo dos recursos de produção disponíveis, visando a satisfazer da melhor forma possível um dado conjunto de critérios.

A programação geralmente se refere a um período de tempo como uma semana ou um mês e tem como propósitos básicos adequar o processo de produção para atender à demanda do mercado, coordenar as operações, tanto as internas à firma quanto aquelas que dependem de fornecedores, e estimular o aperfeiçoamento nas operações [HAL88].

### 2.2 CLASSIFICAÇÃO DO PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO

Existem várias formas de classificação, feitas com o intuito de delinear as características de cada ambiente produtivo, suas restrições e objetivos. As classificações descritas aqui encontram-se notadamente no âmbito da pesquisa operacional. Graves [GRAB1] propõe três critérios para classificar o problema de programação da produção:

- Geração de pedidos
- Complexidade do processo produtivo
- Critério de programação

O problema, quanto à geração de pedidos, pode ser um "open shop", quando os pedidos a serem programados são gerados diretamente por ordens de compra de clientes, ou um "closed shop", quando o pedido tem origem numa decisão de substituição de estoques vendidos.

O critério relativo à complexidade do processo produtivo refere-se ao número de etapas de processamento associadas a cada item ou tarefa. Distinguem-se quatro casos:

- Um estágio, um processador - O problema mais simples, onde as tarefas são realizadas num mesmo processador, completadas em uma operação.

- Um estágio, processadores paralelos - Similar ao caso anterior, cada tarefa é completada em uma operação. Pode ser executada em qualquer dos equipamentos disponíveis.

- Múltiplos estágios, "flow shop" - Cada tarefa pode ser executada num mesmo conjunto de equipamentos, com idêntica ordem de precedência das etapas de processamento. É o caso típico da produção em linha.

- Múltiplos estágios, "job shop" - É o tipo de problema mais geral. Não existe um caminho obrigatório nas etapas de processamento de um item e rotas alternativas são permitidas. Em geral, "job shop" refere-se a um número grande de tarefas com roteiros diversos e competindo por um conjunto de máquinas comum. O problema de tamanho de lote, neste caso, está resolvido ou é irrelevante [GER66].

Os dois primeiros casos, embora sejam assunto constante de artigos de pesquisa operacional, dificilmente serão observados num ambiente real de manufatura.

Quanto ao critério de programação, distinguem-se como parâmetros de avaliação do escalonamento produzido:

- Custo - Associa a um programa de produção os custos relacionados: custos fixos de ajuste de máquina ("setup"), excesso ou falta de estoques, etc., além do próprio custo de geração e monitoramento do programa de produção.

- Desempenho - Pode ser medido de várias formas, sendo comuns: nível de utilização dos recursos de produção, percentagem de tarefas atrasadas, tempo médio de processamento das tarefas, tempo de atraso máximo ou médio, etc.

Segundo Graves [GRA81], na maioria dos ambientes de fabricação são considerados ambos critérios. A literatura para "open shops" trata principalmente do critério de desempenho, enquanto que para "closed shop" concentra-se no critério de custo.

Graves [GRA81] coloca ainda duas outras dimensões de classificação:

- Natureza da especificação do pedido - a especificação dos pedidos pode ser determinística ou estocástica. Por exemplo, para "open shop", o tempo de processamento de cada etapa da tarefa pode ser conhecido ou ter distribuição de probabilidade conhecida. Para "closed shop", a demanda, que gera a necessidade de repor estoques, pode ser estocástica ou determinística.

- Ambiente de programação - pode ser estático (os pedidos de produção são conhecidos e não sofrem mudança) ou dinâmico (podem ser adicionados pedidos e a programação deve contar com estes possíveis pedidos adicionais futuros).

Uma forma de classificação, bastante prática para estabelecer uma nomenclatura é proposta por Conway et alii [CON67]. Segundo os autores, um problema específico de programação da produção é descrito por quatro tipos de informação:

- 1) As tarefas e operações a serem processadas.
- 2) O número e tipo de máquinas que compõem a fábrica.
- 3) Restrições à maneira como fazer as atribuições.
- 4) Critérios sob os quais um programa será avaliado.

Usa-se, então, uma notação de quatro parâmetros para representar o problema de programação da produção: A/B/C/D, onde:

A Descreve o processo de chegada de tarefas. Em ambientes estáticos, onde as tarefas chegam simultaneamente a uma fábrica ociosa e disponível, A representará o número de

tarefas. Quando o ambiente for dinâmico, onde as tarefas tem chegada intermitente, A especificará a distribuição de probabilidade dos tempos entre chegadas.

- B Descreve o número de máquinas. Se for usado  $m$ , representa um número arbitrário de máquinas.
- C Descreve o fluxo de produtos dentro da fábrica.
- D Descreve o critério sob o qual o programa será avaliado.

Assim, um problema descrito pelos parâmetros  $n/2/F/F_{max}$ , conhecido como 'problema de Johnson', indica: programação de um número arbitrário de tarefas, em duas máquinas, ambiente *flow shop*, de modo a minimizar o máximo tempo de fluxo de processamento. O problema geral *job shop*, programação de  $n$  tarefas em  $m$  máquinas, com rotas arbitrárias e de modo a completar a última tão logo quanto possível, é representado por:  $n/m/G/F_{max}$ .

### 2.3 NÍVEIS HIERÁRQUICOS DE PROGRAMAÇÃO

Na seção 2.1, a programação da produção foi definida no âmbito da pesquisa operacional, relacionando-a com o seqüenciamento de operações nas máquinas e com suas datas de início e término.

Na prática e na literatura de administração da produção, porém, o enfoque é diferente, levando em conta o fato de que muitas indústrias não chegam a preparar programas com seqüenciamento de operações, seja por decisão estratégica ou por não haver disponibilidade de um sistema de programação aliado a um sistema de informação eficiente.

A perda de detalhe e precisão, neste caso, pode ser compensada por um ganho em agilidade e eliminação de burocracia.

Burbidge [BUR81] destaca três níveis de programação e controle da produção:

No nível de planejamento da produção é elaborado o Plano Mestre de Produção (PMP), que recebe informações sobre produtos vendidos, previsão de vendas, estratégia de marketing, etc. e tem como saída um plano com os volumes de produtos acabados (finais) a serem produzidos a cada período.

No nível de emissão de ordens são planejadas as entradas de materiais de fornecedores e os volumes de produção de peças de cada setor produtivo. São expedidas ordens de fabricação e requisições de compra de material, indicando: peça, quantidade e data prevista para estar disponível para montagem. Este plano é feito por um sistema do tipo MRP (Materials Requirements Planning - Planejamento de Necessidades de Materiais, descrito na seção 2.5).

No nível de liberação é feito o seqüenciamento nas máquinas ou centros de trabalho e, em alguns casos, planejamento de datas de início e término de operações. Um programa de carga define o volume de produção por máquina para atender as datas previstas. A programação neste nível está coerente com a definição dada pela pesquisa operacional, que considera os níveis anteriores como planejamento.

A figura 1 apresenta um diagrama mostrando esta hierarquia, desde o nível estratégico:

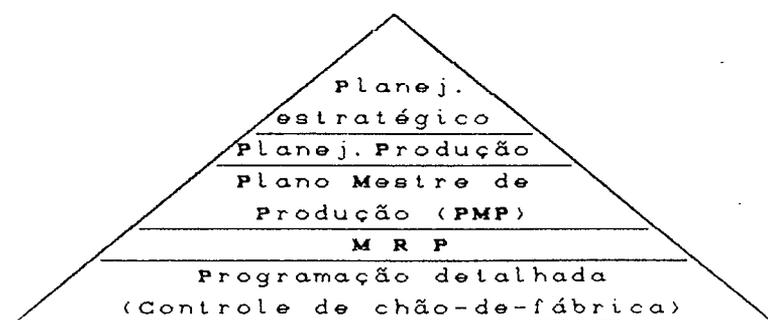


FIGURA 1 - NÍVEIS DE PROGRAMAÇÃO

FONTE: (TER85)

Previsão, análise de mercado e dados reais sobre pedidos de clientes são utilizados para executar o planejamento estratégico da produção. A utilização de dados reais, exclusivamente ou não, vai depender do tipo de indústria. Quando os produtos devem ser supridos pelo estoque (ítems fabricados para estoque) ou em tempo menor do que seu *lead time*, o plano mestre de produção é desenvolvido à base de previsão de demanda [TER85].

Tersine [TER85] distingue planejamento da produção de plano mestre de produção, ressaltando que pode ser difícil separá-los. O

primeiro trata de planejamento agregado, famílias de produtos, considerando a capacidade total da fábrica. O Plano Mestre de Produção mostra itens finais, produtos específicos, considerando as capacidades dos departamentos e centros de trabalho. O horizonte de tempo considerado é, em geral, semanas ou meses.

O PMP é concluído por tentativas (*rough cut capacity planning*), até que seja concluído um plano viável. Seus dados irão alimentar, então, um sistema do tipo MRP (Planejamento de Necessidades de Materiais).

Feito o planejamento de datas e quantidades de itens finais, um sistema de Planejamento e Controle da Produção (PCP) e controle de estoques trata do controle de chão-de-fábrica.

O problema de programação tratado aqui, como definido em 2.1, é aquele referente à programação no nível de liberação, para uso no chão-de-fábrica. O programa de produção é o mais detalhado dentre os programas expostos acima.

## 2.4 FILOSOFIAS DE FABRICAÇÃO

As empresas ocidentais têm, caracteristicamente, uma estrutura concebida a partir das idéias de Ford, Taylor e Fayol. São baseadas na economia de escala, priorizando a utilização da estrutura existente na empresa e necessitando de previsões de vendas para o planejamento da produção.

A produção obedece às previsões, constituindo estoques, buscando a minimização dos custos totais da estrutura de produção.

A partir de meados da década de 70 [ANT89], estruturou-se uma nova filosofia de administração da produção e dos materiais, ganhando espaço de discussão e provocando iniciativas práticas entre empresas ocidentais, inclusive no Brasil, sobretudo devido ao sucesso observado nas empresas japonesas. As técnicas e metodologias usadas pelos japoneses passaram a influenciar o modo ocidental de fabricar.

Nesta seção, apresentam-se as duas grandes linhas existentes quanto a filosofias de fabricação: A filosofia tradicional ou *Just-In-Case* e a filosofia *Just-In-Time* ou "Justo-A-Tempo", também identificada pelo acrônimo JIT.

Antes do surgimento e difusão do JIT, não se encontrava na literatura menção ao termo "filosofia de fabricação", uma vez que as técnicas e métodos existentes serviam, de modo geral, aos objetivos tradicionais da maneira ocidental de administrar a produção.

Ainda hoje o termo "filosofia" desagrada a uns e outros. Voss [VOSXX] procura esclarecer a questão com a seguinte colocação: "O sistema JIT não é uma técnica ou mesmo um conjunto de técnicas para manufatura, mas uma abordagem global ou filosofia que abraça novas e velhas técnicas".

A literatura trata programação da produção e filosofias de fabricação geralmente de forma estanque, não evidenciando a forma como os assuntos se interrelacionam.

As filosofias de fabricação são apresentadas em suas linhas gerais, destacando-se as implicações relativas à programação da produção.

#### 2.4.1 Filosofia Tradicional ou "Just-In-Case"

A filosofia tradicional de fabricação foi a base do desenvolvimento das grandes empresas da América do Norte e Europa, baseada nas idéias de Ford, Taylor e Fayol.

A estrutura produtiva de uma empresa, segundo esta filosofia, está dividida por funções e as linhas de montagem são dedicadas.

O objetivo é obter ganho de escala, com a fabricação em grandes lotes, padronização e redução de variedade, acarretando a formação de estoques. Este objetivo é atingido maximizando a utilização dos meios de produção envolvidos, minimizando sua ociosidade.

É comum, neste tipo de empresa, o uso do conceito de "lote econômico" para cada item, desde os materiais comprados, os produtos intermediários fabricados, até os produtos finais.

Outro aspecto importante é que, dada a organização rígida do sistema produtivo e a ênfase na redução de custos, ocorre uma tendência a fabricar somente uma gama limitada de produtos, visto que a diversificação traz consigo custos (ajustes de máquinas, troca de ferramentas, etc.) e uma perda da produtividade devida ao

efeito de escala.

São bastante difundidos na América do Norte, como sistemas de operacionalização da filosofia tradicional, os sistemas de informação do tipo MRP (Materials Requirements Planning - Planejamento de Necessidades de Materiais) e MRPII (Manufacturing Resources Planning - Planejamento de Recursos de Manufatura) (vide seção 2.5), ainda pouco usados no Brasil, relativamente.

Estes sistemas não fazem a programação ou o controle da produção, apenas contabilizam dados sobre tempos, instantes em que cada item será necessário e quantidades para cada item, lote e período.

Sendo um sistema de informações externo ao ambiente produtivo, o MRP é uma ferramenta de apoio à decisão extremamente dependente da existência de dados fidedignos e consistentes com a realidade do chão-de-fábrica.

Uma indústria funcionando segundo a filosofia tradicional de fabricação apresenta as características descritas no quadro 1:

#### QUADRO 1 - CARACTERÍSTICAS DA FABRICAÇÃO TRADICIONAL

a) Constituição de estoques de matérias-primas, de produtos em processo e de produtos acabados, como forma de "amortecer" as aleatoriedades do sistema produtivo, como variações na demanda, problemas de qualidade, atrasos. etc.

Estes estoques de segurança tendem a acobertar uma série de problemas intangíveis e de difícil previsão, como des-balanceamento da linha de produção, ineficiência na manutenção dos equipamentos, deficiências no processo de aquisição dos materiais, etc., dificultando a detecção dos problemas.

b) Planejamento e controle centralizados e externos ao sistema produtivo propriamente dito.

O controle de qualidade e a manutenção dos equipamentos são realizados por órgãos específicos, atividades que não agregam valor aos produtos e não contribuem para o compromisso e a responsabilidade dos trabalhadores para com a qualidade do seu trabalho.

c) Necessidade de mão-de-obra especializada e pouco flexível a trocas funcionais.

Isto pode provocar subutilização da capacidade de trabalho dos operários, encorajando a uma rápida e constante rotatividade. Além disso, o sistema produtivo torna-se mais vulnerável à ação de grandes organizações de trabalhadores, que poderão provocar atrasos, especialmente em empresas de grande porte, que são mais sensíveis a oscilações no ritmo de produção.

continua ...

## QUADRO 1 (continuação)

d) Escolha de fornecedores feita pelo critério do mínimo preço em geral, ficando os problemas de qualidade, atrasos nas entregas e deficiências no processo de compra absorvidos pelos estoques de segurança.

Ainda, a inspeção de qualidade é feita baseada em dados estatísticos que aceitam ou rejeitam o lote como um todo. Isto é, pode-se aceitar uma certa quantidade de materiais defeituosos que acarretarão problemas na linha de produção.

e) Pouca flexibilidade quanto às oscilações da demanda do mercado, em virtude de a organização estar baseada em idéias de tarefas e máquinas especializadas e de linha de montagem dedicada.

Fonte: [ANT89]

#### 2.4.2 Filosofia "Just-In-Time" (JIT)

O *just-in-time* tem sua origem na companhia Toyota, do Japão, durante a década de 60 [VOSXX].

O *Kanban*, principal ferramenta do JIT, foi inspirado no supermercado americano, onde as prateleiras são reabastecidas assim que se esvaziam. Uma vez que o espaço para cada item era limitado, mais itens eram trazidos apenas quando houvesse necessidade de tê-los [ESP88].

Voss [VOSXX] define JIT como "uma abordagem que assegura que quantidades certas são compradas e feitas no tempo certo e com qualidade adequada, e que não haja perda."

Desenvolvida a partir de uma realidade que combinava abundância de mão-de-obra e escassez de espaço, matérias-primas e recursos financeiros, a filosofia JIT tem enfoques diferentes daqueles da filosofia tradicional. Dirigido a um mercado geralmente menor do que a capacidade de produção da fábrica, o JIT visa a responder às variações do mercado de forma rápida, ao invés de privilegiar o ganho de escala.

Atividades comuns no sistema tradicional, como o controle de qualidade e a manutenção dos equipamentos feitos por departamentos especializados e a movimentação de materiais entre pontos intermediários de estoque não devem acontecer no ambiente JIT. Com efeito, "o princípio básico da filosofia *just-in-time* na

administração da produção é que não se deve fazer nada que não adicione valor aos produtos" [ANT89].

Uma fábrica tradicional deve passar por uma transformação radical no seu lay out, na forma de administrar a produção e até mesmo nos seus aspectos culturais para que possa ser implementada a filosofia JIT e suas técnicas associadas:

*Esta transformação da fábrica tradicional só será possível através da flexibilização da estrutura produtiva visando, simultaneamente, a uma maior diversificação dos produtos para atender às flutuações do mercado e a uma redução dos custos associados, sem perder de vista a qualidade dos produtos [ANT89].*

A implantação do JIT pressupõe a adoção de uma nova sistemática de controle de qualidade, conhecida como TQC - Total Quality Control (Controle da Qualidade Total), onde cada posto de trabalho deve controlar a qualidade do que lhe é fornecido e ser responsável pela qualidade do que produz. Este controle geralmente é feito pelos próprios operários.

A programação da manutenção preventiva dos equipamentos, preferentemente à corretiva, motivada por falha, evita a interrupção do processo produtivo, sendo por isto considerada também um pré-requisito para a implantação do JIT.

A redução dos tempos de ajuste de máquina (set-up), a produção em lotes pequenos, a capacitação dos operários para realizarem diversas funções e a divisão da fábrica em células de manufatura são requisitos que têm íntima relação com o objetivo de aumento da flexibilidade. As Células Flexíveis de Manufatura, que têm na Tecnologia de Grupo uma base analítica para sua implementação, possibilitam um melhor fluxo da produção do que os sistemas tradicionais, o que é um requisito do JIT. A redução dos tempos de ajuste de máquina (set-up) permite a manufatura em lotes muito pequenos. Lotes pequenos, por sua vez, tornam o fluxo contínuo mais fácil de atingir.

Uma vez que isto tenha sido implementado, a próxima técnica a ser usada é "puxar" a produção. Isto é feito através do Kanban (seção 2.5), uma ferramenta de apoio visual para o controle da produção.

A consecução dos objetivos da filosofia JIT e a implementação das técnicas é feita de forma iterativa, reduzindo

progressivamente os estoques intermediários, resolvendo os problemas criados, dando ênfase sobretudo ao envolvimento das pessoas [VOSXX].

As mudanças culturais necessárias incluem a formação de Círculos de Controle de Qualidade (CCQ's), que motivam seus integrantes a participarem das decisões e da busca pela qualidade. A mentalidade da alta e média administração deverá contribuir para a efetivação da gerência por consenso e para a delegação de maiores responsabilidades em todos os níveis. O quadro 2 compara as empresas japonesas e norte-americanas quanto aos aspectos culturais:

QUADRO 2 - COMPARAÇÃO ENTRE EMPRESAS JAPONESAS E AMERICANAS

| ORGANIZAÇÕES JAPONESAS                          | ORGANIZAÇÕES AMERICANAS                     |
|---|---|
| - EMPREGO VITALÍCIO                             | - EMPREGO A CURTO PRAZO                     |
| - AVALIAÇÃO E PROMOÇÃO LENTAS                   | - AVALIAÇÃO E PROMOÇÃO RÁPIDAS              |
| - TRAJETÓRIAS DE CARREIRA<br>NÃO-ESPECIALIZADAS | - TRAJETÓRIAS DE CARREIRA<br>ESPECIALIZADAS |
| - MECANISMOS DE CONTROLE<br>IMPLÍCITOS          | - MECANISMOS DE CONTROLE<br>EXPLÍCITOS      |
| - TOMADA DE DECISÃO COLETIVA                    | - TOMADA DE DECISÃO INDIVIDUAL              |
| - RESPONSABILIDADE COLETIVA                     | - RESPONSABILIDADE INDIVIDUAL               |
| - INTERESSE HOLÍSTICO                           | - INTERESSE SEGMENTADO                      |

FONTE: OUCHI, William G. - Teoria Z: Como as empresas podem enfrentar o desafio japonês, São Paulo, Nobel, 1985. Citado por [YUK88].

Voss [VOSXX] classifica os benefícios auferidos por uma indústria trabalhando sob a filosofia JIT conforme o quadro 3:

QUADRO 3 - CLASSIFICAÇÃO DOS MAIORES BENEFÍCIOS ALCANÇÁVEIS USANDO JIT.

- |   |
|---|
| <ol style="list-style-type: none"> <li>1 Redução dos estoques intermediários</li> <li>2 A Flexibilidade aumenta</li> <li>3 Redução de Matéria-prima</li> <li>4 Melhoria da Qualidade</li> <li>5 Aumento de Produtividade</li> <li>6 Diminui necessidade de espaço</li> <li>7 Menores Despesas Gerais</li> </ol> |
|---|

FONTE: [VOSXX]

Os princípios JIT podem ser aplicados entre companhias, bem como dentro de uma companhia.

A compra JIT, uma aplicação do JIT externamente à empresa,

requer o suprimento de materiais em quantidades pequenas e exatas, em intervalos frequentes e com qualidade total.

Dentre as técnicas específicas da compra JIT destacam-se: uso de um só fornecedor, uso de containers padronizados, entrega no ponto de utilização, Kanban de compra, certificado de fornecedor e, principalmente, confiança mútua.

Cabe ressaltar que o uso do computador é menos vital do que num sistema de manufatura tradicional. A implantação do JIT pode prescindir do computador [SCHXX].

#### 2.4.3 Quadro Comparativo

No quadro 4, a seguir, está esquematizada a descrição das principais características das duas grandes linhas filosóficas da administração da produção.

Quanto à sua aplicação, certamente elas não são mutuamente exclusivas, especialmente se levarmos em conta o grande número de empresas ocidentais que vêm implantando programas para passar gradualmente a produzir segundo a filosofia JIT.

Como organização, uma empresa funcionando sob filosofia JIT tem melhor homeostase; é capaz de responder, reagir e adaptar-se às mudanças do meio com maior agilidade. A combinação de custos baixos com qualidade alta, mais a redução do volume de material em processo com o conseqüente aumento da circulação de capital dá a estas empresas melhores condições de sobrevivência no meio competitivo.

QUADRO 4: FILOSOFIAS DE FABRICAÇÃO - QUADRO COMPARATIVO.

|            | J I C   | J I T   |
|------------|---|---|
| PRIORIDADE | Para a estrutura de produção da empresa, respeitando as restrições impostas pelo mercado. | Ao mercado, respeitando as restrições impostas pela estrutura de produção da empresa. |

QUADRO 4: (continuação)

|                                   | J I C   | J I T  |
|-----------------------------------|---|--|
| LAY OUT                           | No caso mais comum, organizado por processo e, no caso de produção em série, organizado por produto.  | A fim de minimizar o tempo de movimentação das partes, está organizado por produto.  |
| OPERA-<br>CIONALIZAÇÃO            | Extremamente sensível à precisão dos dados de entrada (tamanho do lote, previsão de vendas, tempo-padrão, etc).                                       | Pouco sensível à precisão dos dados de entrada. O sistema de informações é simplificado e modulado em função da demanda.                                 |
| PREVISÕES<br>DE<br>VENDAS         | Dimensionam o sistema produtivo e governam a programação da produção.   | Dimensionam o sistema produtivo, com a programação feita pela estrutura de produção, em função da demanda.   |
| SISTEMA DE<br>INFORMAÇÕES         | Agiliza e orienta a programação da produção externamente (PCP).   | Agiliza e orienta a programação da produção internamente (painéis visuais).  |
| TAMANHO<br>DOS<br>LOTES           | Produção em grandes lotes, para permitir a otimização da utilização da estrutura de produção.   | Produção em pequenos lotes, para maior adaptação ao mercado consumidor e eliminação de estoques.   |
| ESTOQUES                          | Estoques (iniciais, intermediários e finais) são altos.   | Os estoques são minimizados.   |
| TEMPO DE<br>AGREGAÇÃO<br>DE VALOR | Ao otimizar a estrutura de produção, os tempos mortos são relegados a um segundo plano. Somente 0.5 a 2.5% do tempo são usados na agregação de valor. | Lotes de produção tendendo à unidade, c/ redução drástica nos tempos mortos. Alta taxa de agregação por tempo do produto na fábrica. Lead time reduzido. |

QUADRO 4: (continuação)

|                               | J I C  | J I T   |
|-------------------------------|--|---|
| CONTROLE DE QUALIDADE         | A partir de dados estatísticos, considerando possíveis defeitos. O Lote é aceito ou rejeitado por análise estatística. Índices de defeitos e retrabalhos de 8 a 10% na indústria eletrônica americana [BUS88]. | Controle total e feito pelos próprios empregados. O operador subsequente é um "cliente" que rejeita as peças defeituosas. Os índices de defeitos e retrabalhos na indústria eletrônica japonesa chegam a menos de 1% [BUS88]. |
| MANUTENÇÃO                    | Centralizada em um Depto. específico. A Manutenção é preventiva e corretiva.   | Descentralizada e tem a participação do operador. Teoricamente, manutenção preventiva.  |
| CONTROLE DE PRODUÇÃO          | Externo ao processo produtivo e realizado sobre resultados agregados, dificultando a compreensão das causas reais dos desvios.   | Localizado e instantâneo, identificando causas com ações corretivas rápidas e efetivas. Controle interno fazendo parte do processo produtivo.   |
| PRODUÇÃO                      | "Empurrada". Fabrica-se de acordo com as previsões para a constituição de estoques, buscando minimizar os custos totais da estrutura de produção [ANT89].  | "Puxada". A partir da venda de um produto desencadeia-se, do fim para o início do processo, um sistema de informação que permite a reposição instantânea dos componentes [ANT89].   |
| SIST. USUAL DE OPERACIONALIZ. | MRP II<br>OPT  | KANBAN  |
| CADÊNCIA DE PRODUÇÃO          | Dada pela previsão de vendas, respeitando restrições de estrutura de produção.   | Dada pelo mercado, respeitando restrições de gargalo que definem a capacidade produtiva.  |

Fonte: [PAC90], [ANT89], [YUK88] e [BUS88]

## 2.5 MÉTODOS E FERRAMENTAS USUAIS PARA GESTÃO DA PRODUÇÃO

A programação da produção é feita geralmente por programadores humanos, baseados na sua habilidade e experiência.

Dentre os sistemas utilizados para planejamento, programação e controle da produção é controle de estoques, os mais difundidos são: MRP (Materials Requirements Planning - Planejamento de Necessidades de Materiais) e MRPII (Manufacturing Resource Planning - Planejamento de Recursos de Manufatura), o sistema de produção Just-In-Time (JIT ou justo-a-tempo) e o sistema OPT (Optimized Production Technique - Tecnologia da produção otimizada).

### 2.5.1 Kanban

*Kanban* é uma palavra japonesa que significa cartão, literalmente, uma chapa ou registro visual usado como meio de comunicação, de transmissão de informações e idéias [ESP88].

Entende-se como um sistema que utiliza registros físicos para autorizar todo movimento ou produção de material. As operações são autorizadas somente quando há necessidade de executá-las.

Schonberger [SCHXX] relata um caso onde o kanban é implementado como bolas de golfe coloridas transportadas através de tubos. Outras formas são possíveis como alternativas ao uso de cartões, como sinais luminosos ou manuais.

O kanban pode ser uma ferramenta a ser implementada num ambiente just-in-time, mas não deve ser confundido com este. O JIT é uma filosofia de fabricação que engloba vários elementos e pode ter um sistema kanban como um destes. Porém, o kanban só pode existir no ambiente JIT, pois o que o caracteriza é a maneira de "puxar" a produção, fabricando só a quantidade necessária e no momento certo, e isto o diferencia de qualquer outro sistema que utilize cartões dentro da filosofia tradicional de fabricação.

A princípio, é uma técnica que prescinde de uma programação da produção feita externamente e do uso de computadores.

Tipos de Kanban e funcionamento

Há dois tipos de kanban, basicamente: o de cartão simples e o de dois cartões.

O sistema de dois cartões é o adotado na Toyota. Para cada container há um P-kanban ou kanban de produção para uso no centro de trabalho onde o material será processado. O kanban de transporte ou retirada (C-kanban) autoriza a movimentação entre os centros de trabalho. O sistema está esquematizado na figura 2.

O sistema de cartão simples utiliza apenas o kanban de retirada ou de transporte. É uma combinação das sistemáticas de "puxar" e "empurrar" a produção. Os produtos são fabricados ou montados de acordo com um planejamento, mas o reabastecimento para os centros de trabalho é autorizado pelos cartões de transporte. Os itens são "empurrados", enquanto os centros de trabalho "puxam" seus suprimentos [ESP88].

