

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Escalonamento de um Job Shop:
Análise de um Algoritmo
com Regras Heurísticas**

por

RONALD LOPES DE OLIVEIRA

Dissertação submetida à avaliação,
como requisito parcial para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. Cláudio Walter
Orientador

Porto Alegre, julho de 2001.

CIP – Catalogação na Publicação

Oliveira, Ronald Lopes

Escalonamento de um Job Shop, um Algoritmo com Regras Heurísticas/por Ronald Lopes de Oliveira. – Porto Alegre: PPGC da UFRGS, 2001.

51 p. :il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR – RS, 2001. Orientador: Walter, Cláudio.

1. Sistemas de Manufatura. 2.Escalonamento de Um Job Shop
3.Regras Heurísticas. 4.Modelo de Dados. I. Walter, Cláudio. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof.^a Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor Adjunto de Pós-Graduação: Prof. Philippe Olivier Alexandre Navaux.

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

Agradecimentos

Inicialmente, quero agradecer aos meus pais, por toda a dedicação e amor demonstrados em todos estes anos de estudo, tendo sempre uma palavra de conforto nos momentos mais difíceis.

À minha esposa, Rosangela, o meu agradecimento muito especial, por ter me apoiado durante toda esta jornada, suportando meus desabafos, mau humor, sempre com suas palavras de compreensão e otimismo. Rosangela, te amo.

Ao meu filho Lorenzo, pela compreensão nestes tempos difíceis onde não pude lhe proporcionar a atenção merecida, muito obrigado.

Aos meus amigos, colegas de trabalho e a todos que de certa forma me ajudaram e apoiaram neste desafio, mas que com certeza ocupam um espaço em meu coração. Pelas horas de conversa, desabafos, e também pela descontração prestada.

À Hidrover Equipamentos Oleodinâmicos S.A., pela ajuda prestada, fornecendo informações valiosas para o desenvolvimento do trabalho.

Ao meu professor orientador, Cláudio Walter, pela orientação e dedicação durante esse período.

À Universidade de Caxias do Sul pela oportunidade de realização deste mestrado e apoio prestado durante o desenvolvimento do mesmo.

Sumário

Lista de Abreviaturas.....	6
Lista de Figuras	7
Lista de Tabelas	8
Resumo	9
Abstract	10
1 Introdução	11
1.1 Objetivo do Trabalho.....	12
1.2 Organização do Trabalho.....	13
2 Sistemas de Produção.....	15
2.1 Tipos de Produção.....	16
2.1.1 <i>Produção Intermitente</i>	17
2.1.2 <i>Produção Contínua</i>	17
2.1.3 <i>Produção por Projetos</i>	17
2.2 Técnicas de Gestão da Produção	18
2.2.1 <i>Kanban</i>	18
2.2.2 <i>Planejamento com Capacidade Finita</i>	19
2.2.3 <i>MRP / MRP II</i>	20
2.2.4 <i>OPT</i>	23
3 Escalonamento: Problemas e Soluções.....	24
3.1 Introdução.....	24
3.2 Programação para a Frente e para Trás.....	25
3.3 Programação Empurrada e Puxada.....	26
3.4 Escalonamento num Ambiente <i>Job Shop</i>	26
3.5 Modelos Matemáticos	28
3.6 Modelos Heurísticos	31
4 Desenvolvimento de Sistemas de Escalonamento.....	38
4.1 O Papel do Escalonamento dentro de uma Organização	38
4.2 Projeto, Desenvolvimento e Implementação de Sistemas de Escalonamento..	40
4.3 Módulo de Escalonamento	42
4.4 Interface com o usuário	43
4.5 Aplicações Genéricas Versus Específicas.....	44
4.6 Procedimentos Gerais de Escalonamento	45
5 Trabalhos Relacionados.....	47
5.1 Sistemas Acadêmicos	47

5.1.1	<i>LEKIN [PIN99] e [LEK2001]</i>	47
5.1.2	<i>ISTOP [KER90]</i>	49
5.1.3	<i>CUISE [PIN99]</i>	50
5.1.4	<i>ISIS [FOX90]</i>	51
5.1.5	<i>OPIS [SMI90]</i>	52
5.2	Sistemas Comerciais Genéricos	54
5.2.1	<i>PROVISA [PRO2001]</i>	54
5.2.2	<i>PREACTOR [PRE2001]</i>	55
5.2.3	<i>CAPS [CAP2000]</i>	56
5.3	Sistemas Específicos	57
5.3.1	<i>GATES [PIN99]</i>	57
6	Sistema Proposto	59
6.1	Modelo Conceitual de Dados.....	59
6.1.1	<i>Introdução</i>	59
6.1.2	<i>O Modelo Conceitual de Dados</i>	61
6.1.3	<i>Descrição das classe e seus atributos</i>	63
6.2	Implementação do Algoritmo de Escalonamento.....	78
6.2.1	<i>Introdução</i>	78
6.2.2	<i>Análise da Capacidade Fabril</i>	79
6.2.3	<i>O Algoritmo de Escalonamento</i>	85
6.2.4	<i>Resultados Alcançados</i>	94
7	Considerações Finais	97
	Bibliografia.....	99

Lista de Abreviaturas

CAPS	Computer Aided Process Scheduling
CUISE	Columbia University Interactive Scheduling Editor
ETO	Engineer-to-Order
ERP	Enterprise Resource Planning
FIG.	Figura
GATES	Gate Allocation and Tracking Expert System
JIT	Just in Time
MRP	Material Requirements Planning
MRPII	Manufacturing Resource Planning
MTO	Make-to-Order
MTS	Make-to-Stock
MWR	Most Work Remaining
ODBC	Open DataBase Connectivity
OKP	One-of-a-Kind
OF	Ordem de Fabricação
OO	Orientação a Objetos
OPIS	Oportunistic Intelligent Scheduler
OPT	Optimized Production Technology
SPT	Short Process Time
TAB.	Tabela
UML	Unified Modelling Language

Lista de Figuras

FIGURA 2.1 - Fluxo de Dados em um Sistema MRP.....	22
FIGURA 4.1 - Fluxo das informações em um sistema de manufatura [PIN99]	39
FIGURA 6.1 - Modelo conceitual de dados para escalonamento e controle da produção..	62
FIGURA 6.2 - Fluxo geral do sistema.....	79
FIGURA 6.3 - Tela de capacidade dos gargalos	81
FIGURA 6.4 - Tela de aumento da capacidade fabril	82
FIGURA 6.5 - Tela de redução da capacidade fabril	83
FIGURA 6.6 - Tela de capacidade dos gargalos	84
FIGURA 6.7 - Algoritmo de escalonamento.....	86
FIGURA 6.8 - Tela das restrições das máquinas por família.....	89
FIGURA 6.9 - Tela de início do escalonamento para a regra MWR	90
FIGURA 6.10 - Tela de escalonamento da máquina Serra Fita 291	91
FIGURA 6.11 - Tela de início de escalonamento para a regra SPT.....	92
FIGURA 6.12 - Tela de escalonamento da máquina Serra Fita 529	93

Lista de Tabelas

TABELA 3.1 - Vantagens da programação para a frente e para trás	26
TABELA 3.2 - Matriz de ref. de um <i>job shop</i> geral composto de cinco máquinas [HAX 84]	27
TABELA 3.3 - Aplicação da Regra de Johnson para programar seis tarefas em dois centros de trabalho.....	34
TABELA 3.4 - Regras heurísticas baseadas no tempo de processamento	34
TABELA 3.5 - Regras heurísticas baseadas na data de entrega.....	35
TABELA 3.6 - Regras heurísticas baseadas no número de operações.....	35
TABELA 3.7 - Regras heurísticas baseadas no custo	35
TABELA 3.8 - Regras heurísticas baseadas no tempo de <i>setup</i>	36
TABELA 3.9 - Regras heurísticas baseadas no instante de chegada ou aleatórias	36
TABELA 3.10 - Regras heurísticas baseadas na máquina	36
TABELA 3.11 - Regras heurísticas baseadas na montagem	37
TABELA 6.1 - Atributos da Classe Cliente	63
TABELA 6.2 - Atributos da Classe PedidoVenda	63
TABELA 6.3 - Atributos da Classe ItemPedidoVenda.....	64
TABELA 6.4 - Atributos da Classe Produto	65
TABELA 6.5 - Atributos da Classe Família	66
TABELA 6.6 - Atributos da Classe Dimensão.....	66
TABELA 6.7 - Atributos da Classe DimensãoFamília	67
TABELA 6.8 - Atributos da Classe ValorDimensão	67
TABELA 6.9 - Atributo da Classe Desenho	68
TABELA 6.10 - Atributos da Classe EstruturaProduto	68
TABELA 6.11 - Atributos da Classe RoterioFab.....	69
TABELA 6.12 - Atributos da Classe RoterioFabPosterior	70
TABELA 6.13 - Atributos da Classe OF.....	70
TABELA 6.14 - Atributos da Classe AtividadeOF	71
TABELA 6.15 - Atributos da Classe HistóricoOF.....	72
TABELA 6.16 - Atributos da Classe Funcionário	72
TABELA 6.17 - Atributos da Classe CentroTrabalho	73
TABELA 6.18 - Atributos da Classe HorasAdicionais.....	73
TABELA 6.19 - Atributos da Classe Terceiros.....	74
TABELA 6.20 - Atributos da Classe HorasMaqParada.....	74
TABELA 6.21 - Atributos da Classe Máquina	75
TABELA 6.22 - Atributos da Classe Restrição.....	75
TABELA 6.23 - Atributos da Classe CalendárioFab	76
TABELA 6.24 - Atributos da Classe HorárioFab	76
TABELA 6.25 - Atributos da Classe Feriado	77
TABELA 6.26 - Atributos da Classe CapacidadeFab	77
TABELA 6.27 - Capacidade do centro de trabalho de corte.....	81
TABELA 6.28 - Escalonamento manual da semana 08/05/2000 a 12/05/2000.....	94
TABELA 6.29 - Escal. utilizando o algoritmo na semana 08/05/2000 a 12/05/2000.....	94
TABELA 6.30 - Escalonamento manual da semana 21/08/2000 a 25/08/2000.....	95
TABELA 6.31 - Esc. utilizando o algoritmo na semana 21/08/2000 a 25/08/2000.....	95

Resumo

O presente trabalho visa definir um modelo de alocação dos recursos da produção para centros de trabalho em sistemas baseados em *job shop*, usando a abordagem heurística para garantir uma boa alocação dos recursos. São levados em conta a complexidade de um ambiente de produção, seus aspectos temporais e os modelos de *Job Shop Scheduling* atualmente em uso. Com isso são examinados os aspectos conceituais deste ambiente e proposto um modelo de alocação de recursos para auxiliar no planejamento operacional do mesmo.

Pode-se definir os recursos como todos os elementos necessários à execução das diversas atividades de um processo produtivo, tais como equipamentos, máquinas, mão-de-obra, etc. Por sua vez, os recursos são limitados por natureza, quanto à quantidade de unidades disponíveis, às suas funcionalidades e à capacidade produtiva. O processo de alocação dos recursos pressupõe a designação dos recursos mais satisfatórios para a execução de cada uma das atividades que fazem parte de um projeto.

O modelo proposto é baseado no uso de heurísticas para resolver o escalonamento nos centros de trabalho, também chamados de células de produção, usando restrições e regras entre as ordens de fabricação (peças) e as máquinas, para encontrar uma solução satisfatória ao problema.

O resultado final é uma ferramenta de apoio à decisão no processo de manufatura, permitindo a visualização do melhor escalonamento de produção, visando a redução do ciclo e *setup* de produção no processo, com base nas informações locais do ambiente fabril.

O sistema está implementado numa empresa de componentes hidráulicos, inicialmente no centro de trabalho de corte, composto por quatro máquinas que realizam o corte de diversos tipos de matérias-primas.

Palavras-chave: Sistemas de Manufatura, Planejamento da Produção, Alocação de Recursos, Escalonamento de um *Job Shop*, Regras Heurísticas.

TITLE: “SCHEDULLING OF THE JOB SHOP: ANALISYS OF AN ALGORITHM BY MEANS OF HEURISTICS RULES”

Abstract

This proposal aims to define a model for the allocation of production resources for work stations with systems based on job shop, using the heuristic approach to guarantee a good allocation of resources. The complexity of a production environment, its temporal aspects and current types of job shop scheduling are taken into consideration. In this way, the conceptual aspects of the said environment are examined and a model for the allocation of resources is proposed so as to give support to its operational planning.

We may define the resources as all the necessary elements for carrying out the several aspects of a production process, such as equipment, machine, labor, etc. However, the resources are limited to the units available, their functionalities and productive capacity. The process of allocation of resources presupposes the selection of the best resources for carrying out every activity that is part of a project.

The model proposed is based on the use of heuristics to deal with the problem of scheduling in the work stations, also called production cells, using constraints and rules between manufacturing orders and machines to find a satisfactory solution to the problem.

The final result is a support tool for decision making in the manufacturing process, allowing a better view of production scheduling, seeking the reduction of the cycle and production setup in the process, based on local information about the manufacturing environment.

The system is implemented in a company that manufactures hydraulic components, initially in the cutting workstation, with of four machines that cut several types of raw materials.

Keywords: Manufacturing Systems, Planning of the Production, Allocation of Resources, Job Shop Scheduling, Heuristics Rules.

1 Introdução

Às vezes as pessoas não se dão conta de que o lucro de uma empresa está diretamente relacionado com sua produção. Planejar bem e fazer uma programação eficiente são atitudes fundamentais para que uma empresa se torne competitiva. Certeza nas datas de entrega e diminuição da ociosidade de recursos são algumas das vantagens que uma boa programação pode oferecer.

Sistemas de manufatura são sistemas complexos e abertos, sensíveis a alterações externas e internas, existindo um enorme número de fatores que atuam na decisão e no controle de sua produção. De acordo com [SLA97], o processo de alocação de recursos compreende uma eficaz designação dos recursos para a execução das atividades de um projeto. Tendo em vista, nos dias atuais, o aumento de exigência do consumidor e o crescimento da concorrência, torna-se fundamental um processo decisório cada vez mais inteligente e um sistema de produção que consiga maximizar os recursos e minimizar o tempo de produção, reduzindo o custo de produção para a empresa.

A ferramenta tradicionalmente utilizada por programadores de produção é uma simples lista de ordens com datas de entregas, seguindo uma orientação prévia dos superiores onde os recursos são alocados as operações de produção ao longo do tempo. É uma ferramenta modesta e estática para lidar com um problema de natureza dinâmica.

Para cada alteração que ocorra em chão-de-fábrica obriga o programador a gastar um tempo significativo para realocar os trabalhos, trocar prioridades ou alterar rotas produtivas, na tentativa de cumprir as datas de entrega prometidas.

Do lado da demanda, a quantidade de trabalhos a realizar varia devido à chegada de novos pedidos, às mudanças de prioridade e às variações no estoque de produtos acabados. Imprevistos acontecem a qualquer momento. Isso acontece quando as entregas são realizadas, quando previsões são traduzidas em demanda real ou quando os pedidos são cancelados.

Os diversos modelos e métodos convencionais aplicados no planejamento da produção, utilizados atualmente, tanto com uma sistemática manual ou informatizada, limitam a flexibilidade e a eficiência exigidas. A grande quantidade de elementos envolvida no sistema de manufatura, a falta de estruturação de objetivos e restrições e as constantes mudanças desses elementos podem dificultar ou impossibilitar a aplicação adequada dos métodos convencionais.

O problema fundamental é que modelos tradicionais de programação da produção e alocação de recursos devem lidar com as leis da natureza. O chão-de-fábrica é um local rico em eventos randômicos oriundos das perturbações e, por conseqüência, requer comportamento pró-ativo dos recursos disponíveis [BAS98].

Outro problema que afeta diretamente as empresas que possuem uma grande variedade de peças produzidas em uma mesma célula de produção ou centro de trabalho, é o tempo gasto para a preparação das máquinas, ou seja, o *setup* de produção das máquinas. Muitas empresas não possuem um sistema eficiente para o agrupamento de peças dentro de um limite de produção.

Conforme os pedidos forem sendo incluídos, pelo Departamento de Vendas, a alocação dos recursos de produção é gerada com base na data de entrega. Fica a cargo do encarregado do centro de trabalho ou do próprio operador da máquina a decisão de qual lote de peças deva ser feito primeiro, quando ambos têm o mesmo prazo de entrega. Assim sendo, muitos sistemas não possuem o conhecimento de todo o universo da produção, para saber qual a seqüência de peças que estão chegando na célula. Só possuem conhecimento das peças já disponíveis em áreas de armazenamento, podendo assim vir a realizar diversas preparações (*setups*), elevando os custos de produção e diminuindo a produtividade das máquinas.

Considerando essa visão do problema, pode-se concluir que a solução para a complexidade do planejamento da produção encontra-se na busca de um nível de utilização ideal dos recursos, o que implica fundamentalmente a busca da melhor alocação dos recursos a cada momento, visando atender às demandas existentes.

Para tanto, é proposta a utilização de regras heurísticas que visam decidir qual a melhor seqüência a ser executada, dependendo das restrições das máquinas e das condições atuais da produção.

1.1 Objetivo do Trabalho

O objetivo deste trabalho compreende a investigação, proposição de um modelo de dados e desenvolvimento de um algoritmo para o problema de alocação dos recursos aos processos produtivos de uma organização de manufatura.

A primeira etapa será a realização do levantamento das principais particularidades referentes aos sistemas de produção. Logo após, será realizado um estudo das características técnicas das peças da empresa e das características de corte de cada máquina do centro de trabalho. O conhecimento será adquirido através do acompanhamento da produção e de entrevistas às pessoas que presenciam no seu dia-a-dia os problemas enfrentados para o escalonamento da produção. Através dessa vivência no chão-de-fábrica, conseguiu-se trazer para este trabalho um enriquecimento de detalhes pertinentes às atividades de produção, que apenas a leitura sobre o assunto não conseguiria esclarecer. Após o estudo dos principais problemas envolvendo o escalonamento da produção, serão apresentadas algumas sugestões para solucionar os problemas encontrados.

A segunda etapa deste trabalho é o desenvolvimento de um modelo conceitual de dados de referência e o desenvolvimento de um sistema para o controle da produção em um ambiente do tipo *Job Shop*, este é geralmente caracterizado por permitir diferentes fluxos das ordens de produção entre as máquinas e diferentes números de operações por ordem, que

são processadas apenas uma vez em cada máquina. Essa etapa é considerada a mais importante deste trabalho, pois através dela pretende-se demonstrar ser possível realizar um sistema de escalonamento que utilize as informações do sistema de gestão em uso, sem a necessidade de redundância dos dados. O modelo será desenvolvido utilizando técnicas de orientação a objetos.

Através do estudo realizado, foi construída uma ferramenta que escalona e controla os centros de trabalho em um sistema de manufatura. O conhecimento do sistema fica armazenado através de regras heurísticas embutidas no mesmo e de regras lógicas informadas pelos especialistas, também podendo ser chamadas de restrições operacionais. Essas restrições contêm informações sobre as peças (diâmetro interno, diâmetro externo, comprimento, etc.) e as máquinas (características de produção) disponíveis no sistema.

As regras heurísticas são selecionadas pelo usuário para realizar o escalonamento baseado nas características de produção da empresa. Estas heurísticas possuirão os critérios de alocação para a solução de conflitos de produção gerados entre as ordens e as máquinas. O conflito é gerado quando uma ordem de produção pode ser alocada em mais de uma máquina no mesmo instante.

O objetivo é realizar o melhor escalonamento (*scheduling*) de produção possível, utilizando informações locais para a tomada de decisão. O que se pretende com esse sistema é reduzir o tempo de produção e *setup*, valendo-se da similaridade das peças para realizar o melhor agrupamento das mesmas.

O modelo de dados proposto e a ferramenta foi implementada pela empresa Hidrover Equipamentos Oleodinâmicos S.A., situada em Caxias do Sul, num centro de trabalho piloto composto por quatro máquinas que realizam a operação de corte de diversos tipos de peças. Esse modelo no futuro deverá ser estendido aos demais centros de trabalho da empresa.

1.2 Organização do Trabalho

O presente trabalho está dividido em cinco capítulos, descritos da seguinte forma:

O primeiro capítulo visa apresentar o ambiente no qual o mesmo está inserido e um resumo dos problemas do planejamento e controle da produção.

No segundo capítulo são introduzidos os conceitos e as características básicas dos tipos de produção e técnicas de gestão da produção. Também são abordadas as vantagens e desvantagens de cada tipo de gestão da produção.

No terceiro capítulo são apresentados os problemas de escalonamento e algumas formas de soluções para o mesmo; dentre elas serão abordados modelos de solução ótima e modelos heurísticos com suas características.

No quarto capítulo são identificadas as características dos sistemas de escalonamento, seus objetivos e sua relação com os sistemas de ERP atualmente em uso pelas empresas.

No quinto capítulo são demonstrados alguns sistemas de escalonamento encontrados no meio acadêmico e sistemas comerciais já consagrados pelas empresas.

No sexto capítulo, após estudar os modelos existentes e conhecer os tipos de produção também existentes, é apresentado o modelo conceitual de dados, desenvolvido e implementado com o objetivo de mostrar uma solução de escalonamento em centros de trabalho numa empresa de componentes hidráulicos.

Por fim, o apresentam-se as considerações finais sobre o desenvolvimento do trabalho.

2 Sistemas de Produção

Os sistemas de manufatura têm passado por grandes evoluções no decorrer dos anos e, conseqüentemente, as maneiras de planejamento e controle da produção também.

O Planejamento da Produção se preocupa em planejar, dirigir e controlar o suprimento e a organização de facilidades de produção, bem como projetar os métodos de processamento que serão empregados para gerar os produtos fornecidos pela empresa.

Segundo [SLA97], prover a capacitação de satisfazer a demanda atual e futura é uma responsabilidade fundamental da administração da produção. Um equilíbrio entre a capacidade e a demanda pode gerar altos lucros e clientes satisfeitos, enquanto o desequilíbrio pode ser potencialmente desastroso para a empresa. Embora planejar e controlar a capacidade seja uma das principais responsabilidades dos gerentes de produção, também deveria envolver outros gerentes funcionais. Há diversas razões para isso. Uma delas é que as decisões de capacidade têm um impacto em toda a empresa. Outra é que todas as outras funções fornecem entradas vitais para o processo de planejamento. E, por último, que cada função do negócio normalmente deverá planejar e controlar a capacidade de suas próprias *microoperações* para atender à função principal da produção.

Conforme [BER90], o alvo principal da discussão sobre sistemas de produção deve estar centrado nas relações entre a fabricação dos produtos ao longo do tempo. Entretanto, alguns aspectos devem ser considerados:

- os produtos compartilham recursos de produção para a fabricação de seus itens componentes e sua montagem final. Esses recursos de produção são limitados e possuem restrições de capacidade quanto à sua utilização. Além disso, existem diferentes tempos para que cada um dos recursos de produção necessários às atividades de fabricação dos componentes do produto se torne disponível;
- o planejamento das necessidades de materiais, muitas vezes por razões econômicas (imobilização mínima de recursos financeiros em inventário), baseia-se no conceito de *just-in-time*. Entretanto, o atraso no recebimento de qualquer um dos itens de inventário a ser utilizado no processo produtivo também pode determinar um tempo de espera não previsto, prejudicando o *lead time* do produto final;
- a preparação (*setup*) das máquinas, que consome em alguns casos um tempo bastante considerável, pode influenciar diretamente no *lead time* dos itens componentes e, conseqüentemente, dos produtos finais;
- a existência de recursos de produção gargalo, que necessitam de atenção especial, pois acabam sendo um importante fator na sincronização da programação da produção;
- as flutuações do mercado relativamente à demanda dos produtos determinam dificuldades quanto ao estabelecimento de quais desses produtos ou seus

componentes devem ser fabricados ao longo de determinado período, afetando diretamente na programação da produção e na gestão de materiais.

De acordo com [WAL89], os obstáculos encontrados no desenvolvimento de qualquer abordagem, para a alocação de recursos, estão diretamente relacionados com as características do processo produtivo necessário para atender à demanda dos produtos.

Dois aspectos devem ser considerados devido à complexidade desse processo: alterações de demanda e o tempo de resposta do sistema produtivo.

Quanto maior for o volume de alterações de demanda, maior será a complexidade do sistema produtivo, levando em consideração a freqüente mudança das atividades a serem executadas pelos vários recursos da fábrica. Diversas ordens de fabricação são geradas simultaneamente, ocasionando variados roteiros de fabricação para cada um dos diferentes itens a serem produzidos.

O tempo de resposta do sistema produtivo está diretamente ligado à sua capacidade de atendimento das alterações da demanda. Após um certo tempo de resposta (se o tempo de resposta for superior a determinado valor), o sistema produtivo passa a requerer uma eficiente programação da produção. Se isso não acontecer, há necessidade de formação de estoques, de modo a atender às variações de demanda. A formação de estoques intermediários deve ser evitada, pois aumenta o custo de armazenamento para a empresa, sendo conveniente dessa forma que o produto esteja disponível próximo a seu consumo.

Segundo [BAS98], o problema fundamental é que os modelos tradicionais de programação da produção e alocação de recursos devem lidar com leis da natureza. O chão-de-fábrica é um local rico em eventos randômicos provenientes das perturbações e, por conseqüência, requer atitudes ou intervenções dinâmicas dos recursos disponíveis.

Pode-se assim dizer que, quanto maior for a dinâmica do ambiente de negócios, sob o ponto de vista das perturbações, e ainda, quanto maior for a descentralização e autonomia na apropriação dos recursos às atividades produtivas (grau de liberdade entre os diversos participantes envolvidos na solução do problema), mais difícil torna-se a utilização de modelos de planejamento baseados em um controlador central.

2.1 Tipos de Produção

Cada manufatura utiliza algum sistema produtivo para realizar suas atividades de transformação. Segundo [LOP96], os tipos de produção podem ser identificados pelas características operacionais, e não pelo produto acabado. A seguir, serão abordados os tipos de produção existentes.

2.1.1 Produção Intermitente

Esse tipo de produção é característico de sistemas de produção baseados em *job shop*, onde há uma diversificação dinâmica do processo produtivo.

Nesse tipo de produção, o arranjo físico é conhecido como funcional ou por processo em função de os equipamentos e as habilidades dos trabalhadores estarem agrupados. Ele pode ser dividido em duas áreas: por encomenda e por lote.

Na produção por encomenda agregam-se materiais, partes e componentes, a fim de fabricar, montar ou vender uma peça do equipamento ou do sistema de produção.

Na produção por lotes fabrica-se um produto em lotes, através de uma seqüência de operações, sendo cada uma delas executada no lote inteiro antes que qualquer operação subsequente seja iniciada.

