

ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL DA  
TESE DEFENDIDA POR Pythagoras Grangeiro  
Furtado E APROVADA PELA  
COMISSÃO JULGADORA EM 22/03/2000  
marcius fabius  
ORIENTADOR

**UNIVERSIDADE ESTADUAL DE CAMPINAS**  
**FACULDADE DE ENGENHARIA MECÂNICA**

# **Ambiente para Planejamento da Produção de Sistemas**

Autor: **Pythagoras Grangeiro Furtado**  
Orientador: **Marcus Fabius Henriques de Carvalho**

39/00

UNICAMP  
BIBLIOTECA CENTRAL  
SEÇÃO CIRCULANTE



**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
DEPARTAMENTO DE PROJETO MECÂNICO**

## **Ambiente para Planejamento da Produção de Sistemas**

**Autor: Pythagoras Grangeiro Furtado**  
**Orientador: Marcius Fabius Henriques de Carvalho**

Curso: Engenharia Mecânica  
Área de Concentração: Projeto Mecânico

Dissertação de mestrado apresentada à Comissão de Pós Graduação da Faculdade de Engenharia Mecânica, como requisito para a obtenção do título de Mestre em Engenharia Mecânica.

Campinas, 2000  
SP - Brasil

UNICAMP  
BIBLIOTECA CENTRAL  
SEÇÃO CIRCULANTE



886 HT 0000 R

UNIDADE	BC
N.º CHAMADA:	T/UNICAMP
	F984a
V.	Ex.
TOMBO BC/	42454
PROC.	161278100
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇOS	738,00
DATA	05/01/00
N.º CPD	

CM-00147184-6

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

F984a Furtado, Pythagoras Grangeiro  
Ambiente para planejamento da produção de sistemas / Pythagoras Grangeiro Furtado.--Campinas, SP: [s.n.], 2000.

Orientador: Marcius Fabius Henriques de Carvalho.  
Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Planejamento da produção. 2. Teoria dos grafos.  
3. Otimização matemática. I. Carvalho, Marcius Fabius Henriques de . II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
DEPARTAMENTO DE PROJETO MECÂNICO

DISSERTAÇÃO DE MESTRADO

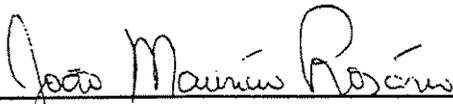
## Ambiente para Planejamento da Produção de Sistemas

Autor: Pythagoras Grangeiro Furtado  
Orientador: Marcius Fabius Henriques de Carvalho



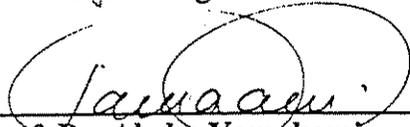
---

Prof. Dr. Marcius Fabius Henriques de Carvalho, Presidente  
Faculdade de Engenharia Mecânica, UNICAMP



---

Prof. Dr. João Maurício Rosário  
Faculdade de Engenharia Mecânica, UNICAMP



---

Prof. Dr. Akebo Yamakami  
Faculdade de Engenharia Elétrica e de Computação, UNICAMP

Campinas, 22 de março de 2000

UNICAMP  
BIBLIOTECA CENTRAL  
SEÇÃO CIRCULANTE



## **Dedicatória**

Dedico este trabalho a Francisco e Francisca, meus pais.



## **Agradecimentos**

Ao Marcius, pela confiança,

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), pelo apoio financeiro,

E a todos que colaboraram, direta ou indiretamente, para a realização deste trabalho.