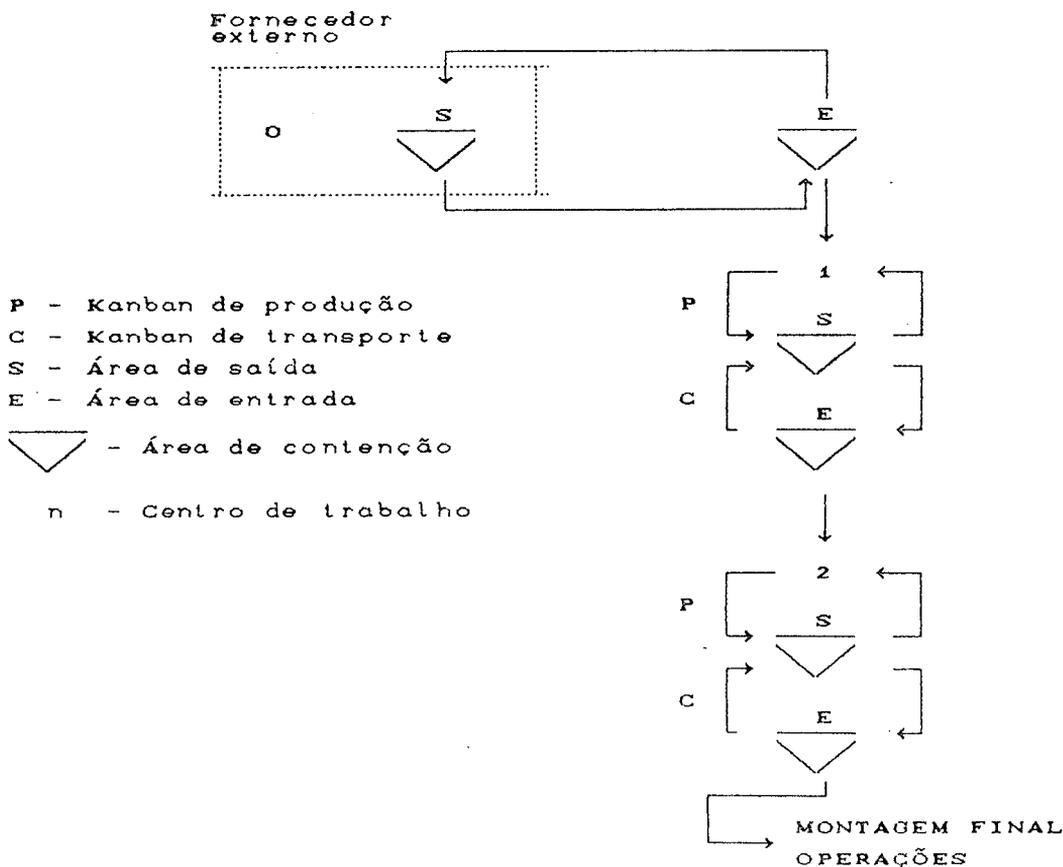


FIGURA 2 - SISTEMA KANBAN SIMPLIFICADO

FONTE: [ESP88]

Esparrago [ESP88] cita ainda dois tipos de Kanban que combinam características do MRP: *Synchro MRP*, criado pela Yamaha Motors, onde os cartões são específicos para cada centro de trabalho, feitos por um computador que também gera um plano diário para cada centro de trabalho. Há também o *Micro Kanban*, um módulo JIT que pode ser adicionado a um sistema MRP já instalado.

#### Benefícios do sistema kanban

Um dos objetivos do sistema kanban é reduzir o estoque de material em processamento até um nível no qual cada centro de trabalho esteja trabalhando com somente um container. A quantidade de estoque intermediário pode ser diminuída também através da redução do tamanho dos containers.

O uso do kanban também facilita a identificação de problemas que ficam acobertados pelos estoques intermediários nos sistemas tradicionais. Os próprios operários podem tomar as decisões de quando emitir kanban de produção ou de transporte, quantos containers usar, etc., favorecendo a melhoria do moral da equipe de trabalhadores.

Uma revisão sistemática da estrutura do Kanban, seus processos e fluxos, permite à empresa realizar melhorias contínuas na qualidade. O sistema força a revelação dos problemas. Quando se atinge um fluxo harmônico de produção, o número de cartões emitido pode ser reduzido e os efeitos observados.

#### Limitações do sistema

Sendo uma técnica desenvolvida na cultura e ambiente japoneses, sua aplicação nas empresas ocidentais encontrará mais limitações. Porém, mesmo que o modelo japonês não seja seguido tão rigidamente, a estrutura básica do Kanban pode ser adaptada às características comportamentais e necessidades ocidentais.

Só existe Kanban dentro do JIT, como um sistema de "puxar" a produção. Assim, só faz sentido se o tempo de "puxar" os materiais para os centros de trabalho é adequado, o que requer tempos de ajuste de máquina (setup) e tamanhos de lote pequenos.

O Kanban se aplica à produção discreta, não a processos contínuos. É mais adequado, também, aos materiais com maior volume de utilização; os itens pouco utilizados podem ser reabastecidos através de técnicas ocidentais (MRP, estoque de segurança, etc.) [SCHXX].

Ítems muito caros ou muito grandes não devem ser incluídos no Kanban. Seus pedidos e entregas devem ter o controle muito próximo de um planejador ou comprador [SCHXX].

Enfim, não é um sistema de uso global. Mesmo a Toyota não utiliza Kanban em todos os itens, supõe-se que em apenas em torno de 60% destes [BEL87].

Como técnica exclusiva do ambiente JIT, o Kanban só deve ser implantado quando já estiverem em funcionamento a manutenção preventiva, o controle de qualidade total e o lay out adaptado para operação just-in-time.

### 2.5.2 MRP (Materials Requirements Planning)

"É uma técnica de gestão de materiais que permite determinar a quantidade e o momento em que são necessários os materiais, num processo de manufatura" [GUI90].

O MRP não produz uma programação da produção a nível de atribuição de máquinas e demais recursos. Antes, é um sistema que contabiliza recursos e tempos.

Para o uso do MRP são necessárias informações sobre roteiros de fabricação, calendário fabril, estrutura dos produtos, dados sobre estoques de matéria-prima, semi-elaborados e acabados, cronograma de compras e lista de pedidos. O planejamento é feito a partir das datas de entrega, escalonando para trás, determinando quais, quando e quantos itens necessitam ser liberados, de modo a atender o Plano Mestre de Produção para um certo horizonte de tempo.

"Embora o conceito básico que está por trás de MRP já existisse na década de 50, foi a partir de 1970 que esta técnica começou a ser difundida, em parte graças ao trabalho de divulgação da APICS (American Production and Inventory Control Society)" [FEN89].

O termo MRP é entendido de várias maneiras, dependendo de onde é empregado. Pode representar desde o simples processo de explosão da demanda dos produtos finais para planejar a demanda dependente até o envolvimento de todo o sistema de gestão da produção.

Num trabalho apresentado por Fensterseifer e Bastos [FEN89], dentre quarenta e seis empresas brasileiras usuárias de MRP, a grande maioria (68,4%) implantou seu sistema a partir de 1980, com uma defasagem de uma década em relação à maioria das empresas americanas. Além disso, observou-se uma incidência bem maior de usuárias de MRP entre as empresas de capital estrangeiro do que entre as nacionais, públicas ou privadas.

Existe na literatura a sugestão de que o MRP pode ser usado tanto sob a filosofia justo-a-tempo quanto sob a filosofia tradicional de fabricação. Segundo Sneller [SNE86], "um sistema MRP funcionando apropriadamente gera um estoque justo-a-tempo para o processo de produção". As empresas japonesas, de fato, começam a utilizar-se mais amiúde do MRP para administração de materiais. Entretanto, a prática de utilização deste tipo de sistema evidencia o MRP como a ferramenta clássica de operacionalização da filosofia tradicional de fabricação.

### Objetivos

Segundo Tersine [TER85], são três:

- Assegurar a disponibilidade de materiais, componentes e produtos para a produção planejada e entregas.
- Manter o nível mínimo possível de estoques.
- Planejar atividades de manufatura, escalonamento de entregas e compras.

### Tipos de MRP

De acordo com a frequência de replanejamento, o sistema MRP pode ser [TER85]:

- Regenerativo: Todo o MRP é recalculado periodicamente (em geral, semanalmente), com base no último Plano Mestre de Produção. Essa modalidade é recomendada em ambientes estáveis, permite o uso eficiente do equipamento de processamento de dados, com menos erros devido à checagem periódica.

- Sistema Net change : Aplicável em ambientes voláteis, apenas faz adições e subtrações ao Plano Mestre de Produção. Faz uma explosão parcial, apenas para os elementos afetados. É um planejamento de alta frequência, podendo ser feito instantaneamente ou no final do dia.

### Entradas

- Plano Mestre de Produção: É o escalonamento da demanda independente, ou seja, dos produtos finais, que dependem dos pedidos de clientes. Não deve extrapolar a capacidade da fábrica. É executado baseado em pedidos dos clientes e também previsão de demanda.

- Estado de estoques: Arquivo com informações sobre estoques de todos os itens. Deve estar atualizado e pode conter informações sobre lead times, tamanhos de lote ou outras peculiaridades.

- Estruturas de produtos (Lista de Materiais): Contém todos os itens necessários a cada montagem ou submontagem e os estágios de processo na manufatura de cada item. A Lista de Materiais também deve estar atualizada, pois nem sempre o produto é feito como foi projetado.

### Saídas

O MRP explode a Lista de Materiais por requisições do PMP (simples operações aritméticas), obtendo as necessidades de cada item. Calcula níveis de estoque e emite relatórios detalhando o quê, quando e quanto fazer, quais ordens expedir, reter ou cancelar.

A liberação de pedidos é usada para compor os perfis de carga.

Os principais documentos produzidos são: ordens de compra e fabricação e relatórios para a gerência e atualização de estoques.

### CRP (Capacity Requirements Planning)

O MRP não considera a capacidade da fábrica ao escalonar as liberações de materiais. Anteriormente ao MRP é feito um ajuste grosseiro da carga contra a capacidade, o rough-cut capacity planning. No MRP, para ajustar a produção demandada à capacidade

instalada, introduziu-se um módulo CRP - Capacity Requirements Planning ou Planejamento de Necessidades de Capacidade.

O MRP escalona os pedidos para trás e alimenta o CRP com seus dados. O CRP avalia as cargas nos centros de trabalho, determinando se é ou não excessiva. Se não houver capacidade suficiente, o perfil de carga deverá ser mudado, retirando as partes excedentes para o período imediatamente mais cedo. Quando o plano for viável, determina-se para cada centro de trabalho o número de horas necessárias para cumprir o Plano Mestre de Produção.

### MRP II - Manufacturing Resources Planning

O MRP II, Planejamento dos Recursos de Manufatura, é um sistema gerencial dos recursos de manufatura, surgido no início dos anos 80 nos Estados Unidos [GUI90].

O MRPII surge da união dos subsistemas MRP e CRP (Planejamento das Necessidades de Capacidade) [GUI90]. É um sistema que se propõe a abranger todos os recursos de manufatura e não apenas as necessidades de materiais. Utilizando sua capacidade de simulação, é uma ferramenta de auxílio à tomada de decisão estratégica e tática.

### Implementações Práticas

A literatura aponta um grande número de insucessos na utilização do MRP. Sneller, em um estudo sobre implementação de sistemas MRP [SNE86], cita vários autores cotando o nível de insucesso do MRP em 80 e até 90%. Entretanto, empresas que trabalhem com um número grande de itens dificilmente poderão prescindir da utilização de um sistema do tipo MRP. Os problemas geradores de insucesso nas implementações de MRP costumam estar relacionados à negligência quanto aos aspectos psicológicos e organizacionais. A dificuldade em manter registros de dados fiéis sobre estoques é um dos sintomas dos sistemas MRP funcionando ineficientemente.

A tabela 1, a seguir, mostra o estudo do impacto da introdução de sistemas MRP em relação a diversos problemas enfrentados por empresas brasileiras.

Observe-se que os problemas cotados como os mais importantes

tiveram uma melhora sensível a partir da introdução dos sistemas MRP, embora não estejam completamente resolvidos, e credita-se ao sistema a causa da melhoria em percentuais superiores a 80%, no caso do excesso de estoques de materiais e interrupções por falta de materiais ou componentes, os dois problemas inicialmente cotados como mais importantes.

**TABELA 1**  
Avaliação dos Aspectos Problemáticos da Gestão da Produção  
Antes e Depois da Implantação do Sistema MRP

| Problemas   | Grau de importância * |                       | A alteração é em função do sistema? (%Sim) |
|---|-----------------------|-----------------------|--|
|   | Antes da implantação  | Depois da implantação |  |
| Estoque excessivo de materiais  | 3,2                   | 1,8                   | 96,9                                       |
| Estoque excessivo de produtos em processo   | 2,5                   | 1,7                   | 72,7                                       |
| Estoque excessivo de produtos acabados  | 1,6                   | 1,0                   | 35,5                                       |
| Atraso nos prazos de entrega ao cliente   | 2,6                   | 1,8                   | 68,7                                       |
| Excesso de produtos defeituosos   | 1,1                   | 1,0                   | 27,6                                       |
| Custos de produção demasiadamente altos   | 2,3                   | 1,9                   | 55,2                                       |
| Ciclo de fabricação demasiadamente longo  | 2,4                   | 2,0                   | 81,3                                       |
| Ciclo de fabricação frequentemente interrompido por falta de materiais ou componentes | 3,2                   | 2,0                   | 81,3                                       |
| Insuficiente flexibilidade na utilização da capacidade de produção                    | 2,2                   | 1,7                   | 45,2                                       |

Fonte: [FEN89]

\* Medida do grau de importância do problema (escala = [0,4] : 0 - Sem a menor importância; 4 - Extremamente importante).

### 2.5.3 OPT (Tecnologia da produção otimizada)

De acordo com Goldratt [GOL88], um de seus criadores, o sistema OPT (Optimized Production Technology - Tecnologia da Produção Otimizada) surgiu no final da década de 70, sendo que seus criadores não tinham vivência dos problemas cotidianos de chão-de-fábrica, o que é encarado por ele como vantagem.

O OPT é um sistema computacional que não necessita de grandes modificações no lay out da fábrica. A evolução do sistema OPT é

descrita como sendo, no seu estágio inicial (1978), similar a um Kanban computadorizado [GOL88]. Somente num estágio posterior passou a ser considerada a hipótese de lotes variáveis e maior dispêndio de tempo de setup (ajuste de máquina) nas operações não-gargalo, para se atingir um fluxo mais suave de produção [GOL88].

A programação é feita primeiramente identificando os gargalos da produção (recursos sobrecarregados ou com utilização máxima) através de uma simulação para trás, em função das datas previstas para a entrega dos pedidos [KLE89]. A seguir, os gargalos e as operações que os sucedem são escalonados para frente pelo módulo BRAIN. O módulo SERVE escala para trás as demais operações e trata também dos estoques ou "pulmões de tempo" que precedem às operações-gargalo. A figura 3 exemplifica o funcionamento do OPT, separando as operações feitas nos recursos-gargalo e suas sucessoras das operações feitas em recursos menos carregados.

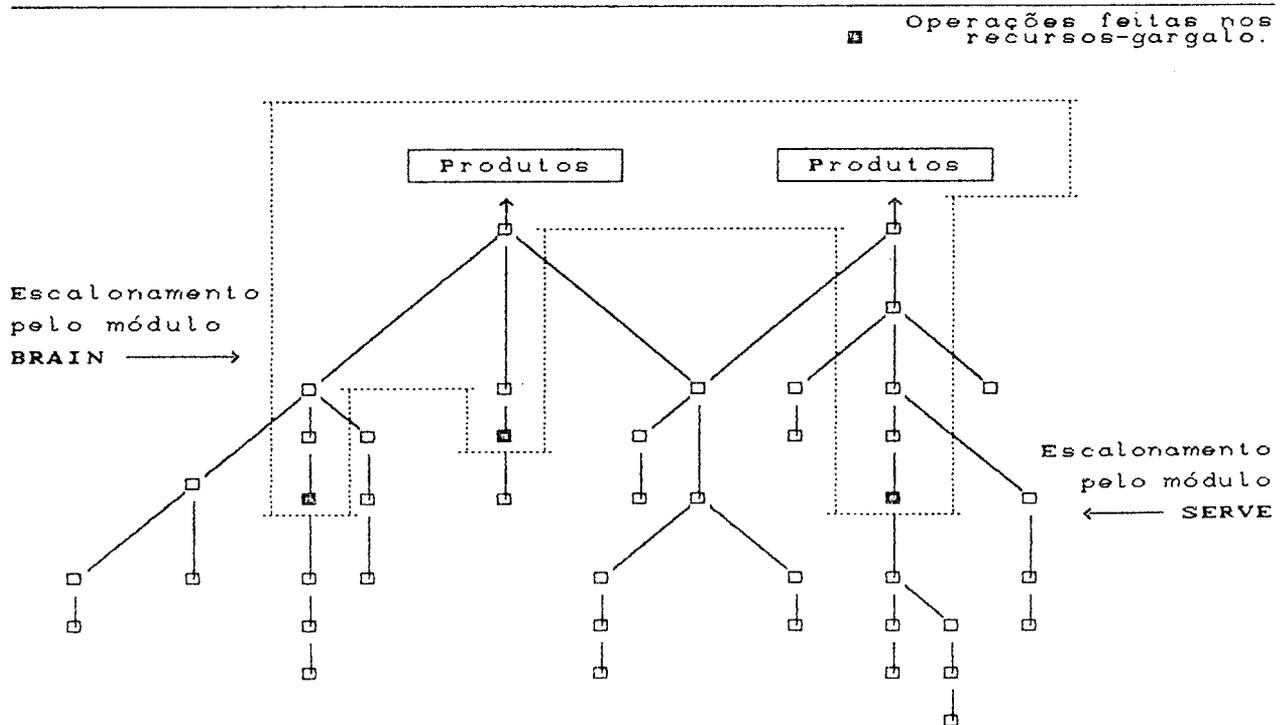


FIGURA 3 - Ilustração do funcionamento do OPT.

FONTE: [GOL88]

De acordo com Goldratt [GOL88], "se os gargalos são tais que têm que produzir todo o tempo, eles devem ser escalonados ignorando a capacidade dos não-gargalos. Uma vez pronta sua programação, todos os outros recursos não-gargalos devem ser

escalonados para trás no tempo, para sustentar a programação dos gargalos."

Goldratt [GOL88] afirma que "a superioridade do software não reside no seu algoritmo, mas principalmente nos conceitos em que está baseado". Porém, o algoritmo utilizado não é detalhado na literatura pelos seus autores.

Os conceitos citados estão no livro "A meta - excelência na manufatura" [GOL86], escrito em forma de romance, onde se estabelece que uma empresa deve ganhar dinheiro (meta mais imediata), obtendo retorno satisfatório sobre seu investimento e com um fluxo de caixa adequado. Em "A corrida - pela vantagem competitiva" [GOL89], mais técnico e esquemático, está explicitado o modelo "Tambor-Pulmão-Corda", representado na figura 4, onde o Tambor é o recurso principal com restrição de capacidade, cujo índice de produção marca o ritmo da fábrica. O Pulmão refere-se ao estoque na entrada dos recursos com restrição de capacidade, para proteger o valor agregado das vendas da empresa contra interrupções, pois "uma hora perdida na operação-gargalo é uma hora de perda para o sistema como um todo" [ANT89]. A Corda, nesta analogia, é a limitação para o estoque frente ao gargalo, para que este não cresça além do nível imposto pelo Pulmão. A Corda deve estar presa à operação inicial, ou seja, a operação inicial poderá liberar material no ritmo que o gargalo está produzindo, e não maior [GOL89].

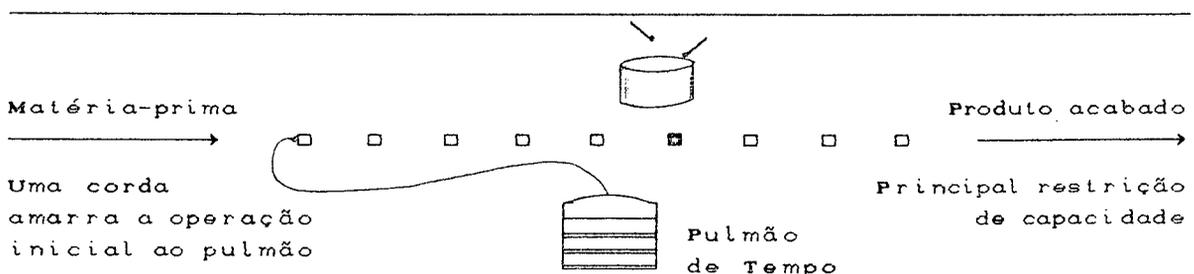


FIGURA 4 - ANALOGIA DO SISTEMA TAMBOR-PULMÃO-CORDA

[GOL89]

Embora não requeira reorganização da fábrica, a implementação do pacote OPT enfrenta resistências: uma vez que a programação dos recursos não-gargalo é feita de modo a manter um fluxo suave de produção nos gargalos, com mais tempo dispendido no ajuste de máquinas, a contabilidade tradicional desaconselhará esta maneira de programar a produção, pois mais tempo de ajuste é encarado como mais custo, embora se esteja utilizando o mesmo pessoal e

equipamentos. Da mesma forma, os capatazes ou encarregados de chão-de-fábrica, que estão acostumados a serem avaliados segundo sua produtividade, não verão com bons olhos uma programação que deteriora a produtividade dos recursos não-gargalo.

O sistema OPT pode ser usado num ambiente de fabricação tradicional e também dentro do JIT, em conjunto com suas técnicas, mas sua origem está na realidade das empresas ocidentais e sua criação é considerada uma resposta à onda causada pelo JIT e as técnicas japonesas de administração da produção.

### III - ABORDAGENS AO PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO

A adequabilidade de uma abordagem à programação da produção pode ser avaliada segundo a relação entre as características do ambiente produtivo a ser modelado e as características e pressupostos do modelo ou abordagem.

Os objetivos de programação colocados na abordagem devem apontar para uma representação o mais fiel possível dos objetivos reais da indústria. As abordagens existentes atingem subconjuntos variáveis destes objetivos.

Kanet e Adelsberger [KAN87] distinguem duas abordagens fundamentalmente diferentes para problemas de programação da produção: abordagens algorítmicas e de reformulação.

Na primeira abordagem, a formulação do problema é feita no contexto da programação matemática clássica, incluindo programação dinâmica e inteira, simulação, métodos branch-and-bound e heurísticos.

Os métodos reformulativos podem ser formulados de modo que o objetivo seja encontrar uma solução "satisfatória" e, além disso, o problema seja reformulado quando não há solução viável ou o esforço computacional é considerado excessivo. Enquanto a abordagem algorítmica propõe simplificar o problema modelado para permitir o uso de um método particular, uma abordagem reformulativa procuraria enriquecer o modelo para tornar o uso de um método específico menos crítico.

Bensana et alii [BEN88] fazem uma divisão mais específica, dividindo as abordagens em:

- Otimização
- Análise baseada em restrições
- Técnicas de liberação heurísticas com checagem de eficiência baseada em simulação.

Além destas, a metodologia de sistemas especialistas é destacada como uma nova abordagem emergente.

As seções seguintes detalham cada uma destas abordagens.

### 3.1 OTIMIZAÇÃO

A otimização compreende, como abordagem de pesquisa operacional, a definição do problema, construção de um modelo na forma de variáveis, equações de restrição e função objetivo e, por fim, a dedução de uma solução e a execução de testes a partir do modelo criado.

Para o problema de programação da produção, com efeito, "um dos primeiros passos ao se defrontar com problema deste gênero é conhecer, para cada produto em particular, todas as operações que devem ser executadas, bem como as máquinas que possam executá-las, de forma a determinar os possíveis fluxos a serem seguidos pela matéria-prima através das diversas máquinas, para obtenção do produto final desejado" [MAL75].

Dentre as técnicas utilizadas no âmbito da pesquisa operacional através de otimização, distinguem-se três tipos de ataque ao problema de programação da produção [CON67]: Os métodos algébricos (onde se incluem a programação inteira, *branch-and-bound*, programação dinâmica, soluções específicas como a de Johnson para o problema  $n/2/F/\bar{F}_{max}$ ), métodos probabilísticos, que tratam o *job shop* como um processo estocástico e o terceiro tipo de abordagem, simulação Monte Carlo, que, embora de fácil formulação, dificilmente capta de modo fiel as características do ambiente fabril.

#### 3.1.1 Formulação por programação inteira

A literatura apresenta vários modelos de formulação de problemas de programação da produção. O modelo seguinte deve-se a Manne<sup>1</sup>, descrito em [CON67]. O modelo é o de seqüenciamento do problema geral *job shop* utilizando programação inteira. Neste exemplo, cada tarefa requer processamento uma e somente uma vez para cada máquina. A formulação de um problema mais geral resulta em maior complexidade.

As variáveis envolvidas são:

<sup>1</sup>MANNE, A. S. - *On the Job Shop Scheduling Problem*. Operations Research 8, número 2, março 1960.

III - ABORDAGENS AO PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO

---

$p_{ik}$  = Tempo de processamento da tarefa  $i$  na máquina  $k$   
 $r_{ijk} = 1$ , se a  $j$ -ésima operação da tarefa  $i$  requer a máquina  $k$   
 $0$ , caso contrário  
 $T_{ik}$  = Data de início da tarefa  $i$  na máquina  $k$

A restrição de que uma e apenas uma tarefa pode ser processada em uma máquina num dado instante de tempo é expressa, para duas tarefas  $I$  e  $J$ , por uma das expressões:

$$T_{Ik} - T_{Jk} \geq p_{Jk} \quad \text{ou} \quad T_{Jk} - T_{Ik} \geq p_{Ik}$$

Ou seja, ou  $J$  precede  $I$ , ou  $I$  precede  $J$ . Este tipo de restrição 'ou exclusivo' não pode ser tratado por programação linear ordinária, requerendo a introdução de variáveis inteiras:

$Y_{Ijk} = 1$ , se a tarefa  $I$  precede (imediatamente ou não) a tarefa  $J$  na máquina  $k$   
 $Y_{Ijk} = 0$ , caso contrário

Logicamente, não é necessário definir  $Y_{Ijk}$  e  $Y_{Jik}$ . Uma é suficiente. As duas restrições excludentes acima podem ser escritas, então, como restrições independentes, ambas necessariamente válidas:

$$(M + p_{Jk}) Y_{Ijk} + (T_{Ik} - T_{Jk}) \geq p_{Jk}$$

$$(M + p_{Ik})(1 - Y_{Ijk}) + (T_{Jk} - T_{Ik}) \geq p_{Ik}$$

A constante  $M$  é escolhida suficientemente grande para que apenas uma das restrições seja obrigatória quando  $Y_{Ijk} = 1$  ou  $0$  (por exemplo,  $M = \sum_i \sum_k p_{ik}$ ).

As restrições de precedência de operações são escritas, dado que  $\sum_k r_{ijk} T_{ik}$  é a data de início da  $j$ -ésima operação da tarefa  $i$ , da seguinte forma, para todas as operações menos a última:

$$\sum_k r_{ijk} (T_{ik} + p_{ik}) \leq \sum_k r_{i,j+1,k} T_{ik}$$

Finalmente, escolhendo como critério de otimização a minimização do tempo médio de fluxo de processamento, usa-se a função objetivo equivalente de minimização do somatório das datas de início da última operação de cada tarefa:

$$\min \sum_i \sum_k r_{imk} T_{ik}$$

A função objetivo estabelece o parâmetro a ser otimizado pelo modelo. Entre os parâmetros de desempenho comumente usados figuram: minimização do atraso médio de pedidos, minimização do tempo de fluxo máximo dos pedidos, maximização do uso dos recursos, etc.

Para o problema em questão, com  $m$  máquinas e  $n$  tarefas, o modelo é o seguinte, com as quantidades correspondentes de variáveis e equações:

$$\min \sum_i \sum_k r_{imk} T_{ik}$$

s. a

número:

$$\sum_k r_{ijk} (T_{ik} + p_{ik}) \leq \sum_k r_{i,j+1,k} T_{ik} \quad (m-1)n$$

$$(M + p_{jk}) Y_{ijk} + (T_{ik} - T_{jk}) \geq p_{jk} \quad m \frac{n(n-1)}{2}$$

$$(M + p_{ik}) (1 - Y_{ijk}) + (T_{jk} - T_{ik}) \geq p_{ik} \quad m \frac{n(n-1)}{2}$$

$$T_{ik} \geq 0$$

$$Y_{ijk} = 0 \text{ ou } 1$$

$$m \frac{n(n-1)}{2} \quad mn$$

Esta formulação tem as quantidades de restrições e variáveis descritas pelas fórmulas à sua direita, no modelo acima. Alguns exemplos numéricos para as  $n$  tarefas e  $m$  máquinas estão na tabela 2:

III - ABORDAGENS AO PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO

| tarefas | máquinas | número de variáveis | número de restrições |
|---------|----------|---------------------|----------------------|
| 2       | 3        | 9                   | 10                   |
| 10      | 3        | 165                 | 290                  |
| 100     | 3        | 15.150              | 29.900               |
| 100     | 30       | 151.500             | 299.900              |
| 1000    | 30       | 15.015.000          | 29.999.000           |
| 1000    | 300      | 150.150.000         | 299.999.000          |

TABELA 2 - Variáveis e equações envolvidas em uma formulação do problema geral job shop por programação inteira.

O problema de programação inteira resultante cresce exageradamente com o número de máquinas e tarefas. É um problema do tipo NP-completo, cuja solução demanda tempo não-polinomial para sua solução. A aplicação de otimização em um ambiente real de fabricação, mesmo obtendo um modelo satisfatório, dificilmente conseguirá produzir programas de produção em tempo adequado.

Outra limitação à aplicação da otimização é a dificuldade em modelar matematicamente variáveis de difícil quantificação, como: qualidade, satisfação do cliente, satisfação do trabalhador.

Embora ainda sejam feitas publicações sobre otimização aplicada à programação da produção, seu auge foi na década de 60, sendo progressivamente abandonada como abordagem viável ao problema.