2.1.2 Produção Contínua

Esse tipo de produção é característico de sistemas de produção baseados em *flow shop*, onde todos os produtos fabricados passam por todas as máquinas componentes de seu roteiro de fabricação, seguindo sempre a mesma ordem.

Nessa abordagem os equipamentos executam continuamente as mesmas operações, e o material se move com pequenas interrupções entre uma e outra máquina, até chegar aos locais de armazenamento ou estoque de produtos acabados. Nesse tipo de produção encontram-se a contínua pura e em massa.

Na contínua pura existe uma só linha de produção, e os produtos finais são todos exatamente iguais. Nesse caso, pode-se citar como exemplo a indústria do petróleo e derivados, energia elétrica, produtos químicos, etc.

Na produção contínua em massa ocorre a fabricação, em larga escala, de poucos produtos com grau de diferenciação relativamente pequeno. É o caso das indústrias de automóveis, geladeiras, fogões, etc.

A produção contínua em massa pode ser chamada de massa pura quando existe uma linha ou conjunto de equipamentos específicos para um produto final, e de massa com diferenciação quando adaptações em uma linha permitem a fabricação de produtos com algumas diferenças entre si. Por exemplo: a produção de livros requer o mesmo processo produtivo, porém existe uma diversidade de livros que requer adaptações das máquinas.

2.1.3 Produção por Projetos

Esse tipo de produção também é conhecida como ETO (*engineer to order*) ou OKP (*one-of-a-kind production*), e seu principal objetivo é atender uma necessidade específica

dos clientes, com todas as suas atividades voltadas para essa meta. Cada projeto é um produto único. Nesse tipo de processo encontram-se a produção de navios, aviões, grandes estruturas, etc.

Empresas de sucesso nessa área percebem que, apesar de cada pedido ser diferente, usualmente a sua especificação varia dentro de uma linha limitada de parâmetros e valores. Certamente essas empresas vão fabricar produtos similares que possuem uma grande semelhança entre si e se utilizam de técnicas de parametrização e normalização para obter uma produtividade e um lucro satisfatórios.

Dependendo do sistema produtivo adotado, a indústria poderá empregar técnicas de produção permitindo seguir uma filosofia que auxilie a coordenar as diversas fases do sistema de manufatura. A seguir, apresentam-se algumas das técnicas de produção utilizadas atualmente nas indústrias.

2.2 Técnicas de Gestão da Produção

As técnicas de gestão da produção foram, ao longo do tempo, acompanhando a evolução dos processos produtivos e, em consequência foram surgindo várias técnicas que permitem atender a diferentes tipos de produção, conforme observado anteriormente. Segundo [COR93] e [SLA97], conceituam-se a seguir as técnicas comumente observadas nas indústrias: Kanban, Planejamento com Capacidade Finita, MRP/MRP II e OPT.

2.2.1 Kanban

Segundo [SLA97], o kanban é um método de regulação da produção utilizável principalmente quando se tem demanda constante de um item (ou conjunto de itens semelhantes, denominado família ou grupo) e um conjunto de equipamentos dedicados à produção dessa família. Esse método utiliza um sistema de realimentação visual, por cartões de demanda circulantes, denominados kanban.

A sua grande vantagem é o fato de realizar a própria regulação, que se integra ao ambiente de chão-de-fábrica, aumentando a responsabilidade e, portanto, a motivação das pessoas diretamente envolvidas com o sistema fabril.

Em sua forma mais simples, tem o seguinte funcionamento: um contentor (recipiente) com peças na entrada de um centro de trabalho **A** só pode realizar a respectiva atividade quando receber uma **licença**, ou **ordem de produção** que é materializada por um cartão, o **kanban**. Esse kanban, por sua vez, veio de um contentor de peças prontas do mesmo tipo (que já sofreram essa atividade), e que foram requisitadas pelo centro de trabalho seguinte, **B**. Em outras palavras, para que um centro de trabalho realize uma atividade sobre um lote de peças, é necessário que haja um kanban que autorize. Depois que as peças saíram do centro de trabalho **A**, o kanban fica preso ao contentor. Quando o centro de trabalho seguinte, **B**, começar a utilizar as peças, o kanban é liberado, e volta para **A**, autorizando a realização da atividade sobre mais um lote. Esse método, portanto, tem a

função de manter um estoque aproximadamente constante de cada item, entre dois centros de trabalho.

Em um sistema de manufatura que adota kanban, determinado centro de trabalho possui, próximo ao local em que são realizadas as atividades sobre as peças, um painel chamado porta-kanbans, que serve para colocar os cartões que autorizam a produção. Possui também um local onde é estocado o material necessário para fabricar as peças (área de entrada) e um local onde são estocadas as peças já concluídas em termos de quantidade de peças. Um contentor corresponde a um lote de transferência.

Em outras palavras, o kanban é uma solução simples, para sistemas fabris também simples. Às vezes o problema é simples por natureza, como por exemplo, linhas de produção dedicadas ao atendimento de uma única ou poucas peças.

Alguns requisitos são fundamentais para o bom funcionamento do kanban. Entre eles:

- o tempo de preparação (*setup*) das máquinas deve ser baixo;
- o centro de trabalho deve ser pouco compartilhado;
- o consumo deve ser homogêneo;
- deve existir uma padronização de itens;
- as flutuações devem ser baixas;
- baixo tempo de reposição.

Segundo [SLA97], cabe um comentário sobre o conceito de *just-in-time*, geralmente associado ao método kanban. Como objetivo principal, todas as técnicas de engenharia de produção são *just-in-time*, pois nenhuma pretende acumular estoques por generosidade: para cada caso há uma técnica adequada.

2.2.2 Planejamento com Capacidade Finita

Conforme [WAL90], no planejamento com capacidade finita os algoritmos localizam cada atividade de forma que não haja superposição com outra no mesmo recurso. São mais complexos e geram resultados melhores, pois, havendo capacidade suficiente, o plano é realizável por construção, tendo já sido resolvidos os conflitos pela alocação de recursos. Entretanto, nem sempre são viáveis e sua implantação depende:

- da existência de um modelo adequado;
- da existência de um algoritmo de capacidade finita para esse modelo.

A existência de um modelo adequado depende da existência de atividades com tempos conhecidos (inclusive *lead-time*). Isso já não ocorre, por exemplo, na fabricação de bens de capital volumosos, com alta participação de mão-de-obra. Nesse caso, a duração da atividade (*lead-time*) depende fortemente da quantidade de pessoas alocadas à tarefa. Essa decisão, por sua vez, dificilmente pode ser deixada a um algoritmo; a qualidade de sua decisão provavelmente seria inferior à que seria obtida por um planejador humano.

Mesmo que o algoritmo seja possível, ele ainda precisa ser realizado. Os algoritmos de capacidade finita dependem da existência de modelos bastante precisos dos sistemas de manufatura, e do desenvolvimento dos respectivos algoritmos. Nesse nível, os sistemas de manufatura são bastante heterogêneos, e o desenvolvimento muitas vezes tem que ser realizado caso a caso.

Os algoritmos de capacidade finita utilizam como parâmetro básico o tempo de trabalho, ou tempo padrão das atividades realizadas pelos recursos. O resultado pode ser avaliado através de uma função objetivo composta por um critério ou uma média ponderada de diversos critérios, como por exemplo: menor estoque, menor *lead-time*, etc. Espera-se com isso apenas um resultado ótimo ou bom, quando medido por determinada função-objetivo.

A distribuição das atividades obedece a um conjunto de restrições:

- a cada instante, um recurso pode realizar uma atividade, no máximo;
- cada atividade só pode ser iniciada quando suas atividades precedentes estiverem concluídas.

2.2.3 MRP / MRP II

O MRP (*Material Requirements Planning* ou Planejamento das Necessidades de Materiais) e sua extensão o MRP II (*Manufacturing Resource Planning*) constituem, provavelmente, as ferramentas mais difundidas para o planejamento da produção desde 1970.

Os principais objetivos desses sistemas são permitir o cumprimento dos prazos de entrega com a mínima formação de estoques, planejando as compras e a produção de itens componentes, para que ocorram apenas nos momentos necessários e nas quantidades também necessárias.

O princípio básico do MRP/MRP II é o cálculo das necessidades, das quantidades e dos momentos em que são necessários os recursos da manufatura (materiais, pessoas, equipamentos, etc.), para que se cumpram os programas de entrega de produtos com o mínimo de formação de estoque. Esse cálculo é feito a partir das necessidades dos produtos finais.

O MRP/MRP II é um sistema hierárquico, no qual os planos de produção de longo prazo ou agregados são sucessivamente detalhados até se chegar ao nível do planejamento de componentes e máquinas específicas.

Nesse sistema as quantidades a serem produzidas de cada componente final, nas diferentes datas, são determinadas através do Plano Mestre da Produção, representado abaixo na figura 2.1. O subsistema de Planejamento das Necessidades de Materiais, com base na Lista de Materiais, explode os itens finais do Plano Mestre de Produção,

estabelecendo as ordens de fabricação ou de compra a serem realizadas para cada peça componente ou produto final. No subsistema Planejamento das Necessidades de Capacidade, com base nas ordens de fabricação geradas pelo subsistema anterior, gera a programação do número de horas impostas a cada centro de trabalho para a execução das mesmas.

Os sistemas MRP/MRP II utilizam o *lead time* de cada componente final (previsto no Plano Mestre de Produção) e de seus respectivos itens componentes para a elaboração do processamento do Planejamento das Necessidades de Materiais. O *lead time* é o tempo que decorre desde a entrada das matérias-primas (de um item) na fila dos centros de trabalho de seu roteiro, até a saída de um lote desse item. Sendo que o tempo de espera nas filas depende da demanda.

Ao se estimar o *lead time* de fabricação de um item, é preciso valer-se de valores históricos. Esses valores muitas vezes se referem a uma situação de demanda média, não à demanda real. Quando a demanda é inferior à média estimada, se estará realizando cada atividade antes do momento necessário; portanto, ter-se-á excesso de estoques de produtos em processo. Quando a demanda é superior, o resultado é o atraso na data de entrega.

Como o *lead time* é um dado histórico, independente da situação de carga real do sistema de manufatura, o MRP/MRP II não indica a viabilidade de execução das ordens de fabricação pelos centros de trabalho existentes, o que pode invalidar o resultado.

O subsistema de Planejamento das Necessidades de Capacidade é responsável por determinar a programação do número de horas imposto a cada centro de trabalho da fábrica, necessário para a fabricação das ordens. Esse planejamento é estabelecido período a período, baseado nas horas de máquina e horas de mão-de-obra a serem consumidas para a fabricação de cada item componente dos itens finais. Também é levado em conta cada um dos centros de trabalho que façam parte do roteiro daquele item. O somatório de todas as necessidades impostas a cada centro de trabalho, para cada um dos itens usuários do mesmo, resulta na necessidade global de um centro de trabalho.

Logo após a execução da programação das necessidades de capacidade é analisada, de acordo com as limitações de cada centro de trabalho, se existe uma sobrecarga ou subcarga em algum deles, de maneira a verificar se o Planejamento das Necessidades de Materiais, e conseqüentemente, o Plano Mestre de Produção são realísticos. Se não for possível a execução de alguma(s) ordem(s) de fabricação e nenhuma solução (utilização de horas extras, atraso ou antecipação de ordens e fabricação) seja encontrada para tornar viável essa execução, o Plano Mestre de Produção deve ser alterado.

Existe uma variação de sistemas do tipo MRP/MRP II que utilizam baldes de tempo (*time buckets*) no planejamento da capacidade dos recursos. Nesse caso, é estabelecida a capacidade de cada recurso para cada balde de tempo. Esse processo garante que a capacidade de cada recurso de produção não será ultrapassada, passando a considerar cada centro de produção como tendo capacidade finita.

Mesmo que se utilize esse mecanismo de planejamento, as atividades de produção continuam a ser livremente distribuídas dentro do intervalo de tempo disponível pelo recurso. Assim sendo, o calendário permanece sendo agregado. Não existem restrições estabelecidas em função do escalonamento das atividades, o que significa dizer que, mesmo que o recurso disponha de capacidade, nada garante que uma determinada atividade de produção, necessária a determinado item, seja realizada no instante em que a mesma é requerida, para que haja um sincronismo com a sua atividade seguinte, e garanta o cumprimento de sua data de entrega.

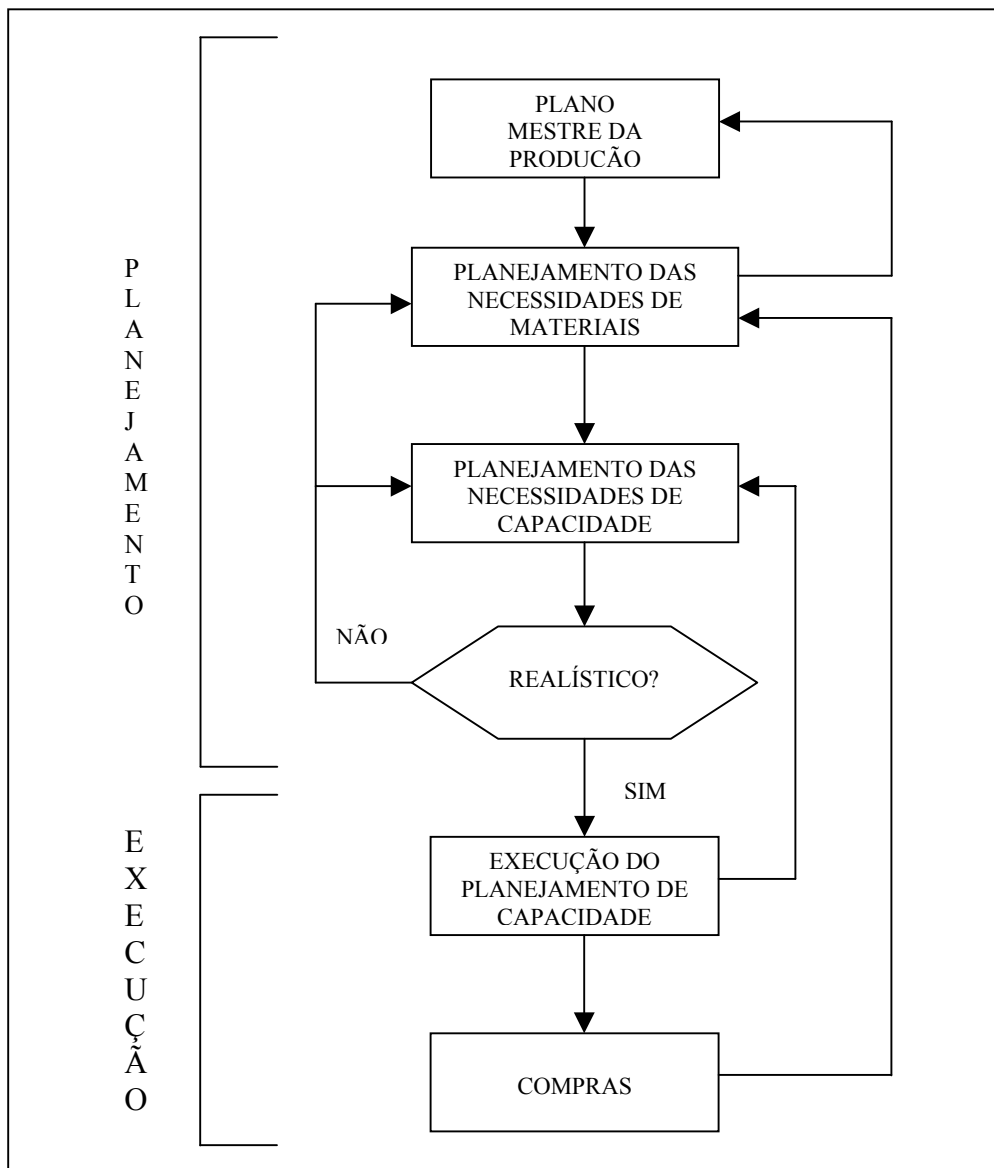


FIGURA 2.1 - Fluxo de Dados em um Sistema MRP

2.2.4 OPT

OPT significa *Optimized Production Technology* (Tecnologia de Produção Otimizada). É um método de gestão da produção criado nos Estados Unidos, em 1978, por E. Goldratt [SLA97]. Esse método é baseado na gestão de empresa a partir dos seus gargalos de produção. Um gargalo numa empresa pode ser definido como um recurso, uma máquina, um centro de trabalho cuja capacidade real de produção é inferior à sua demanda.

Segundo [SLA97], existem algumas regras a serem seguidas para o bom funcionamento do OPT:

- é necessário equilibrar o fluxo e não as capacidades, ou seja, é preciso ajustar o fluxo conforme a demanda;
- o nível de utilização de um não gargalo não é determinado pelo seu próprio potencial, mas sim por outros recursos do sistema;
- o conceito de utilização e a plena utilização de um recurso não são idênticos;
- uma hora perdida num recurso gargalo é uma hora perdida em todo o sistema;
- uma hora ganha num não gargalo não é mais do que um engano, pois quem determina o fluxo é o recurso gargalo;
- os gargalos determinam tanto as saídas como também os níveis de estoque;
- muitas vezes, o lote de transferência não deve ser igual ao lote de produção.

O OPT é um método que utiliza conceitos de outros métodos, como *just-in-time*, MRP, planejamento com capacidade finita. O OPT realiza uma análise da empresa baseada nos seus gargalos de produção.

O software OPT, cujo valor é considerado bastante elevado, baseia-se num sistema do tipo MRP, para posteriormente localizar os gargalos de produção por um algoritmo bastante sofisticado. Define, então, as condições da produção para a empresa a partir dos gargalos.

3 Escalonamento: Problemas e Soluções

3.1 Introdução

De acordo com [WAL88], o escalonamento das atividades, ou *scheduling*, é uma das atividades que compõe o planejamento da produção, nele é definida a ordem em que os lotes de peças serão fabricados. Ele deve alocar os recursos necessários para que as operações de produção desses lotes sejam executadas dentro dos prazos estabelecidos, conforme prioridades e também a custos viáveis.

Conforme [MOR93], a programação da produção, ou o escalonamento, envolve a consideração de uma série de elementos que disputam vários recursos por um período de tempo, e que possuem capacidade limitada. Os elementos a serem processados são chamados ordens de fabricação, ou *jobs*, e são compostos de partes elementares chamadas atividades ou operações.

Para [PIN99], o seqüenciamento ou escalonamento são formas de tomada de decisão, as quais possuem um papel crucial nas empresas, tanto de manufatura como de serviços. No atual ambiente competitivo, o efetivo escalonamento se tornou uma necessidade para sobrevivência no mercado. Empresas devem esforçar-se ao máximo para cumprirem as datas firmadas com seu clientes; o fracasso desse comprometimento pode resultar em uma perda significativa da imagem da empresa perante os clientes.

Escalonamento significa a alocação dos recursos limitados para tarefas sobre um determinado tempo. É um processo de tomada de decisão que tem como objetivo a otimização de um ou mais objetivos.

Os recursos e as tarefas podem ter diversas formas. Os recursos podem ser máquinas no chão-de-fábrica, pessoas em um local de construção, unidades de processamento em um ambiente computacional, etc. As tarefas podem ser operações em um processo de produção, etapas de um projeto de construção, execuções de um programa de computador, etc. Cada tarefa deve possuir um diferente nível de prioridade, uma data de início o mais cedo possível e uma data-limite. Os objetivos também podem ter diversas formas. Um possível objetivo é a minimização do tempo de conclusão da última tarefa, outro pode ser a minimização do número de tarefas finalizadas depois de seu prazo-limite.

A atividade de escalonamento da produção é uma das mais complexas atividades no gerenciamento da produção, segundo [SLA97]. Primeiramente, os programadores têm que lidar com diversos tipos de recursos simultaneamente. As máquinas terão diferentes capacidades, o pessoal terá diferentes habilidades. Dessa forma, o número de alternativas possíveis cresce rapidamente à medida que o número de atividades e processos aumenta. Por exemplo, se uma máquina possui cinco diferentes tarefas a processar, qualquer das cinco tarefas pode ser processada em primeiro lugar e pode ser seguida de qualquer uma das quatro tarefas restantes. Isso significa que se pode ter:

$5 \times 4 \times 3 \times 2 = 120$ combinações.

Ou seja, para n tarefas há $n!$ (n fatorial) maneiras de programação das tarefas em um processo simples.

Pode-se agora imaginar o impacto que teria se, na mesma situação, houvesse mais de um tipo de máquina. Ao tentar minimizar o número de preparações de duas máquinas, não haveria razão para a seqüência da máquina 1 ser a mesma da máquina 2. Ao considerar as duas tarefas de seqüenciamento independentes, para duas máquinas, haveria:

$120 \times 120 = 14.000$ programações possíveis de duas máquinas e cinco tarefas.

Ao se relacionar isso a uma situação real, onde podemos ter 120 tarefas em 40 máquinas, em roteiros que cada tarefa individual usa cinco máquinas diferentes, pode-se imaginar que a tarefa de programação da produção se torna realmente complexa. Dentre um vasto número de programações, há muitas opções aceitáveis, como há vários roteiros e seqüências apropriadas para qualquer conjunto de tarefas.

A tarefa de programação tem que ser repetida freqüentemente para permitir resposta às variações de mercado e às mudanças no *mix* de produtos. Lembre-se que, mesmo mudanças menores no *mix* dos produtos podem fazer com que os recursos limitadores de capacidade, dentro das instalações, mudem bastante em um tempo curto, assim, gargalos podem mover-se pela fábrica bastante rapidamente.

3.2 Programação para a Frente e para Trás

Conforme [SLA97], a programação para frente envolve iniciar o trabalho logo que ele chega. A programação para trás envolve iniciar o trabalho no último momento possível sem que ele tenha atraso.

A escolha entre a programação para frente ou para trás depende das circunstâncias. A tabela 3.1 mostra algumas vantagens e desvantagens de cada abordagem. Na teoria, tanto o Planejamento das Necessidades de Materiais (MRP – *Material Requirements Planning*) como o *Just in Time* (JIT) usam a programação para trás, somente começando a realizar as tarefas quando necessário. Na prática, os usuários do MRP tendem a alocar tempos demasiados para cada tarefa ser completada, por essa razão cada tarefa não é iniciada no último momento possível. Em contrapartida, o JIT começa em cima da hora.

TABELA 3.1 - Vantagens da programação para a frente e para trás

Vantagens da Programação para Frente	Vantagens da Programação para Trás
Alta utilização do pessoal – as pessoas sempre começam a trabalhar para se manterem ocupadas, mesmo que realizem atividades com muita antecedência.	Custos mais baixos com materiais – os materiais não são usados até sua data mais tarde de produção, reduzindo assim o custo de inventário na fábrica.
Flexível – as folgas de tempo no sistema permitem que trabalhos inesperados sejam programados, quando a demanda for menor que a capacidade.	Menos exposta a risco no caso de mudanças na programação pelo cliente.
	Tende a focar a operação nas datas firmadas com o cliente.

3.3 Programação Empurrada e Puxada

Em um sistema de planejamento e controle empurrado, as atividades são programadas por meio de um sistema central e completadas em linha com as instruções centrais, como em um sistema MRP. Cada centro de trabalho empurra o trabalho para o próximo centro de trabalho, sem levar em consideração se esse centro pode realizar o processamento. Na prática, há muitas razões pelas quais as condições reais diferem das planejadas. Como conseqüências: tempo ocioso, estoque e filas freqüentemente caracterizam sistemas empurrados. Nesse sistema a probabilidade de aumento do estoque entre os centros de trabalho é mais acentuada, pois um centro pode ter capacidade de produção diferenciada de outro.

Em um sistema de planejamento e controle puxado, a seqüência e as especificações do que é feito são estabelecidas pelo centro de trabalho do “consumidor”, que “puxa” o trabalho da célula de trabalho antecedente (fornecedor). O consumidor atua como sendo o gatilho do movimento. Se uma ordem (requisição) não é passada para trás, do consumidor para o fornecedor, o fornecedor não é autorizado a produzir nada ou mover qualquer material. Os sistemas puxados de programação são menos suscetíveis de resultar em criação de estoques entre os centros de trabalho no sistema de manufatura, pois somente será processado o que for solicitado.

3.4 Escalonamento num Ambiente *Job Shop*

Existem diversos modelos utilizados na solução de programação da produção intermitente do tipo *job shop*, incluindo tanto os que fornecem solução ótima, quanto os modelos heurísticos mais recentes, baseados em métodos de busca estendida.

O ambiente conhecido como *job shop* é geralmente caracterizado por permitir diferentes fluxos das ordens de produção entre as máquinas e diferentes números de operações por ordem, que são processadas apenas uma vez em cada máquina.

Um ambiente *job shop* geral pode ser caracterizado por uma matriz de transferência, conforme o exemplo da Tabela 3.2. Os valores das células representam a probabilidade de uma ordem (estando na máquina da linha) ter sua operação executada na máquina da coluna correspondente na tabela.

TABELA 3.2 - Matriz de ref. de um *job shop* geral composto de cinco máquinas [HAX 84]

		PARA					
		Maq 1	Maq 2	Maq 3	Maq 4	Maq 5	Saída
DE	Entrada	.3	-	.2	.1	.4	
	Maq 1			.3	-	.5	.2
	Maq 2	.1		-	.6	-	.3
	Maq 3	.5	-		.1	.1	.3
	Maq 4	-	.1	.2		-	.6
	Maq 5	.1	.1	.1	.6		.1

Os objetivos geralmente tratados no problema de escalonamentos podem ser resumidos no atendimento de prazos (datas de entrega), na minimização da utilização da capacidade disponível, ou mesmo na combinação desses.