*“Só sei que nada sei.”  
(Sócrates)*



## Resumo

FURTADO, Pythagoras Grangeiro, *Ambiente para Planejamento da Produção de Sistemas*, Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2000. 52 p. Dissertação (Mestrado)

Este trabalho caracterizou-se por duas atividades principais: 1) desenvolvimento de ambiente computacional para modelagem de sistemas produtivos, visando o planejamento da produção de sistemas de manufatura; 2) implementação de um algoritmo especial para aproveitamento da estrutura de fluxo em redes do problema relacionado com o sistema produtivo em questão, visando melhor desempenho deste sistema. O objetivo do trabalho foi criar uma nova ferramenta de software com uma interface amigável, constituída por uma linguagem visual/simbólica, possibilitando ao usuário não-especialista construir modelos de sistemas produtivos, segundo a abordagem de fluxo em redes. Faz parte desta ferramenta um algoritmo para obtenção da solução otimizante do problema matemático inerente ao modelo proposto pelo usuário. Este algoritmo ou *solver* considera características especiais da estrutura esparsa das matrizes relacionadas com o problema.

### *Palavras Chave*

Planejamento da Produção, Fluxo em Redes, Otimização



## **Abstract**

FURTADO, Pythagoras Grangeiro, *Production Planning Environment*, Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2000. 52 p. Dissertação (Mestrado)

This work considers two main topics: 1) the development of modeling environment for production planning; 2) the implementation of special algorithm to explore the structure of the production planning problem. The objective of the first topic is the development of an environment composed of a man-machine user-friendly interface based on a visual language that allows the users to represent production planning problems through graph models. These models are transformed into mathematical network flows models. The second topic try to take advantage of the special structure of the mathematical model, regarding to the sparse characteristic of the constraints matrices related with the problem and to implement a special algorithm (solver). Finally, a performance comparison shows some advantages and drawback of this approach.

### *Key Words*

Production Planning, Network Flows, Optimization



# Índice

Lista de Figuras	ii
1 Introdução	1
2 Modelagem de Sistemas de Produção por Fluxo em Redes	6
3 Algoritmo para Resolução de Problemas com Restrições Adicionais	15
4 Interface para Modelagem	32
5 Aplicação de Tecnologia de Grupo	39
6 Conclusões e Sugestões para Próximos Trabalhos	47
Referências Bibliográficas	49



## Lista de Figuras

1.1 Hierarquia de Decisão	1
2.1 Esquema de Linha de Produção Serial	7
2.2 Representação de Linha Serial por Grafo	9
2.3 Conservação de Fluxo nos Nós	9
2.4 Representação do Problema Multiproduto por Grafo	12
2.5 Compartilhamento Parcial em Célula de Manufatura	12
2.6 Representação por Grafo de Seqüências de Produção Diferentes	13
2.7 Estrutura da Matriz de Restrições	13
3.1 Modelo de Fluxo em Redes e Árvore Básica	23
3.2 Base e Restrições Adicionais do Modelo de Fluxo em Redes	25
3.3 Comparação entre Soluções Heurísticas e Soluções Ótimas	26
3.4 Restrições Adicionais Modificadas	28
3.5 Alteração do Tempo de Execução do RETRA devido à Variação do Parâmetro $\mu$	29
3.6 Comparação dos Desempenhos dos Algoritmos	30
4.1 Tela para Definição do Layout de Fábrica	34
4.2 Definição de Seqüência para o Produto 1, a partir do Layout de Fábrica	35
4.3 Definição de Seqüência para o Produto 2	35
4.4 Tela para Definição de Custos	36
4.5 Tela para Definição de Capacidades	37
4.6 Tela para Definição de Capacidades Mútuas	37



4.7 Tela para Definição de Matéria-Prima e Demanda	38
5.1 Células de Manufatura	40
5.2 Matriz Binária de Operações (a) Original e (b) Rearranjada	41
5.3 Agrupamento com Movimentos Inter-Células	42
5.4 Alternativas para Eliminação de Movimentos Inter-Células	43
5.5 Agrupamentos Associados às Semanas 1 e 2	45