Sobre a frustração na busca de uma solução efetiva para o problema de programação da produção job shop com  $n$  tarefas (jobs) e  $m$  máquinas, escrevem Conway et alii [CON67]:

*"... É um desafio fascinante. Embora fácil de enunciar e visualizar o que é requerido, é extremamente difícil fazer qualquer progresso na direção da solução. Muitas pessoas competentes têm considerado o problema, e todas o têm abandonado de mãos vazias. Uma vez que esta frustração não é relatada na literatura, o problema continua a atrair investigadores, que simplesmente não podem acreditar que um problema estruturado de forma tão simples seja tão difícil, até que o tenham experimentado".*

### 3.2 Propagação de Restrições

Outra técnica de pesquisa operacional aplicável à programação da produção é a propagação de restrições. É característica desta abordagem o fato de que dificilmente existem restrições em número

suficiente e não excessivo para se chegar a um programa de produção completo. Geralmente, é uma abordagem sub-restrita.

Smith et alii [SMI86] dividem os fatores que influenciam as decisões de programação em duas grandes classes de restrições: (1) as restrições de programação, que delineiam o espaço de possibilidades de programação e (2) as preferências de programação.

As restrições de programação incluem:

Restrições causais: São restrições de precedência que limitam a maneira como determinar as rotas percorridas pelas ordens. Uma restrição deste tipo diz quando uma operação deve preceder ou suceder outra. Cada operação possui um conjunto bem definido de recursos a um dos quais será alocada.

Restrições físicas: Cada máquina em um job shop tem aplicações específicas que restringem os tipos de operações que pode realizar. Cada máquina tem características particulares (por exemplo, velocidade de corte, procedimentos de ajuste) que limitam a quantidade de trabalho que pode ser realizado em um certo período de tempo. Generalizando, as restrições físicas definem as limitações funcionais de recursos específicos na fábrica.

Indisponibilidade de recursos: Restrições dinâmicas, fora do controle do programador, que podem acontecer de forma imprevista, como um defeito de máquina, por exemplo.

As preferências de programação são classificadas como:

Objetivos organizacionais: Refletem preocupações e objetivos globais relativos à operação da fábrica. Implicam em critérios gerais, com os quais serão comparados os programas de produção propostos. São estabelecidos segundo várias dimensões de performance: cumprir datas de entrega, cujo atraso afeta a satisfação do cliente e a probabilidade de fechar futuros negócios, minimizar o tempo de permanência do estoque, porque o investimento em matérias-primas só é recuperável através da venda dos produtos, tornando a minimização dos estoques um objetivo importante, maximizar utilização de recursos, buscando aumentar o throughput da fábrica, mantendo as máquinas críticas operando a maior parte possível do tempo, e manter a estabilidade da fábrica, evitando mudanças de última hora com alterações no programa.

Preferências organizacionais: Refletem o conhecimento heurístico presente num ambiente de fabricação, no nível de decisões de programação, como: seleção de operações, recursos e

intervalos de tempo específicos.

A indisponibilidade de recursos relativamente às preferências de programação, em contraste com o mesmo item relativo às restrições de programação, refere-se às restrições introduzidas pelo programador. Estas restrições devem ser encaradas como preferências, pois podem ser refeitas em função de motivos que as contrariem, como o recebimento de um pedido de alta prioridade, por exemplo.

As restrições não devem ser tratadas necessariamente como regras inflexíveis, mas podem ser ponderadas e relaxadas, levando em conta a existência de alternativas de programação que satisfazem as restrições, a utilidade ou o grau de aceitabilidade de cada alternativa e a elasticidade destas restrições, ou seja, o quanto é possível violar uma restrição e a que custo.

Este conjunto de regras e diretrizes de programação pode ser usado para delinear o programa de produção, propagando as restrições segundo as relações de precedência existentes entre operações relativas à mesma tarefa (dadas pelos roteiros de fabricação), ou entre operações relativas à mesma máquina, com as relações de precedência dadas pelo seqüenciamento de máquinas em curso (seqüenciamento parcial).

O exemplo mais notório deste tipo de abordagem são os sistemas PERT/CPM, que manipulam restrições de tempo. Os sistemas especialistas para programação da produção OPAL [BEN88] [BEL88] e MASCOT também utilizam propagação de restrições de tempo como estratégia principal de inferência.

### 3.3 REGRAS E HEURÍSTICAS DE LIBERAÇÃO

Ainda no âmbito da pesquisa operacional, outra abordagem ao problema de programação da produção é a aplicação de regras de prioridade de operações.

Uma regra de liberação seleciona a próxima tarefa a ser processada em uma máquina dentre um conjunto de tarefas esperando processamento, servindo como base para a tomada de decisão sobre a ordem de processamento de um conjunto de operações.

As regras podem ser simples ou complexas. Exemplo de uma regra simples: "Selecionar uma tarefa aleatoriamente". Uma regra complexa pode ser: "Selecionar a tarefa com a data de entrega mais

próxima cujo estoque do cliente é menor do que uma determinada quantidade". Há uma grande quantidade de regras e variações de regras. Blackstone et alii [BLA82] definem formalmente 34 regras de liberação, o que não é uma lista exaustiva.

Um fator que tem impulsionado o estudo e pesquisa sobre regras de liberação é o fato de que não é possível identificar uma regra com performance superior em todas as circunstâncias, porém várias regras deram indicação de boa aplicabilidade para situações particulares.

Regras de liberação são usadas visando geralmente a minimizar o custo total de estoque. A natureza da regra empregada influencia os custos de atraso, custos de estoques finais e em processamento e também os custos de ajuste de máquinas [BLA82].

Note-se que o uso de regras de liberação não constitui propriamente uma atividade de programação da produção, pois as regras não produzem um seqüenciamento de todas as tarefas na fila, mas apenas determinam qual a tarefa que será processada primeiro. É um tipo de solução parcial para o problema de programação da produção. Além disso, refletem apenas um ou alguns tipos específicos de critério, o que limita sua aplicabilidade.

A dificuldade do problema de liberação decorre do fato de que, dadas  $n$  tarefas na fila de um posto de trabalho, há  $n!$  maneiras de escalonar estas tarefas e, além disso, uma condição relativa a outro posto de trabalho pode influenciar a seqüência ótima de tarefas no posto de trabalho presente [BLA82].

### 3.3.1 Aplicação e Critérios de Medida para Avaliação

Um ambiente de fabricação job shop pode ser visto como uma rede de filas, o que possibilita o uso de técnicas de teoria das filas para seu estudo. Porém, como as tarefas passam pela rede segundo uma variedade de padrões muito grande, o estudo de job shops por métodos analíticos torna-se extremamente complexo, mesmo para ambientes reduzidos. Por este motivo, praticamente todos os estudos sobre regras de liberação em job shops têm usado simulação, ao invés de métodos analíticos [BLA82].

Para medir a efetividade de uma regra de liberação, parece apropriado avaliar o efeito de sua aplicação nos custos, sejam de estoque, de ajuste de máquinas ou, principalmente, custos de

atraso. Entretanto, como a estrutura dos custos de atraso difere grandemente conforme a empresa, muitos pesquisadores preferem utilizar medidas de performance que não envolvem custos [BLA82].

As medidas de folga de tempo entre a data de entrega e a data de término da tarefa, tempo de permanência de uma tarefa no sistema e atraso da tarefa são os critérios mais comumente usados para avaliar regras de liberação. A busca de otimização destas medidas não é, entretanto, perfeitamente concordante. Por exemplo, a minimização do tempo de permanência da tarefa não minimiza o atraso. A maximização da folga média pode gerar o término de tarefas com grande antecedência, o que não é desejável. Neste caso, a minimização do atraso médio é preferível. Todavia, nenhuma destas medidas minimiza custos.

Um ponto a destacar quanto à performance das regras é que o tamanho da fábrica não a influencia<sup>2</sup>. Isto propicia o uso de modelos de tamanho pequeno para simulação.

### 3.3.2 Classificação das Regras de Liberação

Blackstone et alii [BLA82] dividiram as regras de liberação em quatro grupos: regras envolvendo tempo de processamento, regras envolvendo data de entrega, regras simples que não envolvem tempo de processamento nem data de entrega, e regras envolvendo duas ou mais das três primeiras classes.

#### 3.3.2.1 Regras Envolvendo Tempo de Processamento

Uma regra pesquisada desde as primeiras publicações sobre o assunto é a regra SI ou SIO (Shortest Imminent Operation), que seleciona para processamento a tarefa cuja operação pode ser completada no menor tempo. Outro exemplo é a regra SR (Shortest Remaining Processing Time), definida como o somatório dos tempos de processamento nas máquinas pelas quais a tarefa ainda deverá passar.

<sup>2</sup>BAKER, C. T. & DZIELINSKY, B. P. - *Simulation of a simplified job shop*. Management Science 6, pg. 311, 1960. Citado em [BLA82].

3.3.2.2 Regras Envolvendo Data de Entrega

As datas de entrega podem ser atribuídas segundo quatro métodos [BLAB2]: a) externos: 1 - Constante (o vendedor tem cota de entrega para um período uniforme no futuro). 2 - Aleatório (o comprador estabelece a data de entrega). b) internos: 3 - baseado na quantidade total de trabalho. 4 - baseado no número de operações.

As regras deste grupo geram menor variância do atraso de tarefas do que as regras baseadas em tempo de processamento, e frequentemente um número menor de tarefas atrasadas [BLAB2].

Pertencem a este grupo as regras: Data de entrega mais cedo (EDD - Earliest Due Date), Folga mínima (Slack - definida como o tempo que resta até a data de entrega menos o somatório dos tempos de processamento restantes da tarefa) e Taxa de folga da tarefa (igual à folga dividida pelo tempo restante até a data de entrega).

Gere [GER66], usando atraso como medida de performance, listou a seguinte ordem de efetividade de regras baseadas em data de entrega para ambientes dinâmicos:

Sem heurísticas:

- (1) Folga por operação
- (2) Folga
- (3) Taxa de folga modificada
- (4) Taxa de folga da tarefa

Com heurísticas:

- Folga por operação
- Taxa de folga da tarefa
- Folga
- Taxa de folga modificada

A folga pode ser definida de duas formas: folga estática, calculada segundo o tempo de entrada de uma tarefa na estação de trabalho, e folga dinâmica, calculada segundo o tempo presente, ao invés do tempo de entrada.

A regra 'taxa crítica', uma variação da regra de folga dinâmica por operação, requer uma estimativa da demora do sistema. Esta regra tem a forma geral:

$$\text{Taxa crítica: } \frac{[\text{Data de entrega} - \text{Data atual}]}{\text{Lead Time restante}}$$

3.3.2.3 Regras Simples que não envolvem tempo de processamento nem data de entrega

Dentre estas regras figuram:

-RANDOM : A seleção é aleatória.

-FIFO (First-In-First-Out): Regra da fila, a primeira tarefa a entrar na fila é a primeira a ser processada. Em geral, apresenta a mesma performance que a seleção aleatória quanto à folga ou tempo de processamento médios, mas gera uma variância menor. Sua virtude principal é a simplicidade de uso e definição.

-Número de operações restantes: tem desempenho muito pior do que todas as outras regras testadas quanto ao atraso médio.

As regras deste tipo não costumam ter desempenho melhor do que as baseadas em tempo de processamento ou data de entrega.

#### 3.3.2.4 Regras Combinadas envolvendo duas ou mais das três primeiras classes

A composição de regras é feita geralmente por uma soma ponderada dos índices de prioridade gerados pelas regras que estão sendo combinadas.

Um bom exemplo é a regra COVERT (custo de atraso sobre tempo restante). O uso desta regra requer uma estimativa dos custos reais de atraso para cada tarefa. Outros exemplos são: soma ponderada da folga por operação e tempo de processamento e Função de Prioridade (uma soma ponderada sobre vários fatores, como: folga, data de entrega, tempo de operação, etc.).

Segundo o estudo dirigido por Blackstone [BLA82], as pesquisas feitas com regras combinadas não conseguiram obter resultados que indicassem uma vantagem significativa de uso em relação às regras simples.

#### 3.3.3 Regras Heurísticas

As regras de liberação são procedimentos de um passo, selecionam segundo uma medida imediata feita entre as operações executadas em uma mesma máquina, enquanto que um programador, face a um problema com  $n$  tarefas e  $m$  máquinas, buscaria apoio em procedimentos muito mais complexos.

A utilização de regras que captem as características do

ambiente fabril de forma mais ampla do que as regras locais de liberação é o objeto da pesquisa sobre regras heurísticas, também referidas como 'heurísticas secundárias de liberação' [BLA82]. Estas heurísticas são procedimentos que podem ser usados para identificar circunstâncias sob as quais a regra de liberação seja violada, isto é, para que a tarefa indicada pela regra de liberação para ser processada primeiro não o seja.

O uso de heurísticas melhorou a performance de todas as regras testadas [GER66].

Gere [GER66] listou oito heurísticas: Alternar operação, Look Ahead, Inserir, Programação transcendendo o tempo, Subconjuntos de tarefas críticas, Refazer com datas de entrega ajustadas, Flexibilidade e Manipulação.

#### Alternate Operation (Alternar operação):

Programa a operação de acordo com a regra e checa se isto torna alguma outra tarefa "crítica", isto é, faz sua folga ficar negativa ou abaixo de um certo nível. Se isto acontece, cancela a última operação e programa a próxima operação da tarefa crítica. Checa outra vez se há atraso. Caso não houver, confirma a tarefa crítica. Se também houver atraso com esta, programa a tarefa que havia sido selecionada primeiro pela regra.

#### Look Ahead:

Esta regra estuda o efeito causado pela programação de uma tarefa escolhida por uma regra de liberação, antevendo efeitos e programando outra tarefa, se esta for mais crítica do que a selecionada pela regra.

#### Insert (Inserir):

Usa a heurística Look Ahead. Insere uma tarefa na programação da máquina consecutiva, no caso de a duração desta tarefa inserida permitir que seja completada antes do fim da tarefa que está sendo processada na primeira máquina.

#### Programação transcendente ao tempo:

É um procedimento que tenta antecipar conflitos através de iterações. Cada iteração inicia a alocação reavaliando as prioridades das operações, inicialmente permitindo a alocação com quebra de restrições de tempo.

Subconjuntos de tarefas críticas:

Programa segundo 'subconjuntos de tarefas críticas' selecionados. As operações do subconjunto são programadas primeiro. À medida que as operações vão sendo programadas, os índices de prioridade são atualizados.

Refazer com datas de entregas ajustadas:

Quando um programa é terminado e pelo menos uma tarefa está atrasada, a data de entrega de cada tarefa atrasada é decrementada para o cálculo de novas prioridades, forçando as operações antes atrasadas a serem antecipadas quando o programa é refeito.

Flexibilidade:

Um programa de produção representado por um gráfico de Gantt pode apresentar uma tarefa que não se ajusta bem entre suas sucedâneas. Se esta tarefa puder ser "comprimada" no tempo disponível, a tarefa à qual pertence terá grande benefício e as outras tarefas não sofrerão muito atraso. Esta heurística parte desta violação da restrição rígida de tempo, ajustando os tempos de acordo com a nova ordem.

Manipulação:

É uma maneira de melhorar o programa obtido manipulando o gráfico de Gantt, sem executar as regras uma segunda vez. Pode-se melhorar o aproveitamento do programa de produção atacando especificamente os intervalos ociosos que aparecem no gráfico de Gantt.

Existem críticas a estas primeiras pesquisas sobre heurísticas [CON67], referindo-se aos estudos publicados como comparações de regras apenas um pouco mais complexas do que as regras de liberação. No entanto, representaram um avanço no sentido de aproximar os modelos computacionais da realidade fabril.

Para verificar se o custo de implementação compensa a melhoria, deve ser feita uma análise de efetividade usando simulação, por exemplo.

### 3.3.4 Sistemas limitados por máquinas e trabalho

As regras de liberação tratam apenas de selecionar tarefas a serem processadas considerando restrições de máquinas. Considerando um ambiente com M máquinas e T trabalhadores, com  $T < M$ , o sistema fica bem mais difícil de modelar.

Um sistema capaz de manipular esta restrição dual de recursos precisa de uma regra que designe trabalhadores para máquinas e também uma regra de liberação que selecione a próxima tarefa a processar em uma estação de trabalho.

### 3.3.5 Resultados

A pesquisa sobre regras de liberação ainda está em fase inicial [BLAB2]. Algumas regras referidas como tendo boa performance geral são: SIO, Data de entrega mais cedo, Menor folga dinâmica, Taxa crítica, FIFO e COVERT.

## 3.4 SISTEMAS ESPECIALISTAS

Uma abordagem recente ao problema de programação da produção é a dos sistemas especialistas, surgida na década de 70, seguindo a evolução da pesquisa em inteligência artificial (IA) de sistemas generalistas, de senso comum, para sistemas especializados, abrangendo uma área de conhecimento bem definida e tendo como pontos-chave a representação do conhecimento envolvido e os métodos de busca das soluções. Um sistema deste tipo é mais viável do que um que utilize "bom senso". Os sistemas especialistas são "o primeiro e mais avançado produto comercialmente viável de IA" [SCH87][FIS87].

Pode-se definir sistema especialista como "uma ferramenta que tem a capacidade de entender o conhecimento sobre um problema específico e usar este conhecimento inteligentemente para sugerir alternativas de ação" [KUMB6].

Sistemas especialistas são também conhecidos como sistemas baseados em conhecimento. As duas expressões são usadas como sinônimos, porém, alguns autores fazem distinções: "Um sistema

especialista não pode ser construído sem o auxílio de pelo menos um especialista, que deverá estar disposto a gastar muito tempo para transferir seu conhecimento para o sistema" [RIC88]. A figura 5 exibe uma descrição de cada conceito:

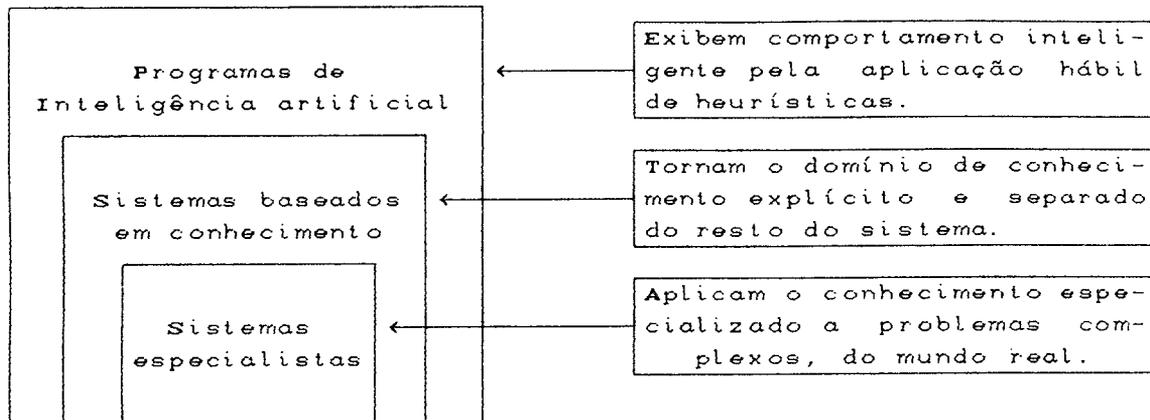


FIGURA 5 - Sistemas especialistas são sistemas baseados em conhecimento. FONTE: [WAT86], também citado em [KLE89].

Embora exista esta diferença de definição, a distinção entre um sistema baseado em conhecimento e um sistema especialista não existe na prática.

Esta nova abordagem contrasta com os sistemas comuns porque representa e manipula conhecimento ao invés de dados, apenas. Também, a existência de uma estratégia de inferência de caráter heurístico difere do processo algorítmico dos sistemas usuais de processamento de dados.

A concepção e construção de sistemas especialistas parte do aproveitamento da prática dos profissionais da área. Por exemplo, quanto à programação da produção:

*"Observa-se, na prática, a predominância de sistemas de programação da produção puramente manuais, especialmente em ambientes produtivos relativamente simples, envolvendo somente alguns passos de processamento. Estes sistemas contam primariamente com o conhecimento especializado de uns poucos programadores experientes que constroem, revisam e mantêm o programa de produção usando não mais do que algum instrumento de auxílio como um gráfico de Gantt"* [GRAB1].

O especialista é uma pessoa que, através de treinamento e experiência, alcançou um alto grau de conhecimento e competência para a solução de problemas sobre determinado assunto. No processo de construção de sistemas especialistas, denominado engenharia de

conhecimento, tenta-se captar e representar o conhecimento sobre o assunto de forma a manipulá-lo por um sistema computacional. Desta forma, o conhecimento adquire um caráter não-perecível, ainda que não seja usado de forma criativa, como no caso do especialista humano.

Quanto à adequabilidade do uso de sistemas especialistas, há vantagem em utilizar técnicas de I.A. em problemas mal estruturados, com grande quantidade de conhecimento envolvido e de caráter eminentemente não-quantitativo, especialmente quando os problemas no domínio de conhecimento não podem ser bem definidos analiticamente, ou os problemas podem ser formulados analiticamente mas o número de soluções alternativas é muito grande, como no caso de explosão combinatorial. Neste caso, o conhecimento precisa ser usado seletivamente, segundo uma estratégia preestabelecida.

A aplicação de sistemas especialistas à programação da produção é recente e não atingiu ainda o nível de desempenho necessário para tratar os problemas reais, de características dinâmicas e de difícil modelamento. Os esforços existentes esbarram, quando não em restrições técnicas, no receio dos especialistas em fornecer as informações necessárias para a construção de um sistema.

Os sistemas especialistas futuros possivelmente empregarão mais técnicas de otimização, bem como a pesquisa operacional está mudando da otimização pura para abordagens que incluem heurísticas [OKE86]. As aplicações de pesquisa operacional (PO) podem ser utilizadas para problemas que sejam bem específicos ou parte de um problema maior. Doukidis & Paul [DOU90] assinalam algumas restrições à utilização de inteligência artificial em PO:

- tempo necessário para implementar
- limitações tecnológicas
- custo
- falta de uso prático
- falta de assistência (pessoal competente)
- falta de conhecimento das potencialidades da I.A.
- relutância dentro da organização

Alguns dos sistemas especialistas existentes empregam

técnicas algorítmicas, como a propagação de restrições de tempo [BEN88] [BEL88], ou simulação [BRU86].

### 3.4.1 Estrutura de um Sistema Especialista

A construção de um sistema especialista deve considerar três aspectos básicos [KUS88]:

- Representação de conhecimento
- Motor de inferência
- Aquisição de conhecimento

A eficiência do sistema dependerá da existência de uma interface de comunicação adequada e amigável, tanto para a aquisição de conhecimento junto ao especialista, quanto na utilização pelo usuário.

Os três aspectos são detalhados a seguir:

#### 3.4.1.1 Representação do conhecimento

O conhecimento no sistema consiste de fatos e regras de inferência relativos ao domínio de conhecimento. Os fatos fazem parte da base de dados ou memória de trabalho, enquanto as regras e heurísticas ficam no módulo chamado base de conhecimento.

As bases de dados e conhecimento podem ser alteradas através da aquisição de conhecimento, descartando informações ou extraíndo conhecimento do especialista, e podem ser manipuladas pelo motor de inferência, segundo as consultas do usuário. As regras, localizadas na base de conhecimento e disparadas pelo motor de inferência, utilizam os fatos na base de dados para efetuar a prova.

#### 3.4.1.2 Motor de inferência

O mecanismo de inferência de um sistema especialista contém a estratégia de controle do sistema. Inclui um interpretador que

aplica as regras e o mecanismo de ordenação que ordena a aplicação das regras. Deve possuir, também, um justificador, que explica o sistema de raciocínio, detalhando o caminho para chegar a uma certa inferência.

Intimamente ligada à representação de conhecimento escolhida para o sistema, a estratégia de inferência pode estar baseada em propagação de restrições, encadeamento para frente ou controle dirigido por dados ('forward chaining'), encadeamento para trás ou controle dirigido por objetivos ('backward chaining'), ou ainda alguma estratégia mista, redução do problema, geração e teste, etc.

#### 3.4.1.3 Aquisição de conhecimento

A aquisição de conhecimento supre o sistema especialista com as informações necessárias à resolução de problemas. O conhecimento adquirido deve ser compatibilizado com a estrutura de representação existente na base de conhecimento.

Uma vez que os sistemas especialistas para planejamento e programação da produção lidam com ambientes dinâmicos, deve ser considerada a aquisição de conhecimento dinâmico, que tem incerteza associada.

O conhecimento pode ser definido como sendo "público" ou "privado" [HAY83]. Uma fonte de conhecimento privado são os especialistas na área. Conhecimento público pode ser obtido através da literatura especializada, modelos matemáticos, bancos de dados globais e programas de simulação.

#### 3.4.1.4 Esquema de funcionamento

A figura 6 apresenta um diagrama de funcionamento de um sistema especialista. Nem todos os sistemas existentes possuem formalmente as estruturas descritas na figura. Porém, as funções desempenhadas por estas deverão ser satisfeitas. Por exemplo, um sistema pode funcionar sem um módulo de aquisição de conhecimento, mas os construtores do sistema deverão fazer a aquisição e controlar a consistência das informações incluídas ou retiradas das bases de fatos e de conhecimento.

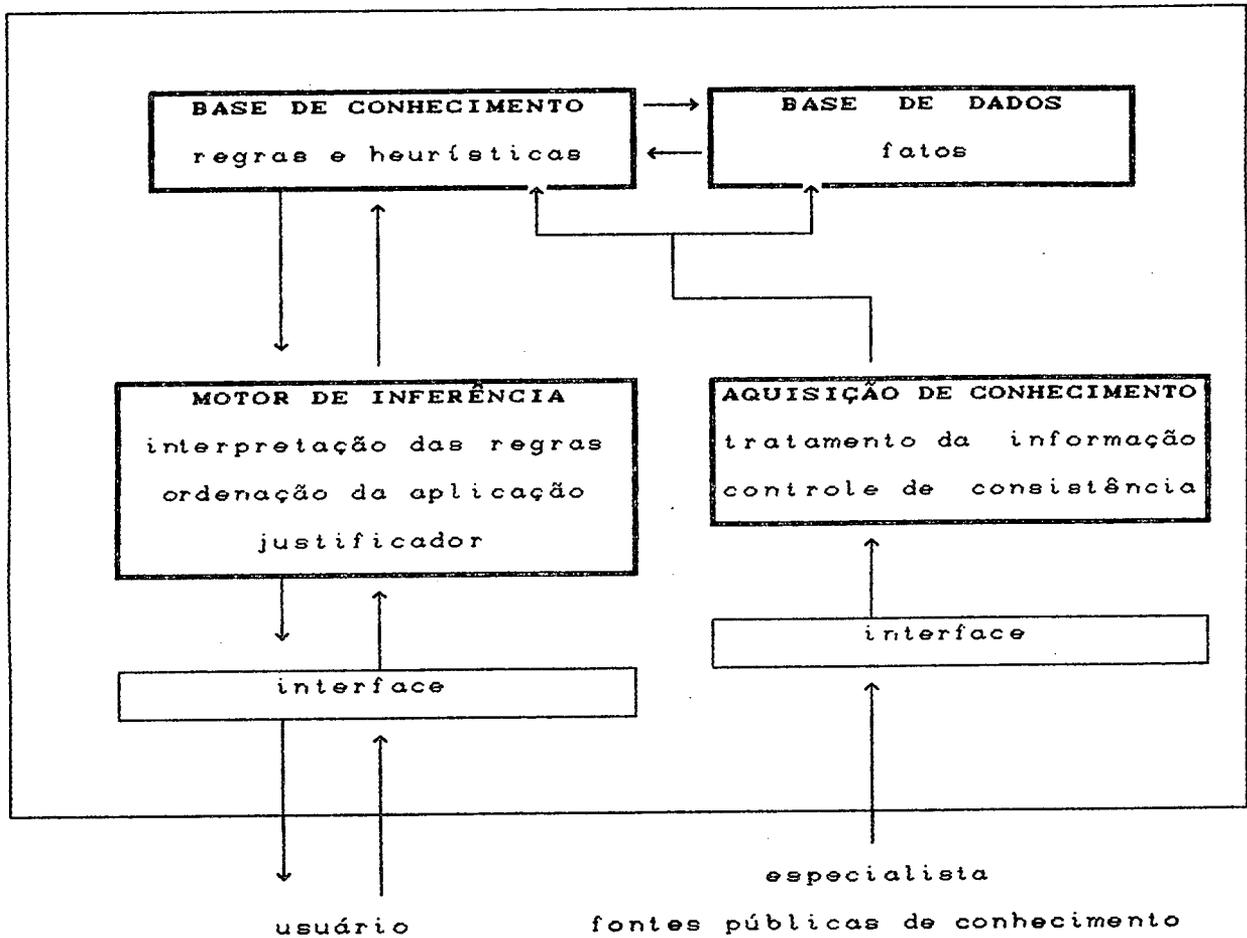


FIGURA 6 - ESTRUTURA DE UM SISTEMA ESPECIALISTA

### 3.4.2 Etapas de construção

A construção de um sistema especialista consome tempo e não é trivial. Pode-se dividir a tarefa de construção nas cinco fases seguintes [KUM86]:

1. Definição do problema;
2. Aquisição de conhecimento, representação e coordenação;
3. Mecanismo de inferência;
4. Implementação e
5. Aprendizagem.