De acordo com [HAX84], os problemas de *job shop scheduling* devem levar em consideração os seguintes aspectos:

- número de máquinas envolvidas: para uma ou duas máquinas, existem algoritmos simples que garantem a solução ótima para determinados objetivos. Nos demais casos, apenas a enumeração explícita ou implícita de todas as alternativas garante a solução ótima (problema NP-completo);
- quanto à forma de atendimento da demanda: o *job shop* pode ser classificado em aberto, fechado ou misto. No caso de *job shop* fechado, os produtos ou serviços são conhecidos e suas demandas podem, em princípio, ser previstas. A produção pode ser feita em lotes maiores do que os pedidos, sendo o excesso estocado (MTS, ou *make-to-stock*). No caso de *job shop* aberto, os produtos são fabricados sob encomenda, ou seja, fabricados de acordo com os pedidos de clientes (MTO, ou *make-to-order*). *Job shop* mistos combinam as características dos *job shop* abertos e fechados;
- quanto ao processo de chegada das ordens: podem ser estáticos, quando as ordens a serem programadas estão disponíveis no início da produção, ou dinâmicos, quando as ordens vão chegando ao longo do tempo. Chegadas dinâmicas podem ser determinísticas, quando os instantes de chegada são conhecidos antecipadamente, ou estocásticas, quando os instantes seguem uma distribuição de probabilidade;
- quanto ao tempo de processamento das ordens: classificam-se em determinísticos, quando os tempos de processamento das operações são fixos e conhecidos, ou estocásticos, caso em que os tempos de processamento das operações são representados por uma distribuição de probabilidade;

- quanto ao tempo de *setup*: classificam-se em dependentes ou independentes. Dependentes são aqueles nos quais o tempo de preparação da máquina, para executar uma operação, depende de qual operação tenha sido executada anteriormente. Independentes não dependem da operação anterior para realizar a preparação;
- quanto à existência de restrições de dependência entre as operações: classificam-se em tecnologicamente dependentes (devido à seqüência ou à espera) ou independentes. Operações tecnologicamente dependentes por seqüência são aquelas nas quais sua execução numa determinada seqüência torna-se desejável por critérios tecnológicos. Operações dependentes por espera são aquelas nas quais, efetuada determinada operação numa ordem, deve-se obrigatoriamente executar a próxima operação dentro de um determinado tempo preestabelecido.

Alguns desses fatores podem estar em conflito. O escalonamento deve ser capaz de julgar a importância de cada um desses itens, para que o plano gerado reflita de forma satisfatória as influências de cada um. Essas restrições atuam diretamente como fatores limitantes na escolha das operações, dos recursos e intervalos de tempo empregados no plano de escalonamento.

O escalonamento de produção deve também levar em consideração a dinâmica do ambiente de produção. Deve possuir mecanismos para tomada de decisão em tempo real, necessário para responder a perturbações ou imprevistos que ocorram durante a execução de um plano. Essas perturbações podem ser modificações de programas de produção, quebra de máquinas, atrasos na entrega de matéria-prima, etc.

Conforme [PAC99], o problema de *job shop scheduling* é um problema combinatorial, na maioria das vezes NP-completo (enumeração explícita ou implícita de todas as alternativas possíveis para garantir a solução ótima). Dessa forma, algoritmos otimizantes são computacionalmente viáveis quando aplicados a problemas reais pequenos, com objetivos limitados. Para problemas de porte similar aos encontrados no ambiente real, costuma-se sacrificar a obtenção de uma solução ótima por métodos subótimos heurísticos, com tempo computacional aceitável.

Os modelos podem ser classificados em duas classes: os modelos matemáticos (solução ótima) e os heurísticos, mas a escolha do modelo mais adequado à solução de problemas reais deve levar em consideração alguns aspectos, como: o número de máquinas envolvidas, os roteiros das ordens, o regime de chegada das ordens, a variabilidade dos tempos de processamento, entre outras características.

3.5 Modelos Matemáticos

Os modelos matemáticos empregam abordagens algorítmicas ou de otimização para realizar os escalonamentos. Seus métodos enfatizam a obtenção de resultados ótimos em função de algum parâmetro de desempenho. Esse pode ser, por exemplo, a minimização dos tempos de produção ou a maximização do uso dos recursos.

Devido ao grande número de soluções possíveis e à complexidade do problema de escalonamento, torna-se difícil e até impossível modelar todas as variáveis envolvidas no processo. Além disso, o tempo de resposta desses sistemas eleva-se consideravelmente, tornando impraticável a obtenção de soluções ótimas em tempos satisfatórios.

São usadas técnicas genéricas para a solução de problemas combinatoriais, sendo que essas técnicas são procedimentos enumerativos que não utilizam informação ou conhecimento de características específicas do problema para direcionar a busca.

Dentre os modelos de solução ótima enumerativos, a enumeração pode se dar explicitamente (todas as alternativas têm o resultado calculado) ou implicitamente (algumas combinações não são exploradas por possuírem resultado parcial deduzido como inferior a outro já obtido).

Entretanto, essa abordagem tende a ser aplicada em problemas bem específicos, onde simplificações podem ser introduzidas. Como exemplo, podem ser citados os métodos de *Branch and Bound*, *Johnson*, etc.

Outra solução encontrada nesses modelos são os algoritmos eficientes que particularizam-se como uma subclasse dos modelos otimizantes por gerarem soluções ótimas, sem a necessidade de enumeração de todas as alternativas. Infelizmente, poucos são os problemas cuja solução pode ser encontrada por algoritmos eficientes e, mesmo existentes, possuem hipóteses pouco aderentes aos problemas normalmente encontrados na realidade.

Os modelos matemáticos de otimização partem do princípio de que todas as hipóteses, sobre as quais forem construídos, devem estar de acordo com o mundo real. Na prática, essas hipóteses, baseadas em características determinísticas, não se aplicam à realidade do sistema produtivo.

Segundo [WAL90], existem limites na aplicabilidade dos modelos matemáticos otimizantes, tendo em vista a complexidade e magnitude do problema de escalonamento dos sistemas de produção. Observando os modelos definidos pela Pesquisa Operacional, como por exemplo: a programação linear e a programação inteira, verifica-se a sua não aplicabilidade devido a alguns aspectos, como:

- os sistemas de manufatura nem sempre apresentam comportamento linear;
- a quantidade de equações para modelá-los atinge volumes enormes;
- a função objetivo é de difícil expressão;
- o tempo de resolução do sistema de equações seria proibitivo.

A seguir, apresentam-se dois métodos para a solução de problemas de escalonamento:

Método *Branch and Bound*: Esse modelo utiliza uma estrutura de busca em árvore para a construção da solução, analisando, a cada ramificação, quais os nós que são qualificados, isto é, apresentam um limite inferior da solução (*lower-bound* ou LB), inferior ao limite superior (*upper-bound* ou UB).

O *upper bound* representa a solução inicial, ou a melhor solução completa já encontrada. Os nós que não atendem a esse requisito são descartados, eliminando-se todas as ramificações deles decorrentes. Os cálculos dos LB costumam ser simplificados. Os nós qualificados são desdobrados em novas ramificações, de modo a se buscar soluções completas melhores que o UB. Caso sejam encontradas, o valor UB é atualizado e a busca prossegue, verificando-se implicitamente todas as alternativas, até que nenhum nó possua valor LB inferior ao UB atual. Maiores detalhes podem ser vistos em [MOR93].

Método de *Johnson* [SLA97]: método que pode ser utilizado na programação de n tarefas, em dois centros de trabalho. O critério de otimização é a minimização da extensão total de tempo, também chamado de *makespan*, que é o intervalo entre o início da primeira atividade sobre a primeira peça e o final da última atividade sobre a última peça. A figura 3.1 ilustra o uso dessa regra. No exemplo abaixo, temos dois centros de trabalho com seis tarefas a serem processadas.

A regra é simples, primeiro busque o menor tempo de processamento, se esse tempo está associado ao primeiro centro de trabalho, programe essa tarefa como sendo a primeira, ou a mais perto possível do primeiro lugar. Se o próximo dos menores tempos está associado ao segundo centro de trabalho, programe essa tarefa por último, ou o mais perto possível do último.

Assim que uma tarefa tiver sido programada, elimine essa tarefa da lista. Continue alocando trabalhos até que a lista esteja completa. Nesse exemplo, o menor tempo de processamento é 30 minutos para o processamento da tarefa B. Devido a essa tarefa estar no primeiro centro, a tarefa B é designada para a primeira posição na programação. O seguinte menor tempo de processamento é 35 minutos para a tarefa D, no segundo centro de trabalho. Devido a essa tarefa estar no segundo centro de trabalho, ela é programada para ser a última da fila. O seguinte menor tempo depois das tarefas B e D é 41 minutos da tarefa A, no segundo centro de trabalho. Devido a mesma estar no segundo centro de trabalho, ela é programada o mais perto possível da última que, em nosso exemplo, será o quinto lugar. Esse processo continua até que todas as tarefas tenham sido programadas. A programação resultante é demonstrada na figura 3.1.

TABELA 3.3 - Aplicação da regra de Johnson para programar seis tarefas em dois centros de trabalho

Tarefas	Tempo de processo do primeiro centro	Tempo de processo do segundo centro
A	55	5° 41
B	1° 30	60
C	60	4° 53
D	52	6° 35
E	2° 45	55
F	3° 48	65

Ordem da Fila	B	E	F	C	A	D

A idéia principal é que as tarefas com menor tempo de processamento fiquem associadas ao primeiro centro de trabalho, e sejam ordenadas por primeiro na fila, enquanto as tarefas com menor tempo de processamento, associadas ao segundo centro de trabalho, sejam ordenadas no fim da fila. A justificativa para essa otimização é inicialmente alocar as tarefas curtas para o primeiro centro, para que o segundo centro logo tenha o que fazer e, no fim, deixar tarefas curtas para o segundo centro, para que ele termine pouco antes do primeiro centro. Dessa maneira, maximiza-se o paralelismo do trabalho e minimiza-se o *makespan*.

3.6 Modelos Heurísticos

O modelo heurístico, que consiste em regras práticas (indutivas) de escalonamento, permite resolver problemas grandes em tempos de computação viáveis. Tais métodos são ditos satisfatórios, pois não garantem que a solução encontrada seja a ótima, mas muito próxima desta.

Usando conhecimentos ou experiências heurísticas previamente obtidas de especialistas humanos, é possível restringir o número de soluções alternativas até que seja encontrada uma única solução viável. Como exemplo, pode-se citar o algoritmo de Palmer [PAL65].

Citam-se a seguir, alguns exemplos de modelos heurísticos de interesse para este trabalho:

Redes Neurais: são heurísticas que utilizam uma malha de uma ou mais camadas de nós para a tomada de decisões. Cada nó fornece um valor de resposta de acordo com a sua função de ativação (cujos parâmetros controlam sua sensibilidade), e com o estímulo

recebido (valor de entrada, que é obtido pelo somatório das respostas dos nós antecedentes) [ZAH91].

Seu valor de resposta, por sua vez, influencia o comportamento dos nós sucessores. O comportamento de cada nó, explicitado pelo valor dos parâmetros que controlam a sensibilidade da função de ativação, é aprendido pela malha, utilizando-se um conjunto de problema-resposta. Uma vez treinada a malha e ajustados os parâmetros, o processo de tomada de decisões é simples, em termos de cálculos.

Sistemas Especialistas: método heurístico e interativo que visa decidir qual a melhor seqüência a ser executada, dependendo das condições atuais da produção. Esse método mantém armazenadas, em bases de dados de conhecimento, também chamadas *knowledge bases*, a configuração, as restrições do problema, bem como um conjunto de regras lógicas a serem utilizadas na decisão. A configuração e as restrições do problema dizem respeito, por exemplo, aos tempos das operações, ao número e as características das máquinas, restrições tecnológicas de produção, etc. Regras lógicas traduzem a forma de raciocínio do especialista (tamanho médio de lotes, preferências de seqüências de operações, etc.) que são muitas vezes de difícil compreensão e formalização[HAY83].

Bases de conhecimento são estruturas complexas, geralmente compactas, modulares e hierarquizadas. Os módulos de dados armazenados nas bases são denominados *frames*.

Dessa forma, é possível imitar as regras utilizadas por especialistas humanos. O algoritmo computacional, que seleciona hierarquicamente quais regras devem ser utilizadas numa determinada condição, é chamado motor de inferência. *Shells* são ambientes que facilitam a construção de bases de conhecimento e sua varredura (por motores de inferência). Algumas formas de se obter regras, ou seja, aquisição de conhecimentos, são [MOR93]:

- estudar a forma de trabalho de um tomador de decisões humano. Seu raciocínio é formalizado em regras que são incorporadas ao sistema especialista após serem validadas pelo tomador de decisões humano;
- uma série de regras é proposta para o problema por especialistas em pesquisa operacional, implementadas pelo construtor de software e validadas pelo tomador de decisões humano;
- o construtor do sistema simplifica heurísticas existentes, criando regras simples, tal como o tomador de decisão humano provavelmente faria. Tais regras são validadas pelo mesmo.

Vistos sob o aspecto combinatorial do problema de escalonamento, os sistemas especialistas efetuam uma busca na árvore de alternativas, auxiliados por regras que descartam e priorizam seqüências, conforme o estado do sistema.

Métodos de Gargalo: heurísticas que se concentram nos gargalos (*bottlenecks*) do sistema, ou seja, nos recursos limitantes de capacidade para a determinação das seqüências

das ordens nas máquinas. É um método bastante utilizado em problemas cujo objetivo é a minimização de tempo total de processamento.

Esse tipo de sistema tem como seu principal representante os sistemas OPT (*Optimized Production Technology*). Um dos princípios fundamentais no OPT é que, uma hora perdida na atividade-gargalo é uma hora de perda para o sistema como um todo, o que significa dizer que, quando uma atividade-gargalo está funcionando em plena carga, qualquer parada da mesma causará perdas no conjunto da fábrica de forma diretamente proporcional ao número de horas não utilizadas.

O primeiro passo no sistema OPT, quanto à programação da produção, é modelar o processo produtivo como se fosse uma rede, contendo todos os recursos, os dados relativos à produção e os dados da demanda. Logo após, são identificados os recursos ou as atividades gargalo. A rede então é dividida em duas áreas, uma com os recursos críticos, representados pelos gargalos, e outra pelos recursos não-críticos.

O planejamento é realizado em função dos recursos críticos e o algoritmo do OPT resulta em uma programação otimizada, considerando as restrições de capacidade dos recursos críticos.

A principal desvantagem do OPT é o fato de ele não possuir memória. De fato, a cada execução do algoritmo é feito um replanejamento completo. Desse modo, os sucessivos replanejamentos podem apresentar resultados totalmente distintos, ocasionando constantes modificações no chão-de-fábrica.

De acordo com [WAL89], nos modelos heurísticos quando há mais de uma atividade disputando um recurso, cabe a uma regra selecionar a atividade a atender primeiro, de forma a otimizar o programa de produção, de acordo com algum critério.

Para os *job shops*, as regras mais adequadas são as do tipo conflito, pois as decisões são tomadas localmente, no nível de atividade, e não no nível de peça. Ou seja, a cada instante, haverá para cada recurso um grupo de produtos a serem processados, cabendo à heurística decidir qual produto deve ser selecionado primeiro.

De modo geral, as regras podem ser classificadas em dois tipos:

- regras que privilegiam o critério de *makespan* mínimo. Sendo que *makespan* é o intervalo de tempo que decorre do início da primeira peça até o final da última atividade sobre a última peça. Minimizar esse intervalo corresponde a maximizar o paralelismo;
- regras que privilegiam o critério do tempo do fluxo médio (*average flow-time*), que é a média dos tempos de permanência de cada peça dentro do sistema de manufatura.

Para minimizar-se o *makespan* utilizam-se regras que priorizam as peças que têm ainda mais trabalho pela frente. Começando-se antes o trabalho mais longo, evita-se, assim,

que ao final do cronograma sobrem “restos” de trabalhos longos, que estendem o tempo total.

Para minimizar-se o tempo de fluxo médio, utilizam-se regras que dão prioridade às peças com tarefas curtas, de modo a liberá-las o mais cedo possível do sistema de manufatura, diminuindo a média de permanência das peças.

As regras heurísticas são baseadas em critérios, dentre eles:

- tempo de processamento;
- data de entrega;
- número de operações;
- custo;
- instante de chegada;
- máquina ou tráfego;
- montagem.

As regras heurísticas baseadas no tempo de processamento caracterizam-se por escolher tarefas com o ciclo de processamento menor ou maior e também com o menor ou maior número de tarefas ainda não concluídas. Na tabela 3.3 estão representados alguns exemplos dessas regras.

TABELA 3.4 - Regras heurísticas baseadas no tempo de processamento

Regra	Descrição da Regra
SPT	<i>Shortest Processing Time</i> – escolhe a atividade mais rápida, ou seja, com o menor tempo de processamento. O algoritmo de Johnson citado anteriormente é um caso particular.
LI	<i>Largest Imminent Operation Time</i> – escolhe a atividade com o maior tempo de processamento.
MWR	<i>Most Work Remaining</i> – seleciona a peça com a maior soma de durações operacionais ainda não efetuadas. Essa regra privilegia a minimização dos atrasos de entrega das peças.
LWR	<i>Least Work Remaining</i> – seleciona a peça com a menor soma de durações operacionais de tarefas ainda não efetuadas. A regra SPT possui resultado semelhante e é mais utilizada.

As regras heurísticas baseadas na data de entrega possuem maior funcionalidade para a produção do tipo MTO e ETO. As datas são muitas vezes geradas por outros algoritmos, como por exemplo o MRP. Na tabela 3.4 estão representados alguns exemplos destas regras.

TABELA 3.5 - Regras heurísticas baseadas na data de entrega

Regra	Descrição da Regra
EDD	<i>Earliest Due Date</i> – seleciona a atividade cujo produto tem a data de entrega mais cedo.
MST	<i>Minimum Slack Time</i> – seleciona a atividade do produto que tem a menor folga, sendo que a folga é a diferença entre o tempo que falta para a data de entrega e o tempo requerido para executar as atividades que restam.
ODD	<i>Earliest Operation Due Date</i> – seleciona a atividade que possuir o instante previsto de término mais próximo. Sendo que esse instante foi previamente calculado por outro algoritmo, como por exemplo: o MRP.
OST	<i>Operation Slack Time</i> – seleciona a atividade que possui a menor folga, sendo que seu instante previsto de conclusão foi determinado através de outro algoritmo.

As regras heurísticas baseadas no número de operações caracterizam-se por escolher tarefas com o menor ou maior número de operações ainda não finalizadas. Na tabela 3.5 estão representados alguns exemplos dessas regras.

TABELA 3.6 - Regras heurísticas baseadas no número de operações

Regra	Descrição da Regra
MOR	<i>Most Operations Remaining</i> – escolhe a atividade pertencente ao produto com o maior número de tarefas ainda não executadas.
LOR	<i>Least Operation Remaining</i> – escolhe a atividade pertencente ao produto com o menor número de tarefas ainda não executadas.

As regras heurísticas baseadas no custo se caracterizam por escolher tarefas com o maior custo ou lucro. Estas regras são muito utilizadas por empresas que vinculam em seus contratos cláusulas de multa por atraso dos pedidos. Na tabela 3.6 estão representados alguns exemplos destas regras.

TABELA 3.7 - Regras heurísticas baseadas no custo

Regra	Descrição da Regra
VALUE	Seleciona a atividade que implica o maior valor adicionado, ou lucro.
1/C	Seleciona a atividade que pertence ao produto com maior custo unitário por penalidade de atraso.

As regras heurísticas baseadas no tempo de *setup* ou preparação das máquinas se caracterizam por escolher as tarefas onde o produto possua o menor *setup*, nenhum *setup* ou produto anterior similar, minimizando assim o tempo de preparação. Essas regras podem ser bastante satisfatórias para as máquinas consideradas como gargalos no sistema produtivo. Na tabela 3.7 estão representados alguns exemplos dessas regras.

TABELA 3.8 - Regras heurísticas baseadas no tempo de *setup*

Regra	Descrição da Regra
NSUT	Seleciona a atividade cujo produto não requer <i>setup</i> .
MINSEQ	Seleciona a atividade cujo produto possui o menor tempo de <i>setup</i> .
NB	Seleciona a atividade que possuir o menor tempo de <i>setup</i> relativo à atividade que terminou.

As regras heurísticas baseadas no instante de chegada ou aleatória caracterizam-se por escolher tarefas de acordo com sua chegada no sistema de produção. Na tabela 3.8 estão representados alguns exemplos dessas regras.

TABELA 3.9 - Regras heurísticas baseadas no instante de chegada ou aleatórias

Regra	Descrição da Regra
FIFO	<i>First In, First Out</i> – a primeira atividade que chegar, é a primeira a ser atendida.
RANDOM	Seleciona aleatoriamente a tarefa a ser executada.
LIFO	<i>Last In, First Out</i> – a última atividade que chegar, é a primeira a ser atendida.
FASFO	<i>First at Shop, First Out</i> – a atividade do primeiro produto a entrar na fábrica é a primeira a ser selecionada.

As regras heurísticas baseadas nas máquinas ou no fluxo das atividades caracterizam-se por procurarem escolher o tráfego no sentido de ocupar preferencialmente as máquinas mais livres. Essas regras tentam evitar a formação de filas extensas. Na tabela 3.9 estão representados alguns exemplos dessas regras.

TABELA 3.10 - Regras heurísticas baseadas na máquina

Regra	Descrição da Regra
NINQ	Seleciona a atividade na qual a atividade seguinte vai para a máquina com a menor fila.
WINQ	Seleciona a atividade na qual a atividade seguinte será realizada na máquina com o menor tempo de processamento na fila.

As regras heurísticas baseadas na montagem se caracterizam por escolher as tarefas dos produtos que possuem montagem, pois os atrasos na montagem implicam em maiores estoques intermediários e também as peças já prontas ficarão aguardando até a chegada das demais para a montagem final do produto. São normalmente combinadas com outras regras já citadas anteriormente, dando-se prioridade as atividades de montagem, não mais existindo tarefas para montagem, utilizam para desempate as regras FASFO, FIFO, LIFO out SPT. Na tabela 3.10 estão representados alguns exemplos destas regras.

TABELA 3.11 - Regras heurísticas baseadas na montagem

Regra	Descrição da Regra
ASMF-RAN	<i>Assembly Jobs First - Random</i> – os produtos de montagem são selecionados primeiro. Os demais são selecionados aleatoriamente.
ASMF-FASFO	Os produtos de montagem são selecionados primeiro. Os demais são selecionados pela regra FASFO.
ASMF-FIFO	Os produtos de montagem são selecionados primeiro. Os demais são selecionados pela regra FIFO.
ASM-LIFO	Os produtos de montagem são selecionados primeiro. Os demais são selecionados pela regra LIFO.
ASMF-SPT	Os produtos de montagem são selecionados primeiro. Os demais são selecionados pela regra SPT.

4 Desenvolvimento de Sistemas de Escalonamento

Segundo [PIN99], o escalonamento é um processo de tomada de decisão que existe em muitos sistemas de manufatura e produção, bem como em muitos ambientes de processamento da informação. Também são encontrados em outros tipos de empresas, como de serviços, sistemas de transporte e distribuição.

O escalonamento possui dificuldades tanto do ponto de vista técnico como de sua implementação. As dificuldades encontradas no aspecto técnico são similares às dificuldades encontradas em outros ramos da otimização combinatorial e em modelos estocásticos. As dificuldades encontradas sobre a visão da implementação são de um tipo completamente diferente e estão relacionadas com os problemas de escalonamento do mundo real e da recuperação dessas informações.

4.1 O Papel do Escalonamento dentro de uma Organização

Em um sistema de manufatura, as ordens devem ser atualizadas e devem ser interpretadas como tarefas com datas de entrega associadas. As tarefas devem ser freqüentemente processadas pelas máquinas de um centro de trabalho em uma determinada ordem ou seqüência. As tarefas devem aguardar para serem processadas sobre as máquinas que estão ocupadas, e alocações imediatas devem acontecer quando tarefas prioritárias chegarem às máquinas. Para que o escalonamento detalhado das tarefas seja executado em um sistema de produção, é necessário manter a eficiência e o controle de operações.

Uma função importante do escalonamento é servir de interface com diversas outras funções importantes da organização. Ele é afetado pelo processo de planejamento da produção, tanto a médio como a longo prazos de planejamento para toda a organização. Esse processo deve considerar níveis de estoque (inventário), previsões e necessidades de recursos para otimizar a diversidade dos produtos e a alocação dos recursos por um longo tempo. Decisões que são executadas por essa função de planejamento devem ter um impacto sobre o escalonamento. O escalonamento também recebe informações do controle de chão-de-fábrica. Eventos inesperados sobre o chão-de-fábrica, como máquinas paradas ou tempos de processamento longos, que são antecipados, devem ser levados em conta, porque eles podem causar um impacto considerável sobre o programa de produção atual [PIN99].

A figura 4.1 representa o fluxo das informações em um sistema de manufatura.

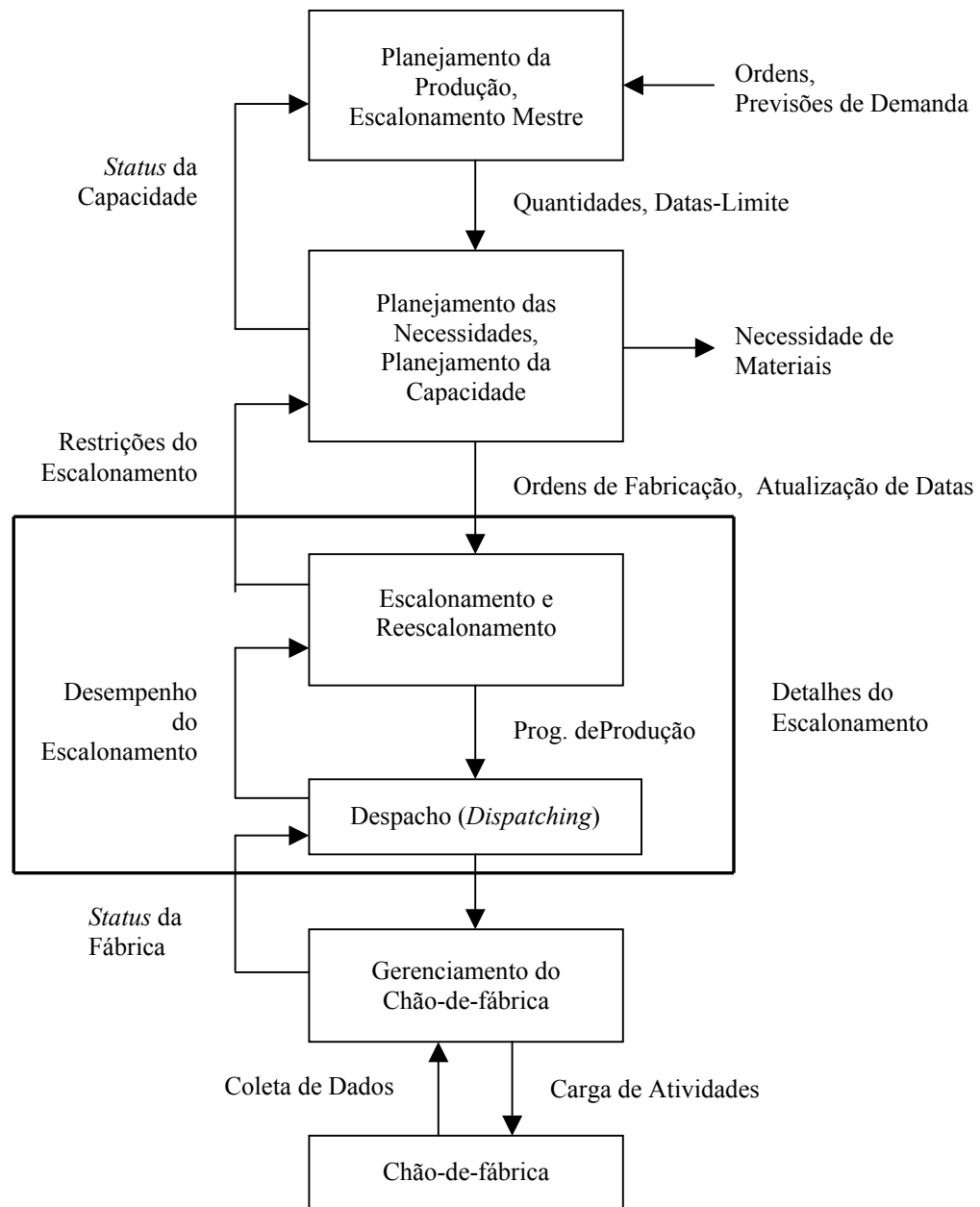


FIGURA 4.1 - Fluxo das informações em um sistema de manufatura [PIN99]

A maioria das empresas possui um computador central e um banco de dados centralizado. Conectado a esse computador central existem redes locais de computadores pessoais, estações de trabalho e terminais de entrada de dados, que podem ser usados para recuperar dados do banco de dados ou entrar com novos dados. A função do escalonamento normalmente é realizar sobre um computador pessoal ou em uma estação de trabalho a comunicação da fábrica com o computador principal.