#### 3.4.2.1 Definição do problema

Esta fase envolve a compreensão do problema, identificação de suas características, determinação dos objetivos do processo de solução do problema e definição da metodologia a ser usada.

### 3.4.2.2 Aquisição, representação e coordenação do conhecimento

Inclui aquisição e representação do conhecimento, projeto do motor de inferência, seleção de ferramentas de programação e coordenação do conhecimento.

A maneira de representar o conhecimento tem relação direta com a habilidade de resolver problemas. Tal representação deve ser implementável no computador e permitir a manipulação do conhecimento relevante para a evolução da solução. O conhecimento representado é uma combinação de estruturas de dados e procedimentos interpretativos.

Há vários esquemas de representação. As regras de produção são esquemas de representação de conhecimento procedurais. Existem esquemas declarativos, sendo os mais difundidos as frames e as redes semânticas [KUS88].

As regras de produção assumem uma das formas:

*SE <condições>*

*ENTÃO <conclusões> (com certos fatores)*

ou

*SE <condições>*

*ENTÃO <ações>*

A premissa, primeira parte da regra, contém as condições a serem satisfeitas para a ativação da segunda.

Qualquer uma das formas pode ser ainda classificada em duas categorias: regras de primeira ordem, específicas da aplicação, e regras específicas do problema ou meta-regras, cujas conclusões são ordens de ativação de outras regras. O metac conhecimento contido nas meta-regras é muitas vezes referido como "conhecimento sobre o conhecimento".

Frames são estruturas de conhecimento complexas que comportam relações de hierarquia. Os elementos relacionados a uma frame são armazenados em 'slots' que descrevem atributos. Estes atributos podem ser herdados pelas frames sucessoras desta, que estão em nível inferior. O exemplo seguinte ilustra esta forma de representação:

```

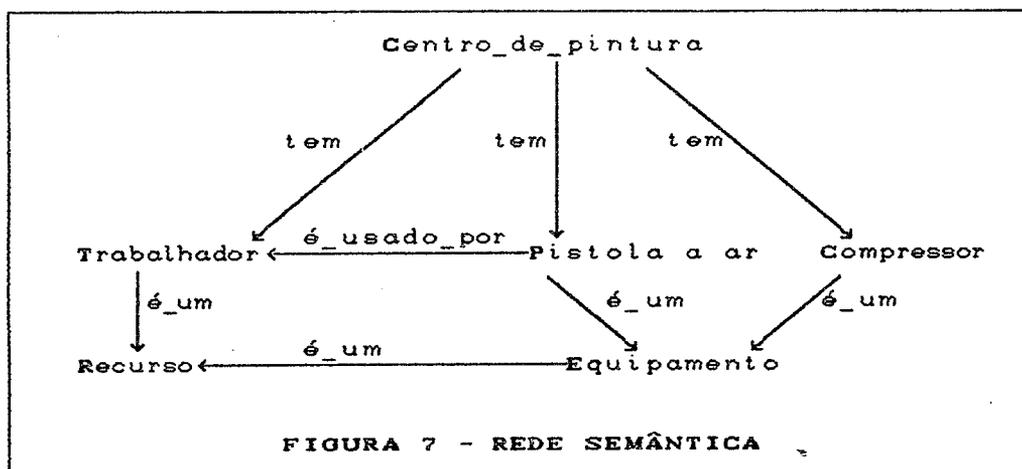
FRAME: Pintura
É_UM(A): Operação

DURAÇÃO: Tempo [default = 18 min]
CENTRO_DE_TRABALHO: Centro_de_pintura
OPERAÇÃO_ANTERIOR: Solda
PRÓXIMA_OPERAÇÃO: Acabamento_plásticos

```

Neste caso, a frame **Pintura** herda os atributos da frame **Operação**.

Redes semânticas representam o conhecimento como nós em uma rede, descrevendo objetos relacionados. A figura 7 ilustra uma estrutura deste tipo:

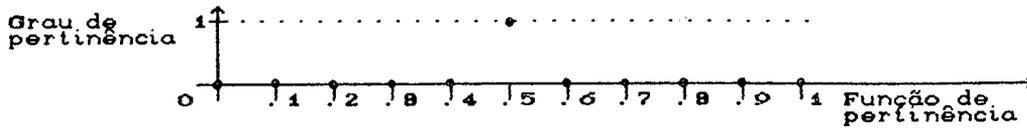


É muito desejável que o sistema especialista possua, junto ao módulo de aquisição de conhecimento, um controlador de consistência para atuar quando o conhecimento é incorporado ou removido do banco de conhecimento. O problema de inconsistência é comentado por Chang & Ibbs [CHA90], na descrição de um sistema especialista difuso para programação da construção.

Uma regra existente neste sistema estabelece:

SE o grau de exposição da atividade à chuva é **baixo**,  
ENTÃO o grau de impacto devido à chuva é **moderado**.

No caso de ter-se uma atividade não exposta à chuva (grau de exposição zero), a definição de **baixo** existente no banco de dados abordará a definição de zero como o caso extremo **absolutamente baixo** (usando o *Modus Ponens Difuso*), concluindo que o grau de impacto devido à chuva é **absolutamente moderado**, o que será representado, no caso, pelo seguinte conjunto difuso:

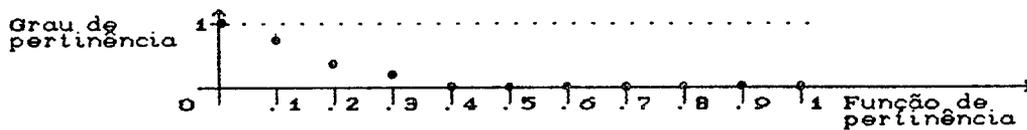


A interpretação do conceito difuso representado no gráfico acima é: No intervalo  $[0,1]$  escolhido para a função de pertinência, **absolutamente moderado** tem grau de pertinência 1 para a função igual a 0,5 e 0 para os demais pontos.

Se o sistema admitir a colocação de outra regra, com o enunciado:

SE o grau de exposição da atividade à chuva é zero,  
ENTÃO o grau de impacto devido à chuva é baixo.

Para o mesmo caso de atividade não exposta à chuva, a conclusão seria que o grau de impacto devido à chuva é baixo, representado no sistema pelo conjunto difuso:



Esta nova regra cria um conflito semântico com a regra anterior, pois o mesmo fato gera respostas completamente diferentes. Esta inconsistência deve ser tratada impedindo que o sistema aceite a entrada de uma regra conflitante com alguma já existente ou implantando um esquema para detectar e resolver os conflitos.

#### 3.4.2.3 Mecanismo de inferência

O motor de inferência do sistema especialista detém a estratégia de geração de alternativas e busca de soluções através da manipulação do conhecimento embutido no sistema. Este processo de geração de caminhos alternativos deduzindo conclusões via mecanismo de raciocínio através da base de conhecimentos é denominado inferência.

Intimamente ligada à representação de conhecimento escolhida para o sistema, a estratégia de inferência pode estar baseada em

III - ABORDAGENS AO PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO

propagação de restrições, encadeamento para frente ou controle dirigido por dados (forward chaining), encadeamento para trás ou controle dirigido por objetivos (backward chaining), ou ainda alguma estratégia mista, redução do problema, geração e teste, etc.

As figuras 8 e 9 mostram os processos de encadeamento para frente e para trás:

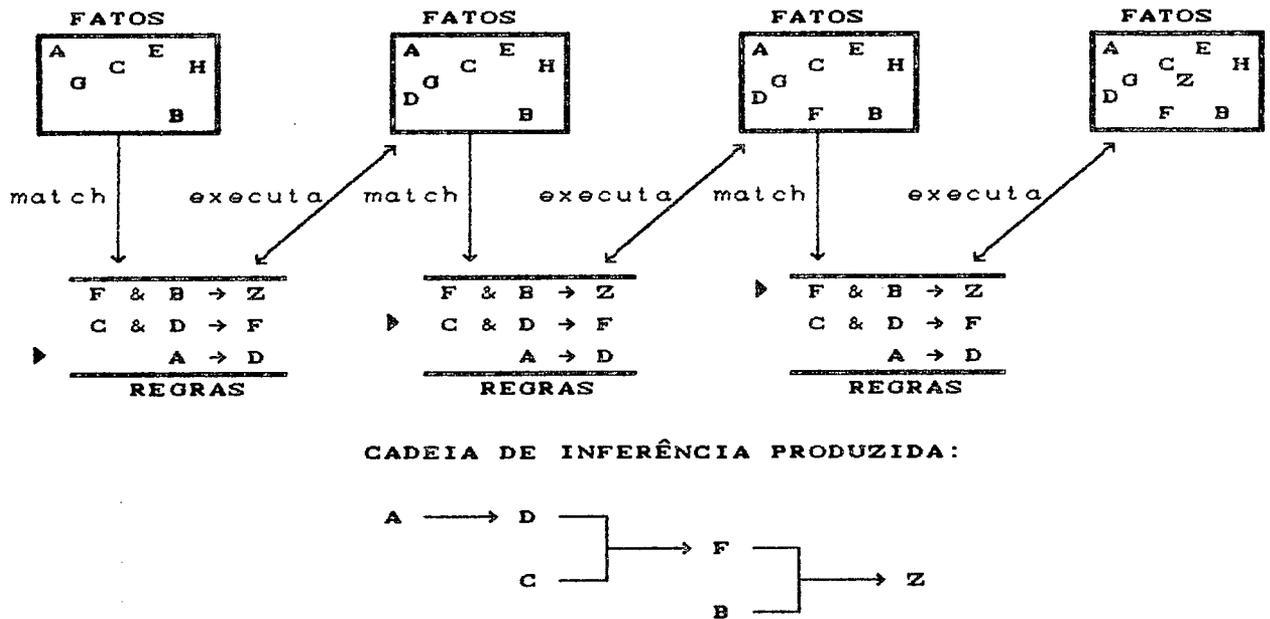
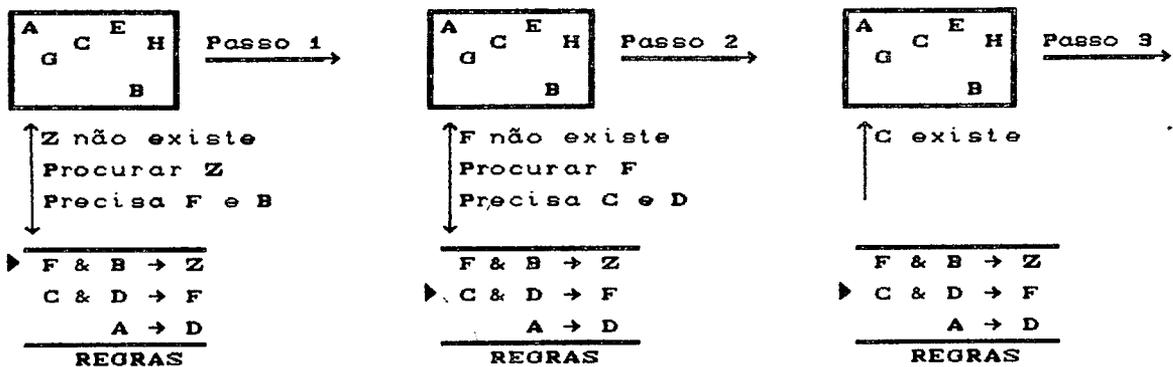


FIGURA 8 - Exemplo de inferência por encadeamento para frente  
 FONTE: [WAT86]

FIGURA 9



continua...

III - ABORDAGENS AO PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO

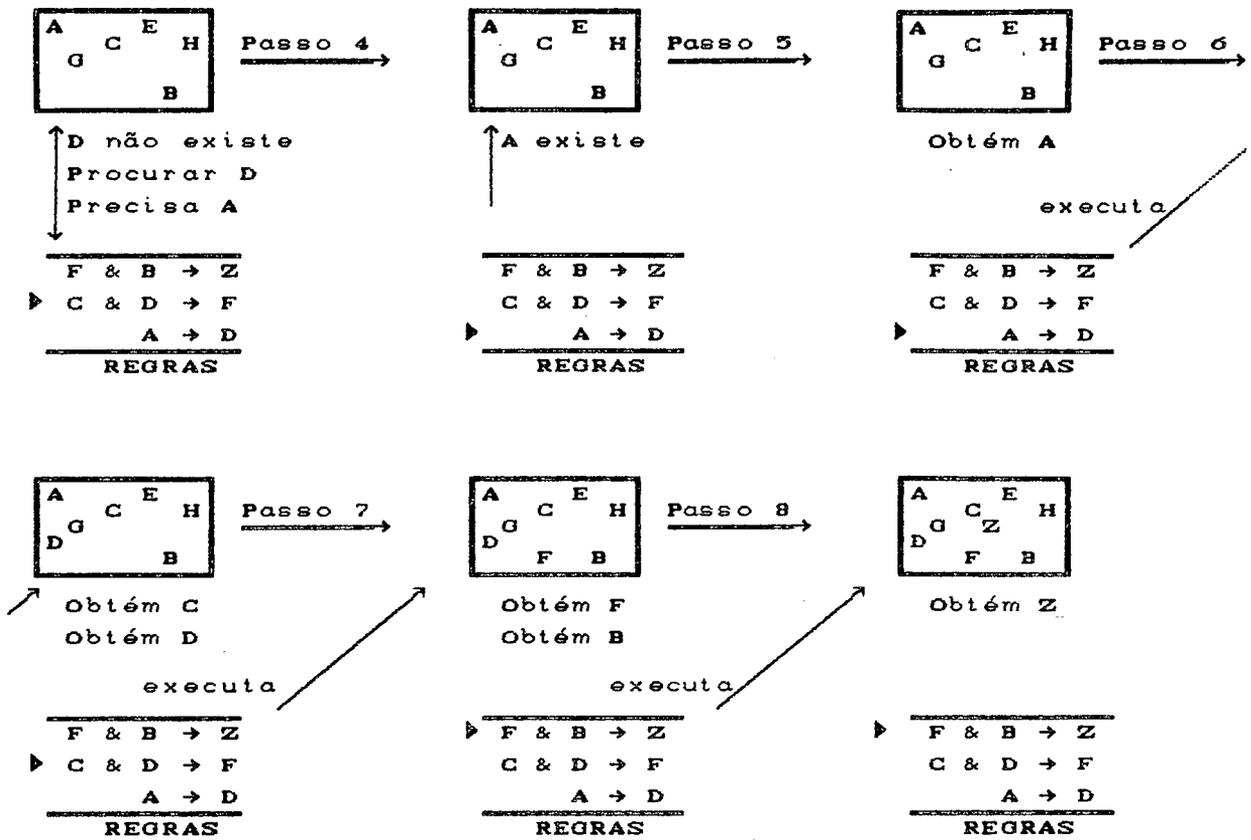


FIGURA 9 - Exemplo de inferência por encadeamento para trás  
 FONTE: [WAT86]

A maior preocupação quanto à adição de novas regras é quanto à consistência, conforme o exemplo dado em 3.4.2.2. O sistema especialista deve possuir um controlador de consistência e um resolvidor de conflitos.

3.4.2.4 Implementação

Um dos primeiros passos para a implementação é decidir sobre a linguagem de programação a ser usada. A primeira linguagem a conquistar popularidade foi o LISP. Em meados da década de 80, o Prolog, uma linguagem baseada em regras construídas sobre um provador de teorema de lógica de predicados, passou a ser a linguagem mais usada para a construção de sistemas especialistas. Atualmente, a tendência é utilizar ferramentas mais específicas conforme a aplicação, como as shells.

A tabela 3 dá uma estimativa da utilização de diversas ferramentas para construção de sistemas especialistas, de acordo com um estudo entre os membros da Operational Research Society:

## III - ABORDAGENS AO PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO

| Ferramentas e linguagens utilizadas                           | %  |
|---|----|
| Shells  | 62 |
| Prolog  | 30 |
| Outras linguagens de alto nível<br>(Pascal, C, Fortran, etc.) | 23 |
| LISP  | 18 |
| Environments  | 12 |
| POPLOG  | 4  |

TABELA 3 - FONTE: [DOU90]

Há características específicas numa linguagem voltada para a construção de sistemas de I.A. que facilitam o trabalho do programador, embora seja formalmente possível escrever qualquer programa em qualquer linguagem [RIC88].

Escolhida a linguagem para implementação, o conhecimento representado deve ser implementado na linguagem escolhida.

Doukidis & Paul estimam os níveis de utilização dos esquemas de representação de conhecimento através da tabela 4:

| Uso de técnicas de representação de conhecimento: |    |
|---|----|
| Técnicas  | %  |
| Regras  | 62 |
| Árvores   | 18 |
| Redes Semânticas                                  | 16 |
| Frames  | 15 |
| Representação híbrida                             | 15 |

TABELA 4 - FONTE: [DOU90]

O projeto da interface com o usuário deve ter estabelecido o nível de detalhe desejado. Supõe-se que o sistema especialista irá auxiliar um usuário pouco experiente, então, deverá ter uma interface amigável, com explicações concisas, claras e seguindo o processo de raciocínio. No caso de o sistema servir de apoio a um especialista, deverá ser ágil e evitar demoras e explicações excessivas.

Por fim, o mercado para sistemas especialistas está altamente condicionado ao equipamento existente nos clientes potenciais, como demonstra a tabela 5:

| Hardware utilizado   | Desenv.% | Usuários% |
|----------------------|----------|-----------|
| PC                   | 85       | 86        |
| Minicomputadores     | 24       | 12        |
| Workstations de I.A. | 6        | 10        |
| Mainframes           | 6        | 4         |

TABELA 5 - FONTE: [DOU90]

#### 3.4.2.5 Aprendizagem

Poucos resultados nesta área têm sido relatados.

Alguns tipos de aprendizagem mencionados na literatura são [KUM86]: aprendizagem por dicionário, tomada de recomendação, indução, aprendizado por matching, por analogia e aprendizagem baseada em regras.

A aprendizagem automática, nos sistemas onde está implementada, ainda se faz em níveis muito elementares, como a acumulação de sinônimos na base de conhecimento, no caso da aprendizagem por dicionário.

#### 3.4.3 Abordagens de Sistemas Especialistas

Diversas arquiteturas e abordagens têm sido usadas para construir sistemas especialistas para programação da produção. De acordo com Kusiak [KUS87], as seguintes abordagens têm sido usadas:

- Hierárquica
- Não-hierárquica
- Baseada em script (esqueleto)
- Oportunística
- Dirigida por restrições
- Múltipla ou Tandem

##### 3.4.3.1 Abordagem hierárquica

Na programação com abordagem hierárquica, o problema de programação é resolvido em vários níveis de abstração. O espaço de estados é reduzido pelo programador, nos primeiros níveis, usando dados agregados e, então, expandindo este programa agregado em programas mais detalhados. A qualidade da solução gerada depende fortemente de como essa redução do espaço de estados é feita.

A utilização desta abordagem apresenta dificuldades quando ocorre interação entre os subproblemas. Esta dificuldade pode ser eventualmente sanada através da representação das interações por restrições que são propagadas durante o processo de programação.

#### 3.4.3.2 Abordagem Não-hierárquica

Na abordagem não-hierárquica o programa é desenvolvido sem decomposição do problema. A principal desvantagem é que o espaço de estados pode ser muito grande e a importância e nível de detalhe dos dados podem ser subestimados.

#### 3.4.3.3 Abordagem Baseada em Script

Na abordagem baseada em script, estruturas básicas de programação (esqueletos) são armazenadas em um banco de dados, podendo ser chamadas, ao invés de geradas para cada situação. Naturalmente, as estruturas de programação podem variar quanto ao seu grau de detalhamento. Esta abordagem funciona em dois passos: Primeiro, um esqueleto é encontrado para definir o problema, então, os operadores de resolução de problema são usados para preencher os espaços vazios na estrutura de programação. Esta abordagem, embora simplifique o acompanhamento da solução, carece da flexibilidade necessária aos sistemas de programação.

#### 3.4.3.4 Abordagem Oportunística

A abordagem oportunística, também conhecida como "quadro-negro", possui um modo de representação e coordenação de conhecimento baseado no uso de grupos independentes de regras que fazem o papel dos especialistas, chamados fontes de conhecimento, que se comunicam por um banco de dados central chamado quadro-negro [WAT86].

As fontes de conhecimento são módulos de programação que tratam da geração independente de soluções parciais, combinando-as e avaliando-as. O quadro-negro faz a interligação entre as fontes de conhecimento.

O termo "oportunística" vem do fato de que as decisões tomadas são as que se apresentam mais promissoras para a solução.

#### 3.4.3.5 Abordagem Dirigida por Restrições

A abordagem dirigida por restrições pode ser vista como uma técnica de busca heurística, onde o domínio de conhecimento é

representado por restrições que limitam e guiam a busca por uma solução viável.

O programa de produção evolui através de estados que representam soluções parciais. As restrições podem variar de preferências de programação a limitações físicas ou tecnológicas.

No caso de restrições conflitantes, algumas são relaxadas e as alternativas aceitáveis são selecionadas.

#### 3.4.3.6 Abordagem Múltipla ou Tandem

A abordagem de otimização, que tem como elementos-chave um modelo e um algoritmo, tem seguidores principalmente na área da pesquisa operacional.

Kusiak [KUS87] propõe uma arquitetura múltipla para programação da produção. Esta arquitetura, chamada tandem, tem dois componentes distintos:

- a) Sistema especialista
- b) Modelo e algoritmo

Esta abordagem busca tirar vantagem das abordagens de sistemas especialistas e pesquisa operacional. Para problemas que podem ser quantificados e formulados, a abordagem de otimização é designada. Problemas qualitativos são melhor abordados pela inteligência artificial.

Um sistema sob esta abordagem trabalha com dados do sistema de manufatura e do ambiente de fabricação. Baseado nestes dados, o sistema gera os dados para o modelo. A solução gerada pelo algoritmo é avaliada pelo sistema especialista. No caso de a solução ser aceitável, será mandada para o sistema de manufatura, caso contrário, o ciclo é repetido.

Podem existir vários modelos e algoritmos de programação no sistema. Neste caso, o sistema especialista seleciona um modelo apropriado para receber os dados.

#### 3.4.4 Sistemas Desenvolvidos

"A pesquisa sobre programação da produção tem tido, até o momento, pouco impacto sobre o problema real de programação da produção" [SMI86].

Relativamente poucos resultados de pesquisa têm sido

### III - ABORDAGENS AO PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO

documentados na literatura. A Universidade Carnegie-Mellon (EUA) tem liderado a pesquisa com ISIS, PTRANS, OPIS e PATRIARCH, entre outros.

Dentre os sistemas existentes, ISIS é o melhor exemplo conhecido de aplicação de técnicas de IA para o problema de programação de job shop. A representação de conhecimento usada no ISIS supera em muito o domínio de informações encontrado na literatura de PO [McK88].

O quadro 5 classifica os estágios de desenvolvimento de sistemas especialistas:

**QUADRO 5 - EVOLUÇÃO DOS SISTEMAS ESPECIALISTAS**

| Estágio de desenvolvimento | Descrição   |
|----------------------------|---|
| Protótipo de demonstração  | O sistema resolve uma parte do problema, sugerindo que a abordagem é viável e o desenvolvimento é possível.           |
| Protótipo de pesquisa      | O sistema abarca o problema por completo, mas pode ser frágil devido à revisão e teste incompletos.                   |
| Protótipo de campo         | O sistema mostra boa performance com confiabilidade adequada, além de ser testado exhaustivamente no ambiente de uso. |
| Modelo de produção         | O sistema exibe qualidade, confiabilidade e velocidade altas e desempenho eficiente no ambiente de uso.               |
| Sistema comercial          | O sistema é um modelo de produção sendo usado em uma base comercial regular.  |

FONTE: [WAT86]

Poucos sistemas especialistas têm atingido o nível de modelo ou protótipo de produção. Segundo Waterman [WAT86], um protótipo de produção típico baseado em regras poderá conter de 500 a 1500 regras e levar de dois a quatro anos para desenvolver. O XCON, um sistema especialista que configura sistemas de computador VAX-11/780, um dos primeiros sistemas especialistas comerciais, tem mais de 3000 regras, chega a conclusões corretas em 90 a 95% das vezes e levou seis anos para ser desenvolvido.

### III - ABORDAGENS AO PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO

O quadro 6 descreve resumidamente cerca de 40 trabalhos feitos na área de planejamento, programação e controle da produção, não necessariamente industrial, assinalando seus esquemas de representação de conhecimento, estratégias de inferência, abordagem e propósitos.

QUADRO 6

| QUADRO-RESUMO DAS APLICAÇÕES DE SISTEMAS ESPECIALISTAS NO PLANEJAMENTO, PROGRAMAÇÃO E CONTROLE DA PRODUÇÃO |   |
|--|---|
| NOME<br>Autores  | CARACTERÍSTICAS   |
| ISIS<br>ISIS-II<br>Fox &<br>Smith  | <ul style="list-style-type: none"> <li>- Sistema para programação da produção, ambiente job shop de grande escala.</li> <li>- Utiliza raciocínio dirigido por restrições.</li> <li>- O conhecimento consiste principalmente de restrições e experiência do programador.</li> <li>- As restrições são organizadas em categorias: objetivos organizacionais, restrições físicas, causais, de disponibilidade e de preferência.</li> <li>- Auxilia gerentes de fábrica a manter a consistência dos programas de produção e identificar decisões que resultam em restrições não satisfeitas.</li> <li>- O sistema seleciona uma seqüência de operações necessárias para completar um pedido, determina tempos de início e fim e atribui recursos a cada operação.</li> <li>- A busca é organizada em quatro níveis: seleção de ordens, análise de capacidade, programação de recursos e seleção de reservas.</li> <li>- Programado em LISP.</li> <li>- Usa linguagem SRL (Schema Representation Language) para a representação das frames, que contêm três slots: condição, função de avaliação e peso.</li> <li>- Ocupa em torno de 10 MB de área de disco.</li> <li>- Desenvolvido na Universidade Carnegie-Mellon, EUA, em projeto conjunto com a Westinghouse Electric Corporation, em três versões, atingindo o status de protótipo de pesquisa.</li> <li>- O protótipo não foi implementado, em parte devido à dificuldade para integrar o sistema com os bancos de dados e sistemas de informação existentes.</li> <li>- Referências: [FOX84] [SMI86] [SMI90] [KUM86] [KUS88] [KLE89] [PAC90] [ROD88] [WAT86]</li> </ul> |
| OPIS<br>Smith<br>et alii   | <ul style="list-style-type: none"> <li>- Usa uma extensão do conceito de busca dirigida por restrições do ISIS.</li> <li>- Utiliza abordagem oportunística ou "quadro-negro".</li> </ul>  |

continua...

- Adota uma representação de conhecimento semelhante à do ISIS, utilizando frames.
- Usa propagação de restrições.
- Possui dois tipos de fontes de conhecimento: de análise e de programação.
- Fontes de conhecimento de análise: Analisador de Capacidade (CAN) e Analisador de Conflitos (CONAN).
- Fontes de conhecimento de programação: Programador de Pedidos (OSC), Programador de Recursos (RSC), Right Shifter (RSH), Permutador de Demanda (DSW).
- Desenvolvido na Universidade Carnegie-Mellon, EUA.
- Referências: [SMI90] [SMI86] [KUM86] [KLE89] [PAC90] [ROD88]

MASCOT  
Erschler &  
Esquirol

- Sistema especialista para programação da produção, ambiente job shop.
- Conhecimento representado através de regras.
- Estratégia de controle consiste de meta-regras e análise baseada em restrições.
- Escrito em Prolog.
- Referências: [KUS88]

OPAL  
Bensana  
et alii

- Sistema para escalonamento de operações. Não determina roteiro de fabricação.
- Representação de Conhecimento: Descreve a estrutura física através de abordagem centrada em objeto, o escalonamento em curso através de um grafo de precedência, a propagação de restrições de tempo foi escrita em Prolog e o conhecimento empírico e prático é expresso por regras IF...THEN.
- O motor de inferência usa duas estratégias, alternadamente: Análise Baseada em Restrições e Regras manipuladas por lógica difusa.
- O programa de produção é expresso segundo um grafo de precedência de operações e segundo suas datas-limite para início e término de operações.
- A linguagem de implementação é o COMMON LISP, com funções DEFSTRUCT.
- Desenvolvido no Centre d'Études et Recherches de Toulouse, França.
- Referências: [BEN88] [BEL88] [KUS88]

KBRS  
Ben-Arieh  
et alii

- Sistema baseado em conhecimento que trata de roteiros de fabricação e dispositivos de montagem na produção automatizada em ambiente job shop. Voltado ao chão-de-fábrica.
- O conhecimento é de dois tipos: regras sobre produção e conhecimento procedural.
- Utiliza uma abordagem hierárquica, em três níveis: planejamento de pré-liberação, controle de entrada e controle operacional.