Computadores em locais estratégicos devem ser conectados ao computador de escalonamento para prover informações aos departamentos envolvidos no processo. Eles também devem fornecer informações relevantes ao sistema de escalonamento, como por exemplo, o estado das máquinas, as alterações nas datas das atividades, as atividades em atraso, etc.

4.2 Projeto, Desenvolvimento e Implementação de Sistemas de Escalonamento

O sistema de escalonamento tem interação com diversos outros sistemas da organização. Ele necessita informações de um sistema de planejamento da produção de alto nível, onde as ações são apropriadas para médio ou longo prazos, com respeito à troca de escalonamentos, manutenção preventiva de escalonamento, etc [PIN95].

Também pode receber informações de um sistema MRP (*Material Requirements Planning*) para saber as datas limite das ordens de fabricação. O sistema de escalonamento também se relaciona com o sistema de chão-de-fábrica para receber informações atualizadas sobre a disponibilidade das máquinas, sobre o andamento das atividades, etc.

O próprio sistema de escalonamento consiste de diferentes módulos. Entre os mais importantes:

- o banco de dados e a base de conhecimento;
- a geração do escalonamento e do reescalonamento;
- a interface com o usuário.

Um módulo gerenciador do banco de dados deve ter a capacidade para manipular os dados, várias formas de análises estatísticas de desempenho devem ser disponibilizadas ao usuário, através de alguma interface gráfica, e também os dados devem ser representados na forma de gráficos. Alguns sistemas, como os citados no capítulo cinco possuem a base de conhecimento separada do banco de dados corporativo, montada especificamente para a realização do escalonamento, mas na maioria dos sistemas a base de conhecimento do escalonamento está incorporada ao sistema da empresa, juntamente com o módulo de escalonamento [PIN99].

Um módulo de escalonamento envolve a formulação de um modelo satisfatório com funções objetivas, restrições, regras, bem como com os algoritmos e heurísticas.

O módulo de interface com o usuário é muito importante, especialmente durante o processo de implementação. Sem uma interface amigável e de fácil manuseio, o sistema tem uma grande probabilidade de não ser usado satisfatoriamente pelos usuários. A interface deve oferecer gráficos na forma de Gantt [PIN95]. O gráfico de Gantt é um tradicional gráfico de barras horizontais (inventado por H. L. Gantt em 1917), onde o eixo x representa o tempo e o eixo y representa neste caso as diversas máquinas do escalonamento. Uma cor ou um padrão de código pode indicar uma característica ou um atributo da atividade correspondente; por exemplo, atividades que são finalizadas após a

sua data limite no escalonamento atual são mostradas em vermelho no gráfico. Também é importante que a interface ofereça tabelas que habilitem o usuário a editar o escalonamento gerado pelo sistema, permitindo assim adicionar informações de última hora.

Dentro do banco de dados uma distinção pode ser feita entre dados estáticos e dinâmicos. Dados estáticos são todos os dados das atividades e máquinas que não dependem do escalonamento. Nesses tipos de dados, podem ser citados a quantidade a ser produzida na ordem de fabricação, a data limite de entrega, o tempo total de produção da ordem e possíveis restrições no processo. As prioridades das atividades também são consideradas como dados estáticos que não dependem do escalonamento. Os dados estáticos das máquinas incluem a velocidade das máquinas, o tempo de manutenção das máquinas, etc.

Os dados dinâmicos são todos os dados que dependem do escalonamento como, por exemplo, o tempo de início e fim das atividades (ordens), o tempo inativo das máquinas, o tempo que a máquina está gastando com a preparação (*setup*), as seqüências de tarefas que serão processadas sobre as máquinas, o número de tarefas que estão em atraso, etc.

A função de calendário é normalmente parte integrante do sistema de banco de dados; ela possui informações cruciais para o escalonamento, como os feriados, os dias de manutenção das máquinas, o número de trocas disponíveis, etc.

O modelo de uma base de conhecimento, em contraste com o modelo de um banco de dados, tem um impacto importante sobre a arquitetura global do sistema. O aspecto mais importante da base de conhecimento é a representação do conhecimento.

Um das formas de se representar o conhecimento é através de regras. Existem diversos formatos de declarar regras. Um formato comum é através da declaração *se-então* (*if-then*), que significa dizer: *se* uma determinada condição for satisfeita, *então* uma ação específica tem que ser realizada. É fácil codificar uma regra na maioria das linguagens de programação, como Fortran, Pascal, C, C++, VisualBasic, Delphi, etc.

Exemplo:

```
IF um vôo não tem continuação
  THEN o vôo deve ser designado para um portão remoto
```

Outra forma de declarar regras é através da lógica de predicados (*predicate logic*), que é baseada no cálculo proposicional. Uma linguagem de programação que produz regras nesse formato é o Prolog.

Exemplo:

```
PORTAOOK(P,V):-not(coninue(V)),portao-remoto(L),pessoas(P,V)
```

Uma segunda forma de representação do conhecimento é através de *frames* ou esquemas de *frame*, ou também esquema que prove uma representação estruturada de um objeto ou classe de objetos.

Exemplo: Um esquema que representa uma restrição de data de entrega genérica de uma tarefa.

```

    {{DUE DATE CONSTRAINT
      {IS=A CONTINUOUS-CONSTRAINT
        DOMAIN:dates
        PIECE=WISE=LINEAR=UTILITY: cost function }}}
  
```

Cientistas e pesquisadores em inteligência artificial têm experimentado outras formas de representação do conhecimento. O uso de redes neurais, que com o passar do tempo vão aprendendo com o sistema, pode ser classificado como a terceira forma de representação do conhecimento [SMI90].

4.3 Módulo de Escalonamento

Técnicas atuais de geração de escalonamentos estão unindo o pensamento de várias universidades nos dias de hoje. Determinada universidade que possui o ensino voltado para engenheiros industriais e pesquisa operacional se referencia predominantemente por algoritmos matemáticos para resolver o problema de escalonamento. Em outra, que possua seu ensino voltado para a ciência da computação e inteligência artificial, prevalece o uso de bases de conhecimentos e técnicas de inteligência artificial [PIN99].

Recentemente essas duas linhas de pesquisa estão convergindo para a solução do problema. Recentemente, no desenvolvimento de alguns sistemas híbridos, foram utilizadas bases de conhecimento e heurísticas razoavelmente sofisticadas. Determinados segmentos de empresas são mais voltados para o uso de técnicas algorítmicas, com cálculo matemático. Outros segmentos são projetados para a visão de bases de conhecimento para gerar o escalonamento.

Tanto a solução baseada em técnicas algorítmicas matemáticas quanto a técnica usando bases de conhecimento possuem vantagens e desvantagens na sua utilização.

A abordagem algorítmica possui vantagens quando:

- o problema permite uma fórmula matemática precisa;
- o número de tarefas envolvidas é grande;
- a quantidade de alternativas no ambiente é mínima;
- algumas formas de otimização têm que frequentemente ser feitas em tempo real;
- as regras gerais estão sendo seguidas constantemente, sem muitas exceções.

A desvantagem dessa abordagem é que, se o ambiente de escalonamento (as tarefas e máquinas) for muito variável, a reprogramação pode ser significativa.

A solução baseada em bases de conhecimento deve ser utilizada somente se for praticável do ponto de vista do tempo de resposta. Alguns desenvolvedores de sistemas acreditam que mudanças no ambiente do escalonamento ou regras podem ser incorporadas mais facilmente em sistemas de bases de conhecimento do que os baseados em algoritmos com soluções matemáticas. Outros, entretanto, acreditam que o esforço requerido para modificar qualquer sistema é principalmente em função de como o algoritmo foi documentado e organizado, assim, o esforço para realizar a modificação não depende da técnica utilizada.

A principal desvantagem de se usar bases de conhecimento é que, para se obter um escalonamento razoável, pode-se levar um tempo substancial de processamento em relação à técnica algorítmica.

O tempo de processamento deve ser um item primordial a ser levado em conta para a escolha da técnica de geração do escalonamento. Muitas aplicações precisam de informações em tempo real, outras possuem intervenção humana no processo de escalonamento, necessitando que o computador processe o escalonamento em segundos ou minutos.

Como foi dito anteriormente, essas duas linhas estão convergindo, e muitos sistemas atualmente têm sido desenvolvidos usando ambos os elementos. Uma linguagem muito utilizada atualmente para a programação é o C++, pois é uma linguagem fácil para codificar procedimentos algorítmicos e também possui extensões para orientação a objetos.

4.4 Interface com o usuário

As interfaces utilizadas com o usuário são uma parte muito importante do sistema. Essas interfaces podem determinar se o sistema vai ser usado ou não. As interfaces que utilizam janelas amigáveis que permitam ao usuário interagir com o sistema, têm grandes chances de se tornarem altamente produtivas.

O usuário deve ter a possibilidade de visualizar o escalonamento gerado de diversas maneiras através das interfaces do sistema. As interfaces para manipulação do escalonamento determinam o nível básico de um sistema, pois elas são usadas extensivamente pelo usuário do sistema. As diferentes formas das interfaces de manipulação do escalonamento dependem do nível de detalhe exigido, bem como do horizonte de planejamento a ser considerado.

Quatro tipos de interfaces são mais usadas :

- a interface com gráfico de Gantt;
- a interface com a lista de despacho (*dispatching list*);
- a interface com as capacidades de tempo das máquinas e centros de trabalho;

- a interface com o diagrama de processamento.

4.5 Aplicações Genéricas Versus Específicas

Sistemas de escalonamento como os citados no próximo capítulo podem ser implementados em diferentes indústrias com algumas modificações. Observa-se, no entanto, que o esforço envolvido para a personalização de tais sistemas pode ser bastante substancial. O código desenvolvido para a customização do sistema e a implementação de novos algoritmos pode não justificar a sua utilização.

Por outro lado, alguns sistemas possuem configurações muito sofisticadas que lhes permitem ser customizados a diferentes tipos de indústrias sem muito esforço de programação. Esses sistemas são altamente modulares e possuem interfaces que permitem ajustes específicos para determinadas empresas [PRO2001], [PRE2001].

Nos sistemas genéricos, quando possuem capacidade de modularização, a inclusão de um algoritmo especial de escalonamento também é permitida. Especialistas podem desenvolver esse algoritmo, e o software de escalonamento fornece as interfaces necessárias para o bom funcionamento do mesmo.

Diversos sistemas genéricos de escalonamento são desenvolvidos sobre banco de dados comerciais, e com isso podem ser conectados com os sistemas de MRP utilizados pelas empresas [CAP2000]. Dessa forma, o sistema de escalonamento pode realizar alterações nas datas das ordens de produção e informações que sejam úteis ao sistema de escalonamento.

Estes sistemas genéricos possuem rotinas de escalonamento para gerar um primeiro escalonamento das atividades, baseado no disparo de regras. O escalonamento gerado, então, fica disponível para o usuário, que raramente se satisfaz com o seqüenciamento apresentado. A alteração do escalonamento se faz pela liberação de interfaces na qual o usuário interage manualmente com o sistema para alterar a seqüência apresentada.

Alguns sistemas genéricos possuem procedimentos de programação mais complicados, como realizar o escalonamento para a frente (*forward loading*) ou para trás (*backward loading*), como foi citado anteriormente no capítulo 3. O escalonamento para a frente pressupõe que as ordens serão inseridas pelas datas limite, começando no início do escalonamento. Escalonamento para trás insinua que a seqüência é gerada a partir da última data limite de entrega, retrocedendo até a primeira data limite.

Também existem sistemas que primeiro identificam os centros de trabalho ou máquinas gargalo, calculam o tempo gasto para o processamento das atividades nessas máquinas e então escalonam as atividades nessas máquinas, através de algum algoritmo baseado em regras preestabelecidas. Após o escalonamento nas máquinas gargalo, o sistema escalona o restante das ordens nas demais máquinas, utilizando o escalonamento para frente ou para trás, como já foi citado anteriormente [WAL90].

Segundo [PIN 99], sistemas genéricos possuem grandes vantagens sobre sistemas específicos. Se um problema de programação é bastante tradicional (*standard*) e somente uma pequena customização do sistema genérico basta, então essa opção normalmente é menos cara que o desenvolvimento de um sistema específico. Outra vantagem é que uma empresa estabelecida comercialmente, e com sistemas já em funcionamento, irá realizar as manutenções necessárias e atualizações no sistema.

Em muitos casos, porém, sistemas de escalonamento genéricos não são satisfatórios, e sistemas específicos têm que ser desenvolvidos. Uma razão para o desenvolvimento de um sistema específico é que o problema de escalonamento é freqüentemente muito grande para um sistema de escalonamento genérico baseado em PC. Os bancos de dados podem ser muito grandes e necessitarem de interfaces entre o controle de chão-de-fábrica e o sistema de escalonamento que possa estar além das capacidades de um sistema de programação genérico.

Uma segunda razão é que às vezes o ambiente de processo possui certas restrições que são difíceis de construir em um sistema genérico, por exemplo: certas máquinas de um centro de trabalho iniciam com o processamento de diferentes tarefas ao mesmo tempo, ou um grupo de máquinas às vezes age como uma única máquina, e em outro momento, como máquinas separadas.

A terceira razão para o desenvolvimento de sistemas específicos de escalonamento é que o usuário pode insistir em ter o código de fonte e, dessa forma, ele próprio ou a organização será capaz de realizar as manutenções necessárias. Uma importante vantagem para sistemas específicos é que a manipulação do escalonamento é normalmente considerada mais rápida e fácil com esses sistemas do que com sistemas genéricos.

4.6 Procedimentos Gerais de Escalonamento

Alguns problemas de escalonamento são relativamente fáceis. Alguns deles podem ser formulados com programação linear e solucionados através do uso de algoritmos eficientes. Esses algoritmos eficientes são normalmente referenciados como algoritmos de tempo polinomial. Quando um problema pode ser solucionado através de método eficiente, algoritmos de tempo eficiente permitem que exemplos muito grandes desse problema, com centenas ou até mesmo, milhares de trabalhos, ainda podem ser resolvidos em um computador em um tempo relativamente curto.

No entanto, diversos problemas de escalonamento são intrinsecamente muito complexos. Esses problemas são chamados NP-completos (enumeração explícita de todas as alternativas para a obtenção do melhor resultado). Esses não podem ser formulados com programas lineares e nenhum algoritmo ou regras simples rendem ótimas soluções em um período limitado de tempo computacional. Pode ser possível resolver esses problemas, mas isso acarretará num elevado tempo computacional. Fica-se, então, normalmente satisfeito com uma solução aceitável, não distante da solução ótima.

Tratar-se-ão a seguir algumas técnicas gerais de escalonamento que se mostraram úteis em sistemas de programação industrial. Muitos desses métodos ou técnicas não garantem uma solução ótima, seu objetivo é encontrar uma solução razoável em um tempo computacional relativamente pequeno.

Regras de Despacho (*Dispatching Rules*)

Sempre que uma máquina estiver livre, uma regra de disparo verifica as ordens que estão esperando na fila e seleciona a ordem que possui a prioridade mais alta.

As regras podem ser classificadas de diversas formas. Uma diferenciação pode ser feita entre regras estáticas e regras dinâmicas. As regras estáticas não são dependentes de tempo. Elas somente possuem uma função dos dados da tarefa, da máquina ou de ambas. Por outro lado, as regras dinâmicas são dependentes de tempo. Um exemplo de uma regra dinâmica é a MS – *Minimum Slack* (Menor Folga), na qual as ordens das tarefas são definidas de acordo com seus tempos restantes de folga. Nesse tipo de regra se pode, em algum momento, ter a tarefa com prioridade maior sobre outra tarefa, e em outro determinado momento, a prioridade pode se inverter devido à folga existente entre as tarefas.

Outra maneira de se classificar as regras é de acordo com a origem das informações. Essas regras podem ser locais ou globais. As regras locais usam somente informações pertinentes à fila onde o trabalho está aguardando ou a máquina onde o trabalho será feito. As regras globais podem usar informações pertinentes às outras máquinas, como, por exemplo, o tempo de processamento e de fila nas próximas máquinas do roteiro da peça.

A utilização de somente uma regra para a solução do escalonamento é muitas vezes limitada quando objetivos complexos devem ser alcançados. Para esses casos a utilização de regras compostas apresenta resultados relativamente melhores.

As regras compostas são uma combinação de duas ou mais regras que, quando utilizadas em conjunto, em determinada função, possuem resultados bem melhores que a utilização de uma simples regra.

5 Trabalhos Relacionados

Centenas de sistemas de escalonamento foram desenvolvidos nos últimos anos. Esses sistemas estão localizados em várias indústrias e universidades localizadas em diversos países. Não existe uma lista ou uma referência bibliográfica com todos os sistemas, suas aplicações e características. Entretanto, foram escritos artigos e livros que descrevem alguns sistemas, bem como, na Internet também é possível localizar referências sobre esses sistemas.

Neste capítulo serão apresentados alguns sistemas que fornecem soluções para o problema de escalonamento. Os sistemas apresentados serão divididos em: sistemas acadêmicos ou protótipos de pesquisa, sistemas comerciais genéricos e específicos, tentando com isso apresentar uma visão geral de alguns sistemas existentes e suas aplicações. Porém, não é objetivo para este estudo comparar os sistemas, mas apresentar suas principais características e as abordagens utilizadas em cada sistema, tanto os que utilizam bases de conhecimento ou algoritmos para gerarem o escalonamento.

Entre os sistemas a seguir apresentados foi possível apenas testar e realizar experimentos com o LEKIN [LEK2001], que se mostrou bastante satisfatório quanto à apresentação gráfica dos resultados do escalonamento e a facilidade de operação, entretanto, por se tratar de um sistema acadêmico não possui integração com outros sistemas ou banco de dados externos ao mesmo.

Nos sistemas comerciais citados a seguir não foi possível realizar testes e simulações, pois as empresas que comercializam os mesmos não disponibilizam para implantações e testes sem um vínculo comercial.

5.1 Sistemas Acadêmicos

5.1.1 LEKIN [PIN99] e [LEK2001]

O LEKIN é um sistema de escalonamento que foi desenvolvido na Stern School of Business de Nova Iorque. A versão acadêmica do sistema de escalonamento LEKIN é uma ferramenta de aprendizagem para o escalonamento em seis ambientes de produção: uma única máquina, máquinas em paralelo, *flow shop*, *flexible flow shop*, *job shop* e *flexible job shop*. O sistema foi projetado para rodar sobre o ambiente Windows 95 ou superior e Windows NT.

O sistema possui vários algoritmos de escalonamento e heurísticas, e também foi desenvolvido para permitir que o usuário utilize e teste suas próprias heurísticas, disponibilizando, desse modo, que posteriormente os resultados possam ser comparados com as heurísticas e os algoritmos existentes no sistema. O sistema possui limitação para controle de até dezoito atividades, suporta até dez centros de trabalho, sendo que nesses centros se pode ter até quatro máquinas.

Inicialmente o usuário deverá informar os dados, manualmente, dos centros de trabalhos, das máquinas e atividades a serem escalonados. O sistema apresenta simultaneamente quatro janelas, todas com as informações organizadas em estrutura de árvore, com as informações sobre o escalonamento; essas janelas são descritas a seguir:

- Janela do Conjunto das Máquinas (*Machine Park Window*): essa janela mostra informações sobre a capacidade dos centros de trabalho e suas respectivas máquinas;
- Janela do Ambiente de Trabalho (*Job Pool Window*): essa janela mostra informações sobre o tempo de início, tempo de conclusão, e demais informações com respeito a cada tarefa que está sendo executada. Primeiro as tarefas são listadas. Se o usuário selecionar uma tarefa específica, então imediatamente uma lista com as operações pertencentes a essa tarefa será apresentada;
- Janela de Seqüenciamento (*Sequence Window*): essa janela mostra a lista de todas as tarefas na ordem em que elas serão processadas nas máquinas. Primeiramente aparecem as máquinas; também é permitida a seleção detalhada de um máquina, então aparecerá todos as tarefas alocadas para essa máquina com seus tempos de *setup*, início e fim de processo;
- Janela do Gráfico de Gantt (*Gantt Chart Window*): essa janela mostra o tradicional gráfico de Gantt, mas o usuário também pode selecionar uma operação do gráfico e abrir uma outra janela com as informações detalhadas dessa operação.

Após a seleção de uma heurística ou do algoritmo, o escalonamento é gerado e pode ser visualizado nas janelas citadas acima. O escalonamento mostrado na janela do gráfico de Gantt é também chamado escalonamento semi-ativo. Um escalonamento semi-ativo é caracterizado pelo fato de que o tempo de início de qualquer operação de qualquer atividade sobre qualquer máquina, é determinado pela conclusão da operação precedente da mesma atividade em uma máquina diferente ou a conclusão de uma operação de outra atividade na mesma máquina.

O sistema possui algoritmos para diversos ambientes de máquinas e funções objetivos. Esses algoritmos possuem as seguintes características:

- regras de despacho: incluem algumas regras citadas no capítulo três. Essas regras podem ser aplicadas tanto para ambientes com máquinas em série ou máquinas que trabalham paralelamente. A cada momento que uma máquina se torna livre, o sistema verifica quais as atividades devem ser as seguintes, nessa máquina, utilizando alguma regra pré-selecionada;
- heurísticas do tipo gargalo: essas heurísticas são bastante genéricas e foram desenvolvidas para diversos ambientes de máquinas e funções objetivos, não sendo específicas para determinado ambiente ou função objetivo;
- técnicas de pesquisa local: o sistema também possui uma rotina de procura local que é aplicável para ambientes do tipo *flow shop* e *job shop* (mas não

para flexíveis *flow shop* ou flexíveis *job shop*) com o critério de makespan ou de maior tempo de atraso como objetivo;

- uma heurística para flexíveis *flow shop*, tendo como objetivo o maior tempo de atraso. Essa rotina é uma combinação da heurística do tipo gargalo com a rotina de pesquisa local.

O usuário pode alterar manualmente o escalonamento gerado, tendo opção de deixar o escalonamento semi-ativo ou não, o sistema também possui mecanismos para a inclusão da seqüência de tarefas individualmente para cada máquina.

Os escalonamentos gerados podem ser armazenados em arquivos de *log*, para posteriormente serem comparados com outros escalonamentos gerados através de uma ou mais funções objetivo. Essa facilidade do sistema é demonstrada através de gráficos.

Uma vantagem adicional desse sistema é que o usuário pode adicionar seu próprio algoritmo ao sistema. Esse recurso permite que o usuário utilize as interfaces do sistema para testar seu algoritmo. O processo para fazer um programa externo reconhecível pelo sistema consiste de duas etapas: a preparação do programa ou código (programação, compilação e *debuging*) e a união do código ao sistema de escalonamento.

5.1.2 *ISTOP* [KER90]

ISTOP é um sistema que tem como objetivo integrar o planejamento estratégico em nível de fábrica, com o planejamento tático em nível de célula ou de centro de trabalho, usando sistemas baseados em conhecimento local.

No nível operacional, o sistema opera dentro de um *framework* de planejamento global para executar as análises, gerando informações ou detalhando os planos locais de ação, baseados na integração do conhecimento oriundos de diversas fontes. Já que muitas restrições são brandas, uma característica dessa descentralização é que há uma negociação de restrições leves entre sistemas baseados em conhecimento local, ao invés de rígidas restrições impostas pelos níveis mais altos, como ocorre em sistemas do tipo MRP.

Uma importante característica do *ISTOP* é a mínima intervenção pelo plano estratégico nas operações detalhadas da fábrica. O plano estratégico é suficientemente flexível para permitir que as áreas operacionais individuais reajam aos problemas de modo mais apropriado, usando o conhecimento local das situações e heurísticas para resolver os problemas localmente, que podem estar sintonizados com os principais objetivos e estratégicos atualmente em uso.

O *ISTOP* deve ser usado como um sistema de planejamento para o nível de chão-de-fábrica e seus principais componentes são:

- um banco de dados integrado com previsão de ordens, lista de materiais com planos de processo alternativo para cada peça, previsão de vendas, planejamento e atualização das ordens de fabricação com datas de entrega e

conclusão do escalonamento, visão atualizada dos processos em produção e níveis de estoque, avaliação dos recursos de manufatura e suas atuais reservas para as diferentes ordens de fabricação;

- um módulo de planejamento estratégico que possui uma base de conhecimento, um sistema de suporte à decisão, permitindo ao usuário analisar entre as diferentes políticas em relação aos níveis de capacidade de produção e inventário, a média de *lead times*; permite também examinar as implicações das diferentes políticas nos objetivos estratégicos da empresa;
- um módulo de planejamento mestre que possui uma base de conhecimento, um sistema de apoio à decisão que ajuda no desenvolvimento de um produto final, construindo um escalonamento para conhecer a demanda do cliente, e que é tanto consistente como possível em nível de fábrica, e no conjunto de políticas no nível estratégico;
- um conjunto de módulos no planejamento tático, em nível de célula individual ou de estação de trabalho, possuindo bases de conhecimento responsáveis pelo planejamento e escalonamento do trabalho em cada célula, de modo consistente com as políticas fixadas pelo módulo de planejamento estratégico. As datas de entrega dos produtos finais são requisitadas pelo módulo de planejamento mestre e pelos planos desenvolvidos através dos centros vizinhos.

O ISTOP foi construído sob um banco de dados orientado a objetos, onde as principais classes das entidades são os recursos, os processos e as ordens. Recursos são divididos em recursos materiais (matéria-prima, componentes, produtos finais, etc.) e recursos de produção (máquinas, ferramentas, operadores, etc.). Os processos consistem dos vários tipos de produção, montagem e operações de inspeção nos quais a conversão de matéria-prima para produtos acabados ocupa lugares. As ordens são divididas em ordens de clientes (pedidos), ordens de fabricação e ordens de compra, e se elas estão em previsão, planejadas, em desenvolvimento, completas, estocadas, etc.

Os relacionamentos entre os recursos e os processos são também representados. Isso resulta em uma rede de recursos de manufatura que serve para integrar a lista de material e as rotas de informações para o planejamento de fábrica.

5.1.3 *CUISE* [PIN99]

O *CUISE* (*Columbia University Interactive Scheduling Editor*) foi desenvolvido na Universidade de Columbia em Nova Iorque por volta de 1990. O software foi escrito na linguagem C++ para o ambiente Windows, usando a metodologia de orientação a objetos.

O principal objetivo do projeto foi criar um sistema de escalonamento iterativo que pudesse ser facilmente adaptado aos diversos ambientes de manufatura existentes. Inicialmente, o software concentrou seus esforços para a solução de escalonamentos nos ambientes do tipo *flow shop*, deixando para futuras implementações os ambientes mais complexos do tipo *job shop*.

O sistema é dividido em dois módulos, o módulo de armazenamento das informações e o módulo de escalonamento. O módulo de manutenção dos dados é dividido em informações das atividades ou dos trabalhos e das informações do ambiente de produção, ou seja, das máquinas. Os conceitos de dados estáticos e dinâmicos são levados em conta, conforme discussão já apresentada no capítulo anterior. Para cada processo específico de manufatura, um conjunto de arquivos serão criados para definir os roteiros de fabricação e as máquinas, bem como o formato das atividades a serem escalonadas. A edição dos tipos de máquinas e atividades é permitida, mas mudanças no número de estágios ou fases sem a alteração dos processos de manufatura não são permitidas.

O módulo de escalonamento é dividido em quatro grupos:

- módulo de geração do escalonamento: esse módulo é responsável pelo escalonamento de todas as atividades. Isso inclui o escalonamento global sobre todas os ambientes fabris e também do escalonamento local de algum ambiente específico da empresa. O resultado do escalonamento gerado é enviado ao módulo de análise de viabilidade para realizar a análise;
- o módulo de análise de viabilidade: esse módulo desempenha duas funções: assegura que os tempos de início e fim de cada atividade estejam calculados corretamente e confere a viabilidade. Também possui dois conjuntos de restrições: as restrições rígidas e brandas. As restrições rígidas são aquelas que nunca podem ser violadas, como uma atividade que começa mais cedo que o seu tempo final em um estágio anterior. As restrições brandas podem ser violadas, um exemplo seria não escalonar algumas atividades depois de outras, devido ao grau de dificuldade da atividade. Todas as violações existentes são armazenadas e ficam disponíveis para o usuário realizar a verificação;
- módulo de edição do escalonamento: esse módulo permite que as atividades sejam alteradas dentro do escalonamento gerado. Entretanto, as alterações efetuadas são verificadas pelas restrições previamente estabelecidas. A interface com o usuário nesse módulo pode ser tanto gráfica como a caráter; essas interfaces permitem visualizar, filtrar e editar os escalonamentos, bem como acessar as medidas de desempenho;
- o módulo de análise de desempenho: esse módulo suporta vários métodos para análise de um escalonamento de acordo com as medidas de desempenho especificadas. Existem duas maneiras de realizar a análise de desempenho, através de uma função objetivo que pode ser criada pelo usuário ou através de uma biblioteca de medidas de desempenho; essa biblioteca possui rotinas que mostram os dados tanto no formato texto como gráfico.

5.1.4 ISIS [FOX90]

ISIS é um sistema de *job shop scheduling* desenvolvido na Universidade de Carnegie-Mellon. Sua razão está baseada na observação dos vários *job shops* escalonados por humanos para procurar por um escalonamento que satisfaça os diversos tipos de

restrições existentes. ISIS usa uma estratégia de procura direta das restrições para encontrar satisfatoriamente a solução ao invés de escalonamentos ótimos.

ISIS usa uma linguagem de descrição orientada a objetos, chamada SRL, para a representação das ordens de produção, operações em processo, máquinas, ferramentas e operadores na forma de um tipo de rede semântica. Essa linguagem foi subsequenteiramente usada como base de programação no ambiente de Inteligência Artificial, definida pelo grupo Carnegie com o nome de '*Knowledge Craft*'. O uso de SRL se fez necessário para representar o nível exigido de conhecimento das operações de chão-de-fábrica e para produzir escalonamentos realísticos e detalhados, sendo um bom exemplo tanto para representar o poder das linguagens orientadas a objetos, como a complexidade do conhecimento que precisa ser representado.

A representação do ISIS utiliza diferentes tipos de entidades para capturar os principais conceitos e as relações de conhecimento no domínio do chão-de-fábrica. A primeira descrição refere-se aos diferentes tipos de objetos como ordens, peças, máquinas, operadores, etc., e estes podem ser organizados em hierarquias de classificação.

O escalonamento em construção no ISIS possui hierarquicamente quatro níveis diferentes. O primeiro nível ou superior tem por objetivo a seleção das ordens de produção para o escalonamento. O segundo nível está interessado em executar uma análise de capacidade para descobrir os possíveis gargalos de produção. O terceiro nível tem o objetivo de detalhar a alocação dos recursos para as ordens, e, finalmente, no quarto e mais baixo nível, é feita a reserva dos recursos individuais detalhada para as faixas de tempo nos quais eles são requeridos.

Cada nível possui duas fases: a primeira fase procura encontrar uma solução ao problema desse nível, e a segunda fase realiza a análise de pós-pesquisa que determina a aceitabilidade da solução e tenta melhorá-la. A solução a um determinado nível gera um conjunto de restrições para a solução do próximo nível.

A versão inicial do ISIS foi testada em 1984 na Westinghouse em Winston-Salem, na Carolina do Norte, demonstrando que os escalonamentos gerados foram superiores aos executados pelo escalonador humano. O sistema não foi completamente instalado, porque sua arquitetura não era suficientemente robusta e não apresentava uma interface satisfatória. Algumas das limitações predominantemente baseadas nas ordens e na estratégia de decomposição de problemas do ISIS ficaram visíveis. Tornando necessário avaliar estratégias alternativas, com por exemplo, o escalonamento reativo (*reactive scheduling*) para a solução dos problemas encontrados.

5.1.5 OPIS [SMI90]

O OPIS (*Oportunistic Intelligent Scheduler*) é um sistema desenvolvido na Universidade de Carnegie Mellon para explorar a noção de escalonamento reativo (*reactive scheduling*). Contingências do chão-de-fábrica não planejadas, como máquinas paradas, ferramentas perdidas, atraso em entregas e retrabalhos significam que qualquer

escalonamento preditivo (*predictive scheduling*) requererá contínuas atualizações nesse ambiente.

OPIS usa a mesma forma do modelo de representação da fábrica apresentado no ISIS, baseado na definição de ordens, operações, recursos, restrições, etc. nos termos do esquema SRL. Porém, difere do ISIS no fato de que é possível diferentes formas de decomposição do problema no nível superior, dependendo principalmente das várias restrições aplicadas ao problema.

Esse sistema possui a visão de que, quando algum evento inesperado acontece no chão-de-fábrica, invalida o escalonamento existente. Há necessidade de solucionar as inconsistências no escalonamento anterior, nesse momento também deve haver alternativas para melhorar o escalonamento, através da remoção de restrições que existiam previamente. Para tirar proveito desse mecanismo, é necessário focalizar o esforço do escalonamento reativo (*reactive scheduling*) sobre um conjunto de restrições, no qual a maioria será afetada pelas mudanças.

A posição adotada pelo OPIS é de que deveria ser escolhida aquela perspectiva de escalonamento de acordo com o conjunto de restrições que é mais importante para o aspecto particular do escalonamento em questão. Isso permitirá a avaliação mais efetiva das causas no chão-de-fábrica, tanto para a resolução de conflitos como do ponto de vista oportunístico, focalizando no componente (ordem ou recurso) do escalonamento que melhor cerca o problema em questão.

Ao escalonar operações em recursos que são gargalos, uma perspectiva de escalonamento, baseado em recursos deve ser usada para fazer um ótimo uso destes recursos. Os tempos de operação sobre os recursos gargalo então servem como restrições sobre o escalonamento, orientado as ordens no restante das operações.

OPIS usa uma arquitetura de quadro negro (*blackboard architecture*), adotando diferentes fontes de conhecimento. Atualizações do estado da fábrica são comunicadas ao sistema através de uma interface externa. Os efeitos da mudança serão propagados para todo o sistema, e a mudança introduzirá um conjunto de conflitos ou oportunidades que existem no escalonamento. Essas alterações serão processadas pelo propagador de restrições (*propagator constraint*) e pelo verificador de consistência (*consistency checker*) e posteriormente para o gerente de procura (*search manager*), como um controlador de eventos. O gerente de procura analisa o controle de eventos usando um conjunto de regras, e invoca o escalonamento estratégico de acordo com o tipo de evento. O escalonador de ordens (*order scheduler*) proporciona um conjunto de métodos para revisar as operações escalonadas.

O escalonador de recursos (*resource scheduler*) possui métodos para revisar o escalonamento de um recurso alocado, e assume que a competição para o recurso é alta, eliminando assim a necessidade de introduzir fatias de tempo entre as operações. O trocador de escalonamento (*scheduler shifter*) implementa o menos sofisticado meio de reação, movendo os tempos escalonados das operações individuais adiante para uma quantidade alocada. O trocador de demanda (*demand swapper*) implementa outro método

reativo relativamente simples, trocando a porção afetada das ordens escalonadas, com outra ordem do mesmo tipo que possa não ser requerida urgentemente, explorando assim qualquer flexibilidade que possa existir nos escalonamentos das ordens existentes.

O analisador de capacidade (*capacity analyser*) executa uma análise para identificar gargalos e assim indicar quando o escalonador de recursos deve ser invocado no lugar do escalonador de ordens. Isso provê meios para estruturar o problema de escalonamento global, baseando-se em escalonamento orientado em recursos onde os recursos com gargalos têm prioridade mais alta.

No ambiente dinâmico do chão-de-fábrica, a habilidade do sistema para responder rapidamente às perturbações é obviamente essencial, e, nesse contexto situa-se o OPIS, com a decomposição dos múltiplos problemas direcionados oportunamente do nível mais baixo até o mais alto, possuindo assim uma vantagem clara sobre o ISIS.

5.2 Sistemas Comerciais Genéricos

5.2.1 *PROVISA* [PRO2001]

PROVISA é um sistema de escalonamento baseado na capacidade finita dos recursos de produção. Ele é composto de sete módulos:

- importação e exportação: o *Provisa* realiza a integração com sistemas de ERP/MRP através de arquivos texto predefinidos, também possui ligação com planilhas eletrônicas;
- banco de dados: o banco de dados padrão utilizado pelo *Provisa* é o *Acess* da Microsoft;
- editor de regras: além do padrão de regras oferecido pelo *Provisa*, ele suporta adicionar lógicas, facilitando a customização da lógica modelada. Dessa maneira o usuário não fica restrito a uma metodologia de escalonamento, ou tem necessidade de chamar novamente o vendedor do sistema para alterar a lógica, quando sua fábrica ou os objetivos do sistema de manufatura forem alterados;
- escalonador: o algoritmo de escalonamento não é divulgado pelo fornecedor do sistema, apenas é citado o esforço e os anos de testes gastos pela equipe da *Lanner Group* para que ele realmente seja um verdadeiro sistema de escalonamento confiável e rápido;
- quadro de planejamento: um gráfico de Gantt, com capacidade de arrastar e soltar (*drag and drop*), permite que o *Provisa* mostre a visão tradicional do escalonamento. Essa visão é ampliada com a habilidade de visualizar tabelas iterativas do escalonamento e reescalonar ordens individuais, permitindo assim analisar facilmente os aspectos individuais do escalonamento;
- relatórios: tendo o *Acess* como banco de dados padrão do *Provisa*, todos os recursos de geração de relatórios desse aplicativo podem ser explorados.

Também são fornecidos relatórios tradicionais do sistema que podem ser facilmente modificados conforme a necessidade do usuário;

- comandos de macro: uma característica importante do Provisa é permitir a geração de macros, onde vários comandos são inseridos em apenas um ícone na tela do sistema. Isso permite, por exemplo, importar dados de um arquivo texto, executar o escalonamento e gerar um determinado relatório com apenas um *click* no ícone definido pelo usuário.

Uma importante característica do Provisa é sua capacidade de comparar escalonamentos usando diferentes lógicas, mas com a mesma base de dados. O Provisa permite a escolha de até cinco critérios para realizar as comparações. Logo após a seleção, uma outra janela é apresentada com os resultados, tornando mais fácil para o usuário decidir qual o melhor escalonamento a ser utilizado.

5.2.2 **PREACTOR [PRE2001]**

O PREACTOR é um software inglês baseado no conceito de capacidade finita de produção. Gera a programação da produção confiável, respeitando a disponibilidade efetiva dos recursos produtivos e a existência de restrições.

A flexibilidade do Preactor admite sua utilização em sistemas produtivos de diversas naturezas:

- produção sob encomenda;
- lotes repetitivos;
- plantas e processos;
- serviços de logística e outros.

Desenvolvido para ambiente Windows (95, 98 e NT), possui interface gráfica e recursos como arrastar e soltar, gráfico de Gantt e demais funcionalidades necessárias para um bom sistema de escalonamento. Comercializado em quatro versões, sendo que cada uma se destina para cada tipo de empresa e necessidade de programação.

- Preactor Lite: quadro eletrônico de programação com configuração fixa. Indicado para empresas com problemas mais simples de seqüenciamento. Integrável com planilhas eletrônicas e permite a programação para frente, para trás e bidirecional;
- Peactor 200: oferece recursos e flexibilidade para configurações e customizações mais complexas, que demandem matrizes de seqüenciamento dependente de *setups*, quebras de lotes, controle de recursos secundários. Permite a integração com outros softwares.
- Preactor 300: indicado para situações que demandem soluções avançadas de programação em capacidade finita. Indicado para sistemas produtivos de indústrias de processos, aplicações com múltiplas restrições, montagens e sub-montagens complexas;

- Preactor 400: recomendado para aplicações de planejamento e programação avançadas, com alocação automática de materiais e que demandem regras complexas de seqüenciamento, para gerar gargalos dinâmicos, minimização de materiais em processo, seleção de seqüência com minimização de tempos de *setup*. Permite que os usuários criem suas próprias regras de otimização.

Todas as versões utilizam os mesmos arquivos de configuração do sistema. Em caso de atualização ou *upgrade* de versões, não é necessário fazer reconfigurações.

O Preactor pode ser associado com planilhas eletrônicas, sistemas MRP ou ERP, sistemas de coleta de dados no chão-de-fábrica, pacotes financeiros e de vendas e outros bancos de dados. A associação com sistemas de MRP ou ERP será feita através de arquivos textos gerados por esses sistemas e atualizado no Preactor, com *layouts* pre-definidos. Além disso, o Preactor pode ser usado antes e/ou depois da execução do MRP. Por exemplo, os pedidos com informações dos produtos são carregados no sistema, que produzirá a programação. O MRP lerá a programação e fará a explosão das listas de materiais e o planejamento das necessidades de matéria-prima. Em seguida o Preactor cria listas de trabalhos, cartões de rota e relatórios para controle diário da programação. Se necessário o usuário fará as alterações durante o processo.

5.2.3 CAPS [CAP2000]

O CAPS (*Computer Aided Process Scheduling*) é um software desenvolvido no *Prolog Development Center* em Copenhague na Dinamarca, utilizando a linguagem VisualProlog. Também utiliza o conceito de capacidade finita de produção para gerar o escalonamento.

O CAPS é utilizado para o escalonamento e controle das ordens de fabricação ou atividades executadas em um ou mais recursos, observando as restrições existentes, como por exemplo, capacidade e disponibilidade dos recursos, seqüência das atividades e datas limite das ordens. O sistema também combina o planejamento automático com a iteração manual, através de uma interface gráfica na qual o usuário pode simular rotas alternativas e alterações no escalonamento gerado.

Os escalonamentos gerados podem ser visualizados no tradicional gráfico de Gantt, onde as ordens e atividades são mostradas. O gráfico dos recursos mostra as atividades alocadas por recursos, quando esses estão disponíveis ou ativos. O gráfico de ordens mostra as ordens com seu planejamento no horizonte de tempo, com o horário mais cedo para início e o último horário para término. Se uma atividade ou ordem não satisfazem todas as condições requeridas, elas recebem uma condição de alerta (a atividade fica em vermelho no gráfico) e a informação sobre a restrição não satisfeita é apresentada para o usuário.

O gráfico de PERT [PER58] também está disponível no CAPS, onde cada barra do gráfico representa uma tarefa e seu ciclo de atividades. Esse gráfico é muito usado para visualizar estruturas muito complexas com diversos níveis de dependência.

O escalonamento pode ser executado automaticamente utilizando algoritmos baseados em recursos, atividades ou eventos. A técnica baseada em recursos também pode ser realizada utilizando somente os recursos selecionados como gargalos, devido a restrições específicas de tempo, ou devido à dependência na seqüência das atividades para as operações de *setup*. O escalonamento pode escolher entre os diferentes recursos em uma ordem de prioridade predefinida ou como primeiro recurso disponível.

A capacidade dos recursos é definida através de um calendário individual, estabelecido por turnos semanalmente. Cada turno pode apresentar um ou mais horários, variando dia a dia, onde jornadas de trabalhos variáveis, períodos de férias e feriados devem ser levados em conta. A disponibilidade de um recurso pode ser modificada posteriormente através de uma função, no caso de existir uma manutenção planejada.

O reescalonamento somente é realizado em consequência de uma inevitável atualização ou mudança, e o sistema tenta limitar a extensão de reescalonar as atividades mais necessárias, para manter o escalonamento global tão inalterado quanto possível. Isso pode implicar que tempos ou restrições de recursos não específicos sejam ultrapassados, mas isso é mostrado imediatamente no gráfico de Gantt como uma atividade crítica. Então o usuário pode decidir sobre as ações corretivas através de intervenção manual.

O CAPS pode ser utilizado com ODBC padrão e permite a importação e exportação de arquivos para a integração com outros sistemas; desse modo as atualizações podem ser feitas tanto em tempo real como posteriormente.

5.3 Sistemas Específicos

5.3.1 GATES [PIN99]

O GATES (*Gate Allocation and Tracking Expert System*) foi desenvolvido para as tarefas diárias de alocação e realocação dos aviões nos portões da TWA, nos terminais do aeroporto JFK em Nova Iorque.

O principal problema é a variedade no tamanho dos aviões e no tamanho dos portões de acesso, nos quais geram restrições de quais tipos de aviões podem ocupar determinados portões ou que determinado portão não pode receber determinados aviões. Os aviões que sofrem translações, e o tempo de permanência dos aviões no solo também devem ser levados em consideração.

O sistema foi desenvolvido em Prolog que representa o conhecimento através de predicados lógicos que demonstram as regras. A versão implementada consiste de cinco passos:

1. Um vôo não alocado deve ser considerado. Um vôo é selecionado através de um conjunto de regras de vôo que são organizadas de acordo com alguma prioridade. As prioridades são determinadas pelo tamanho da aeronave, bem como do horário de sua chegada e partida;

2. Uma alocação do portão para selecionar o voo deve ser considerada. A regra é selecionada usando um conjunto de regras do tipo GATEOK, citadas no capítulo anterior. Essas regras são organizadas em uma certa prioridade, que é determinada pelo local, tamanho e pela posição do portão;
3. A correta alocação do portão de embarque deve ser verificada. Isso é feito verificando a alocação através de um conjunto de regras de conflito;
4. Se um voo não pode ser alocado, ajustes devem ser feitos através do relaxamento de restrições;
5. Após todos os ajustes terem sido feitos, as alocações são ajustadas para maximizar a utilização do portão de embarque e minimizar a carga de trabalho da aeronave.

Esses passos são executados recursivamente. O sistema pode necessitar de dois, três ou quatro passos para completar a alocação das aeronaves nos portões de embarque. A interface do usuário é um gráfico de Gantt vertical que apresenta o tempo que a aeronave gasta no portão, seus horários de chegada e partida e também os números dos voos para chegada e partida.

O escalonamento de um portão é construído todos os meses para um novo voo escalonado. O controlador de voo revisa o escalonamento gerado pelo computador e o aceita ou modifica através de uma interface gráfica. Uma vez satisfeito, o controlador submete o escalonamento final para o computador principal, onde o mesmo é distribuído para os demais departamentos.

O escalonamento mensal do portão é usado como base para a geração das alocações diárias do mesmo. Diariamente o sistema está sujeito a mudanças imprevistas, como atrasos na chegada, cancelamentos de voos, mudanças de aeronaves, e assim por diante, cujas realocações nos portões de acesso são necessárias. No processo de realocação, o escalonador diário tem que levar em consideração o número limitado de portões, porque o software está em comunicação com o computador principal, uma nova informação do portão é imediatamente transmitida para as outras partes do sistema, incluindo monitores, que são responsáveis por retransmitir as informações dos portões para os passageiros.

6 Sistema Proposto

6.1 Modelo Conceitual de Dados

6.1.1 Introdução

Segundo [BOO2000] um modelo é uma simplificação da realidade.

A utilização de um modelo conceitual de dados é de vital importância para o desenvolvimento de um sistema, pois é através de um modelo de dados que podemos abstrair a representação de parte da realidade que está sendo analisada para posterior implementação.

Os modelos fornecem uma cópia do projeto de um sistema. Os modelos poderão abranger planos detalhados, assim como planos mais gerais com uma visão panorâmica do sistema considerado. Um bom modelo inclui aqueles componentes que têm ampla repercussão e omite os componentes menores que não são relevantes em determinado nível de abstração. Desta forma construímos modelos para compreender melhor o sistema que estamos desenvolvendo.

Através da modelagem se pretende atingir alguns objetivos como:

- Visualizar o sistema como ele é ou como desejamos que ele seja;
- Permitir especificar a estrutura ou o comportamento de um sistema;
- Proporcionar um guia para a construção do sistema;
- Subsidiar as decisões tomadas.

O primeiro passo para o desenvolvimento de um modelo de dados é a definição dos requisitos necessários para este modelo. Para se obter estes requisitos é necessário analisar cuidadosamente o funcionamento do ambiente a ser modelado, as suas características, é também os principais problemas apresentados neste ambiente.

De acordo com [BOO2000] o uso da modelagem tem uma rica história em todas as disciplinas de engenharia. Essa experiência sugere quatro princípios básicos de modelagem, que são:

- a escolha dos modelos a serem criados tem profunda influência sobre a maneira como um determinado problema é atacado e como uma solução é definida;
- cada modelo poderá ser expresso em diferentes níveis de precisão;
- os melhores modelos estão relacionados à realidade;
- nenhum modelo único é suficiente. Qualquer sistema não-trivial será melhor investigado por meio de um pequeno conjunto de modelos quase independentes.

Anteriormente foram apresentados conceitos básicos sobre os tipos de produção e algumas técnicas de gestão da produção, detalhando o funcionamento do controle da produção, com o objetivo de auxiliar na modelagem proposta.

Com a coleta das informações concluídas e a compreensão do ambiente no qual o modelo atuará, a seguir será apresentado o modelo de dados de referência para o escalonamento da produção num ambiente com capacidade finita.

Para o desenvolvimento do modelo de dados apresentado a seguir, optou-se pelo modelo de dados orientado a objetos, utilizando a técnica de modelagem de objetos - UML do inglês *Unified Modelling Language*.

Não é objetivo deste trabalho definir e detalhar todos os significados da notação UML, mas sim apresentarmos como essas entidades são criadas, simbolizadas, organizadas e como serão utilizadas dentro de um desenvolvimento utilizando a UML. Maiores detalhes sobre os aspectos de modelagem da UML podem ser encontrados em [ERI98], [FOW2000], [LAR2000] e [RUM99].

Alguns motivos podem ser citados para a utilização de orientação a objetos (OO):

- a orientação a objetos é uma tecnologia para a produção de modelos que especifiquem o domínio do problema de um sistema.
- quando construídos corretamente, sistemas orientados a objetos são flexíveis a mudanças, possuem estruturas bem conhecidas e provêm a oportunidade de criar e implementar componentes totalmente reutilizáveis.
- a orientação a objetos permite uma maior abstração do domínio do problema, concentrando-se nos aspectos essenciais de um problema, limitando-se no que um objeto é e faz, antes de implementá-lo.
- modelos orientados a objetos são implementados convenientemente utilizando uma linguagem de programação orientada a objetos. A engenharia de software orientada a objetos é muito mais que utilizar mecanismos de sua linguagem de programação, é saber utilizar da melhor forma possível todas as técnicas da modelagem orientada a objetos.
- as técnicas baseadas em objetos permitem o compartilhamento em vários níveis, utilizando os conceitos de herança das estruturas de dados;
- a orientação a objetos não é só teoria, mas uma tecnologia de eficiência e qualidade comprovadas, usada em inúmeros projetos e para construção de diferentes tipo de sistemas.
- a orientação a objetos requer um método que integre o processo de desenvolvimento e a linguagem de modelagem com a construção de técnicas e ferramentas adequadas.

É importante salientar que o desenvolvimento de um modelo conceitual de dados é uma das etapas no desenvolvimento de um sistema. Após a definição do modelo conceitual, ainda é necessário seguir alguns passos até a implementação do mesmo. Entre estes passos podemos citar:

- definição dos diagramas de fluxo de dados, de forma a definir o modelo funcional;
- definição dos métodos e operações, descrevendo o comportamento funcional de cada classe;
- desenvolvimento do projeto do sistema.

Quando se está desenvolvendo um modelo de dados, é preciso ter a noção que ele é uma abstração do que o sistema deve fazer, e não de como deverá fazê-lo.

6.1.2 O Modelo Conceitual de Dados

Este capítulo destina-se a apresentação do modelo desenvolvido. Este modelo tem por objetivo auxiliar no desenvolvimento do sistema de controle e escalonamento da produção. As atividades relativas a orçamentos, modificações dos produtos, compras de materiais e manutenção fabril não serão abordadas neste modelo.

A ferramenta utilizada para o desenvolvimento do modelo conceitual de dados foi a *Rational Rose* [RAT2000]. A escolha foi feita pelo fato desta ferramenta suportar a modelagem orientada a objetos, podendo ser utilizados todos os conceitos de orientação a objetos.