- Conflitos são resolvidos através do ordenamento das regras.
  - Escrito em Prolog, para máquina VAX 11/78D em ambiente Unix. Testado em um ambiente simulado. O subsistema de simulação foi escrito em Slam II. A base de dados dinâmica possui conhecimento algorítmico escrito em Prolog e Pascal.
  - Referências: [KUM86] [KUS88] [KLE89]
- ISA**  
Orciuch  
& Frost
- Intelligent Scheduler Assistant.
  - Desenvolvido para executar planejamento e programação mestre da produção na Digital Equipment Corporation.
  - Conhecimento representado através de regras.
  - Contém aproximadamente 300 regras.
  - Possui uma estratégia para tratar ordens de difícil programação.
  - Referências: [KAN87] [ROD88] [KUS88]
- IMACS**  
O'Connor
- Sistema especialista para planejamento de capacidade, gerenciamento de estoques, etc.
  - Conhecimento representado através de regras.
  - Utiliza mecanismo de encadeamento para frente.
  - Implementado em OPS5.
  - Desenvolvido pela Digital Equipment Co., atingindo o status de protótipo de campo.
  - Referências: [KUS88] [WAT86]
- Bruno**  
et alii
- Sistema especialista para programação da produção de lotes de tarefas em sistemas flexíveis de manufatura.
  - Conhecimento representado através de regras.
  - A estratégia de controle é feita parcialmente por simulação.
  - Possui um módulo de simulação para avaliação da saída do sistema especialista.
  - Escrito em OPS5 e Fortran.
  - Desenvolvido no Politecnico di Torino, Itália.
  - Referências: [BRU86] [KUS88] [KLE89]
- GARI**  
Descotte  
& Latombe
- Sistema baseado em conhecimento para planejamento de processos.
  - Restrito à indústria de corte de metal.
  - O conhecimento é representado por regras de produção, onde a parte esquerda contém condições sobre a parte a ser manufaturada e a parte direita contém recomendações relativas a preferências econômicas e tecnológicas, com pesos associados. Como muitas restrições são conflitantes, os pesos são usados para resolver os conflitos.
  - Implementado em MACLISP.
  - Sistema experimental, com mais ou menos 50 regras.
  - Referências: [KUM86]

- PATRIARCH**  
Morton  
& Smunt
- Sistema para planejamento e programação de projetos em sistemas flexíveis de manufatura.
  - Conhecimento representado através de frames e redes-E.
  - Usa uma abordagem hierárquica, com uma estrutura em quatro níveis: planejamento estratégico, planejamento de capacidade, escalonamento e liberação.
  - Estratégia de controle por heurísticas e regras de produção.
  - Possui módulo externo com modelos de pesquisa operacional.
  - Desenvolvido na Universidade Carnegie-Mellon, EUA.
  - Referências: [KUS88] [KLE89]
- SOJA**  
Le Pape  
& Sauve
- Não executa a atribuição de operações a máquinas (tarefa de planejamento).
  - Utiliza uma estratégia de busca dirigida por restrições, envolvendo um processo de seleção de operações que reduz o tamanho do problema.
  - Referências: [BEN88]
- SONIA**  
Sauve
- Sistema para escalonamento de fábrica que trabalha sob encomenda.
  - Utiliza uma abordagem oportunística ou "quadro-negro".
  - Ligado a um sistema MRP.
  - Desenvolvido no Laboratoire de Marcoussis, França.
  - Referências: [KLE89]
- CROPS**  
Lecocq  
et alii
- Sistema para programação e controle on-line de um sistema flexível de manufatura.
  - Utiliza abordagem hierárquica.
  - Utiliza propagação de restrições, construindo uma rede temporal com tempos e ordens de sucessão.
  - Recebe dados de um sistema MRP II.
  - Escrito em OPS5.
  - Desenvolvido no Centre de Recherches Scientifiques et Techniques de l'Industrie des Fabrications Métalliques, Bélgica.
  - Referências: [LEC87] [KLE89]
- FIXER**  
Grant
- Sistema especialista para a programação de trabalho de reparo em aviões.
  - Conhecimento representado através de frames e regras.
  - Estratégia de controle por regras heurísticas.
  - O sistema está embutido em um sistema de gerenciamento de informações e permite incorporar um programa PERT.
  - Escrito em NIAL (Nested Interactive Array Language).
  - Referências: [KUS88]

- NUDGE e BARGAIN Goldstein & Roberts**
- Dois programas que constituem um sistema baseado em conhecimento para programação da produção.
  - Conhecimento representado através de frames.
  - Utiliza técnicas de busca tradicionais.
  - Referências: [KAN87]
- SISEP Kleebank**
- Sistema especialista para escalonamento da produção.
  - Representação de conhecimento baseada em frames.
  - O mecanismo de controle contém um avaliador de regras e um ou mais algoritmos de escalonamento, tendo implementado um algoritmo baseado no OPT, até o momento.
  - Escrito em Prolog.
  - Desenvolvido na Universidade Federal do Rio Grande do Sul.
  - Referências: [KLE89]
- WATPASS McKay**
- Sistema especialista baseado em regras para programação da produção.
  - Recebe dados de um sistema MRP.
  - Referências: [KLE89]
- Steffen & Greene**
- Sistema para programação de processadores paralelos em um sistema flexível de manufatura.
  - O esquema de representação de conhecimento não é descrito.
  - Utiliza busca dirigida por restrições.
  - Protótipo escrito em Virginia Tech Prolog.
  - Referências: [KUS88]
- Kerr & Ebsary**
- Sistema especialista para programação da produção em uma pequena manufatura de equipamento para bombeamento.
  - Possui uma base de conhecimento com 325 regras, depois expandida para 430.
  - Desenvolvido na Universidade de Nova Gales do Sul, Austrália.
  - Referências: [KAN87]
- Shen & Chang**
- Sistema especialista para programação da produção em ambiente job shop generalizado.
  - Conhecimento representado através de frames e regras.
  - Possui módulo externo com algoritmos para detalhar a programação.
  - Não foi relatada aplicação deste sistema.
  - Referências: [KUS88]
- SCORE Chiadini**
- Faz o replanejamento em tempo real, baseado em MRP.
  - Esquema de representação de conhecimento não mencionado.
  - Utiliza busca dirigida por restrições.
  - Utiliza um módulo externo MRP.

- Escrito em Knowledge Craft.
  - Referências: [KUS88]
- YAMS/  
CASCADE  
Parunak**
- Sistemas para manufatura distribuída.
  - Utiliza busca dirigida por restrições.
  - Tem a estrutura de um sistema distribuído de inteligência artificial.
  - Escrito em PropLan.
  - Referências: [KUS88]
- Rahr**
- Pesquisa sobre um sistema de replanejamento da produção baseado em MRP.
  - Tem como objetivos: Detectar eventos críticos, gerar alternativas de programação e facilitar a reprogramação.
  - Referências: [KUS88]
- AAAS  
anônimo**
- Sistema de planejamento e programação de grande escala para uma planta montadora de aviões.
  - Desenvolvido e implementado na Lockheed Georgia Company.
  - Composto de quatro subsistemas, entre eles o Genplan, um sistema especialista construído a partir da experiência de peritos da companhia.
  - Desenvolvido ao longo de 20 anos.
  - Referências: [KUS88]
- DEVISER  
Vere**
- Planejador/programador de propósito geral que gera uma rede de programação, especificando tempos de início de operações.
  - Desenvolvido para planejar e programar ações a bordo de espaçonaves não tripuladas, como o Voyager, que devem trazer fotos de objetos no espaço.
  - Referências: [ROD88]
- Deal  
Chen  
Ignizio  
Jeyakumar**
- Desenvolvido para programar operações relativas a processos de produção em uma refinaria petroquímica (processos contínuos).
  - Construído sobre a shell Exsys.
  - Desenvolvido na Universidade de Houston, EUA.
  - Referências: [DEA90]
- SIPP  
Nau &  
Chang**
- Sistema baseado em conhecimento para planejamento de processo de partes usinadas.
  - Representação de conhecimento baseada em frames.
  - A base de conhecimento consiste de superfícies usináveis e capacidades de várias operações de usinagem e uma estrutura de controle que manipula a base de conhecimento para construir planos de processamento.
  - Usa uma estratégia 'best-first' baseada em branch-and-bound.
  - Programado em Prolog, composto de 55 frames.
  - Referências: [KUM86]

- TOM**  
Matsushima  
et alii
- Sistema baseado em regras.
  - Usa uma variante do encadeamento para trás em seu motor de inferência.
  - Conflitos entre regras são resolvidos usando heurísticas, como: usar a regra encontrada antes ou a última encontrada, aplicar a regra mais frequentemente usada, etc.
  - Possui um módulo de explicação que permite mostrar a linha de encadeamento de regras.
  - O sistema não produz planos completos de operações, mas enfoca planos detalhados para algumas características.
  - Escrito em Pascal, para um computador VAX-11.
  - Desenvolvido na Universidade de Tóquio, Japão.
  - Referências: [KUMB6]
- PTRANS**  
Haley  
et alii
- Sistema baseado em regras para controle da manufatura e distribuição de sistemas de computadores.
  - Utiliza mecanismo de encadeamento para frente.
  - Implementado em OPS5.
  - Desenvolvido na Universidade Carnegie-Mellon, EUA, em conjunto com a Digital Equipment Co., atingindo o status de protótipo de pesquisa.
  - Referências: [WATB6]
- AGMPO**  
Eshel  
et alii
- Sistema baseado em regras para geração automática de esboços no projeto de formas.
  - As regras têm uma organização hierárquica.
  - A ativação das regras é controlada por uma 'regra de gerenciamento' que busca entre as regras aquelas cujo escopo contém o problema.
  - O sistema usa uma técnica de geração e teste para retificar as operações e para formular hipóteses antes de retificar as operações.
  - Programado em Prolog, em ambiente Unix.
  - Referências: [KUMB6]
- EXCAP**  
Davis &  
Darbyshire
- Sistema especialista para gerar planos de processamento de usinagem de componentes rotacionais.
  - Sistema baseado em regras.
  - Usa mecanismo de encadeamento para trás.
  - Desenvolve uma árvore de possíveis seqüenciamentos. Cada operação disponível é uma hipótese na rede de inferência, que o sistema tenta provar ou negar.
  - Programado em Pascal, em um VAX 11/750, sistema operacional VMS.
  - Referências: [KUMB6]
- CUTTECH**  
Barkocy &  
Zdeblick
- Sistema desenvolvido para seleção de ferramentas de corte, tamanhos, velocidades e níveis de alimentação, requerendo conhecimento sobre máquinas.
  - Não é um sistema especialista propriamente dito, mas incorpora ferramentas de IA.

continua...

- A base de conhecimento é composta de regras sobre usinagem e dados sobre máquinas e instrumentos.
  - As regras são armazenadas em tabelas de decisão e algoritmos e aplicadas em ordem decrescente de importância para agrupar uma lista de ferramentas, da mais à menos preferida. A ênfase do sistema é na implementação e seu caráter é prático por natureza.
  - Referências: [KUM86]
- PWA-PLANNER**  
Terwilliger
- É um protótipo para o planejamento da montagem de circuitos impressos.
  - Dados sobre as partes, layout e dimensões são utilizados para atribuir as partes às máquinas montadoras e as seqüências em cada máquina, evitando colisões entre componentes.
  - A base de conhecimento consiste de regras para inserção de componentes, dados sobre máquinas e outros fatos relevantes para o planejamento.
  - Usa o conceito de 'planejamento para frente'.
  - Programado em Prolog, para ambiente Unix.
  - Referências: [KUM86]
- Shaw & Whinston
- Sistema baseado em conhecimento para programação da produção em sistemas flexíveis de manufatura.
  - Utiliza 'método de planejamento não-linear'.
  - Utiliza encadeamento para frente e para trás.
  - Conhecimento representado através de predicados de primeira ordem e operadores.
  - Implementado em LISP.
  - Referências: [SHAB9] [KUS88] [KLE89]
- AURORA**  
Yu
- Sistema para planejamento de missões aéreas.
  - Conhecimento representado através de rede de restrições.
  - Estratégia de controle por matching de padrões.
  - Possui módulo externo de programação linear, ativado quando ocorre matching de um padrão.
  - Escrito em Lisp, Ada e C.
  - Referências: [KUS88]
- Subramanyam & Askin
- Sistema especialista para programação da produção em sistemas flexíveis de manufatura.
  - Conhecimento representado através de regras.
  - A aquisição das regras foi feita baseada principalmente em simulação de ambiente job shop.
  - Possui uma estrutura de decisão com três níveis de hierarquia.
  - Escrito em Prolog.
  - Referências: [KUS88]
- Fellenstein et alii
- Sistema para planejamento da capacidade de um sistema de manufatura.

- Conhecimento representado através de regras (fatos e silogismos).
  - Escrito na shell Syllog.
  - Referências: [KUS88]
- Thesen  
& Lei
- Sistema para programação de robôs em um sistema flexível de eletrodeposição.
  - Conhecimento representado através de regras.
  - Estratégia de controle por meta-regras.
  - A aquisição de conhecimento foi feita usando simulação.
  - Referências: [KUS88]
- Newman  
& Kempf
- Sistema para programação de robôs em um sistema flexível de manufatura.
  - Conhecimento representado através de regras, divididas em seis níveis.
  - Possui aproximadamente 250 regras de produção.
  - A aquisição de conhecimento foi feita usando simulação da célula de manufatura.
  - A performance do sistema foi testada em ambiente simulado.
  - Escrito em Pascal.
  - Referências: [KUS88]
- Wynot
- Sistema baseado em técnicas de IA para o controle de um sistema de montagem.
  - O esquema de representação de conhecimento e a estratégia de controle não são relatados.
  - Composto de dois níveis: 'Mover', que trata da manipulação de materiais, e 'Trim scheduler', que atribui as operações a cada estação.
  - A atribuição das tarefas é feita com base no princípio 'First-In-First-Out' ou outra regra de prioridade.
  - Desenvolvido na Digital Equipment Co.
  - Referências: [KUS88]
- FADES  
Fisher
- Sistema para projeto e planejamento de dispositivos.
  - O conhecimento está representado na forma de regras implementadas em procedimentos lógicos e predicados de primeira ordem.
  - A base de conhecimento abrange as áreas: tecnologia de estações de trabalho e análise econômica de investimentos, desenvolvimento de taxas de relacionamento entre estações de trabalho, seleção de algoritmos de atribuição, preparação das entradas e chamada de algoritmo para planejamento de layout, recuperação da informação necessária para o uso das regras.
  - Apresenta bom desempenho como pré-processador para algoritmos existentes, como o CORELAP.
  - Referências: [KUM86]

continua...

- |                                 |  |
|---------------------------------|--|
| <b>IFLAPS</b><br>Tirupatikumara | <ul style="list-style-type: none"><li>- Sistema multicritério para planejamento e análise de layout de dispositivos.</li><li>- Sistema baseado em regras.</li><li>- Considera critérios como segurança, comodidade dos trabalhadores e satisfação de restrições.</li><li>- Conhecimento representado por predicados lógicos.</li><li>- Inferência por encadeamento para frente.</li><li>- Implementação em Prolog.</li><li>- Usa a abordagem de Redes de Transição Aumentadas (ATN) de processamento de linguagem natural para geração configurações de layout alternativas.</li><li>- Os conflitos entre objetivos são resolvidos por heurísticas baseadas em regras e interação com o usuário.</li><li>- O sistema explica seu raciocínio através da interface com o usuário e adquire novo conhecimento através de aprendizagem baseada em regras.</li><li>- Referências: [KUM86]</li></ul> |
|---------------------------------|--|

**QUADRO 6 - APLICAÇÕES DE SISTEMAS ESPECIALISTAS**

A maioria destes sistemas constitui-se de protótipos. ISIS e GARI foram testados no ambiente industrial [KUM86].

A maior parte dos sistemas está baseada no uso de regras (Mascot, Opal, Isa, Imacs, etc.). Algumas aplicações importantes utilizaram frames (ISIS, Fixer).

Quanto à estratégia de inferência, existe uma tendência à adoção de mecanismos que utilizam múltiplas estratégias ao invés de métodos puros.

As linguagens de implementação mais utilizadas, Prolog e Lisp, vêm cedendo lugar à aplicação de shells, que existem em diversidade cada vez maior e eventualmente conseguem captar melhor aspectos específicos de uma dada aplicação.

#### IV - UM SISTEMA ESPECIALISTA PARA PROGRAMAÇÃO DA PRODUÇÃO

Os sistemas computacionais voltados à programação da produção existentes usam técnicas simples de desdobramento de operações (como os sistemas MRP) e, eventualmente, algoritmos de pesquisa operacional.

Algumas lacunas deixadas por estes sistemas são comentadas por Graves [GRAB1]:

*As técnicas correntes de pesquisa operacional podem não estar à altura, ou serem inadequadas, ou ainda não serem necessárias para muitos ambientes de produção. (...) Temos visto que há uma lacuna entre a teoria e a prática da programação da produção. Os futuros esforços de pesquisa devem ser dirigidos à redução destas diferenças.*

O problema prático de programação da produção é resolvido usualmente por programadores, baseados na própria intuição e experiência. A abordagem de sistemas especialistas busca modelar este conhecimento especializado em um sistema computacional.

Este capítulo delinea a arquitetura de um sistema especialista para programação da produção.

#### 4.1 CARACTERÍSTICAS DE UM PROGRAMA ESCALONADOR DA PRODUÇÃO

Entre as qualidades desejáveis e esperadas de um programa escalonador de um sistema deste tipo, podem-se destacar:

- Flexibilidade, permitindo a reprogramação frente a perturbações no ambiente. Um comentário freqüente ouvido em muitos ambientes de programação é que não há um problema de programação, mas um problema de reprogramação. Pode ser muito fácil construir um escalonamento, o que é difícil é a revisão constante que é imposta pelo ambiente dinâmico [GRAB1].

- Possibilidade de acompanhamento da programação em curso, permitindo visualizar o progresso do programa e a aplicação das estratégias inseridas no sistema;

- Produção de um programa flexível e robusto, pouco sensível a pequenas perturbações. O programa de produção precisa ter a capacidade de absorver e ser adaptado a pequenas variações no

chão-de-fábrica.

- Modularidade, possibilitando ampliações e modificações no sistema.

- Interface amigável, facilitando e agilizando a entradas de dados, edição de dados e regras, acesso a informações e saídas do sistema.

- Funcionamento factível em tempo real, o que tem sido uma barreira à utilização das abordagens ótimas de pesquisa operacional.

Os mesmos requisitos de flexibilidade, modularidade, eficiência e facilidade de comunicação homem-máquina devem ser atendidos pelo banco de conhecimento, manipulável pelo programa escalonador.

#### 4.2 ARQUITETURA DO SISTEMA

A arquitetura apresentada aqui é a de um sistema especialista modular para programação da produção, agregando técnicas de propagação de restrições e decisões de precedência de operações baseadas em regras, utilizando a teoria de conjuntos difusos.

O sistema faz a programação operando 'janelas' de tempo definidas para cada operação, que levam em conta o tempo de processamento e datas-limite para início e fim da operação.

O sistema não executa a atribuição de operações a máquinas, o que é um pressuposto. Havendo máquinas alternativas, a atribuição de operações é considerada tarefa de planejamento anterior à entrada neste sistema.

As restrições de precedência iniciais são definidas pelos roteiros de fabricação, entre operações referentes ao mesmo produto (mesma tarefa). A partir das janelas de tempo e tempos de processamento, o sistema definirá as ordenações entre operações realizadas em uma mesma máquina.

O sistema detalhado a seguir possui uma arquitetura semelhante à dos sistemas OPAL [BEN88] [BEL88] e MASCOT. O motor de inferência utiliza-se de mais de uma estratégia. No caso, um módulo de análise de restrições toma decisões de precedência de operações baseado nas datas-limite para início e término de operações. Um módulo de controle gerencia a alternância entre esta

estratégia e as decisões de precedência tomadas no módulo escalonador difuso, que gera novas restrições utilizando regras manipuladas por lógica difusa. As novas restrições são propagáveis através do módulo de ajuste de datas, que utiliza as relações de precedência existentes para limitar as datas de início e término de cada operação.

A primeira estratégia de inferência, análise de restrições, trata o problema de forma determinística, através de um analisador que atualiza as datas críticas de início e fim de cada operação em função de relações de precedência e de restrições, como: data de entrega de pedidos, datas das operações precedentes e sucessoras, início e fim do período de escalonamento.

O segundo módulo ataca o problema fazendo ordenações de operações que não foram executadas no primeiro módulo, devido ao fato de a abordagem de propagação de restrições ser usualmente sub-restrita. Os critérios de escolha de quais operações ordenar e a forma de ordenar neste segundo módulo devem atender à filosofia de fabricação adotada na empresa e estar de acordo com o ambiente real de manufatura e os objetivos a serem atingidos com o escalonamento.

Desta forma, o tratamento determinístico feito através da análise e propagação de restrições, contendo conhecimento puramente matemático, é integrado ao conhecimento pouco estruturado, empírico ou prático contido nas regras de ordenação do módulo decisor difuso.

O funcionamento do sistema é esquematizado na figura 10 e seus módulos são descritos a seguir:

#### 4.2.1 Banco de Dados

O banco de dados contém a representação do ambiente físico e dos objetivos de produção, ou seja, roteiros de fabricação, recursos, estruturas de produtos, tempos e quantidades de produtos a serem programadas.

As entidades representadas têm múltiplas relações de hierarquia: cada operação é referente a uma tarefa relativa a determinado produto final (com operações precedentes e sucessoras na tarefa), tem tempo de processamento e tempos-limite para início e término, está associada a uma determinada máquina (com relações de precedência relativas às demais operações realizadas na mesma

máquina) e pode ter outros atributos designados pelo usuário, por exemplo, grau de prioridade.

O estágio de desenvolvimento atual do sistema não considera ainda tempos de ajuste de máquina entre operações. A representação de conhecimento é ainda objeto de pesquisa.

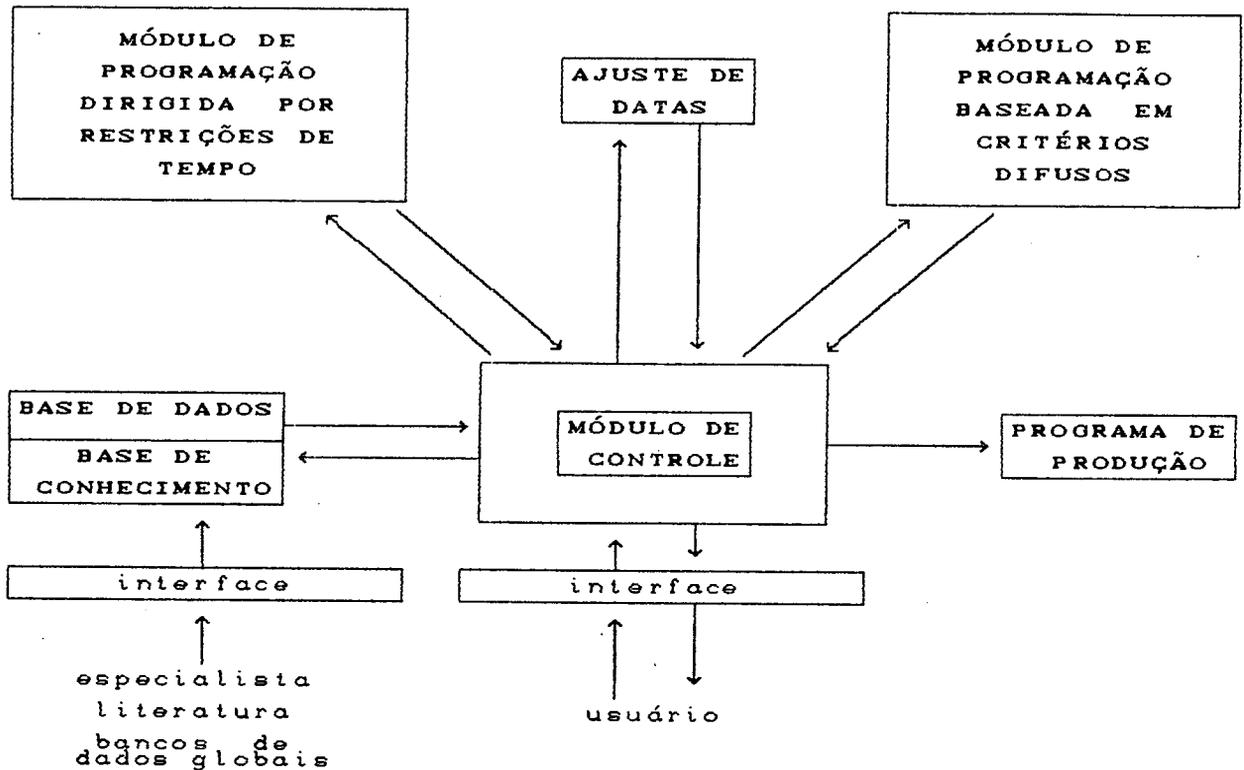


FIGURA 10 - ESQUEMA DE FUNCIONAMENTO DO SISTEMA ESPECIALISTA

#### 4.2.2 Banco de Conhecimento

Contém a descrição das regras e heurísticas de programação.

Enquanto datas-limite de início e fim de operações representam restrições de programação, as regras de liberação e heurísticas usadas pelos programadores expressam preferências, que podem ser tratadas como restrições enunciadas de forma difusa.

As regras quantificam preferências relativas à ordenação de um par de operações, de acordo com um critério próprio.

As regras existentes na base de conhecimento recebem um peso ou grau de ponderação  $P(R)$ , no intervalo  $[0,1]$ , que limita a influência da regra. Assim, objetivos de programação conflitantes poderão ser considerados de forma ponderada.

Cada regra focaliza um tipo específico de argumento de cálculo nas operações. Assim, uma regra definida como "programar

primeiro as operações com menor tempo de processamento" terá como argumento de cálculo os tempos de processamento das operações.

A seguir, define-se a regra "Folga dinâmica":

Regra R01:

Programar primeiro a operação que preserva as maiores folgas.

P(R01): grau de ponderação (definido pelo usuário)

argumento: folga dinâmica (data de entrega - data de início mais cedo - tempo de processamento) - folga existente depois de feita a ordenação do par.

Exemplo:

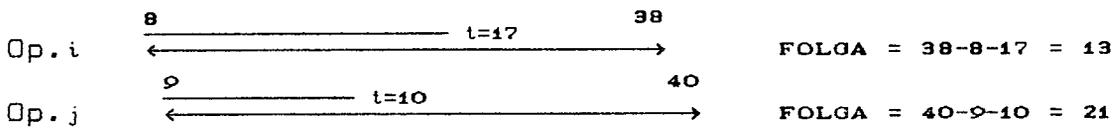
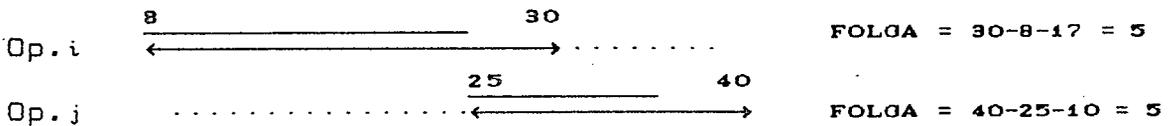


FIGURA 11

O esquema da figura 11 representa as janelas de tempo de duas operações não ordenadas, com os respectivos tempos de processamento e datas-limite para início e término.

Segundo esta regra, deve-se projetar a situação após a ordenação, pois o objetivo da regra é preservar as maiores folgas. Há duas situações possíveis, conforme a figura 12:

Precedência: i → j



Precedência: j → i

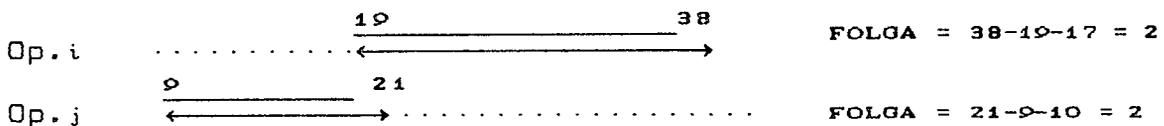


FIGURA 12

De acordo com o argumento de cálculo desta regra, folga dinâmica, a ordenação a ser feita deve ser: Operação i precede Operação j (i → j), pois esta ordenação mantém as operações com 5 unidades de tempo de folga, enquanto que a ordenação alternativa levaria a folgas de 2 unidades de tempo, para ambas operações.

Além das regras de programação de operações, outras regras práticas ou heurísticas referentes à atividade de programação podem ser definidas no banco de conhecimento. Uma regra usada no exemplo do capítulo 5 diz respeito à limitação do número de ordenações feitas no módulo decisor difuso. No sistema OPAL [BEN88], a taxa de 20% do total de operações não ordenadas foi encontrada como a mais eficiente. Este valor é utilizado para construir a regra que limita o número de ordenações no módulo decisor difuso:

$$N_{ord} < 1,5 + 0,20 * (\text{número total de pares não ordenados})$$

Outro critério expresso no banco de conhecimento refere-se à formação de um subgrupo de pares não ordenados para serem submetidos às regras no módulo escalonador difuso. O critério a ser utilizado no exemplo do próximo capítulo é o de selecionar pares de operações nas máquinas mais carregadas, segundo o "índice de flexibilidade absoluta" definido em [BEN88]:

$$I.F._m = \sum_i (dtt_{im} - dic_{im} - d_{im}) / \sum_i d_{im}$$

onde **dtt**, **dic** e **d** são os tempos de término mais tarde, início mais cedo e duração para as *i* atividades na máquina *m*.

Quanto menor o índice, maior a prioridade das ordenações na máquina.

Estas são regras práticas que visam ao bom funcionamento do sistema, evitando a geração excessiva de restrições pelo módulo escalonador difuso.

#### 4.2.3 Ajuste de Datas

É o módulo que propaga as restrições de tempo geradas através das novas relações de precedência entre operações feitas em uma mesma máquina.

As datas são ajustadas para operações feitas na mesma máquina e também propagadas para outras operações, segundo as seqüências de processamento de operações de uma mesma tarefa.

O ajuste tem um princípio semelhante ao dos métodos PERT/CPM, manipulando datas de início mais cedo e término mais tarde. Os novos tempos são calculados para operações sucessivas e podem ser expressos da seguinte forma:

$$dic_{\beta} \geq dic_{\alpha} + d_{\alpha} \qquad dtt_{\alpha} \leq dtt_{\beta} - d_{\beta}$$

definindo-se  $O_\alpha$  e  $O_\beta$  como operações sucessivas (nesta ordem), com duração  $d$ , data de início mais cedo  $dic$  e data-limite para término  $dti$ . A diferença ( $dti - dic$ ) deve ser maior ou igual à duração  $d$  de cada operação.

As expressões têm o sinal de desigualdade para resguardar os casos em que as datas que estão sendo alteradas já estejam restritas por cálculos anteriores. Neste caso, seria um erro atribuir o valor de cálculo da expressão e criar novos graus de liberdade antes inexistentes.

#### 4.2.4 Módulo de Programação por Análise de Restrições de Tempo

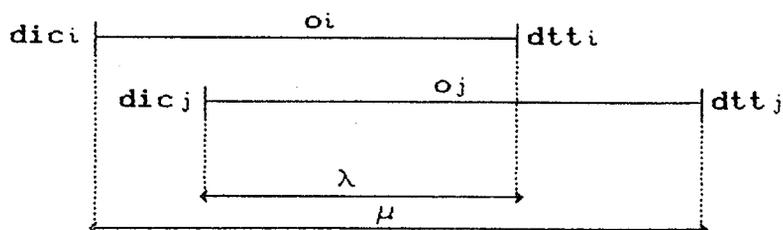
A técnica de propagação de restrições permite manter um programa de produção o mais flexível possível, dentro das restrições existentes, pois a liberdade de escolha de relações de precedência e datas-limite para início e fim de operações só é diminuída quando há restrições que impõem esta decisão.

As decisões de programação tomadas neste módulo envolvem apenas o seqüenciamento de pares de operações executadas em uma mesma máquina, a partir de restrições de tempo preexistentes. O processo de cálculo do quadro 7 é enunciado em [BEN88]:

#### QUADRO 7

##### DEFINIÇÕES:

- $O_i, O_j$  : Duas operações ainda não ordenadas, a serem executadas na máquina  $m$ .
- $d_i, d_j$  : Durações (tempos de processamento) das operações  $i$  e  $j$ , respectivamente.
- $dici, dti_i,$   
 $dic_j, dtj_j$  : Datas de início mais cedo e término mais tarde para as operações  $O_i$  e  $O_j$ .



Com base nos valores:

$$\lambda = dti_i - dic_j$$

$$\mu = dtj_j - dici$$

$$\sigma = d_i + d_j$$

conclui-se:

(a) Se  $\sigma > \lambda$ :

Se  $\sigma \leq \mu$ , então  $O_i$  deve preceder  $O_j$ .

Se  $\sigma > \mu$ , então não há ordem viável para  $O_i$  e  $O_j$ .

O programa deve ser reformulado.

- (b) Se  $\sigma \leq \lambda$ :  
Se  $\sigma > \mu$ , então  $O_j$  deve preceder  $O_i$ .  
Se  $\sigma \leq \mu$ , então nenhuma ordem pode ser atribuída.  
O par continua sem ordenação definida.

---

**QUADRO 7**

Se as janelas de tempo são tais que só permitem uma ordenação viável para o par, esta ordenação será feita. Caso contrário, nenhuma decisão poderá ser tomada, ou as janelas de tempo não permitem nenhum escalonamento viável.

Este procedimento é aplicado a todos os pares de operações não ordenados, máquina por máquina, exaustivamente. Examinados todos os pares e estabelecidas novas relações de precedência, as datas são atualizadas pelo módulo de ajuste de datas e podem forçar novas decisões de precedência de operações. Este procedimento é repetido até que o módulo de análise de restrições não consiga mais gerar nenhuma nova relação de precedência.

As restrições tecnológicas e de tempo iniciais, dadas pela estrutura de produção e pelas seqüências de operações dos produtos, geralmente não são suficientes para gerar um programa de produção completo, deixando indefinidas algumas decisões de precedência de operações numa mesma máquina. Esgotada a capacidade de programação por análise de restrições, se ainda restarem decisões de programação a serem feitas, outra técnica deverá ser utilizada, gerando novas restrições.

Cada par de operações ordenadas tem suas datas ajustadas automaticamente, usando a mesma expressão definida em 4.2.3.

Quando a análise das datas-limite das operações concluir pela inviabilidade de qualquer das ordens de precedência, o programa deverá ser reformulado, cancelando as últimas ordenações e possivelmente relaxando algumas restrições, segundo regras predefinidas.

#### 4.2.5 Módulo Escalonador Difuso

Neste módulo de programação, procura-se utilizar o conhecimento empírico sobre regras de liberação e o conhecimento prático sobre as características do ambiente fabril, como restrições humanas e tecnológicas.

A existência de preferências e objetivos de programação conflitantes ou enunciados de forma imprecisa é uma característica

do ambiente fabril que os modelos existentes têm dificuldade em tratar. Estas preferências, se tratadas qualitativa ou ponderadamente, geram critérios objetivos de programação que podem ser transformados em novas restrições, diminuindo o espaço de estados para a busca da solução e orientando o processo de programação.

Em função da arquitetura do sistema, o módulo escalonador difuso é acionado quando a programação por análise de restrições esgotou as restrições existentes sem ter chegado a um programa completo, ou seja, restam decisões de precedência de operações a serem tomadas.

A tarefa do módulo é ordenar uma parcela do total não ordenado de pares de operações. O motivo de se ordenar alguns e não todos os pares tem relação com o progresso e a qualidade do programa de produção em curso. A ordenação de poucos pares geraria poucas restrições de precedência, resultando numa evolução muito lenta do programa. Ordenando-se muitos pares, geram-se muitas restrições, aumentando a velocidade da evolução do programa. Entretanto, sendo as ordenações baseadas em regras com caráter primordialmente local, as decisões tomadas podem comprometer a flexibilidade do programa, ou mesmo serem excessivas e levarem à inviabilidade.

A escolha do subconjunto de pares a serem ordenados pelo módulo escalonador difuso pode ser definida por regras, como: programar primeiro operações nas máquinas mais carregadas ou com menores índices de folga, escolher para ordenação os pares que envolvem operações de uma tarefa de interesse específico, etc. A seleção do conjunto de pares de operações a serem analisados é anterior ao processo de cálculo propriamente dito.

A seguir, detalha-se a rotina de cálculo a ser seguida para cada par de operações no conjunto selecionado, de acordo com [BEN88]:

Parte A: Aplicação das regras ao par de operações (i, j):

A1) Selecionar, para cada operação, um argumento relativo à regra R em questão. Por exemplo, para uma regra referente a folgas, os argumentos  $arg_i$  e  $arg_j$  serão as folgas das operações.

A2) Calcular a taxa  $T_{ij}$ , que relaciona os argumentos:

$$T_{ij} = arg_i / (arg_i + arg_j)$$

A3) Calcular os graus de preferência para a ordem  $i \rightarrow j$  ( $i$  precede  $j$ ), com base nos conjuntos difusos FAVORÁVEL, INDIFERENTE e DESFAVORÁVEL da figura 13:

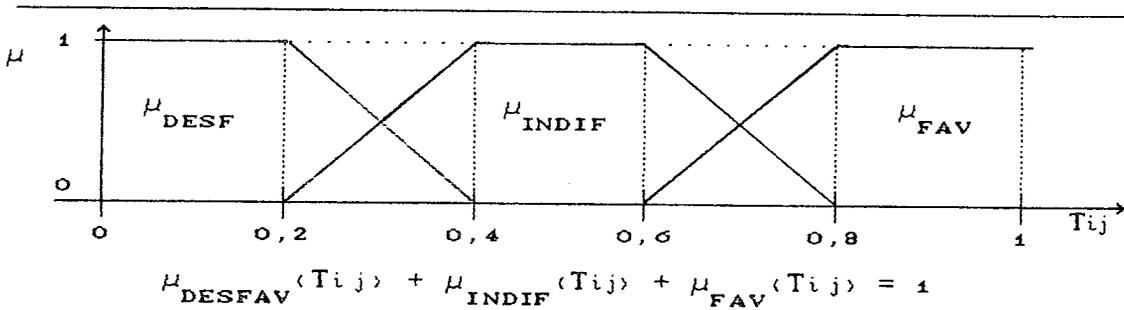


FIGURA 13

A4) Atualizar os graus de preferência com base no grau de ponderação  $P(R)$  conferido à regra pelo usuário, limitando o alcance da regra:

$$\mu_R (i \rightarrow j) = \min ( \mu_{FAV} (T_{ij}) ; P(R) ) : \text{ordem } i \rightarrow j$$

$$\mu_R (i \sim j) = \min ( \mu_{INDIF} (T_{ij}) ; P(R) ) : \text{ordem indiferente}$$

$$\mu_R (i \leftarrow j) = \min ( \mu_{DESF} (T_{ij}) ; P(R) ) : \text{ordem } i \leftarrow j$$

Parte B: Agregação dos cálculos relativos a todas as regras:

B1) Cálculo dos graus de preferência agregados para o par de operações:

$$GPA (i \rightarrow j) = \sum_R \mu_R (i \rightarrow j)$$

$$GPA (i \sim j) = \sum_R \mu_R (i \sim j)$$

$$GPA (i \leftarrow j) = \sum_R \mu_R (i \leftarrow j)$$

B2) Cálculo do índice de preferência  $I(i, j)$  para a decisão de precedência  $o(i, j)$ :

$$i (i \rightarrow j) = \min ( GPA (i \rightarrow j) ; 1 - GPA (i \sim j) )$$

$$i (i \leftarrow j) = \min ( GPA (i \leftarrow j) ; 1 - GPA (i \sim j) )$$

$$I (i, j) = \max ( i (i \rightarrow j) ; i (i \leftarrow j) )$$

$$o (i, j) = \begin{cases} i \rightarrow j, & \text{se } i (i \rightarrow j) > i (i \leftarrow j) \\ i \leftarrow j, & \text{caso contrário} \end{cases}$$

Segundo os pares de operações selecionados e a quantidade de pares que se pretende ordenar, as decisões são tomadas a partir do maior índice de preferência, seguindo uma escala descendente, até alcançar o número de decisões desejado.

Segundo a simulação feita por Bensana et alii [BEN88], usando o sistema OPAL, a taxa de 20% de ordenamentos sobre o total de pares de operações não ordenadas foi encontrada como a mais satisfatória. Este é um valor particular para aquele sistema. Entretanto, será considerado nas futuras simulações utilizando a arquitetura apresentada.

Cada par de operações ordenadas tem suas datas ajustadas automaticamente, usando a mesma expressão definida em 4.2.3.

As novas relações de precedência criadas pelo módulo escalonador difuso propiciam o reinício do trabalho do módulo de análise de restrições.

A abordagem possibilística presente no processo permite o tratamento de um conjunto de critérios e regras cuja definição é muitas vezes conflitante, incompleta ou imprecisa.

#### 4.2.6 Módulo de Controle

O módulo de controle dirige o processo, controlando a entrada de dados e as transições cíclicas entre os módulos: ajuste de tempos → propagação de restrições → ajuste de tempos → decisão baseada em regras; até que uma programação viável seja encontrada.

O módulo de controle funciona como núcleo do motor de inferência do sistema, passando a programação ora para o módulo de análise de restrições, ora para o módulo escalonador difuso.

A rotina de trabalho do módulo, baseada no procedimento descrito em [BEN88], pode ser definida como segue:

a) O primeiro passo é o ajuste de datas das operações, baseado nos roteiros de fabricação. Para cada tarefa, partindo da operação inicial, as datas-limite para início são atualizadas conforme definido em 4.2.3:

$$dic_{\beta} \geq dic_{\alpha} + d_{\alpha}$$

onde o índice  $\alpha$  refere-se à operação precedente e  $\beta$  à operação sucessora.

As datas-limite para término das operações são atualizadas a partir da última operação da tarefa, voltando até a operação inicial:

$$dtt_{\alpha} \leq dtt_{\beta} - d_{\beta}$$

onde  $\beta$  refere-se à operação sucessora e  $\alpha$  é o índice para a data de término mais tarde da operação antecedente.

Segue-se um processo iterativo:

b) Se não houver mais pares de operações não ordenados, o processo está terminado, passando para o passo f). Caso haja, o módulo de análise de restrições é acionado, verificando todos os pares de operações não ordenados, segundo a rotina descrita em 4.2.4. Se for encontrada alguma nova relação de precedência, o processo passa para o item c), caso contrário, vai para o item d) (módulo escalonador difuso):

Para as operações  $O_i$  e  $O_j$ :      Se  $\sigma > \lambda$ :  
 $\lambda = dtt_i - dic_j$       se  $\sigma \leq \mu$ ,  $O_i$  deve preceder  $O_j$ .  
 $\mu = dtt_j - dic_i$       se  $\sigma < \mu$ , não há ordem viável  
 $\sigma = d_i + d_j$       (o programa deve ser reformulado)  
                                  Se  $\sigma \leq \lambda$ :  
                                  se  $\sigma > \mu$ ,  $O_j$  deve preceder  $O_i$ .  
                                  se  $\sigma \leq \mu$ , não há ordem definida  
                                  (o par continua sem ordenação)

c) As datas são atualizadas segundo as seqüências de processamento das tarefas e também segundo as relações de precedência entre operações na mesma máquina, conforme 4.2.3:

$$dic_{\beta} \geq dic_{\alpha} + d_{\alpha} \qquad dtt_{\alpha} \leq dtt_{\beta} - d_{\beta}$$

A programação volta ao módulo de análise de restrições (item b)).

d) O módulo escalonador difuso estabelece um conjunto de novas relações de precedência, conforme 4.2.5. O processo segue em e).

e) As datas são novamente atualizadas segundo as ordens de precedência na tarefa e na máquina:

$$dic_{\beta} \geq dic_{\alpha} + d_{\alpha} \qquad dtt_{\alpha} \leq dtt_{\beta} - d_{\beta}$$

A programação passa para o módulo de análise de restrições, em b).

f) Não há mais pares de operações a ordenar. O programa de produção resultante é expresso em termos de relações de precedência de operações nas máquinas e suas datas-limite para início e término, permitindo pequenas variações.

A situação em que o programa se torna inviável está prevista no módulo de análise de restrições, quando o programa deverá ser reformulado, possivelmente relaxando algumas restrições.

A solução por janelas de tempo confere robustez ao programa produzido, evitando a reprogramação frente a pequenos desvios.

As duas técnicas de programação são exploradas de forma complementar, utilizando as vantagens e desvantagens das duas técnicas. A propagação de restrições é uma técnica limitada pela carência de restrições, mas mantém o programa de produção em curso o mais flexível possível, dentro das restrições existentes. As regras de programação dificilmente conseguem refletir os objetivos globais de programação, mas podem designar uma ordem entre operações segundo critérios predefinidos, gerando novas restrições de programação.

V - APLICAÇÃO

De acordo com a estratégia do módulo de controle descrita no capítulo 4, o exemplo a seguir refere-se à programação de três tarefas (P1, P2, P3) em um ambiente com três máquinas (M1, M2, M3), em 24 unidades de tempo.

No módulo escalonador difuso foram usadas 2 regras: Importância da tarefa (atribuída pelo usuário) e Folga dinâmica, com graus de ponderação P(R) iguais a 0,3 e 0,8, respectivamente.

Foram usadas rotinas em Turbo C++ para a análise de restrições e para a atribuição do índice de preferência da ordenação no módulo escalonador difuso.

A figura 14 mostra os roteiros de fabricação para o exemplo:

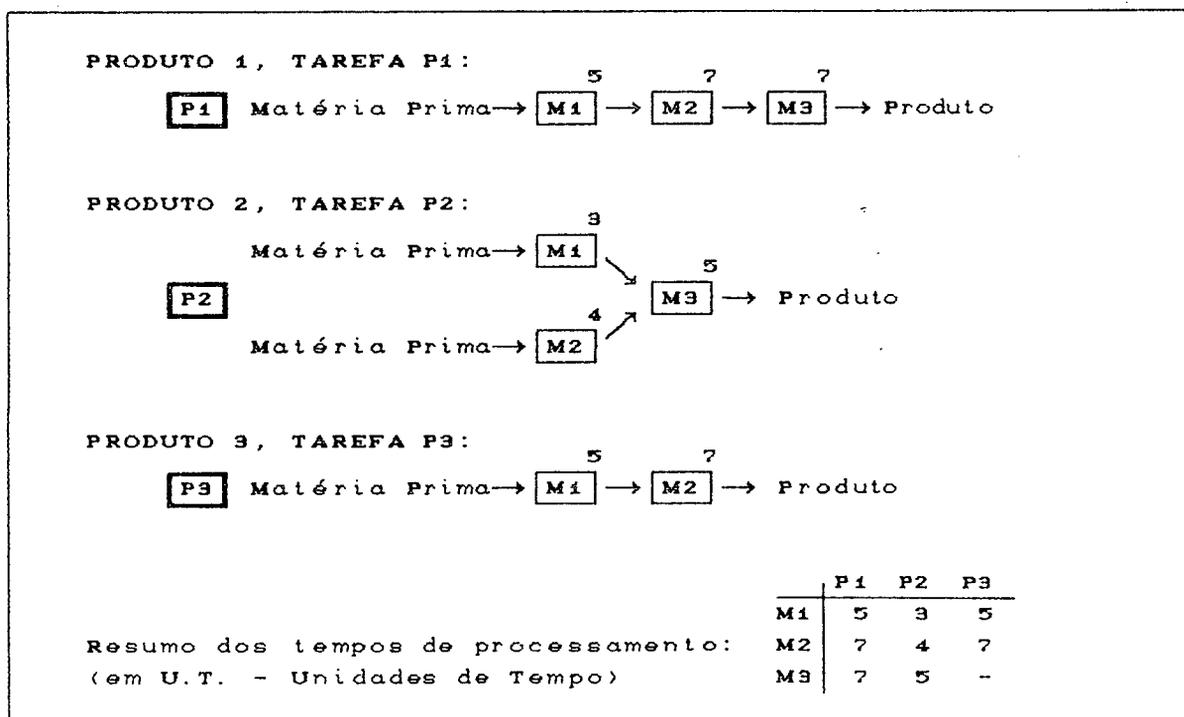


FIGURA 14 - EXEMPLO - ROTEIROS DE FABRICAÇÃO

A tarefa P2 tem um argumento de cálculo quanto à importância da tarefa definido como 0,2 (prioritária), enquanto as demais têm o valor default 0,5.

Nos diagramas a seguir, as janelas de tempo são expressas por barras horizontais, com um padrão para cada tarefa e a duração da

operação em destaque sobre a barra, a partir do início. As partes vazadas nas barras representam restrições de tempo criadas na etapa imediatamente anterior.

Inicialmente, antes de fazer o ajuste de datas segundo os roteiros de fabricação, as janelas de tempo têm como única restrição as datas de início e fim do período de programação (0 e 24 unidades de tempo). As janelas de tempo iniciais são as da figura 15:

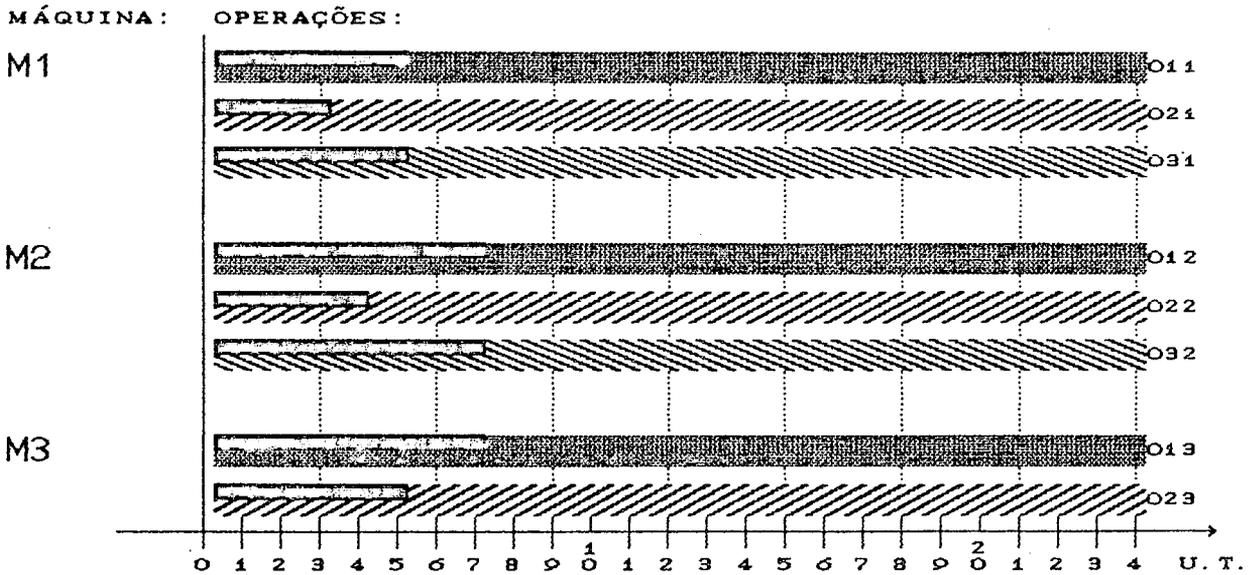


FIGURA 15

O módulo de ajuste de datas faz as alterações resumidas no quadro 8, segundo o processo descrito em 4.2.3, levando às janelas de tempo da figura 16:

QUADRO 8: RESUMO DAS DATAS-LIMITE ALTERADAS AJUSTE DE DATAS SEGUNDO OS ROTEIROS DE FABRICAÇÃO

| dic               | $O_{\alpha}$    | dic <sub><math>\alpha</math></sub> | d <sub><math>\alpha</math></sub> | dic <sub><math>\alpha</math></sub> + d <sub><math>\alpha</math></sub> | Valor antigo | Novo valor |
|-------------------|-----------------|------------------------------------|----------------------------------|---|--------------|------------|
| dic <sub>12</sub> | O <sub>11</sub> | 0                                  | 5                                | 5   | 0            | 5          |
| dic <sub>13</sub> | O <sub>12</sub> | 5                                  | 7                                | 12  | 0            | 12         |
| dic <sub>23</sub> | O <sub>21</sub> | 0                                  | 3                                | 3   | 0            | 3          |
| dic <sub>23</sub> | O <sub>22</sub> | 0                                  | 4                                | 4   | 3            | 4          |
| dic <sub>32</sub> | O <sub>31</sub> | 0                                  | 5                                | 5   | 0            | 5          |
| dtc               | $O_{\beta}$     | dtc <sub><math>\beta</math></sub>  | d <sub><math>\beta</math></sub>  | dtc <sub><math>\beta</math></sub> - d <sub><math>\beta</math></sub>   | Valor antigo | Novo valor |
| dtc <sub>12</sub> | O <sub>13</sub> | 24                                 | 7                                | 17  | 24           | 17         |
| dtc <sub>11</sub> | O <sub>12</sub> | 17                                 | 7                                | 10  | 24           | 10         |
| dtc <sub>22</sub> | O <sub>23</sub> | 24                                 | 5                                | 19  | 24           | 19         |
| dtc <sub>21</sub> | O <sub>23</sub> | 24                                 | 4                                | 19  | 24           | 19         |
| dtc <sub>31</sub> | O <sub>32</sub> | 24                                 | 7                                | 17  | 24           | 17         |

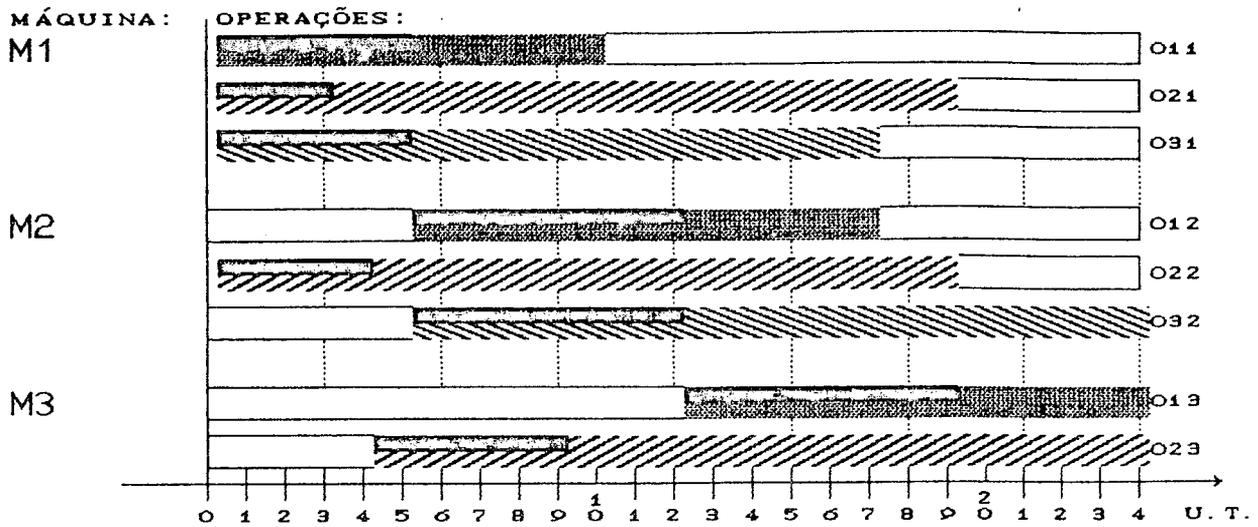


FIGURA 16

Inicialmente, há sete pares de operações não ordenados, sendo 1 na máquina M3, 3 na máquina M1 e 3 na M2. A análise de restrições ordena apenas o par (O12→O32), segundo a rotina descrita em 4.2.4. O cálculo é resumido a seguir.

| O <sub>i</sub> | O <sub>j</sub> | $\lambda$ | $\mu$ | $\sigma$ | Conclusão | Datas: dic <sub>j</sub> | dt <sub>ti</sub> |
|----------------|----------------|-----------|-------|----------|-----------|-------------------------|------------------|
| O12            | O32            | 12        | 19    | 14       | O12→O32   | 12                      | =                |

O algoritmo do módulo de controle executa novo ajuste de datas. Após a primeira passagem pelo módulo de análise de restrições, resultam as janelas de tempo da figura 17:

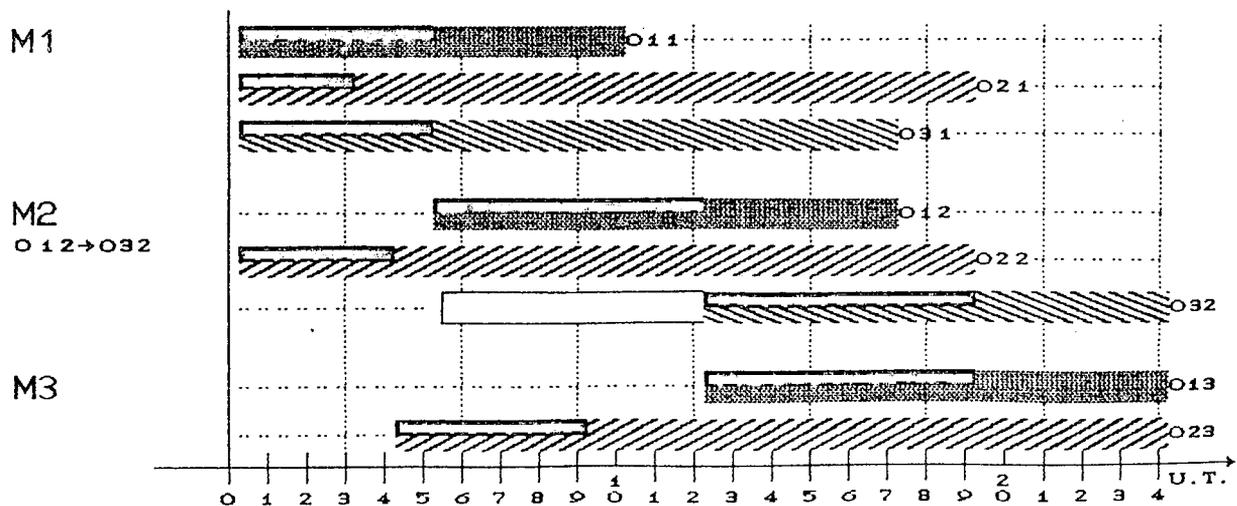


FIGURA 17

Após nova passagem pelo módulo de ajuste de datas, a

programação volta à análise de restrições e uma nova restrição de precedência é estabelecida: (o22→o32). As novas janelas de tempo estão descritas na figura 18:

| O <sub>i</sub> | O <sub>j</sub> | λ | μ  | σ  | Conclusão | Datas: dic <sub>j</sub> | dti <sub>i</sub> |
|----------------|----------------|---|----|----|-----------|-------------------------|------------------|
| O22            | O32            | 7 | 24 | 11 | O22→O32   | =                       | 17               |

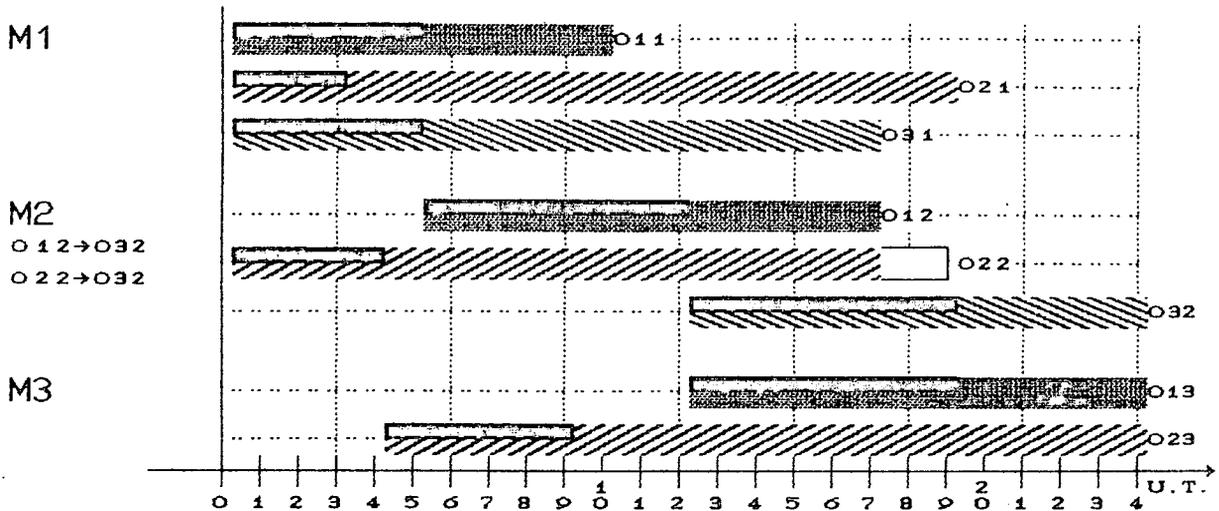


FIGURA 18

Seguindo a rotina do módulo de controle, o ajuste de datas é feito e a inferência é delegada ao módulo de análise de restrições. Dentre os 5 pares de operações não ordenados, nenhuma nova restrição de precedência é descoberta, esgotando os recursos de programação do módulo de análise de restrições.