O modelo desenvolvido apresenta uma breve descrição das classes e seus principais atributos. A nomenclatura utilizada tanto nas classes como nos atributos tenta expressar da maneira mais real o ambiente onde o modelo atuará.

Além dos principais atributos, a semântica de cada classe será apresentada, detalhando seu relacionamento com outras classes do modelo.

O modelo, na íntegra pode ser visualizado na figura 6.1.

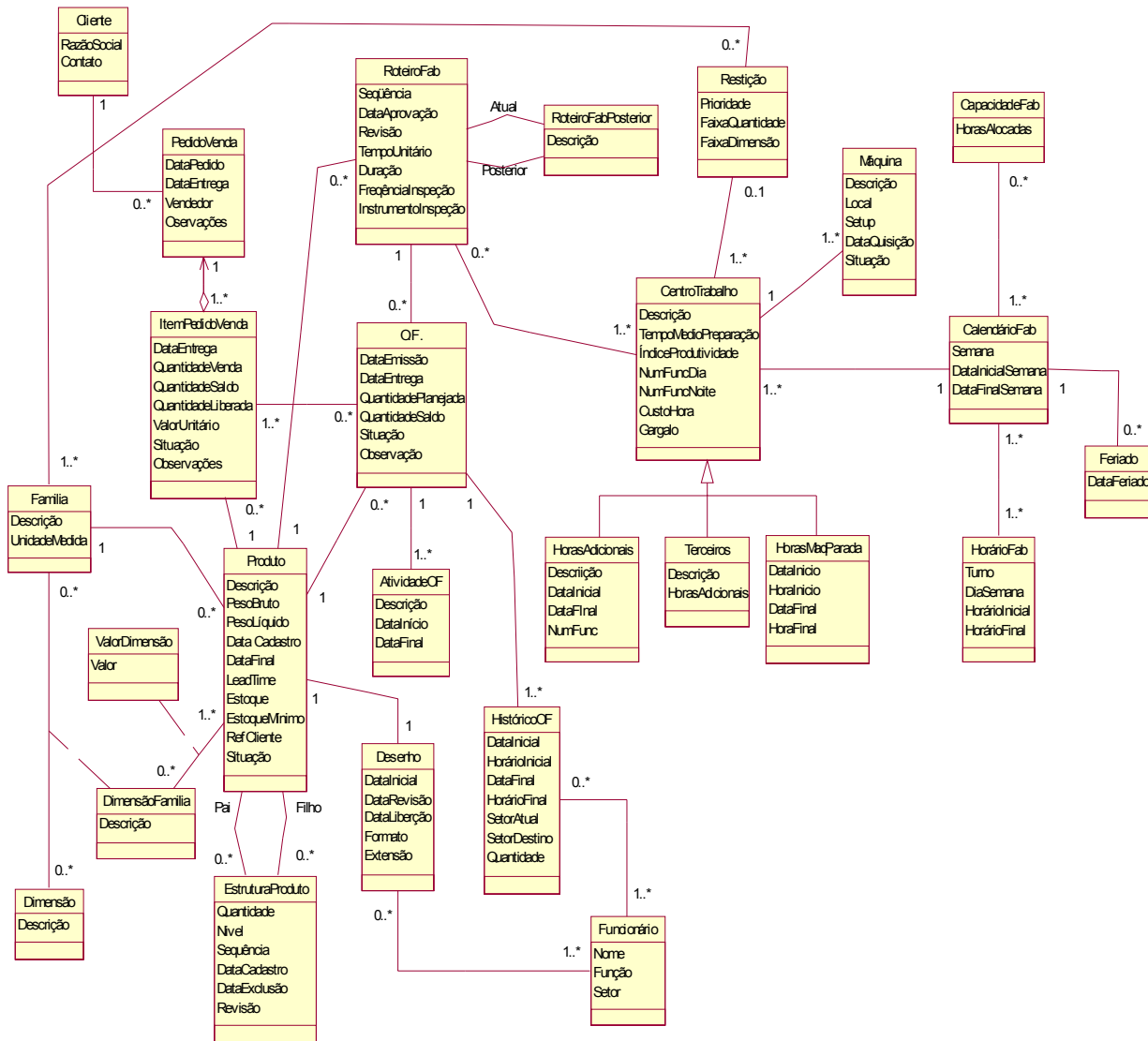


FIGURA 6.1 - Modelo conceitual de dados de uma implementação para escalonamento e controle da produção

6.1.3 Descrição das classe e seus atributos

Classe **Cliente**

Um cadastro de clientes se faz necessário quando ocorre um pedido de venda para determinado produto, caso este cliente ainda não esteja cadastrado. A classe cliente não é fundamental para o escalonamento da produção, ela somente foi adicionada para a inclusão de um pedido.

Os atributos da classe cliente estão representados na tabela 6.1.

TABELA 6.1 - Atributos da Classe **Cliente**

Atributo	Descrição
RazãoSocial	Razão Social do cliente.
Contato	Nome de um contato do cliente.

Semântica:

1. Uma instância da classe **Cliente** pode estar associada a zero ou mais instâncias da classe **Pedido**.

Classe **PedidoVenda**

Nesse modelo não são levadas em consideração as classes de orçamento e especificações técnicas que muitas vezes se fazem necessárias antes da empresa receber o aceite definitivo do pedido ou quando a empresa irá fornecer um produto que não faz parte da sua linha de vendas, ou seja, um pedido para produção sob encomenda.

Após o departamento de vendas verificar a disponibilidade de estoque ou a capacidade de produção para este pedido, informará ao cliente uma possível data de entrega do mesmo. E logo que ocorra a aceitação por parte do cliente, um pedido de venda é gerado confirmando datas e demais condições estabelecidas anteriormente.

Os principais atributos da classe Pedido de Venda estão representados na tabela 6.2.

TABELA 6.2 - Atributos da Classe **PedidoVenda**

Atributo	Descrição
DataPedido	Data de inclusão do pedido de venda.
DataEntrega	Data prevista para entrega do pedido de venda.
Vendedor	Vendedor do pedido de venda.
Observação	Observações a serem descritas sobre o pedido.

Semântica:

1. Uma instância da classe **PedidoVenda** esta associada a uma, e somente uma, instância da classe **Cliente**;
2. Uma instância da classe **PedidoVenda** agrega uma ou mais instâncias da classe **ItemPedidoVenda**.

Classe ItemPedidoVenda

Um pedido de venda pode conter (agregar) um ou mais itens solicitados pelo cliente. Os itens de um pedido podem ter datas de entrega diferentes e observações diferenciadas para cada item.

Os principais atributos da classe itens de um pedido de venda são representados na tabela 6.3.

TABELA 6.3 - Atributos da Classe **ItemPedidoVenda**

Atributo	Descrição
DataEntrega	Data de entrega do item. Preenchido quando a fabricação do item é concluída.
QuantidadeVenda	Quantidade solicitada deste item.
QuantidadeSaldo	Quantidade disponível deste item no pedido.
QuantidadeLiberada	Quantidade liberada deste item para faturamento.
ValorUnitário	Valor firmado na confirmação do pedido junto ao cliente.
Situação	Situação do item no pedido. Opções de preenchimento: L - Liberado para faturamento; C – Cancelado.
Observação	Observações a serem incluídas sobre este item.

Semântica:

1. Uma instância da classe **ItemPedidoVenda** faz parte (é um agregado) de uma, e somente uma, instância da classe **PedidoVenda**;
2. Uma instância da classe **ItemPedidoVenda** está associada a uma, e somente uma, instância da classe **Produto**;
3. Uma instância da classe **ItemPedidoVenda** está associada a zero ou mais instâncias da classe **OF**.

Classe Produto

Um produto é qualquer matéria-prima, peça, embalagem, componente, submontagem, montagem, produto manufaturado ou comprado. Pode ser considerado qualquer entidade consumível em um sistema de manufatura.

Os principais atributos da classe Produto são representados na tabela 6.4.

TABELA 6.4 - Atributos da Classe Produto

Atributo	Descrição
Descrição	Descrição do produto.
DataCadastro	Data de cadastramento do produto.
DataFinal	Data final de validade do produto, ou seja, a partir desta data o item é considerado inativo para o sistema.
PesoBruto	Peso bruto do produto.
PesoLíquido	Peso líquido do produto.
LeadTime	Tempo médio que o produto consome para ser produzido, incluindo o tempo médio de fila.
Estoque	Quantidade de estoque físico do produto.
EstoqueMínimo	Valor informado para estoque mínimo deste produto. Quando o estoque do produto se tornar inferior ao valor deste atributo, deve ser feita nova compra ou fabricação.
ReferênciaCliente	Referência utilizada pelo cliente para compra.
Situação	Atributo para identificar se o produto está em modificação. Oferece as seguintes opções: C – Cancelado; L – Liberado; M – Modificação.

Semântica:

1. Uma instância da classe **Produto** está associada a zero ou mais instâncias da classe **PedidoItem**;
2. Uma instância da classe **Produto** está associada a uma, e somente uma, instância da classe **Família**;
3. Uma instância da classe **Produto** está associada a zero ou mais instâncias da classe **DimensãoFamília**;
4. Uma instância da classe **Produto** está associada a uma, e somente uma, instância da classe **Desenho**;
5. Uma instância da classe **Produto** está associada a zero ou mais instâncias da classe **OF**;
6. Uma instância da classe **Produto** está associada a zero ou mais instâncias da classe **EstruturaProduto** nos papéis de pai e filho dentro da estrutura;
7. Uma instância da classe **Produto** está associada a uma ou mais instâncias da classe **RoteiroFab.**

Classe Família

Todos os produtos devem ser classificados em famílias. Esta classificação é fundamental para realizar o agrupamento dos produtos em função de suas características técnicas e físicas.

Os principais atributos da classe família são representados na tabela 6.5.

TABELA 6.5 - Atributos da Classe **Família**

Atributo	Descrição
Descrição	Descrição da família.
UnidadeMedida	Unidade do produto usada para o sistema.

Semântica:

1. Uma instância da classe **Família** está associada a zero ou mais instâncias da classe **Produto**;
2. Uma instância da classe **Família** está associada a zero ou mais instâncias da classe **Dimensão**;
3. Uma instância da classe **Família** está associada a zero ou mais instâncias da classe **Restrição**.

Classe Dimensão

Toda a família de produtos possui identificadores, que associados a valores descrevem as particularidades de cada produto. Por exemplo, a família de canos possui como identificadores, diâmetro interno, diâmetro externo e comprimento. Estas dimensões informadas em cada família de produtos auxilia na parametrização, busca dos produtos no sistema e na sua melhor alocação nas máquinas para a produção.

Os principais atributos da classe família são representados na tabela 6.6.

TABELA 6.6 - Atributos da Classe **Dimensão**

Atributo	Descrição
Descrição	Descrição da dimensão para a família.

Semântica:

1. Uma instância da classe **Dimensão** está associada a zero ou mais instâncias da classe **Família**.

Classe **DimensãoFamília**

Toda família de produtos pode apresentar um ou mais identificadores de dimensão, assim como um identificador de dimensão pode estar associado a várias famílias. Para expressar corretamente no modelo o relacionamento entre as classes família e dimensão, uma outra classe auxiliar foi adicionada ao modelo para armazenar os atributos de ligação entre as duas classes.

Os principais atributos da classe família são representados na tabela 6.7.

TABELA 6.7 - Atributos da Classe **DimensãoFamília**

Atributo	Descrição
Descrição	Descrição da dimensão dentro da família.

Semântica:

1. Uma instância da classe **DimensãoFamília** está associada a uma, e somente uma instância da classe **Família**;
2. Uma instância da classe **DimensãoFamília** está associada a uma, e somente uma instância da classe **Dimensão**.

Classe **ValorDimensão**

Toda dimensão possui um valor de acordo com a família e o produto que está sendo utilizado. Da mesma forma que foi demonstrado acima optou-se pela criação de atributos de ligação para valorar as dimensões de um produto.

Os principais atributos da classe valor dimensão são representados na tabela 6.8.

TABELA 6.8 - Atributos da Classe **ValorDimensão**

Atributo	Descrição
Valor	Valor informado da dimensão do produto.

Semântica:

1. Uma instância da classe **ValorDimensão** está associada a uma, e somente uma, instância da classe **Produto**;
2. Uma instância da classe **ValorDimensão** está associada a uma, e somente uma, instância da classe **DimensãoFamília**.

Classe Desenho

Quando um produto está em fabricação um desenho se faz necessário para a produção do mesmo. Este desenho pode acompanhar a ordem de fabricação ou estar fixo no centro de trabalho.

Os principais atributos da classe Desenho são representados na tabela 6.9.

TABELA 6.9 - Atributo da Classe **Desenho**

Atributo	Descrição
DataInicial	Data de criação do desenho.
DataRevisão	Data que o desenho foi revisado.
DataLiberação	Data que o desenho foi liberado para a produção.
Formato	Formato utilizado no desenho. Ex.: A3, A4.
Extensão	Tipo de arquivo que está armazenado o desenho. Exemplo: PCX, BMP, PCL.

Semântica:

1. Uma instância da classe **Desenho** está associada a uma, e somente uma, instância da classe **Produto**;
2. Uma instância da classe **Desenho** está associada a uma ou mais instâncias da classe **Funcionário**.

Classe EstruturaProduto

A estrutura de um produto armazena toda a formação do produto, ou seja, todos os componentes necessários para a fabricação do produto final.

Um produto pode ou não ter uma estrutura de produto, assim como um item pode se fazer presente em mais de uma estrutura de produto.

Os principais atributos da classe estrutura de produto são representados na tabela 6.10.

TABELA 6.10 - Atributos da Classe **EstruturaProduto**

Atributo	Descrição
Quantidade	Quantidade de produtos filhos presentes no produto pai.
Nível	Nível do produto na estrutura do produto.
Seqüência	Seqüência ou posição do produto na estrutura.
DataCadastro	Data de inclusão do produto na estrutura.
DataExclusão	Data de exclusão do produto na estrutura.
Revisão	Contém a revisão da estrutura. Uma estrutura de produto pode sofrer alterações durante seu ciclo de vida.

Semântica:

1. Uma instância da classe **EstruturaProduto** está associada a uma ou mais instâncias da classe **Produto**, no papel de pai e filho.

Classe RoteiroFab

Quando um produto é desenvolvido ou criado, também são descritas todas as atividades necessárias para a fabricação deste produto. Isto é, o roteiro das atividades a serem seguidas durante o processo de produção para a elaboração do produto final.

Os principais atributos da classe roteiro de fabricação são representados na tabela 6.11.

TABELA 6.11 - Atributos da Classe **RoterioFab**

Atributo	Descrição
Seqüência	Seqüência do roteiro de fabricação.
DataAprovação	Data de aprovação do roteiro.
Revisão	A revisão do roteiro serve para auxiliar na identificação do mês e ano das alterações efetuadas no roteiro. Exemplo: C00 significa revisão de março de 2000.
TempoUnitário	Tempo médio em minutos gasto para a realização de cada operação.
Duração	Duração em dias de cada atividade.
FreqüênciaInspeção	Quantidade de peças a serem inspecionadas a cada lote produzido.
InstrumentoInspeção	Instrumento a ser utilizado para a inspeção das peças produzidas.

Semântica:

1. Uma instância da classe **RoteiroFab** está associada a uma ou mais instâncias da classe **RoteiroFabPosterior** sob dois papéis: atual e posterior;
2. Uma instância da classe **RoteiroFab** está associada a uma, e somente uma, instância da classe **Produto**;
3. Uma instância da classe **RoteiroFab** está associada a zero ou mais instâncias da classe **OF**;
4. Uma instância da classe **RoteiroFab** está associada a uma ou mais instâncias da classe **CentroTrab**.

Classe RoteiroFabPosterior

A classe roteiro de fabricação pode estar associada a classe roteiro de fabricação posterior. É necessário identificar a seqüência em que as atividades no roteiro de fabricação serão executadas.

Os principais atributos da classe roteiro de fabricação são representados na tabela 6.12.

TABELA 6.12 - Atributos da Classe **RoteiroFabPosterior**

Atributo	Descrição
Descrição	Descrição da atividade e sua atividade posterior.

Semântica:

1. Uma instância da classe **RoteiroFabPosterior** está associada a uma, e somente uma instância da classe **RoteiroFab** sob dois papéis: atual e posterior;

Classe OF (Ordem de Fabricação)

Uma ordem de fabricação é gerada para cada produto fabricado. Nela se encontram as principais informações para a fabricação do produto.

Os principais atributos da classe ordem de fabricação são representados na tabela 6.13.

TABELA 6.13 - Atributos da Classe **OF**

Atributo	Descrição
DataEmissão	Data de emissão da ordem de fabricação. Data em que ela é impressa.
DataEntrega	Data prevista para a conclusão da ordem de fabricação.
QuantidadePlanejada	Quantidade de produtos planejados para produzir.
QuantidadeSaldo	Quantidade de produtos restantes para serem produzidos
Situação	As Ordens de fabricação podem apresentar as seguintes situações: A – aguardando liberação; I – impressa, ou seja, em processo de fabricação; C – concluída; E – excluída.
Observação	Observações a serem descritas sobre a ordem de fabricação.

Semântica:

1. Uma instância da classe **OF** está associada a uma, e somente uma, instância da classe **RoteiroFab**;
2. Uma instância da classe **OF** está associada a uma ou mais instâncias da classe **ItemPedidoVenda**;
3. Uma instância da classe **OF** está associada a uma, e somente uma, instância da classe **Produto**;
4. Uma instância da classe **OF** está associada a uma ou mais instâncias da classe **AtividadeOF**;
5. Uma instância da classe **OF** está associada a uma ou mais instâncias da classe **HistóricoOF**.

Classe AtividadeOF

Os produtos em produção passam por diversas atividades de fabricação. A classe atividades da ordem de fabricação foi criada para armazenar as informações sobre as datas de início e fim de cada atividade no respectivo centro de trabalho.

Os principais atributos da classe atividades da ordem de fabricação são representados na tabela 6.14.

TABELA 6.14 - Atributos da Classe **AtividadeOF**

Atributo	Descrição
Descrição	Descrição da atividade.
DataInício	Data prevista para início da atividade.
DataFinal	Data prevista para finalização da atividade.

Semântica:

1. Uma instância da classe **AtividadeOF** está associada a uma, e somente uma, instância da classe **OF**.

Classe HistóricoOF

Com o intuito de comparar e aperfeiçoar o escalonamento realizado em chão-de-fábrica é necessário armazenar as atividades ocorridas nos centros de trabalho e suas respectivas máquinas. Com isso também é possível verificar o índice de produtividade nos centros de trabalho e nas máquinas.

Os principais atributos da classe histórico das ordens de fabricação são representados na tabela 6.15.

TABELA 6.15 - Atributos da Classe **HistóricoOF**

Atributo	Descrição
DataInicial	Data de início da operação.
HoraInicial	Horário de início da operação.
DataFinal	Data de finalização da operação.
HoraFinal	Horário de término da operação.
SetAtual	Centro de trabalho atual da operação.
SetDestino	Centro de trabalho destino da operação.
Quantidade	Quantidade movimentada da OF.

Semântica:

1. Uma instância da classe **HistóricoOF** está associada a uma, e somente uma, instância da classe **OF**;
2. Uma instância da classe **HistóricoOF** está associada a uma ou mais instâncias da classe **Funcionário**.

Classe Funcionário

A classe funcionário se faz necessária para se identificar os funcionários que realizaram as operações nas ordens de fabricação, projetaram ou realizaram as modificações nos desenhos.

Os principais atributos da classe funcionário são representados na tabela 6.16.

TABELA 6.16 - Atributos da Classe **Funcionário**

Atributo	Descrição
Nome	Nome do funcionário.
Função	Função do funcionário na empresa.
Setor	Setor onde o funcionário exerce suas atividades na empresa.

1. Uma instância da classe **Funcionário** está associada a zero ou mais instâncias da classe **HistóricoOF**;
2. Uma instância da classe **Funcionário** está associada a zero ou mais instâncias da classe **Desenho**.

ClasseCentroTrabalho

Um produto durante seu processo de fabricação atravessa por um ou mais centros de trabalho até se tornar um produto pronto. Um centro de trabalho pode realizar operações em somente um tipo de produto ou vários tipos de produtos dependendo de suas características. Consideramos um centro de trabalho ou célula de produção quando este possuir um conjunto de máquinas similares.

Os principais atributos da classe Centro de Trabalho são representados na tabela 6.17.

TABELA 6.17 - Atributos da Classe **CentroTrabalho**

Atributo	Descrição
Descrição	Descrição do grupo de máquinas.
TempoMédioPreparação	Tempo médio de preparação considerado no centro de trabalho.
ÍndiceProdutividade	Índice de produtividade no centro de trabalho.
NumFuncDia	Número de funcionários que trabalham no turno do dia.
NumFuncNoite	Número de funcionários que trabalham no turno da noite.
CustoHora	Valor da hora dispensada pelo centro de trabalho.
Gargalo	Indica se este centro de trabalho é gargalo ou não.

Semântica:

1. Uma instância da classe **CentroTrabalho** pode ser especializada em instâncias da classe **HorasAdicionais**, da classe **Terceiros** ou da classe **HorasMaqParada**;
2. Uma instância da classe **CentroTrabalho** está associada a zero ou mais instâncias da classe **RoteiroFab**;
3. Uma instância da classe **CentroTrabalho** está associada a uma ou mais instâncias da classe **Máquina**;
4. Uma instância da classe **CentroTrabalho** está associada a zero ou mais instâncias da classe **Restrição**;
5. Uma instância da classe **CentroTrabalho** está associada a uma, e somente uma, instância da classe **CalendárioFab**.

Classe HorasAdicionais

Os centros de trabalho possuem capacidade limitada de produção, uma maneira de aumentar a capacidade do centro é a realização de horas extras, possibilitando assim o incremento de horas disponíveis para o planejamento da produção.

Os principais atributos da classe horas adicionais são representados na tabela 6.18.

TABELA 6.18 - Atributos da Classe **HorasAdicionais**

Atributo	Descrição
DataInicialExtra	Data inicial que será efetuada a hora extra.
DataFinalExtra	Data final que será efetuada a hora extra.
NumFuncExtra	Número de funcionários que irão realizar as horas extras.

Semântica:

1. A classe **HorasAdicionais** é uma especialização da classe **CentroTrabalho**.

Classe Terceiros

A empresa possui centros de trabalho que muitas vezes não conseguem absorver todo o trabalho requerido pelos pedidos, e conseqüentemente pelas OF (Ordens de Fabricação) necessitando desta forma terceirizar algumas horas com empresas prestadoras de serviços. Deste modo os pedidos podem ser entregues conforme o prazo de entrega estabelecido anteriormente.

Esta classe se faz necessário para ampliar as horas disponíveis nos centros de trabalho gargalo, liberando desta forma o planejamento para novos pedidos.

Os principais atributos da classe terceiros são representados na tabela 6.19.

TABELA 6.19 - Atributos da Classe **Terceiros**

Atributo	Descrição
Descrição	Descrição do serviço terceirizado.
HorasAdicionais	Horas a serem terceirizadas do serviço e adicionadas na capacidade do centro de trabalho.

Semântica:

1. A classe **Terceiros** é uma especialização da classe **CentroTrabalho**.

Classe HorasMaqParada

As máquinas podem sofrer paradas repentinas ou manutenções, ocasionando assim uma inatividade do centro de trabalho por tempo indeterminado. Estas horas de inatividade são de suma importância para o cálculo de capacidade da produção e do escalonamento das ordens de fabricação.

Os principais atributos da classe horas máquina parada são representados na tabela 6.20.

TABELA 6.20 - Atributos da Classe **HorasMaqParada**

Atributo	Descrição
DataInicialMaqParada	Data de início que a máquina ficará parada.
HoraInicialMaqParada	Hora de início que a máquina ficará parada.
DataFinalMaqParada	Data de término que a máquina ficará parada.
HoraFinalMaqParada	Hora de término que a máquina ficará parada.

Semântica:

1. A classe **HorasMaqParada** é uma especialização da classe **CentroTrabalho**.

Classe Máquina

Todos os centros de trabalho possuem máquinas e estas possuem características próprias para cada tipo de operação num determinado produto.

Os principais atributos da classe máquina são representados na tabela 6.21.

TABELA 6.21 - Atributos da Classe **Máquina**

Atributo	Descrição
Descrição	Descrição da máquina.
Local	Setor ou fábrica onde se localiza a máquina.
Setup	Setup da máquina, ou seja, o tempo médio de preparação da máquina.
Data de Aquisição	Data de compra da máquina.
Situação	A máquina possui as seguintes situações: A – Ativa P – Parada M – Manutenção

Semântica:

1. A classe **Máquina** está associada a uma, e somente uma, instância da classe **CentroTrabalho**.

Classe Restrição

Nos centros de trabalhos existem máquinas que possuem restrições para realizarem determinadas operações sobre os produtos, como por exemplo, o comprimento ou o diâmetro externo de uma peça não é suportada pela máquina. Também é comum encontrar produtos com um índice de produtividade maior em determinadas máquinas do centro de trabalho.

Os principais atributos da classe restrição podem ser visualizados na tabela 6.22.

TABELA 6.22 - Atributos da Classe **Restrição**

Atributo	Descrição
Prioridade	Prioridade informada para a máquina no centro de trabalho para determinada família.
FaixaQuantidade	Intervalo que compreende a restrição de quantidade.
FaixaDimensão	Intervalo que compreende a restrição de dimensão.

Semântica:

1. A classe **Restrição** está associada a uma ou mais instâncias da classe **Família**;

2. A classe **Restrição** está associada a uma ou mais instâncias da classe **CentroTrabalho**.

Classe CalendárioFab

O classe calendário fabril se faz necessário para a programação da produção por semanas de trabalho.

Os principais atributos da classe calendário fabril são representados na tabela 6.23.

TABELA 6.23 - Atributos da Classe **CalendárioFab**

Atributo	Descrição
Semana	Semana a qual pertence determinada data.
DataInicialSemana	Data inicial da semana no calendário fabril.
DataFinalSemana	Data final da semana no calendário fabril.

Semântica:

1. Uma instância da classe **CalendárioFab** está associada a uma ou mais instâncias da classe **CentroTrabalho**;
2. Uma instância da classe **CalendárioFab** está associado a uma ou mais instâncias da classe **HorárioFab**;
3. Uma instância da classe **CalendárioFab** está associado a zero ou mais instâncias da classe **Feriado**;
4. Uma instância da classe **CalendárioFab** está associado a zero ou mais instâncias da classe **CapacidadeFab**.

Classe HorárioFab

O horário de trabalho de cada dia da semana e turno se faz necessário para o escalonamento das ordens de fabricação.

Os principais atributos da classe horário fabril são representados na tabela 6.24.