O módulo escalonador difuso aplica as regras aos 5 pares não ordenados restantes e estabelece duas ordenações: (o22→o12) e (o23→o13).

Neste módulo, a primeira atividade é a seleção de um subgrupo de pares não ordenados de operações para submeter às regras. A seleção é feita tomando as operações não ordenadas nas máquinas mais carregadas, o que será medido segundo o "índice de flexibilidade absoluta" definido em [BEN88] (enunciado em 4.2.2):

$$I.F._m = \frac{\sum_i (dtt_{im} - dic_{im} - d_{im})}{\sum_i d_{im}}$$

Para cada máquina, o índice dá um valor relativo da quantidade de folgas em relação ao tempo de processamento requerido na máquina.

Há três conflitos na máquina M1, um na máquina M2 e um na máquina M3. Os índices são:

$$\begin{aligned} \text{I.F.}_{M1} &= (9+16+12)/(5+3+5) = 2,54 \\ \text{I.F.}_{M2} &= (5+13+5)/(7+4+7) = 1,28 \\ \text{I.F.}_{M3} &= (5+15)/(5+7) = 1,67 \end{aligned}$$

Pelos índices acima, a máquina  $M_2$  tem menos folgas. O par de operações  $(o_{12}, o_{22})$ , único não ordenado nesta máquina, é selecionado para ser submetido às regras. Sendo 5 pares ao todo, o subconjunto precisa de mais pares para atender ao objetivo de escalonar  $1,5 + 20\%$  do total de pares não ordenados. A próxima máquina a selecionar é  $M_3$ , que tem um par não ordenado:  $(o_{13}, o_{23})$ .

A seguir, resumem-se os cálculos para os dois pares:

#### Parte A: Aplicação das regras:

Par  $(o_{12}, o_{22})$ :

$$\begin{aligned} \text{Regra "Folga dinâmica"} \\ P(R) &= 0,8 \\ T_{ij} &= 5/(5+1) = 0,83 \\ \mu_{\text{DESF}}(T_{ij}) &= 0 \\ \mu_{\text{INDIF}}(T_{ij}) &= 0 \\ \mu_{\text{FAV}}(T_{ij}) &= 1 \\ \mu_R(i \rightarrow j) &= 0 \\ \mu_R(i \sim j) &= 0 \\ \mu_R(i \leftarrow j) &= 0,8 \end{aligned}$$

$$\begin{aligned} \text{Regra "Importância da Tarefa"} \\ P(R) &= 0,3 \\ T_{ij} &= 0,5/(0,5+0,2) = 0,71 \\ \mu_{\text{DESF}}(T_{ij}) &= 0 \\ \mu_{\text{INDIF}}(T_{ij}) &= 0,57 \\ \mu_{\text{FAV}}(T_{ij}) &= 0,43 \\ \mu_R(i \rightarrow j) &= 0 \\ \mu_R(i \sim j) &= 0,3 \\ \mu_R(i \leftarrow j) &= 0,3 \end{aligned}$$

Par  $(o_{13}, o_{23})$ :

$$\begin{aligned} \text{Regra "Folga dinâmica"} \\ P(R) &= 0,8 \\ T_{ij} &= 5/(5+0) = 1 \\ \mu_{\text{DESF}}(T_{ij}) &= 0 \\ \mu_{\text{INDIF}}(T_{ij}) &= 0 \\ \mu_{\text{FAV}}(T_{ij}) &= 1 \\ \mu_R(i \rightarrow j) &= 0 \\ \mu_R(i \sim j) &= 0 \\ \mu_R(i \leftarrow j) &= 0,8 \end{aligned}$$

$$\begin{aligned} \text{Regra "Importância da Tarefa"} \\ P(R) &= 0,3 \\ T_{ij} &= 0,5/(0,5+0,2) = 0,71 \\ \mu_{\text{DESF}}(T_{ij}) &= 0 \\ \mu_{\text{INDIF}}(T_{ij}) &= 0,57 \\ \mu_{\text{FAV}}(T_{ij}) &= 0,43 \\ \mu_R(i \rightarrow j) &= 0 \\ \mu_R(i \sim j) &= 0,3 \\ \mu_R(i \leftarrow j) &= 0,3 \end{aligned}$$

#### Parte B: Agregação dos cálculos:

Par  $(o_{12}, o_{22})$ :

$$\begin{aligned} \text{GPA}(i \rightarrow j) &= 0 + 0 = 0 \\ \text{GPA}(i \sim j) &= 0 + 0,3 = 0,3 \\ \text{GPA}(i \leftarrow j) &= 0,8 + 0,3 = 1,1 \\ i(i \rightarrow j) &= 0 \\ i(i \leftarrow j) &= 0,7 \\ I(i, j) &= 0,7 \\ o(i, j) &= j \leftarrow i \end{aligned}$$

Par  $(o_{13}, o_{23})$ :

$$\begin{aligned} \text{GPA}(i \rightarrow j) &= 0 + 0 = 0 \\ \text{GPA}(i \sim j) &= 0 + 0,3 = 0,3 \\ \text{GPA}(i \leftarrow j) &= 0,8 + 0,3 = 1,1 \\ i(i \rightarrow j) &= 0 \\ i(i \leftarrow j) &= 0,7 \\ I(i, j) &= 0,7 \\ o(i, j) &= j \leftarrow i \end{aligned}$$

Segundo os cálculos, as ordenações (o22→o12) e (o23→o13) têm índices de preferência iguais a 0,7. A regra que estabelece a quantidade de pares a ordenar determina, neste caso, a quantidade:

$$N_{ord} < 1,5 + 0,2 * 5 = 2,5$$

Portanto, os dois pares ficam ordenados pelo módulo escalonador difuso.

Ajustados os tempos, a situação resultante é a representada no quadro 9 e figura 19:

| O <sub>i</sub> | O <sub>j</sub> | Conclusão | Datas: d <sub>i</sub> c <sub>j</sub> | dt <sub>i</sub> |
|----------------|----------------|-----------|--------------------------------------|-----------------|
| O22            | O12            | O22→O12   | =                                    | 10              |
| O23            | O13            | O23→O13   | =                                    | 17              |

QUADRO 9: RESUMO DAS DATAS-LIMITE ALTERADAS

| dt <sub>t</sub>   | O <sub>β</sub> | dt <sub>t β</sub> | d <sub>β</sub> | dt <sub>t β</sub> - d <sub>β</sub> | Valor antigo | Novo valor |
|-------------------|----------------|-------------------|----------------|------------------------------------|--------------|------------|
| dt <sub>t21</sub> | O23            | 17                | 5              | 12                                 | 10           | 12         |

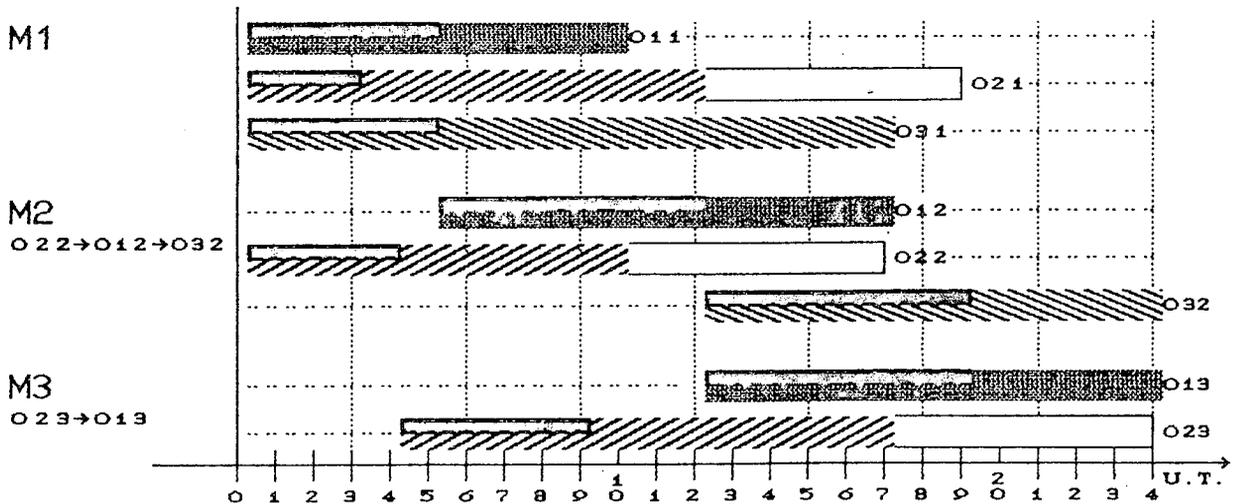


FIGURA 19

Restam 3 pares de operações a ordenar, todos na máquina M1. A programação é feita novamente pelo módulo de análise de restrições, manipulando as novas restrições criadas. A análise dos 3 pares não ordenados não revela nenhuma nova restrição de precedência. O módulo de controle, então, remete o processo de ordenação para o módulo escalonador difuso que, entre os 3 pares de operações não ordenados, faz duas novas ordenações: (o11→o21) e (o11→o31). O processo é descrito a seguir:

Parte A: Aplicação das regras:

Par (011, 021):

Regra "Folga dinâmica"

$$P(R) = 0,8$$

$$T_{ij} = 2/(2+5) = 0,29$$

$$\mu_{\text{DESFAV}}(T_{ij}) = 0,57$$

$$\mu_{\text{INDIF}}(T_{ij}) = 0,43$$

$$\mu_{\text{FAV}}(T_{ij}) = 0$$

$$\mu_R(i \rightarrow j) = 0,57$$

$$\mu_R(i \sim j) = 0,43$$

$$\mu_R(i \leftarrow j) = 0$$

Regra "Importância da Tarefa"

$$P(R) = 0,3$$

$$T_{ij} = 0,5/(0,5+0,2) = 0,71$$

$$\mu_{\text{DESFAV}}(T_{ij}) = 0$$

$$\mu_{\text{INDIF}}(T_{ij}) = 0,43$$

$$\mu_{\text{FAV}}(T_{ij}) = 0,57$$

$$\mu_R(i \rightarrow j) = 0$$

$$\mu_R(i \sim j) = 0,3$$

$$\mu_R(i \leftarrow j) = 0,3$$

Par (011, 031):

Regra "Folga dinâmica"

$$P(R) = 0,8$$

$$T_{ij} = 0/(0+5) = 0$$

$$\mu_{\text{DESFAV}}(T_{ij}) = 1$$

$$\mu_{\text{INDIF}}(T_{ij}) = 0$$

$$\mu_{\text{FAV}}(T_{ij}) = 0$$

$$\mu_R(i \rightarrow j) = 0,8$$

$$\mu_R(i \sim j) = 0$$

$$\mu_R(i \leftarrow j) = 0$$

Regra "Importância da Tarefa"

$$P(R) = 0,3$$

$$T_{ij} = 0,5(0,5+0,5) = 0,5$$

$$\mu_{\text{DESFAV}}(T_{ij}) = 0$$

$$\mu_{\text{INDIF}}(T_{ij}) = 1$$

$$\mu_{\text{FAV}}(T_{ij}) = 0$$

$$\mu_R(i \rightarrow j) = 0$$

$$\mu_R(i \sim j) = 0,3$$

$$\mu_R(i \leftarrow j) = 0$$

Par (021, 031):

Regra "Folga dinâmica"

$$P(R) = 0,8$$

$$T_{ij} = 11/(11+9) = 0,55$$

$$\mu_{\text{DESFAV}}(T_{ij}) = 0$$

$$\mu_{\text{INDIF}}(T_{ij}) = 1$$

$$\mu_{\text{FAV}}(T_{ij}) = 0$$

$$\mu_R(i \rightarrow j) = 0$$

$$\mu_R(i \sim j) = 0,8$$

$$\mu_R(i \leftarrow j) = 0$$

Regra "Importância da Tarefa"

$$P(R) = 0,3$$

$$T_{ij} = 0,2(0,2+0,5) = 0,23$$

$$\mu_{\text{DESFAV}}(T_{ij}) = 0,57$$

$$\mu_{\text{INDIF}}(T_{ij}) = 0,43$$

$$\mu_{\text{FAV}}(T_{ij}) = 0$$

$$\mu_R(i \rightarrow j) = 0,3$$

$$\mu_R(i \sim j) = 0,3$$

$$\mu_R(i \leftarrow j) = 0$$

Parte B: Agregação dos cálculos:

Par (011, 021):

$$GPA(i \rightarrow j) = 0,57 + 0 = 0,57$$

$$GPA(i \sim j) = 0,43 + 0,3 = 0,73$$

$$GPA(i \leftarrow j) = 0 + 0,3 = 0,3$$

$$i(i \rightarrow j) = 0,27$$

$$i(i \leftarrow j) = 0,27$$

$$I(i, j) = 0,27$$

$$O(i, j) = i \rightarrow j$$

Par (011, 031):

$$GPA(i \rightarrow j) = 0,8 + 0 = 0,8$$

$$GPA(i \sim j) = 0 + 0,3 = 0,3$$

$$GPA(i \leftarrow j) = 0 + 0 = 0$$

$$i(i \rightarrow j) = 0,7$$

$$i(i \leftarrow j) = 0$$

$$I(i, j) = 0,7$$

$$O(i, j) = i \rightarrow j$$

Par (O21, O31):

- $GPA(i \rightarrow j) = 0 + 0,3 = 0,3$
- $GPA(i \sim j) = 0,8 + 0,3 = 1,1$
- $GPA(i \leftarrow j) = 0 + 0 = 0$
- $i(i \rightarrow j) = -0,1$
- $i(i \leftarrow j) = -0,1$
- $I(i, j) = -0,1$
- $o(i, j) = i \rightarrow j$

Segundo os cálculos, as ordenações (O11→O31), (O11→O21) e (O21→O31) têm os índices de preferência 0,70; 0,27 e -0,10, respectivamente. A regra que estabelece a quantidade de pares a ordenar determina, neste caso, a quantidade:

$$N_{ord} < 1,5 + 0,2 * 3 = 2,1$$

Portanto, ficam ordenados os pares (O11→O31) e (O11→O21) pelo módulo escalonador difuso.

O ajuste de tempos e as janelas estão no quadro 10 e figura 20:

| O <sub>i</sub> | O <sub>j</sub> | Conclusão | Datas: dic <sub>j</sub> | dt <sub>i</sub> |
|----------------|----------------|-----------|-------------------------|-----------------|
| O11            | O21            | O11→O21   | 5                       | 9               |
| O11            | O31            | O11→O31   | 5                       | =               |

QUADRO 10: RESUMO DAS DATAS-LIMITE ALTERADAS

| dic   | O <sub>α</sub> | dic <sub>α</sub> | d <sub>α</sub> | dic <sub>α</sub> + d <sub>α</sub> | Valor antigo | Novo valor |
|-------|----------------|------------------|----------------|-----------------------------------|--------------|------------|
| dic23 | O21            | 5                | 3              | 8                                 | 4            | 8          |
| dic13 | O23            | 8                | 5              | 13                                | 12           | 13         |

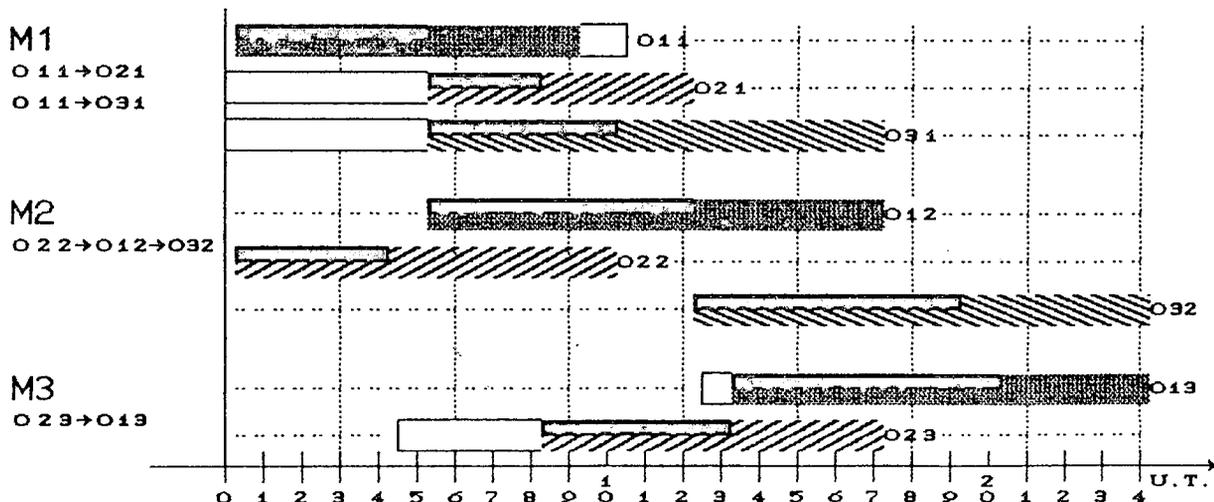


FIGURA 20

Restando um par a ser ordenado, a programação volta ao módulo de análise de restrições, que determina a ordenação: (O21→O31), resultando na situação descrita pelo quadro 11 e figura 21:

| O <sub>i</sub> | O <sub>j</sub> | $\lambda$ | $\mu$ | $\sigma$ | Conclusão | Datas: dic <sub>j</sub> | dt <sub>ti</sub> |
|----------------|----------------|-----------|-------|----------|-----------|-------------------------|------------------|
| O21            | O31            | 7         | 12    | 8        | O21→O31   | 8                       | =                |

QUADRO 11: RESUMO DAS DATAS-LIMITE ALTERADAS

| dic   | O <sub><math>\alpha</math></sub> | dic <sub><math>\alpha</math></sub> | d <sub><math>\alpha</math></sub> | dic <sub><math>\alpha</math></sub> + d <sub><math>\alpha</math></sub> | Valor antigo | Novo Valor |
|-------|----------------------------------|------------------------------------|----------------------------------|---|--------------|------------|
| dic32 | O31                              | 8                                  | 5                                | 13  | 12           | 13         |

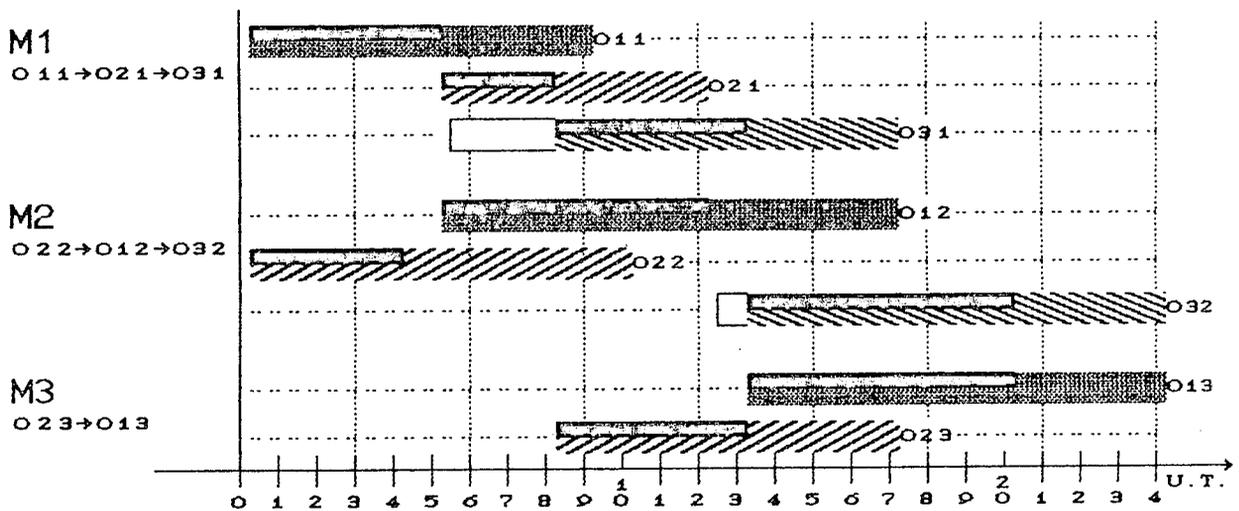


FIGURA 21

O diagrama acima fornece uma programação em termos de datas-limite para início e fim de operações. O programa não precisará ser alterado devido a desvios e variações no programa original até os limites das folgas de tempo das operações.

## VI - CONCLUSÕES

O objetivo desta dissertação de mestrado e do trabalho de pesquisa correspondente é apresentar o projeto de um sistema especialista para programação da produção.

Como parte dos objetivos complementares, grande parte do tempo de pesquisa foi empregado no estudo de tópicos de administração da produção: filosofias de fabricação, ambientes de fabricação, métodos e técnicas de administração da produção e estoques.

O sistema apresentado no capítulo 4 faz a programação a nível de operações elementares, tendo como estratégia principal de inferência a análise e propagação de restrições de tempo. A segunda estratégia de inferência, programação utilizando regras, é aplicada no sentido de resolver a principal deficiência da propagação de restrições, que é a insuficiência de restrições, sem comprometer, no entanto, a principal virtude desta técnica, a de preservar os graus de liberdade do programa em curso até o limite das restrições estabelecidas.

A implementação computacional do sistema conta, até o momento, apenas com as duas rotinas centrais de programação: análise de restrições e aplicação do conjunto de regras. A aplicação exibida no capítulo 5 mostra o comportamento do sistema. Uma avaliação do sistema em termos de rapidez e eficiência requer, ainda, a construção de outros módulos e a definição e implementação de esquemas de representação de conhecimento.

O principal objetivo foi alcançado com o detalhamento de um sistema especialista que agrega técnicas de propagação de restrições, que utiliza conhecimento puramente determinístico, com as decisões de programação através de regras tratadas por matemática difusa, que permitem modelar o conhecimento empírico (heurístico) e prático sobre o ambiente de fabricação.

O módulo de programação baseada em regras permite simulações com diferentes tipos de regras e a verificação do desempenho destas segundo diferentes objetivos de programação. Isto propicia, por exemplo, a busca dos resultados do sistema OPT, cuja

arquitetura não é apresentada na literatura, através da ênfase na programação das máquinas mais carregadas, os "gargalos" de produção.

As primeiras aplicações de sistemas especialistas foram dirigidas a problemas que envolvem diagnósticos. A aplicação a problemas de administração de negócios é muito recente, principalmente na área de programação da produção. O poderio desta nova tecnologia frente aos problemas que pretende resolver é alardeado de forma muito efusiva, às vezes, cabendo aos pesquisadores a devida cautela na divulgação de seus trabalhos (ver, por exemplo, o artigo na Business Week [BUS84]).

De acordo com Kanet [KAN87], o sistema ISA, em uso na Digital Equipment Corporation, dá uma evidência de que sistemas especialistas podem ser desenvolvidos para fazer certas funções de programação de modo tão bom quanto pessoas, com velocidade consideravelmente maior e menos esforço humano. Entretanto, a complexidade do problema tem levado os programadores a produzirem programas sofríveis, com desempenho fraco quanto a atrasos, tempo de estocagem, volume de estoques, etc. Este é um ponto fraco quando se pensa em utilizar a experiência de programadores para construir um sistema baseado neste conhecimento:

*"Se uma abordagem de sistemas especialistas para programação da produção significa 'garimpar' o conhecimento de programadores humanos e desenvolver um sistema que recomenda os mesmos tipos de programa que eles recomendariam, então, os sistemas especialistas não produzirão programas tão pobres quanto aqueles produzidos por programadores humanos? Não se conclui que os sistemas especialistas nunca produziram programações melhores do que as feitas pelos programadores?" [KAN87].*

Com efeito, esta é uma das objeções colocadas à aplicação de sistemas especialistas na programação da produção, juntamente com o alto custo e o tempo necessário para o desenvolvimento.

Por outro lado, o sucesso alcançado pelo método kanban, ferramenta da filosofia de fabricação just-in-time, representa outro bordo de ataque aos sistemas especialistas e a todos os sistemas externos ao ambiente produtivo, em geral. Contudo, sabe-se que o kanban não é aplicável a todos os ambientes e envolve fatores comportamentais com muita intensidade.

Em um artigo clássico, McKay et alii [McK88] argumentam que a

formulação do problema usada pelos investigadores da IA "é essencialmente a mesma dos teóricos do job shop estático ou dinâmico". Segundo os autores, são fatores que dificultam a tarefa de programação a existência de informações incompletas, ambíguas ou errôneas, a pouca estabilidade do ambiente produtivo e a extrema variabilidade dos tempos de processamento de operações e de ajuste de máquinas.

Conforme McKay et alii, ainda, a intuição é usada pelos programadores para ganhar compreensão sobre aspectos da fábrica que não são atingidos por relatórios e listagens de computador.

Baseados no exame de mais de vinte sistemas especialistas para planejamento e programação da produção, Kusiak & Chen [KUS88] apresentam as seguintes conclusões:

1. Sistemas especialistas baseados em regras são os mais frequentemente aplicados ao planejamento e programação da produção, devido a:

- várias shells disponíveis para desenvolvimento.
- as regras IF-THEN são facilmente aceitáveis porque são similares à lógica do bom-senso.
- linguagens baseadas em lógica, como LISP e Prolog, são as mais conhecidas pelos pesquisadores de IA.

2. Técnicas de busca dirigida por restrições (espaço de estados) parecem mais adequadas para programação da produção do que para outras aplicações, como sistemas de diagnóstico. Isto se deve ao fato de que:

- sistemas de manufatura em grande escala têm uma estrutura hierárquica. As regras de produção são organizadas de modo semelhante.
- restrições importantes, relativas a disponibilidade de máquinas e outros recursos de manufatura, não podem ser violadas.

3. Heurísticas são amplamente usadas para atribuir prioridades para tarefas, operações e máquinas.

4. Simulação é amplamente usada no desenvolvimento de sistemas especialistas:

- para simular o sistema de manufatura e assim adquirir conhecimento.

- para simular o ambiente e testar o comportamento do sistema especialista.

- para controlar parcialmente o processo de raciocínio.

5. Sistemas especialistas e pesquisa operacional podem ser combinados com benefícios mútuos:

- métodos de simulação são muito usados, como mencionado acima.

- vários algoritmos para programação estão embutidos em muitos sistemas especialistas.

- algoritmos de programação matemática podem ser embutidos em sistemas especialistas para planejamento.

- podem ser usados métodos de decomposição.

- módulos PERT e relacionados podem ser ligados com um sistema especialista.

Considerando o caso dos sistemas ISIS e OPIS, cotados como as aplicações mais bem sucedidas de sistemas especialistas para programação da produção [McKB8] [KUMB6], desenvolvidos na Universidade Carnegie-Mellon (EUA), em colaboração com a Westinghouse Electric Corporation, conclui-se que a aplicação prática ainda não chegou ao chão-de-fábrica. O sistema ISIS ocupa em torno de 10 megabytes de disco em sua base de conhecimento. Um motivo parcial para a não utilização do sistema na planta da Westinghouse é dificuldade em compatibilizar as bases de dados do ISIS com as bases de dados e sistemas de informações preexistentes.

A implementação computacional das rotinas de programação do sistema apresentado foi feita em Turbo C++, devido às características de velocidade e possibilidade de linkagem desta linguagem e à facilidade para implementar esquemas de representação de conhecimento visando à programação orientada a objeto.

O detalhamento do módulo escalonador difuso foi feito segundo o apresentado por Bensana et alii [BEN88]. Porém, novas formas estão sendo pesquisadas para tornar o cálculo mais rápido e evitar o caráter parcialmente não compensatório das regras, ou seja, para

alguns intervalos de valores, as mudanças nos argumentos de cálculo fornecidos pelas operações não geram mudanças nos índices calculados pelas regras.