TABELA 6.24 - Atributos da Classe **HorárioFab**

Atributo	Descrição
Turno	Turno a qual pertence o horário.
DiaSemana	Dia da semana a qual pertence o horário.
HoraInício	Horário de início da jornada.
HoraFinal	Horário de término da jornada.

Semântica:

1. Uma instância da classe **HorárioFab** está associada a uma ou mais instâncias da classe **CalendárioFab**.

Classe Feriado

Durante o ano existem diversos feriados e também podem existirem feriados programados pela empresa, ocasionando dias de inatividade para o calendário fabril.

Os principais atributos da classe feriado são representados na tabela 6.25.

TABELA 6.25 - Atributos da Classe **Feriado**

Atributo	Descrição
DataFeriado	Data do calendário que representa um dia de inatividade para o calendário fabril.

Semântica:

1. Uma instância da classe **Feriado** está associada a uma, e somente uma, instância da classe **CalendárioFab**.

Classe CapacidadeFab

Considerando que a fábrica possui capacidade finita de produção, é necessário conhecer quantas horas estão alocadas para os centros de trabalho considerados gargalos de produção, pois é através destes centros que controlaremos a capacidade de produção.

Os principais atributos da classe capacidade fabril são representados na tabela 6.26.

TABELA 6.26 - Atributos da Classe **CapacidadeFab**

Atributo	Descrição
HorasAlocadas	Número de horas alocadas na semana para determinado centro de trabalho considerado gargalo.

Semântica:

1. Uma instância da classe **CapacidadeFab** está associada a uma ou mais instâncias da classe **CalendárioFab**.

6.2 Implementação do Algoritmo de Escalonamento

6.2.1 Introdução

Com a finalidade de implementar o modelo de dados apresentado anteriormente, foi desenvolvido um sistema-piloto para o escalonamento do centro de trabalho de corte. O sistema de escalonamento desenvolvido interage com um sistema de gestão em uso na empresa, sendo que, atualmente a mesma utiliza o SGDB ZIM 4.20 como banco de dados e linguagem de programação [ZIM2000].

A ferramenta utilizada para o desenvolvimento do sistema de escalonamento e controle da produção foi o *Visual Basic 5.0*, o qual, através de ODBC (*Open Database Connectivity*), acessa e manipula os dados do sistema de gestão [VIS2001].

Não é objetivo deste estudo demonstrar todo o sistema de produção e os módulos necessários para o seu perfeito funcionamento, mas somente o módulo desenvolvido para controlar a capacidade de produção e o algoritmo utilizado para realizar o escalonamento no centro de trabalho de corte.

De um modo geral, o sistema tem seu início na inclusão de um pedido de vendas, no qual não se possui estoque do produto final para atender. Será necessário produzir o mesmo para suprir a necessidade do cliente. Na inclusão do pedido de vendas, é realizada a verificação da capacidade fabril, somente das células consideradas gargalos de produção. Desse modo o Departamento de Vendas terá condições de aceitar o pedido ou sugerir uma nova data de entrega para o mesmo, respeitando as limitações da fábrica.

A seguir são geradas as ordens de fabricação de cada item componente do pedido, seguindo a estrutura do item cadastrada pela engenharia do produto, com suas respectivas datas-limite de cada processo incluso no roteiro de fabricação. O sistema também informa quais ordens de fabricação não possuem estoque para iniciar a produção. Tem-se então a opção de iniciar a produção parcial da ordem de fabricação ou aguardar a chegada da matéria-prima para iniciar a produção da mesma.

O escalonamento é gerado levando em consideração o centro de trabalho a ser escalonado; no sistema desenvolvido, está sendo testado o escalonamento no centro de trabalho de corte. Também é necessário informar o período de entrega das ordens de fabricação e selecionar a heurística para a resolução dos conflitos. Esse procedimento será explicado em detalhes no item 6.2.3.

O escalonamento também pode sofrer modificações, através de alterações das datas-limite pelo usuário ou através de perturbações ocorridas no chão-de-fábrica, como por exemplo, uma máquina parada ou em manutenção. Mas, para visualizar esse novo ambiente, deve-se gerar novamente o escalonamento. O fluxo geral do sistema de escalonamento está mostrado na figura 6.2.

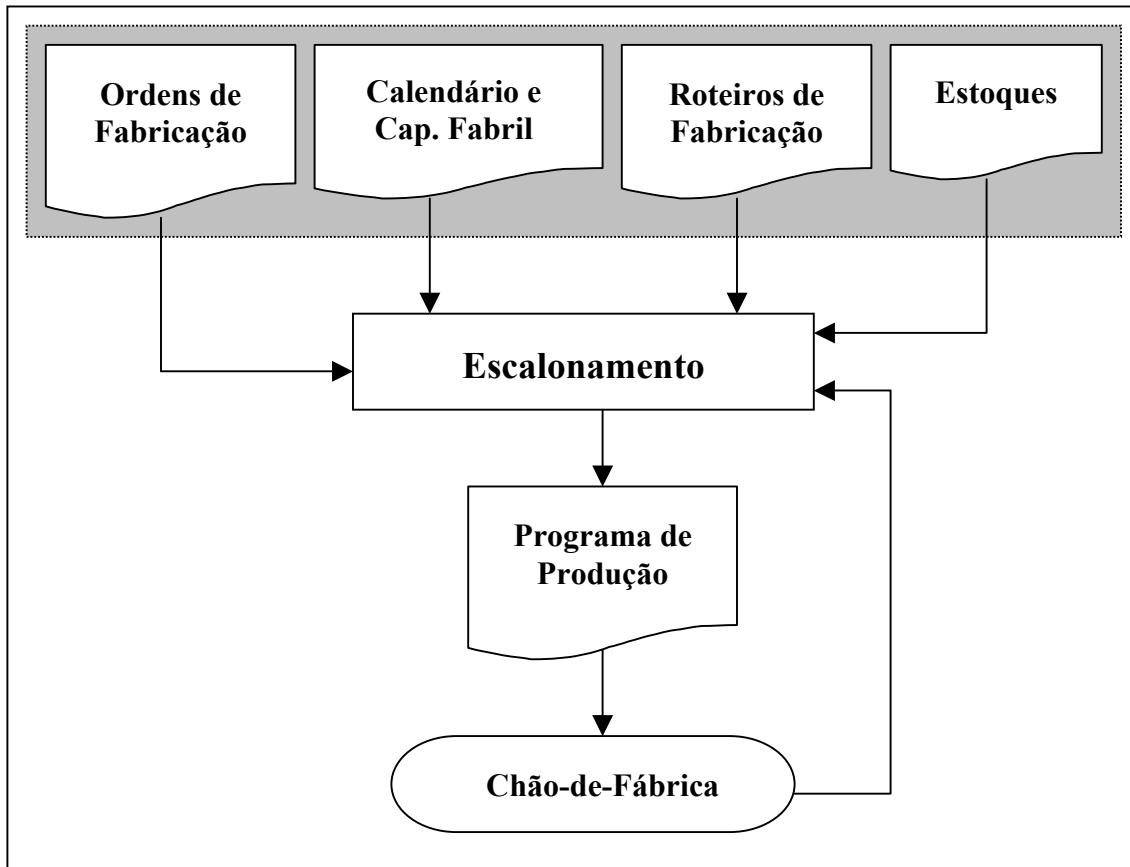


FIGURA 6.2 - Fluxo geral do sistema

6.2.2 Análise da Capacidade Fabril

A capacidade é a quantidade de trabalho alocado para um centro de trabalho. Por exemplo, um determinado centro de trabalho numa fábrica está disponível, em teoria, 429,07 horas por semana, como demonstrado na tabela 6.27. Contudo, isso não significa necessariamente que as 429,07 horas de trabalho podem ser alocadas para esse centro.

As classes do modelo de dados que estão envolvidas no controle da capacidade fabril são: **CalendárioFab**, **Feriado**, **HorárioFab**, **Máquina**, **CentroTrabalho**, **HorasAdicionasis**, **Terceiros**, **HorasMaqParada**.

Por alguns períodos algumas máquinas desse centro podem não trabalhar, um funcionário pode não vir trabalhar, a logística de materiais até as máquinas do centro pode variar. Portanto, a carga sobre o centro de trabalho e, conseqüentemente, sobre as máquinas, deve levar em conta todos esses fatores. Do tempo em que a máquina está disponível para o trabalho, outras tarefas que não as de produção precisam ser desempenhadas, reduzindo muito mais o tempo disponível. Por exemplo, tempo perdido na troca entre fazer um componente e outro diferente.

Além disso, as máquinas podem precisar de limpeza entre as operações ou até mesmo uma manutenção. Essas perdas de tempo também devem ser levadas em consideração quando é formado um plano de quanta carga será posta sobre o centro de trabalho. Se a máquina quebrar, não estará disponível. Se houver dados disponíveis de confiabilidade da máquina, eles também devem ser levados em consideração.

A maioria dos sistemas de planejamento existentes assume como pressuposto que os recursos de produção estarão sempre disponíveis, como se eles tivessem capacidade infinita. Os sistemas MRP (*Material Requirements Planning*) usualmente explodem os produtos em componentes e, com base em seus *lead times* (tempos de passagem), estabelecem quando iniciar a produção de cada um deles. Não levam, entretanto, em consideração se a capacidade dos recursos está efetivamente disponível naquele momento.

Para controlar a capacidade da fábrica foi limitado o número de horas disponíveis por semana nos centros de trabalho considerados como gargalos de produção. O cálculo da capacidade nesses centros leva em consideração o número de máquinas que trabalham no centro de trabalho, o índice de eficiência de cada máquina, o número de horas trabalhadas na semana e a eficiência do mesmo.

A criação de um índice de eficiência foi necessária para se diminuir as horas em que os funcionários não estão diretamente produzindo na máquina, ou seja, eles podem estar realizando atividades que não estão diretamente relacionadas com a produção de algum produto na máquina.

Também se pode ter dentro do centro de trabalho alguma máquina que possua uma velocidade de processamento superior às demais, para a produção da mesma peça. Nesse caso há necessidade de ser atribuído um índice, ou seja, um fator de multiplicação que irá adequar as horas disponíveis no centro.

O centro de trabalho de corte possui quatro máquinas, sendo que uma máquina possui velocidade superior às demais; utilizou-se então o fator de multiplicação de 2,5 nessa máquina para realizar o ajuste das horas disponíveis. No período do dia trabalham quatro homens nas quatro máquinas em tempo integral, e, durante o período da noite, trabalham dois homens, que se revezam em três máquinas. Desse modo se tem 5,5 máquinas operando no turno do dia, que possui 44 horas e 4,5 máquinas operando no turno da noite, que possui 41,77 horas. Multiplicando-se as máquinas do dia com suas respectivas horas disponíveis e as máquinas da noite, obtém-se 429,07 horas de capacidade total na semana. Aplicando-se, após, o índice de eficiência do centro, tem-se 364,71 horas de capacidade real no centro.

A tabela 6.27 representa a capacidade de produção no centro de trabalho de corte.

TABELA 6.27 - Capacidade do centro de trabalho de corte

Centro de Trabalho	Preparação Média	Número de Máquinas Dia/Noite		Horas Disponíveis Dia/Noite		Índice de Eficiência	Capacidade Semanal em Horas	Capacidade Semanal Real em Horas
Corte	7,5 min.	5,5	4,5	44	41,57	0,85	429,07	364,71

A capacidade fabril possui flexibilidade em sua capacidade total, desse modo, é permitido adicionar pedidos que acarretem um acréscimo de até 10 % na capacidade fabril. Essa flexibilidade é concebida devido à possibilidade de horas extras e à prestação de serviços por terceiros.

O sistema permite visualizar a disponibilidade dos centros de trabalho considerados como gargalos; a figura 6.3 demonstra este recurso do sistema. Como o sistema trabalha por semanas, informando-se uma data inicial, o mesmo já localiza essa data na semana do calendário, calcula os atrasos existentes e projeta, cinco semanas para frente, as horas disponíveis e alocadas com as ordens de fabricação já geradas e confirmadas.

O sistema também utiliza um calendário fabril, onde os feriados, as férias coletivas e outras datas previamente cadastradas são levadas em conta para a geração da capacidade dos centros de trabalho. No exemplo demonstrado na figura 6.3, tem-se um feriado no dia 14 de abril de 2001; portanto, a semana quinze do calendário possui sua capacidade reduzida em um dia de trabalho.

Consulta Capacidade dos Gargalos											
Data Inicial:		11/03/2001		Data Final:		14/04/2001					
Atraso		Semana/11		Semana/12		Semana/13		Semana/14		Semana/15	
11/03		11/03-17/03		18/03-24/03		25/03-31/03		01/04-07/04		08/04-14/04	
Máquina	Atr.	Disp.	Ocup.	Disp.	Ocup.	Disp.	Ocup.	Disp.	Ocup.	Disp.	Ocup.
▶ Corte	2,5	364,7	370,3	364,7	225,8	364,7	115,4	364,7	178,9	305,7	138,3
Brunidora	4,27	207,3	195,4	207,3	203,9	207,3	139,2	207,3	41,4	174,5	113

FIGURA 6.3 - Tela de capacidade dos gargalos

Aumentos de capacidade podem ser alcançados: antecipando-se as ordens de fabricação para aproveitar os recursos subutilizados; alterando rotas para evitar gargalos; estendendo turnos de trabalho; subcontratando a produção ou fazendo a divisão de lotes. Por outro lado, reduções de capacidade são causadas por quebras, retrabalhos, tempos de *setup* e pelos inevitáveis gargalos.

No sistema proposto, para a realização do incremento da capacidade de produção criou-se uma tela onde devem ser informados: o centro de trabalho, a máquina onde vai ser realizada as atividades, a data e hora inicial e também a data e hora final.

Na figura 6.4 se está realizando um incremento de capacidade do centro de trabalho de corte, onde estão sendo adicionadas quatro horas e trinta minutos na máquina Serra Fita 291. Com isso se está disponibilizando horas de trabalho para solucionar o atraso que ficou da semana dez, conforme figura 6.3.

Máquina	Descrição
291	Serra Fita - 291
607	Serra Circular
529	Serra Fita - 529
568	Serra Fita - 568

Data Inicial: 10/03/2001 Hora Inicial: 07:15:00
Data Final: 10/03/2001 HoraFinal: 11:45:00

FIGURA 6.4 - Tela de aumento da capacidade fabril

Para a redução da capacidade fabril, criou-se no sistema uma tela idêntica a apresentada anteriormente, com informações relativas às máquinas paradas e também possibilitando a inclusão das manutenções preventivas. Desse modo se garante a redução das horas disponíveis nos centros de trabalhos aos quais essas máquinas pertencem.

Na figura 6.5 foi informada uma manutenção no dia 23 de março de 2001 para a máquina Serra Fita 291, pertencente ao centro de trabalho de corte. Com isso, as horas disponíveis na semana doze do calendário fabril será reduzida.

Máquina	Descrição
291	Serra Fita - 291
607	Serra Circular
529	Serra Fita - 529
568	Serra Fita - 568

Data Inicial: 23/03/2001 Hora Inicial: 17:30:00
Data Final: 23/03/2001 HoraFinal: 20:30:00

FIGURA 6.5 - Tela de redução da capacidade fabril

O acréscimo de horas no centro de trabalho de corte, através da inclusão de quatro horas extras e trinta minutos no dia 10 de março de 2001, como foi demonstrado na figura 6.4, se faz necessário para eliminar as horas que ficaram em atraso da semana dez do calendário fabril e também para amenizar o problema da semana onze, que já está com sua ocupação acima das horas disponíveis, como demonstra a figura 6.3.

Também foi incluída uma manutenção na máquina Serra Fita 291, pertencente ao centro de corte, no dia 23 de março de 2001, gerando uma redução na capacidade desse centro em quatro horas. Essa manutenção está demonstrada na figura 6.5.

Logo após essas inclusões no centro de trabalho de corte, uma nova tela de capacidade produtiva pode ser visualizada, esse novo cenário está demonstrado na figura 6.6, onde não existem mais atrasos, e a semana doze teve sua capacidade disponível reduzida no respectivo centro de trabalho.

Consulta Capacidade dos Gargalos

Data Inicial: 11/03/2001 Data Final: 14/04/2001

	Atraso	Semana/11		Semana/12		Semana/13		Semana/14		Semana/15	
	11/03	11/03-17/03	18/03-24/03	25/03-31/03	01/04-07/04	08/04-14/04					
Centro	Atr.	Disp.	Ocup.	Disp.	Ocup.	Disp.	Ocup.	Disp.	Ocup.	Disp.	Ocup.
✎ Corte	0	364,7	368,3	361,7	225,8	364,7	115,4	364,7	178,9	305,7	138,3
✎ Brunidora	4,27	207,3	195,4	207,3	203,9	207,3	139,2	207,3	41,4	174,5	113

OK Fim

FIGURA 6.6 - Tela de capacidade dos gargalos

6.2.3 O Algoritmo de Escalonamento

O sistema dispõe de um conjunto de pedidos com itens a fabricar, cada um com sua respectiva quantidade e data de entrega. Cada item do pedido possui um roteiro de fabricação, cada atividade é executada em um centro de trabalho (célula de produção), possuindo um tempo médio de fabricação (tempo de passagem).

Obtém-se a data-limite para cada atividade retornando-se da data de entrega do pedido para a primeira atividade e subtraindo-se os tempos de passagem, obtendo-se assim, para cada centro de trabalho, as suas atividades com a respectiva data-limite (data mais tarde) de conclusão.

A metodologia utilizada para o escalonamento das atividades a realizar em um centro de trabalho consiste em construir uma fila inicial selecionando todas as ordens de fabricação que passam pelo respectivo centro e ordenando de modo crescente pela data-limite.

Os conflitos gerados por similaridade de data-limite são resolvidos através de regras heurísticas previamente selecionadas; nesse caso se utiliza o critério de maior *makespan*, ou seja, a peça que possui o maior tempo de produção da primeira atividade até a última atividade.

O sistema também utilizará restrições entre as peças e as máquinas do centro a ser escalonado para a obtenção da melhor alocação da ordem de fabricação ao recurso de produção. Essas restrições levam em conta a prioridade, a quantidade e as dimensões da peça em relação às máquinas do centro de trabalho.

De acordo com as restrições, cada máquina possui sua própria fila, e a cada nova atividade tenta-se primeiro colocá-la logo após a uma outra atividade sobre um item similar. Com isso consegue-se reduzir o tempo de *setup*. Esse mecanismo permite que algumas ordens de fabricação furem a fila, atrasando assim outras ordens já previamente alocadas.

Se o atraso é suportável para as demais ordens da fila, do ponto de vista da data-limite de produção, então a inserção pode ser feita e economiza-se a preparação. Senão, tenta-se inserir a ordem em outro ponto da fila, sempre buscando a similaridade e respeitando a data-limite das demais ordens já inseridas na fila. Na pior das hipóteses, a ordem será inserida no final da fila, e o processo continua automaticamente até se chegar no fim da fila do centro de trabalho a ser escalonado.

O algoritmo está representado da figura 6.7.

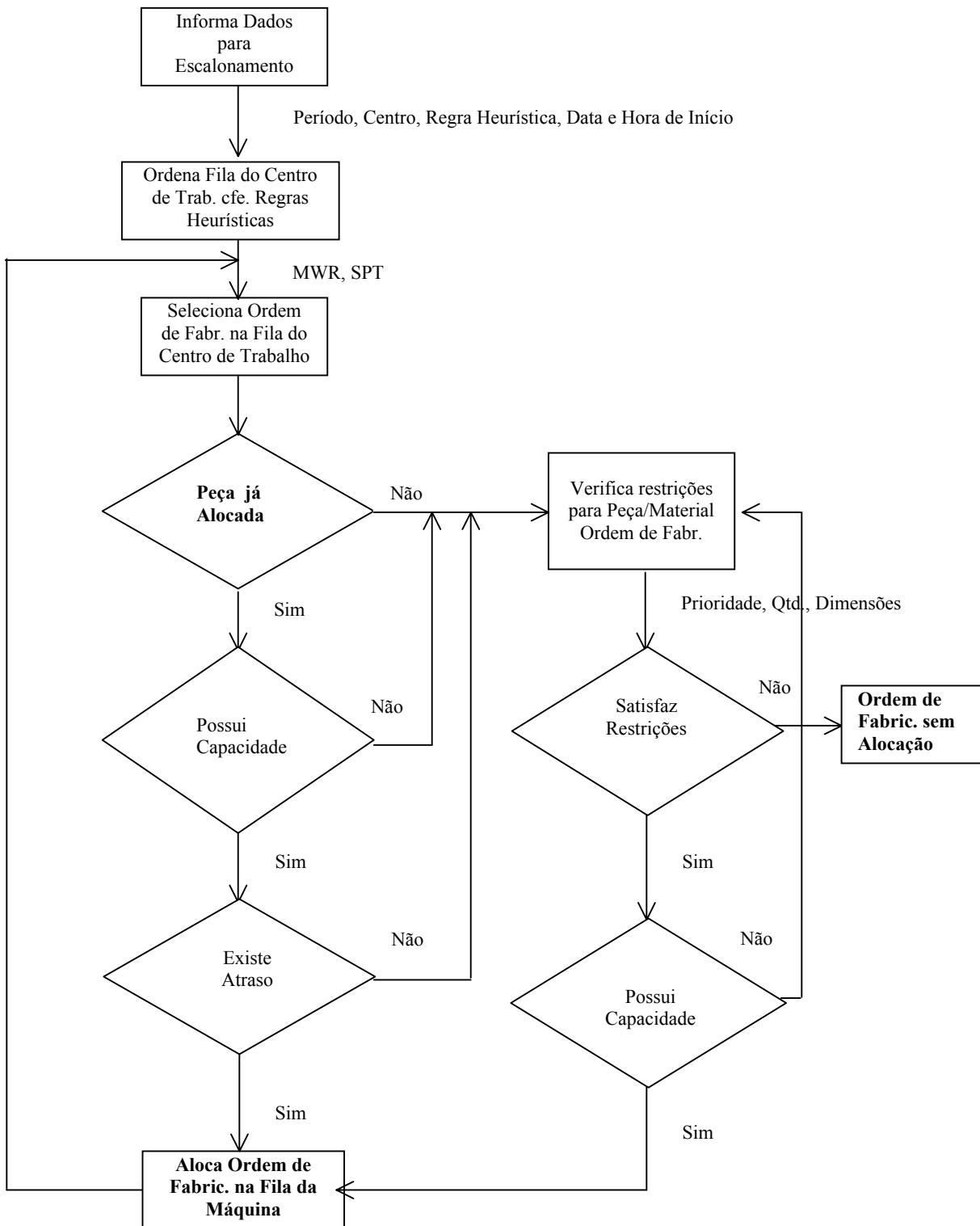


FIGURA 6.7 - Algoritmo de escalonamento

A seguir será detalhado o algoritmo descrito acima.

O primeiro passo do algoritmo de escalonamento é realizar a seleção de todas as ordens de fabricação com datas-limite de corte no período informado para realizar o escalonamento.

- Calcula tempo de cada ordem de fabricação.
Para cada ordem de fabricação é calculado o tempo para a operação de corte e o tempo total de todas as operações restantes.
- Grava ordens de fabricação da fila do centro de trabalho.
Após o cálculo dos tempos das ordens de fabricação, é gerada uma fila do centro de trabalho com todas as ordens e seus respectivos tempos.
- Ordena fila do centro de trabalho
Se a regra heurística é MWR
Então
 Ordena fila do centro de trabalho, posicionando a peça com maior tempo total de processo no início da fila.
Fim Se

Se a regra heurística é SPT
Então
 Ordena fila do centro de trabalho, posicionando a ordem de fabricação com menor tempo total de processo no início da fila.
Fim Se
- Calcula o tempo disponível no período
Conforme o período informado, o sistema verifica, no calendário fabril, as horas extras a serem realizadas e as horas perdidas com o agendamento de máquinas em manutenção. Após, calcula o número de horas disponíveis para realizar o escalonamento em cada máquina do centro de trabalho.
- Grava tempo total para cada máquina
Grava um registro na fila de cada máquina do centro com as informações das horas totais disponíveis, conforme cálculo realizado pela rotina anterior.

Enquanto fila do centro maior que zero

 Seleciona ordem de fabricação

- Testa Matéria-Prima da Ordem de Fabricação
Verifica se a matéria-prima da ordem de fabricação já está alocada em alguma fila das máquinas do centro. Com isso tenta-se reduzir o tempo de preparação gasto na máquina.

Se a matéria-prima existe em alguma fila do centro,

Então

- Verifica a capacidade e o atraso da fila
Rotina que verifica se com a inclusão da ordem de fabricação na fila da máquina, a mesma não ultrapassa a capacidade de horas da fila, conforme cálculo do tempo disponível no período informado. Esta rotina também verifica se com a inclusão da ordem de fabricação na fila, não ocorrerá o atraso nas demais ordens da fila.

Se fila possui capacidade de horas

Então

Se não atrasa as demais ordens da fila

Então

- Aloca ordem de fabricação da fila da máquina

Senão

- Seleciona Máquina

Senão

- Seleciona Máquina

Senão

- Seleciona Máquina

Fim Enquanto

Rotina que seleciona uma máquina para realizar a operação de corte da ordem de fabricação

- Seleciona Máquina
A rotina Seleciona Máquina tem a função de relacionar todas as restrições referentes à peça e à matéria-prima da ordem de fabricação. Também leva em consideração a prioridade de cada matéria-prima com relação as máquinas do centro de trabalho.
- Ordena a prioridade da matéria-prima em sentido crescente dentro do centro

Enquanto restrição for maior que zero

Se satisfazem as restrições

Então

Se possui capacidade na fila da máquina

Então

- Aloca ordem de fabricação da fila da máquina

Senão

- Seleciona Máquina

Senão

- Aloca ordem de fabricação na fila de ordens sem máquina

Fim Enquanto

As restrições das máquinas citadas anteriormente são de suma importância para o bom funcionamento do programa de escalonamento, pois é através dessas restrições que as peças serão alocadas nas máquinas do centro de trabalho.

Para todas as famílias de materiais ou produtos que necessitam ser cortados ou produzidos em algum centro de trabalho, há necessidade de informar as prioridades e as restrições para cada máquina do centro. A tela do sistema que realiza a manutenção dessas restrições está representada na figura 6.8.