Para os exemplos testados, de pequena dimensão, a regra "Folga dinâmica" (programar as operações de forma a preservar as maiores folgas) demonstrou ser eficiente, favorecendo a manutenção da viabilidade do programa em curso.

Há atividades não previstas no detalhamento do sistema, como a atribuição de máquinas alternativas para as operações, a consideração dos tempos de ajuste de máquina e a programação de manutenção preventiva.

O sistema poderá servir-se de dados fornecidos por um sistema externo, como um plano mestre de produção gerado por um sistema MRP.

Uma conclusão óbvia, porém importante, é que o problema de programação da produção é extremamente complexo. As relações de hierarquia entre as entidades que compõem o ambiente produtivo se entrelaçam e tornam a representação deste conhecimento um fator crucial para qualquer sistema que gerencie dados sobre manufatura.

O ambiente fabril oferece um campo vasto de pesquisa na área de representação de conhecimento, especialmente utilizando as técnicas de orientação a objeto e modelos entidade-relacionamento.

O tamanho do problema sugere um melhor aproveitamento dos esforços de pesquisa feitos em equipe, preferencialmente com dotação adequada de recursos financeiros e de apoio (staff, equipamento e suprimentos). Neste sentido, o Programa de Pós-Graduação em Engenharia de Produção da UFSC tem encaminhado aos órgãos de fomento projetos de pesquisa envolvendo empresas industriais [PAC90].

VII - BIBLIOGRAFIA7.1 REFERÊNCIAS BIBLIOGRÁFICAS

- [ALL88] ALLORA, Franz  
*Controle de Produção Unificado e o Computador.* São Paulo, Pioneira, 1988.
- [ANT89] ANTUNES JR., J.A.V., KLIEMANN NETO, F.J. & FENSTENSEIFER, J.E  
Considerações Críticas sobre a Evolução das Filosofias de Administração da Produção: Do "Just-In-Case" ao "Just-In-Time". *Revista de Administração de Empresas*, São Paulo 29(3), 1989.
- [BAT90] BATALHA, Mário O. & DEMORI, Flávio  
*A Pequena e Média Indústria Catarinense.* Florianópolis, Ed. da UFSC, 1990.
- [BEL87] BELT, B.  
MRP and Kanban - A Possible Sinergy? *Prod. and Inv. Management*, First Quarter, 1987.
- [BEL88] BEL, G. et alii  
*A Multi-knowledge-based Approach to Industrial Job-shop Scheduling.* Rapport L.A.A.S. número 88195. Junho, 1988.
- [BEN88] BENSANA, E., BEL, G. & DUBOIS, D.  
OPAL: A Multi-knowledge-based System for Industrial Job-shop Scheduling. *International Journal of Production Research* 5, vol. 26, 1988.
- [BLA82] BLACKSTONE, John H. Jr., PHILLIPS, Don T. & HOGG, Gary L.  
A State-Of-The-Art Survey of Dispatching Rules for Manufacturing Job Shop Operations. *Int. J. Prod. Res.* vol. 20, no. 1, 1982.
- [BRU86] BRUNO, G. et alii  
A Rule-based Ssystem to Shedule Production. *IEEE Computer* 7, vol.19, 1986.
- [BUR81] BURBIDGE, John L.  
*Planejamento e Controle da Produção.* São Paulo, Ed. Atlas, primeira edição, 1981.
- [BUS84] BUSINESS WEEK  
Artificial Intelligence is here. *Business Week*. 9 de julho de 1984, pag. 54.
- [BUS88] BUSINESS WEEK  
The Productivity Paradox. *Business Week*. McGraw Hill, 6 de junho de 1988.

- [CHA90] CHANG, T. C. & IBBS, C. William  
PRIORITY RANKING - A Fuzzy Expert System for Priority Decision Making in Building Construction Resource Scheduling. *Building and Environment* vol. 25, número 3, 1990.
- [CON67] CONWAY, Richard W., MAXWELL, William L. & MILLER, Louis W.  
*Theory of Scheduling*. Addison-Wesley, 1967
- [DEA90] DEAL, Don E. et alii  
An Expert System Scheduler: Some Reflections on Expert Systems Development. *Computers Opns. Res.* vol. 17, no 6, 1990.
- [DOU90] DOUKIDIS, G. I. & PAUL, R. J.  
A Survey of the Application of A.I. Techniques within the OR Society. *J. Opl. Res. Soc.* vol. 41, número 5, 1990.
- [ESP88] ESPARRAGO Jr., R. A.  
*Kanban. Prod. and Inv. Management Journal*, First Quarter, 1988.
- [FEN89] FENSTERSEIFER, J. E. & BASTOS, R. M.  
A Implantação de Sistemas MRP de Gestão da Produção e de Materiais nas Grandes Empresas Industriais do Brasil. *Revista de Administração*, São Paulo 24(1), 1989.
- [FIS87] FISCHLER, M. A. & FIRSCHEIN, O.  
*Intelligence: The Eye, the Brain and the Computer*. Addison-Wesley, 1987.
- [FOX84] FOX, M. S. & SMITH, S. F.  
ISIS - A Knowledge-based System for Factory Scheduling. *Expert Systems* 1(1), 1984.
- [GER66] GERE, W. S., Jr.  
Heuristics in Job Shop Scheduling. *Management Science* 3, vol. 13, 1966.
- [GES85] GESSNER, R. A.  
*Master Production Schedule Planning*. John Wiley & Sons, 1985.
- [GOL86] GOLDRATT, E. & COX, J.  
*A Meta - Excelência na Manufatura*. IMAM, São Paulo, 1986.
- [GOL88] GOLDRATT, E. M.  
Computerized Shop Floor Scheduling. *Int. J. of Prod. Res.* vol. 26, número 3, 1988.
- [GOL89] GOLDRATT, E. & FOX, R. E.  
*A Corrida pela Vantagem Competitiva*. IMAM, São Paulo, 1989.
- [GRAB1] GRAVES, S. C.  
A Review of Production Scheduling. *Operations Research* 29(4), 1981.

- [GUI90] GUIMARÃES, Renato  
*Adaptabilidade das Técnicas Integradas de Gestão da Produção às Pequenas e Médias Indústrias.* Dissertação de Mestrado. Florianópolis, UFSC, 1990.
- [HAL88] HALL, Robert W.  
Cyclic Scheduling for Improvement. *Int. J. Prod. Res.* vol. 26, número 3, 1988.
- [HAY83] HAYES-ROTH, F., WATERMAN, D. A. & LENAT, D. B.  
*Building Expert Systems.* Ed. Addison Wesley, Reading, 1983.
- [KAN87] KANET, John J. & ADELSBERGER, Heimo H.  
Expert Systems in Production Scheduling. *European Journal of Operational Research* 29, 51-59, 1987.
- [KLE89] KLEEBANK, Vivian  
*SISEP: Uma Proposta de Sistema Baseado em Conhecimento para o Problema de Escalonamento da Produção.* Dissertação de Mestrado. Porto Alegre, UFRGS, 1989.
- [KUM86] KUMARA, S. R. T. & alii  
Expert Systems in Industrial Engineering. *Int. J. Prod. Res.* 5, vol. 21, 1986.
- [KUS87] KUSIAK, Andrew  
Designing Expert Systems for Scheduling of Automated Manufacturing. *Industrial Engineering* 7, vol. 19, 1987.
- [KUS88] KUSIAK, Andrew & CHEN, M.  
Expert Systems for Planning and Scheduling Manufacturing Systems. *European Journal of Operations Research* 34, 1988.
- [LEC87] LECOCCQ, P. et alii  
*A Knowledge-based System for On-line Planning and Real-time Control of Flexible Manufacturing Systems.* In: 6th International Conference on Flexible Manufacturing Systems. Turin, Nov 4-6, 1987.
- [LUN91] LUNA, Paulo de Tarso Mendes  
*Sistemas de Apoio à Decisão: Uma Aplicação na Administração do Capital de Giro sob Inflação.* Dissertação de Mestrado. Florianópolis, UFSC, 1991.
- [McK88] McKAY, Kenneth N., SAFAYENI, Frank R. & BUZACOTT, John A.  
Job-Shop Scheduling: What Is Relevant? *Interfaces* 18: 4, July-August 1988.
- [MAL75] MALINOWSKY, Hercílio  
*Um Procedimento para Programação da Produção.* Dissertação de mestrado. Florianópolis, UFSC, 1975.
- [MON91] MONTENEGRO, Fernando Borges  
*Uma Interface em Linguagem Natural para um Sistema Especialista de Administração de Capital de Giro.* Dissertação de Mestrado. Florianópolis, UFSC, 1991.

- [OKE86] O'KEEFE, R. M., BELTON, V. & BALL, T.  
Experience with using expert systems in OR. *Journal of Operational Research Society* 37/7, 657-668, 1986.
- [PAC90] PACHECO, Roberto C. S. et alii  
*Sistema Baseado em Conhecimento para Programação da Produção*. Projeto de pesquisa encaminhado à FINEP - Financiadora de Estudos e Projetos. Programa de Pós-graduação em Engenharia de Produção, UFSC, 1990.
- [PAC91] PACHECO, Roberto C. S.  
*Tratamento de Imprecisão em Sistemas Especialistas*. Dissertação de Mestrado. Florianópolis, UFSC, 1991.
- [RIC88] RICH, Elaine  
*Inteligência Artificial*. McGraw-Hill, São Paulo, 1988.
- [ROD88] RODAMMER, F. A. & WHITE JR., K. P.  
A Recent Survey of Production Scheduling. *IEEE Transactions on Systems, Man and Cybernetics* 6, vol. 18, 1988.
- [ROD90] RODRIGUES Filho, Ílson Wilmar  
*Inteligência Artificial aplicada na Engenharia Rodoviária*. Dissertação de mestrado. Florianópolis, UFSC, 1990.
- [SCH87] SCHILDT, Herbert  
*Advanced Turbo Prolog: Version 1.1*. Osborne McGraw Hill, 1987.
- [SCHXX] SCHONBERGER, R. J.  
Just-in-time - Replacing complexity with simplicity. In: *Just-in-time manufacture*. C.A.Voss (Editor) IFS Ltd., Inglaterra, sem data.
- [SHA89] SHAW, Michael & WHINSTON, Andrew  
An Artificial Intelligence Approach to the Scheduling of Flexible Manufacturing Systems. *IIE Transactions* vol. 21, número 2, 1989.
- [SMI86] SMITH, S. F. et alii  
Constructing and Maintaining Detailed Production Plans: Investigations into the Development of Knowledge-based Factory Scheduling Systems. *AI Magazine* 7(4), Fall 1986.
- [SMI90] SMITH, S. F. et alii  
An Integrated Framework for Generating and Revising Factory Schedules. *Journal of Operational Research Society* vol. 41, número 6, 1990.
- [SNE86] SNELLER, M.  
*Application of a Classical Management Approach to the Implementation of Material Requirements Planning Systems*. Ph.D. thesis. Claremont Graduate School, 1986.

- [THE88] THÉ Neto, Jessé Sotto Maior L.  
*Um Sistema de Treinamento para Operadores de Reservatórios Baseado em Técnicas de Inteligência Artificial.* Dissertação de Mestrado. Florianópolis, UFSC, 1988.
- [TOD91] TODESCO, José Leomar  
*Um Sistema Especialista para Auxiliar na Elaboração de Recomendações para o Controle da Sarna da Macieira.* Dissertação de Mestrado. Florianópolis, UFSC, 1991.
- [TER85] TERSINE, Richard J.  
*Production/Operations Management: Concepts, Structure and Analysis.* Segunda Edição, North-Holland, 1985.
- [VOSXX] VOSS, C. A. (Editor)  
*Just-In-Time Manufacture (International Trends in Manufacturing Technology).* IFS Ltd., Inglaterra, sem data.
- [WAL89] WALTER, Cláudio  
*MRPII, Kanban e Escalonamento da Produção: Uma Análise Comparativa.* Anais do IX ENEGEP - Encontro Nacional de Engenharia de Produção. Porto Alegre, 1989.
- [WAT86] WATERMAN, Donald A.  
*A Guide to Expert Systems.* Addison-Wesley, London, 1986.
- [YUK88] YUKI, M. M.  
*Uma Metodologia de Implementação de Técnicas e Filosofias Japonesas na Gestão de Empresas Brasileiras.* Dissertação de mestrado. Florianópolis, UFSC, 1988.

## 7.2 BIBLIOGRAFIA SUPLEMENTAR

- [AMAB8] AMADOR, M. S. T. & PIMENTÃO, J. P. B.  
*Representação de Conhecimento através de "Frames".* Fonte desconhecida.
- [BICXX] BICHENO, J. R.  
*A Framework for JIT Implementation.* In: *Just-in-time manufacture.* C.A.Voss (Editor) IFS Ltd., Inglaterra, sem data.
- [BOR91] BORLAND INTERNATIONAL  
*Turbo C++ - Programmer's Guide.* Segunda Edição. Borland International, Scotts Valley, CA, 1991.

- [BUH88] BU-HULAIGA, M. I. & CHAKRAVARTY, A. K.  
An Object-oriented Knowledge Representation for Hierarchical Real-time Control of Flexible Manufacturing. *International Journal of Production Research* 5, vol. 26, 1988.
- [BUR88] BURBIDGE, John L.  
Operation Scheduling with GT and PBC. *International Journal of Production Research* 26 (3), 1988.
- [CAM89] CAMPOS, D. F.  
*Planificación de Necesidades de Materiales con Recursos de Fabricación Limitados: Diseño de un Sistema aplicable a Pequeñas Empresas.* Ph.D. Thesis. Universidad Politecnica de Madrid, 1989.
- [CAR88] CARDOZO, E., FERNANDES, J. F. M. & GUIMARÃES, G. A.  
*Expert Systems for Assisting the Operation of Complex Manufacturing Processes.* IV Congresso Latino-Ibero-Americano de Pesquisa Operacional e Engenharia de Sistemas e XXI Simpósio Brasileiro de Pesquisa Operacional - Anais. Rio de Janeiro, 1988.
- [CHA90a] CHANG, T. C. & CRANDALL, Keith C.  
An Algorithm for Solving Expected Possibility and its Applications in Construction Resource Allocation. *Fuzzy Sets and Systems* vol. 34, no. 2, 1990.
- [CON88] CONDOR (Committee On the Next Decade in Operations Res.)  
Operations Research: The next decade. *Operations Research* 4, vol. 36, 1988.
- [COO76] COOPER, D. F.  
Heuristics for Scheduling Resource-constrained Projects: An Experimental Investigation. *Management Science* 11, vol. 22, 1976.
- [DUB80] DUBOIS, Didier & PRADE, Henry  
*Fuzzy Sets and Systems - Theory and Applications.* Academic Press, New York, 1980.
- [FER86] FERNANDES, C. A. O. & GOMIDE, F. A. C.  
*Planejamento e Sequenciamento da Produção em Sistemas Modernos de Manufatura.* In: Congresso Brasileiro de Automática, 6. Belo Horizonte, 1986.
- [FIN86] FININ, T.  
Understanding frame languages. *AI Expert*, Nov. 1986.
- [GAR78] GAREY, Michael R. et alii  
Performance Guarantees for Scheduling Algorithms. *Operations Research* 1, vol. 26, 1978.
- [GAR79] GAREY, Michael R. & JOHNSON, David S.  
*Computers and Intractability - A guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979.

- [GEI87] GEIB, F. O.  
Redes PERT na Preparação da Produção. *Tecnicouro* 9(1),  
Novo Hamburgo, 1987.
- [GEL81] GELDERS, L. F. & VAN WASSENHOVE, L. N.  
Production Planning: A Review. *Eur. J. Op. Res.* 7,  
101-110, 1981.
- [GIL90] GILBERT, J. P.  
The State of JIT Implementation and Development in the  
USA. *International Journal of Production Research*  
28(6), 1990.
- [GOLXX] GOLDRATT, E. M.  
*Devising a Coherent Production/Finance/Marketing  
Strategy using the OPT Rules.* Reprints do IMAM.
- [GRAB6] GRANT, T. J.  
Lessons for O.R. from A.I.: A Scheduling Case Study.  
*Journal of the Operational Research Society* 1, vol. 37,  
1986.
- [HAT83] HATON, J. P.  
*Knowledge-based and Expert Systems in Industrial  
Applications.* In: Artificial Intelligence. Proceedings  
of the IFAC Symposium. Leningrado, 1983.
- [JON86] JONES, C. V. & MAXWELL, W. L.  
A System for Manufacturing Scheduling with Interactive  
Computer Graphics. *IIE Transactions*, September 1986.
- [KAN86] KANDEL, A.  
*Fuzzy Mathematical Techniques with Applications.*  
Addison-Wesley, 1986.
- [LAR84] LARNDER, H.  
The Origin of Operational Research. *Operations Research*  
32(2), 1984.
- [OHA90] O'HARE, G. M. P.  
*Distributed Artificial Intelligence: An Invaluable  
Technique for the Development of Intelligent  
Manufacturing Systems.* Annals of the CIRP Vol.  
39/1/1990.
- [PAN77] PANWALKAR, S. S. & ISKANDER, Wafik  
A Survey of Scheduling Rules. *Operations Research* vol.  
25, número 1, 1977.
- [PAP82] PAPADIMITRIOU, Christos H. & STEIGLITZ, Kenneth  
*Combinatorial Optimization.* Prentice-Hall, New Jersey,  
1982.
- [PEAB4] PEARL, Judea  
*Heuristics: Intelligent Search Strategies for Computer  
Problem Solving.* Addison-Wesley, 1984.

- [RICK88] RICH, Kelly M. & ROBINSON, Phillip R.  
*Using Turbo Prolog*. Second Edition, Osborne McGraw Hill, 1988.
- [SCHXXa] SCHONBERGER, R. J.  
The Kanban System. In: *Just-in-time manufacture*. C.A.Voss (Editor) IFS Ltd., Inglaterra, sem data.
- [STAB6] STABLER JR., E. P.  
Object-oriented Programming in Prolog. *AI Expert*, outubro 1986
- [STAB8] STALK JR., G.  
Time - The Next Source of Competitive Advantage. *Harvard Business Review*, July-August 1988.
- [STAB5] STANGE, P.  
*Introducción a la Automatización de los Sistemas de Producción* (curso). In: Congreso Chileno de Ingeniería Eléctrica, 6. Santiago, 1985.
- [SWAB6] SWANN, D.  
Using MRP for Optimized Schedules (emulating OPT). *Production and Inventory Management*, Second Quarter, 1986.
- [VOSXXa] VOSS, C. A.  
Japanese JIT Manufacturing Management Practices in the UK. In: *Just-in-time Manufacture*. C.A.Voss (Editor) IFS Ltd., Inglaterra, sem data.
- [WILXX] WILSON, I. B.  
*Synchronised Manufacturing: The Route to Zero Inventory and Maximum Profitability*.

ANEXO IGLOSSÁRIO DE TERMOS USADOS EM ADMINISTRAÇÃO DA PRODUÇÃO  
E INTELIGÊNCIA ARTIFICIAL

- Ajuste de máquina - O processo de preparar uma máquina para fazer uma parte específica. O tempo de ajuste de máquina é vital para a determinação do tamanho de lote. Atualmente, há uma forte ênfase na redução dos tempos e, conseqüentemente, dos lotes [GES85].
- Aquisição de conhecimento - Processo de extrair, estruturar e organizar o conhecimento de alguma fonte, em geral um especialista humano, para que possa ser utilizado em um sistema especialista.
- Base (banco) de conhecimento - Parte de um sistema especialista que contém regras e/ou procedimentos sobre a área de conhecimento característica.
- Busca - O processo de analisar um conjunto de soluções possíveis segundo um algoritmo ou método, para encontrar uma solução viável.
- Centro de Trabalho - Grupo de máquinas semelhantes, com custos operacionais e de overhead também semelhantes. Podem ser agrupados em categorias maiores, como "centros de custos".
- CRP (Capacity Requirements Planning) - Planejamento das Necessidades de Capacidade, função referente à medida e ajustamento dos limites ou níveis de capacidade que são consistentes com um plano de produção [SNE86]. Um plano detalhado da capacidade requerida por cada máquina ou centro de trabalho, baseado no planejamento individual das partes, que é dado pelo MRP e arquivos de roteiros. A abordagem consiste em "simular" o fluxo de trabalho em cada centro de acordo com as ordens de fabricação, criando um perfil período a período das necessidades de capacidade em cada centro de trabalho.
- Encadeamento para frente - Método de inferência onde as partes SE das regras são comparadas com fatos, visando a deduzir novos fatos [WAT86].
- Encadeamento para trás - Método de inferência onde o sistema começa por aquilo que se quer provar (a parte ENTÃO das regras) e procura estabelecer os fatos que são premissas para a conclusão desejada. As premissas podem ser fatos na base de dados ou fatos a serem deduzidos de outras premissas, até que a conclusão seja efetivada ou falhe.
- Engenharia de conhecimento - Ramo da engenharia que se ocupa do processo de construção de sistemas especialistas [WAT86].

- Engenheiro de conhecimento - A pessoa que projeta e constrói um sistema especialista [WAT86].
- Espaço de estados (ou espaço de busca) - O conjunto de todas as soluções possíveis para um problema.
- Especialista - Pessoa que adquiriu, por treinamento, experiência e/ou talento, nível elevado de conhecimento e capacidade para a resolução de problemas em determinada área de conhecimento.
- Flow shop - Fabricação em linha, repetitiva. Cada tarefa tem uma rota predefinida e não são admitidas alternativas de percurso.
- Frame - Uma técnica de representação de conhecimento que associa características descritas por atributos e valores a 'nós' ou 'slots' desta frame. Estes nós representam conceitos ou objetos e formam uma rede conectada por relações e organizadas segundo uma hierarquia [WAT86].
- Heurística - Regra intuitiva ou simplificação que limita a busca de uma solução em um domínio muito amplo ou pouco estruturado [WAT86].
- Horizonte - Quantidade de tempo para o futuro. Por exemplo, um plano com horizonte de tempo de três anos significa que o plano se estenderá pelos próximos três anos [GES85].
- IA (Inteligência Artificial) - "Parte da ciência da computação dedicada ao desenvolvimento de programas inteligentes" [WAT86]. Em geral, os sistemas ditos inteligentes comportam-se de uma maneira que seria considerada inteligente se observada em uma pessoa.
- Inferência - O processo de raciocínio e dedução que leva à conclusão sobre um problema.
- Item - Um termo geral. Pode referenciar um produto, uma submontagem, uma parte comprada ou fabricada.
- Job shop - Processo de manufatura onde máquinas similares são agrupadas em centros de trabalho. É caracterizado por grandes filas, considerações sobre ajuste de máquinas e tempos de espera longos [GES85].
- Lead time - Tempo gasto durante o ciclo de fabricação. É o intervalo de tempo entre a liberação da ordem de fabricação e o término da fabricação. Para itens comprados, corresponde ao intervalo entre a colocação da ordem de compra e sua disponibilidade. "Intervalo médio entre duas etapas sucessivas de um roteiro de fabricação" [WAL89]. O tempo que é requerido para fazer um produto ou componente [GES85].
- Lista de Materiais (Bill of Materials) - É uma listagem de todas as submontagens, partes e matérias-primas que compõem uma montagem final, mostrando as quantidades requeridas para fazer a montagem [SNE86]. Os materiais (matérias-primas, partes fabricadas e compradas, submontagens, etc.) que são usados para fazer os produtos. Às vezes estruturada como uma lista completa (total list), uma lista parcial (as coisas requeridas para o próximo nível de montagem), ou uma 'indented structure list' [GES85].

Lógica difusa - Abordagem ao raciocínio aproximado. Usa a técnica de conjuntos difusos e é capaz de tratar variáveis lingüísticas, como: *muito*, *quase nunca*, etc. e dados imprecisos ao invés de apenas dados precisos e dicotomizados, como no caso da lógica clássica.

Manutenção corretiva - Aquela feita por defeito de equipamento.

Manutenção preventiva - Manutenção programada, prevenindo quebras de máquinas e equipamentos. Pode incluir um sistema de gerenciamento de estoques de peças de reposição e uma escala de pessoal.

Metaconhecimento - "Conhecimento sobre o conhecimento". Em um sistema especialista, é o conhecimento sobre como o sistema raciocina.

Meta-regra - Regra que descreve como outras regras devem ser usadas ou modificadas [WAT86].

Modus Ponens - Regra dedutiva que constitui a base da maioria dos mecanismos de inferência presentes em sistemas especialistas [PAC91]. Pode-se expressar segundo o esquema:

REGRA: SE A ENTÃO B.

PREMISSA: A.

CONCLUSÃO: B.

Se A (premissas) é verdadeiro e a regra diz que "se A, então B", é válido inferir que B (conclusões) é verdadeiro.

Modus Ponens Difuso - Regra dedutiva adaptada à inferência no raciocínio aproximado, utilizando lógica difusa. No caso difuso, o *modus ponens* pode ter graus de pertinência associados à regra, à premissa e à conclusão:

REGRA: SE  $A^*$  ENTÃO B.

PREMISSA:  $A^*$ .

CONCLUSÃO:  $B^*$ .

onde os termos A e B na regra podem envolver conceitos difusos. Se A (premissas) é verdadeiro com grau de pertinência  $\mu_p$  e a regra diz que "se A, então B", é válido inferir que B (conclusões) é verdadeiro com grau de pertinência  $\mu_c$ , função de  $\mu_p$ .

Motor de inferência - A parte de um sistema especialista que contém a estratégia de inferência para chegar às conclusões. O motor de inferência processa o conhecimento existente na base de conhecimento para chegar a novas conclusões.

MRP (Materials Requirements Planning) - Planejamento de Necessidades de Materiais, processo que tem como entrada o Plano Mestre de Produção, que é 'explodido' segundo os itens de materiais e componentes para cada produto final (lista de materiais), resultando as necessidades de cada item, em quantidade e momento apropriado.

MRPII (Manufacturing Resources Planning) - Planejamento dos Recursos de Manufatura, evolução do MRP que compreende, além das funções de contabilização da manufatura do MRP, o sistema de Planejamento das Necessidades de Capacidade (CRP). É um sistema que se propõe a abranger todos os recursos de manufatura, incluindo integração com funções de marketing, transformando o MRP numa ferramenta de auxílio à tomada de decisão empresarial.

NP-completo - Classe de problemas combinatoriais cuja complexidade computacional determina sua intratabilidade inerente. Nenhum problema NP-completo pode ser resolvido por qualquer algoritmo polinomial conhecido (um algoritmo polinomial é aquele cujo número de passos para se chegar à solução cresce como um polinômio em relação ao tamanho da entrada). Entretanto, não é fato provado a inexistência de um algoritmo polinomial para resolver um (e, por conseqüência, todos) problema NP-completo. O problema de programação da produção, o problema do caixeiro viajante e o de programação linear inteira são exemplos de problemas sabidamente NP-completos. Do ponto de vista computacional, qualquer algoritmo que resolva este tipo de problema irá requerer um tempo de solução crescendo de forma exponencial, no mínimo, em relação ao crescimento do número de variáveis de entrada. Isto torna a solução impraticável, exceto para problemas muito pequenos. Uma vez reconhecido um problema como NP-completo, é recomendável que o pesquisador operacional detenha sua atenção na busca por soluções aproximadas ou satisfatórias ou relaxe as restrições do problema, economizando o esforço infrutífero de procurar por uma solução ótima e exata.

Período - Porção de tempo no qual é feito o planejamento. Em geral, corresponde a um dia, semana, mês, etc. [GES85].

Plano Mestre de Produção - Plano de produção de produtos finais para um certo período, chamado horizonte de tempo. É executado a partir de informações de previsões de vendas, marketing, engenharia, vendas realizadas, etc.

PO (Pesquisa Operacional) - Uma abordagem científica para problemas operacionais. O objetivo da PO é ser um auxílio na tomada de decisão, de acordo com informações quantitativas analisadas cientificamente. A utilização de PO na resolução de um problema pressupõe a construção de um modelo do problema a ser analisado por uma ou mais das técnicas. Técnicas de PO são ferramentas matemáticas, como: técnicas de otimização (programação linear, não-linear, inteira, dinâmica, etc.), teoria dos jogos, teoria das filas, probabilidade, estatística, teoria dos grafos, otimização em redes, etc.

Produto - Ítem final, o que é despachado para o cliente.

Raciocínio simbólico - Resolução de problemas baseada na aplicação de estratégias e heurísticas para manipular símbolos referentes a conceitos relativos ao problema [WAT86].

Regra de produção - Regra do tipo SE-ENTÃO, que pode ser satisfeita por condições existentes em uma base de dados e pode gerar ações que modifiquem a base de dados.

Representação - Processo de formular ou visualizar um problema, de modo que ele seja fácil de resolver [WAT86].

Representação de conhecimento - Forma de organizar ou estruturar o conhecimento. Em um sistema especialista, esta representação deve ser adequada à forma como o sistema manipulará o conhecimento.

Roteiro - A seqüência de operações através de vários centros de trabalho [GES85].

Rough-cut capacity planning - Planejamento de recursos feito no nível do Plano Mestre de Produção, sobre centros de trabalho críticos quanto às necessidades de materiais.

Sistema Especialista - Um sistema computacional que usa conhecimento especializado para atingir níveis altos de performance na resolução de problemas em uma área limitada de conhecimento [WAT86]. Em geral, deve ser capaz de explicar o processo de raciocínio que levou a uma conclusão.

Tamanho de lote - Quantidade a ser produzida, referente a uma ordem de fabricação.