Máquina	Descrição	Prioridade	Faixa Qtde.	Faixa Dim.
291	Serra Fita - 291	1	0 A 99999	00.00 A 99999.99
529	Serra Fita - 529	2	26 A 99999	24.99 A 88.90
568	Serra Fita - 568	3	0 A 99999	00.00 A 99999.99
607	Serra Circular	3	0 A 25	00.00 A 99999.99

FIGURA 6.8 - Tela das restrições das máquinas por família

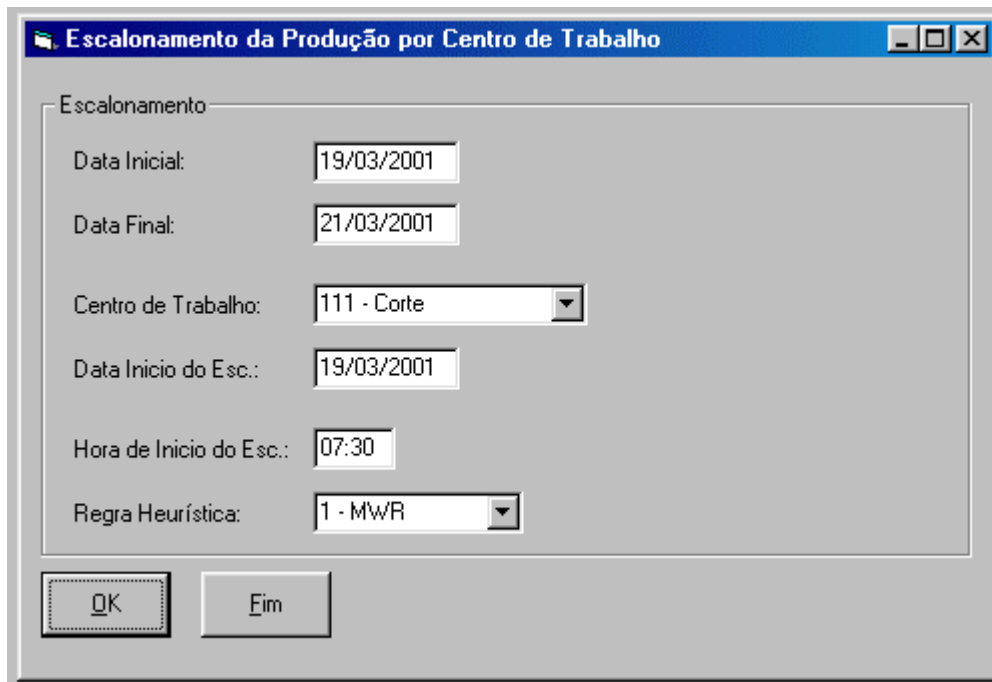
Na figura representada acima, o centro de trabalho corte é composto por quatro máquinas, sendo que cada máquina do centro possui prioridades e restrições para cada família de materiais. No exemplo da figura 6.8 foi selecionada a família de tubos trefilados e foram informadas suas restrições.

A máquina Serra Fita 291 possui prioridade maior que as demais máquinas do centro e não apresenta restrições de quantidade e dimensão para os produtos que a empresa produz. Por outro lado, a máquina Serra Fita 529 possui a segunda prioridade no centro para essa família e possui a restrição de produzir lotes acima de 26 peças e a restrição de dimensão. A máquina Serra Fita 568 possui a terceira prioridade dentro do centro e também não apresenta restrições de quantidade e dimensão. A máquina Serra Circular 607 possui a

terceira prioridade dentro do centro para a família selecionada, e apresenta somente a restrição de produzir lotes até 25 peças.

Através das prioridades e restrições, é possível verificar qual máquina será mais eficiente para determinada família de produtos e menos eficiente para outra família. Também uma peça nunca deixará de ser alocada a alguma máquina, mesmo que determinada máquina do centro esteja em manutenção ou parada.

Após o preenchimento de todas as informações relativas às máquinas e famílias no centro de trabalho, o sistema estará apto para iniciar o escalonamento de forma eficiente. A tela do sistema que possui as informações referentes ao início do escalonamento está representada na figura 6.9.



A imagem mostra uma janela de software intitulada "Escalonamento da Produção por Centro de Trabalho". O formulário contém os seguintes campos:

- Data Inicial: 19/03/2001
- Data Final: 21/03/2001
- Centro de Trabalho: 111 - Corte (menu suspenso)
- Data Inicio do Esc.: 19/03/2001
- Hora de Inicio do Esc.: 07:30
- Regra Heurística: 1 - MWR (menu suspenso)

Na base da janela, há dois botões: "OK" e "Fim".

FIGURA 6.9 - Tela de início do escalonamento para a regra MWR

Na figura 6.9 foi informada a data inicial e final para seleção das ordens de fabricação que possuem suas datas-limite no período informado. Também deve ser informado o centro de trabalho no qual se deseja realizar o escalonamento das peças; nesse projeto-piloto está sendo utilizado somente o centro de trabalho de corte, devendo posteriormente ser estendido a todos os demais centros de trabalho da planta fabril. A data e hora de início do escalonamento também são necessárias para o sistema.

Outra informação fundamental para o sistema são as regras heurísticas utilizadas para a realização do escalonamento. No exemplo acima foi escolhida a regra denominada MWR (*Most Work Remaining*), ou seja, durante o escalonamento, quando existir um

conflito entre duas ordens de fabricação com a mesma data-limite, o sistema irá priorizar a peça que possua o maior tempo de processo até a sua última atividade. A figura 6.10 representa o escalonamento gerado através das informações representadas na figura 6.9.

	NumOP	NumOF	Atr.	CodPro	Descrição	Qtde.	Dt.Início	Hr.Início	Dt.Fim	Hr.Fim
▶	76107	217295		20709	Camisa	40	19/03/01	07:30:00	19/03/01	10:04:00
	75852	215942		33385	Haste Ref	4	19/03/01	10:04:01	19/03/01	10:42:00
	75855	216042		39300	Haste Ref	4	19/03/01	10:42:01	19/03/01	11:02:00
	75846	215742		29547	Haste Ref	4	19/03/01	11:02:01	19/03/01	11:16:00
	76132	218040		28306	Camisa Re	20	19/03/01	11:16:01	19/03/01	12:00:00
	76132	218040		28306	Camisa Ref	20	19/03/01	13:30:00	19/03/01	14:15:00
	76129	217880		36245	Camisa	20	19/03/01	14:15:01	19/03/01	15:36:00
	76840	223304		17630	Camisa	30	19/03/01	15:36:01	19/03/01	16:59:00
	76106	217273		24874	Camisa	50	19/03/01	16:59:01	19/03/01	17:57:00
	76135	218143		20806	Camisa Ref	20	19/03/01	17:57:01	19/03/01	19:11:00
	76839	223279		25417	Camisa Ref	30	19/03/01	19:11:01	19/03/01	20:02:00
	75851	215903		33388	Camisa	4	19/03/01	20:02:01	19/03/01	20:26:00
	75854	216003		39292	Camisa	4	19/03/01	20:26:01	19/03/01	20:44:00
	75846	215733		29542	Camisa Ref	4	19/03/01	20:44:01	19/03/01	21:06:00
	75847	215764		29542	Camisa Ref	4	19/03/01	21:06:01	19/03/01	21:32:00
	75852	215933		33380	Camisa	4	19/03/01	21:32:01	19/03/01	21:48:00
	75853	215964		33380	Camisa	4	19/03/01	21:48:01	19/03/01	22:04:00

FIGURA 6.10 - Tela de escalonamento da máquina Serra Fita 291

Após a execução do programa o sistema permite escolher em qual máquina do centro de trabalho se deseja visualizar o escalonamento. Na figura 6.10 foi escolhido demonstrar o escalonamento da máquina Serra Fita 291, pertencente ao centro de trabalho de corte.

O sistema também permite selecionar outra regra heurística para a realização do escalonamento. Na figura 6.11 são informados os mesmos dados da figura 6.9, apenas selecionando a regra heurística SPT (*Short Process Time*), onde a resolução de conflitos prioriza a ordem de fabricação que possuir o menor tempo de fabricação.

Escalonamento da Produção por Centro de Trabalho

Escalonamento

Data Inicial: 19/03/2001

Data Final: 21/03/2001

Centro de Trabalho: 111 - Corte

Data Início do Esc.: 19/03/2001

Hora de Início do Esc.: 07:30

Regra Heurística: 2 - SPT

OK Fim

FIGURA 6.11 - Tela de início de escalonamento para a regra SPT

A regra heurística selecionada procura minimizar o tempo de fluxo médio, pois prioriza as peças com menor tempo total de processamento. Desse modo as peças são finalizadas o mais cedo possível no sistema de manufatura, baixando a média de permanência das peças na fábrica.

Também não se pode esquecer, que as regras heurísticas produzem resultados razoavelmente bons, mas não ótimos, ou seja, não se garante que não haja uma solução ainda melhor.

A figura 6.12 representa o escalonamento gerado na máquina Serra Fita 529, através das informações representadas na figura 6.11.

Atualmente no sistema estão implementadas somente duas regras heurísticas; MWR (*Most Work Remaining*) e SPT (*Short Process Time*). A escolha inicialmente destas regras se deve às características de produção onde o sistema está implementado e também devido aos testes do sistema estarem sendo realizados na célula de produção de corte, onde é iniciado o processo de produção.

Consulta o Escalonamento por Máquina

Código da Máquina
000529 - Serra Fita

	NumOP	NumOF	Atr.	CodPro	Descrição	Qtde.	Dt.Início	Hr.Início	Dt.Fim	Hr.Fim
▶	77105	225354		4778	Calco	1	19/03/01	07:30:00	19/03/01	07:44:00
	77105	225369		7847	Guia Ref	1	19/03/01	07:44:01	19/03/01	08:00:00
	77096	225310		26121	Tirante Ref	1	19/03/01	08:00:01	19/03/01	08:14:00
	75031	211102		11732	Conexao R	2	19/03/01	08:14:01	19/03/01	08:30:00
	77552	227998		6872	Cabeça	1	19/03/01	08:30:01	19/03/01	08:46:00
	75031	211100		4702	Fundo Ref	2	19/03/01	08:46:01	19/03/01	09:06:00
	76349	220629		41403	Base Ref	40	19/03/01	09:06:01	19/03/01	10:00:00
	75035	211391		13582	Anel Ref	1	19/03/01	10:00:01	19/03/01	10:21:00
	77400	227147		37608	Terminal	20	19/03/01	10:21:01	19/03/01	10:54:00
	75846	215737		19082	Fundo	4	19/03/01	10:54:01	19/03/01	11:19:00
	75852	215937		19082	Fundo	4	19/03/01	11:19:01	19/03/01	11:39:00
	75855	216037		19082	Fundo	4	19/03/01	11:39:01	19/03/01	11:58:00
	75847	215768		18918	Fundo	4	19/03/01	11:58:01	19/03/01	12:00:00
	75847	215768		18918	Fundo	4	19/03/01	13:30:00	19/03/01	13:45:00
	75856	216068		18918	Fundo	4	19/03/01	13:45:01	19/03/01	14:19:00
	76353	220680		29982	Embolo	6	19/03/01	14:19:01	19/03/01	14:54:00
	76060	217105		29262	Embolo	5	19/03/01	14:54:01	19/03/01	15:15:00

OK Fim Realizado

FIGURA 6.12 - Tela de escalonamento da máquina Serra Fita 529

6.2.4 Resultados Alcançados

Os resultados alcançados com a utilização do algoritmo foram obtidos através de tabelas comparativas, onde foram coletados no chão-de-fábrica duas semanas do escalonamento executado manualmente pelos operadores do centro de trabalho de corte, e comparados com a execução do algoritmo desenvolvido. A primeira amostragem teve como referência a semana de 08/05/2000 a 12/05/2000.

Após a coleta dos dados, os mesmos foram transcritos para uma planilha eletrônica, onde foram calculados os tempos totais de *setup* e de processo em cada uma das máquinas do centro de trabalho. A tabela 6.28 representa as horas obtidas da primeira amostragem executada pelo escalonamento manual.

TABELA 6.28 - Escalonamento manual da semana 08/05/2000 a 12/05/2000

Escalonamento Manual – Semana de 08/05/2000 a 12/05/2000				
Máquinas	Tempo de Setup (horas)	Tempo de Corte (horas)	Tempo de Setup e Corte (horas)	Percentual de Setup
SF0529	5,70	38,68	44,38	12,84 %
SF0568	4,70	36,56	41,26	11,39 %
SC0607	8,60	97,39	105,99	7,89 %
SF0291	5,10	42,45	47,55	10,73 %
Totais	24,10	215,08	239,18	10,08 %

As mesmas ordens de fabricação escalonadas manualmente foram selecionadas para serem executadas pelo algoritmo citado anteriormente, utilizando a regra heurística MWR. A escolha desta regra heurística pela empresa, para a solução dos conflitos entre as ordens, deve-se ao fato de se priorizar as ordens de produção com a maior soma de tempos operacionais ainda não efetuados, permitindo assim amenizar o problema de atraso se algum imprevisto ocorrer durante o ciclo de fabricação da peça. A tabela 6.29 representa as horas de *setup* e o processamento gasto pelas respectivas máquinas do centro de trabalho.

TABELA 6.29 - Escal. utilizando o algoritmo na semana 08/05/2000 a 12/05/2000

Algoritmo - Semana de 08/05/2000 a 12/05/2000				
Máquinas	Tempo de Setup (horas)	Tempo de Corte (horas)	Tempo de Setup e Corte (horas)	Percentual de Setup
SF0529	8,70	77,92	86,62	10,04 %
SF0568	4,00	36,58	40,58	9,86 %
SC0607	8,00	77,28	85,28	9,38 %
SF0291	1,60	23,30	24,90	6,42 %
Totais	23,30	215,08	237,38	9,42 %

Durante os testes realizados com o algoritmo, observou-se a necessidade de alterações nas restrições das máquinas com relação às famílias de materiais e na prioridade das mesmas, pois o algoritmo selecionava peças que não eram compatíveis com a máquina ou muitas vezes selecionava a máquina que não era a mais indicada para tal peça.

Uma nova amostragem foi realizada na semana do dia 21/08/2000 a 25/08/2000, e o mesmo processo descrito anteriormente foi realizado. A tabela 6.30 representa as horas obtidas pelo escalonamento manual.

TABELA 6.30 - Escalonamento manual da semana 21/08/2000 a 25/08/2000

Escalonamento Manual - Semana de 21/08/2000 a 25/08/2000				
Máquinas	Tempo de Setup (horas)	Tempo de Corte (horas)	Tempo de Setup e Corte (horas)	Percentual de Setup
SF0529	4,50	32,45	36,95	12,18 %
SF0568	9,30	45,22	54,22	17,15 %
SC0607	6,40	194,33	200,73	3,19 %
SF0291	6,70	44,75	51,45	13,02 %
Totais	26,90	316,75	343,35	7,83 %

Da mesma maneira descrita no experimento da primeira semana, as ordens de fabricação foram selecionadas para que o algoritmo pudesse gerar o escalonamento nas respectivas máquinas do centro de trabalho. A tabela 6.31 representa as horas de *setup* e o processamento gastos pelas respectivas máquinas.

TABELA 6.31 - Esc. utilizando o algoritmo na semana 21/08/2000 a 25/08/2000

Algoritmo - Semana de 21/08/2000 a 25/08/2000				
Máquinas	Tempo de Setup (horas)	Tempo de Corte (horas)	Tempo de Setup e Corte (horas)	Percentual de Setup
SF0529	9,10	111,95	121,05	7,52 %
SF0568	2,00	21,87	23,87	8,32 %
SC0607	4,20	113,15	117,35	3,58 %
SF0291	2,30	69,78	72,08	3,19 %
Totais	17,60	316,75	334,35	5,26 %

Através das tabelas representadas acima, pode-se observar que o algoritmo se mostrou satisfatório quanto à redução do *setup* e também ocorreu uma melhor distribuição das horas de processamento entre as máquinas do centro de trabalho.

Após estes resultados obtidos o algoritmo ganhou credibilidade por parte da empresa e dos usuários, tornando-se uma importante ferramenta de trabalho. Com a utilização constante do algoritmo espera-se também uma redução dos estoques intermediários e do *lead time* das peças. A redução do *lead time* será alcançada devido a uma melhor distribuição e escalonamento das peças nas filas das máquinas em seus respectivos centros de trabalho. A empresa utiliza o tipo de produção intermitente e a técnica de gestão da produção é o MRP II, como foi citado no capítulo dois, com a utilização desta técnica o *lead time* depende da demanda atual e o algoritmo tentará amenizar estas oscilações.

Se os objetivos citados acima forem atingidos, teremos também uma maior pontualidade na entrega dos produtos finais e conseqüentemente uma maior satisfação dos clientes.

7 Considerações Finais

Em virtude da grande diversidade dos sistemas de gestão existentes, os chamados ERP muitas vezes não conseguem representar em seu modelo as peculiaridades do processo produtivo de cada empresa, limitando assim a flexibilidade e a eficiência exigida. Portanto, torna-se muitas vezes necessário o desenvolvimento de ferramentas que se integrem a esses sistemas de gestão em uso e passem a ser um diferencial competitivo da empresa perante o mercado.

Um dos principais problemas na área de manufatura é o escalonamento e controle dos processos de produção. Vários sistemas e pesquisas têm sido realizados envolvendo essa área, levando em consideração os diferentes ambientes de produção. No entanto, as particularidades encontradas em cada tipo de empresa e a integração com os sistemas de gestão se tornam difíceis ou ocorre um volume muito grande de dados em duplicidade.

Assim, uma das preocupações deste estudo foi a abordagem das particularidades do ambiente de produção do tipo *job shop*, com especial atenção às soluções existentes para o problema de escalonamento. Outro aspecto importante a ressaltar foi o acompanhamento e entendimento dos problemas de escalonamento existentes na empresa, bem como as restrições existentes.

Para aprofundar o conhecimento em sistemas de escalonamento e conhecer as técnicas utilizadas para a solução do problema de escalonamento, alguns sistemas foram pesquisados, e estão citados no capítulo cinco.

Após o levantamento das informações e a aquisição do conhecimento através do acompanhamento da produção e de entrevistas às pessoas que presenciam no seu dia-a-dia os problemas enfrentados para o escalonamento da produção, o desenvolvimento do modelo de dados foi proposto e, logo a seguir, o desenvolvimento do sistema piloto, que está em utilização na empresa.

A linguagem de programação Visual Basic 5.0 [VIS2001] foi escolhida para implementar a ferramenta, devido à sua facilidade de representação do conhecimento através de regras, onde declarações do tipo se-então representam as mesmas, conforme exemplos citados no capítulo quatro.

A funcionalidade do sistema dependerá principalmente do conhecimento dos especialistas e de sua visão sobre o sistema de manufatura adotado pela empresa, tendo em vista que, para cada centro de trabalho, existem diversas máquinas e as mesmas possuem suas particularidades de produção, exigindo restrições e prioridades diferenciadas.

No decorrer dos estudos, algumas características importantes foram observadas e contribuíram para a solução do problema. Uma delas é a utilização de restrições e prioridades nas famílias de materiais e peças para as máquinas individualmente dentro de cada centro de trabalho. Através da similaridade das peças, foi possível realizar o melhor

agrupamento das mesmas. Dessa maneira, os índices de produtividade das máquinas foram melhorados, e, conseqüentemente, as horas gastas para a preparação das mesmas foram reduzidas. Também observou-se uma redução dos estoques intermediários em chão-de-fábrica, diminuindo assim o custo operacional.

O sistema está implementado na empresa Hidrover Equipamentos Oleodinâmicos S.A., situada em Caxias do Sul, inicialmente no centro de trabalho de corte, composto por quatro máquinas que realizam o corte de diversos tipos de matérias-primas.

Após a implantação do sistema-piloto, percebem-se alguns aspectos que podem ser citados como possíveis melhoramentos e sugestões para futuros trabalhos:

- desenvolvimento de rotinas com outras regras heurísticas;
- desenvolvimento de interfaces gráficas, como os gráficos de Gantt [PIN95] e Pert [PER58] para a visualização das atividades e seu fluxo no sistema;
- implementação do escalonamento nos demais centros de trabalho da empresa;
- implementação do algoritmo para também verificar o tamanho das filas nas demais máquinas que fazem parte do roteiro da peça.

Bibliografia

- [BAS98] BASTOS, R. M. **O Planejamento de Alocação de Recursos Baseado em Sistemas Multiagentes**. 1998. Tese (Doutorado em Ciência da Computação) - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [BER90] BERTRAND, J.W.M.; WORTMANN, J.C.; WIJNGAARD, J. **Production Control: a Structural and Desing Oriented Approach**. Amsterdam: Elsevier, 1990.
- [BOO2000] BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML – Guia do Usuário**. São Paulo: Campus, 2000.
- [BRU98] BRUCKER, Peter. **Scheduling Algorithms**. Berlin: Springer-Verlag, 1998.
- [CAP2000] CAPS – Computer Aided Process Scheduling. Disponível: em <<http://www.caps.com.htm>>. Acesso em: 15 dez. 2000.
- [COA92] COAD, Peter; YOURDON, Edward. **Análise Baseada em Objetos**. São Paulo: Campus, 1992.
- [COA92a] COAD, Peter; YOURDON, Edward. **Projeto Baseado em Objetos**. São Paulo: Campus, 1992.
- [COR93] CORREA, H.L.; GIANESI, I.G. **Just-in-Time, MRP-II e OPT: um Enfoque Estratégico**. São Paulo: Atlas, 1993.
- [ERI98] ERIKSSON, Hans-Erik; PENKER, Magnus. **UML – Toolkit**. [S.l.]: Ed. Wiley, 1998.
- [FOX90] FOX, M. S.; SMITH, S. F.; OW, P.S. The ISIS system for job shop scheduling. In: KERR, R. (Ed.). **Knowledge-Based Manufacturing Management: Applications of Artificial Intelligence to the Effective Management of Manufacturing Companies**. Sydney: Addison-Wesley, 1990. p. 239-254.
- [FOW2000] FOWLER, Martin; SCOTT, Kendall. **UML Essencial: um breve guia para a linguagem-padrão de modelagem de objetos**. 2. ed. Porto Alegre: Bookman, 2000.
- [HAX84] HAX, A.; CANDEA, D. **Production and Inventory Management**. Englewood Cliffs, N.J.: Prentice-Hall, 1984.

- [HAY83] HYES-ROTH, F.; WATERMAN, D.A.; LENAT, D.B. **Building Expert Systems**. Reading: Addison Wesley, 1983.
- [KER90] KERR, R.; WALKER, N. ISTOP – an intelligent strategic and operational planning system. In: KERR, R. (Ed.). **Knowledge-Based Manufacturing Management: Applications of Artificial Intelligence to the Effective Management of Manufacturing Companies**. Sydney: Addison-Wesley, 1990. p. 229-239.
- [LAR2000] LARMAN, Craig. **Utilizando UML e padrões: uma Introdução à Análise ao Projeto Orientado a Objetos**. Porto Alegre: Bookman, 2000.
- [LEK2001] LEKIN. **Flexible Job-Shop Scheduling**. Disponível em: <<http://www.ieor.columbia.edu/~andrew/scheduling/Lekin.html>>. Acesso em: 20 jan. 2001.
- [LOP96] LÓPEZ, V.; OSCAR, C.; CURY, R.M. **Gestão de Sistemas de Produção**. Florianópolis: Programa de Pós-Graduação em Engenharia de Produção e Sistema, UFSC, 1996. Apostila de aula.
- [MOR93] MORTON, T.E.; PENTICO, D.W. **Heuristic Scheduling Systems**. New York: John Wiley & Sons, 1993.
- [PAC99] PACHECO, R.F.; SANTORO, M.C. Proposta de Classificação Hierarquizada dos Modelos de Solução para o Problema de *Job Shop Scheduling*. **Gestão e Produção: Revista do Departamento de Engenharia de Produção**, São Carlos, p. 1-15, abr. 1999.
- [PAL65] PALMER, D.S. Sequencing Jobs Through a Multi-Stage Process in the Minimum Total Time: A Quick Method of Obtaining a Near Optimum. **Operational Research Quarterly**, [S.l.], v.10, p. 101-107, 1965.
- [PER58] PERT. **Program Evaluation Research Task**. Phase I Summary Report, Special Projects Office, Bureau of Ordnance, Department of Navy. Washington, D.C., 1958.
- [PIN95] PINEDO, M. **Scheduling: theory, algorithms and systems**. New Jersey: Prentice-Hall, 1995.
- [PIN99] PINEDO, M.; CHAO, X. **Operations Scheduling With Applications in Manufacturing and Services**. New York: Irwin McGraw-Hill, 1999.
- [PRE2001] PREACTOR. **The Path to Scheduling Excellence**. Disponível em: <<http://www.preactor.com.htm>>. Acesso em: 12 jan. 2001.

- [PRO2001] PROVISA. **Finite Capacity Scheduling Solutions**. Disponível em: <<http://www.lanner.co.uk/products/provisa.htm>>. Acesso em: 12 jan. 2001.
- [RAT2000] RATIONAL ROSE. **UML Resource Center**. UML Documentation. Disponível em: <<http://www.rational.com/index.jsp>>. Acesso em: 15 dez. 2000.
- [RUM99] RUMBAUGH, James; BOOCH, Grady; JACOBSON, Ivar. **The Unified Modeling Language Reference Manual**. New York: Addison-Wesley, 1999.
- [SLA97] SLACK, Nigel et al. **Administração da Produção**. São Paulo: Atlas, 1997.
- [SMI90] SMITH, S. F. Reactive Scheduling the OPIS System. In: KERR, R. (Ed.). **Knowledge-Based Manufacturing Management: Applications of Artificial Intelligence to the Effective Management of Manufacturing Companies**. Sydney: Addison-Wesley, 1990. p. 254-257.
- [VIS2001] VISUAL BASIC. **Language Vbasic**. Disponível em: <<http://www.microsoft.com/vbasic/>>. Acesso em: 12 jan. 2001.
- [WAL88] WALTER, C.; KLEEBANK, V. Técnicas de Inteligência Artificial aplicadas ao Planejamento da Produção. In: CONGRESSO NACIONAL DE AUTOMAÇÃO INDUSTRIAL - CONAI, 3., 1988. **Anais...** São Paulo:SUCESU, 1988.
- [WAL89] WALTER, Cláudio. Um Algoritmo e uma Heurística para a Programação da Produção em Sistemas de Manufatura. In: CONGRESSO NACIONAL DE INFORMÁTICA, 22., 1989. **Anais...** São Paulo:SUCESU, 1989.
- [WAL90] WALTER, Cláudio. Um método para o Escalonamento Finito em Sistemas Fabris. In: CONGRESSO NACIONAL DE AUTOMAÇÃO INDUSTRIAL - CONAI, 4.,1990. **Anais...** São Paulo:SUCESU, 1990.
- [ZAH91] ZAHEDI, F. An Introduction to Neural Networks and a Comparison with Artificial Intelligence and Expert Systems, **Interfaces**, [S.l.], v.21, p.25-38, 1991.
- [ZIM2000] ZIM. **Technology**. Disponível em: <<http://www.zim.biz>>. Acesso em: 18 nov. 2000.