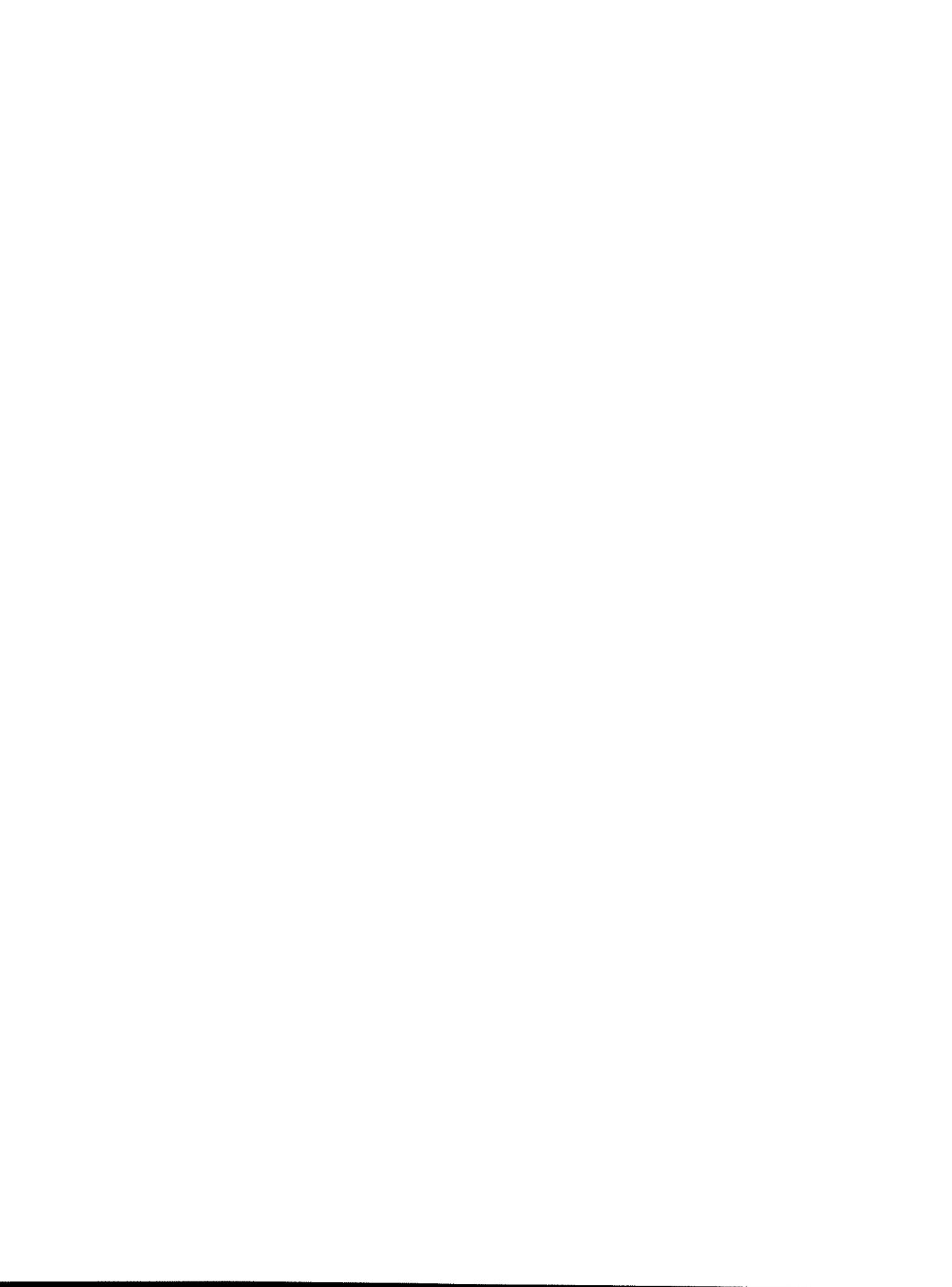
# Capítulo 1

## Introdução

A consistência do planejamento da produção depende da integração dos três níveis decisórios (estratégico, tático e operacional, em ordem hierárquica) da organização. O nível estratégico envolve decisões agregadas de longo prazo (anos, discretizados em meses). No nível tático, o plano de produção agregado (traduzido pelo *planejamento mestre da produção*) é desagregado em planos mais detalhados (Özdamar, 1998), discretizados em períodos semanais ou diários. Por fim, no nível operacional, as metas definidas pelo planejamento tático são utilizadas para a determinação precisa do escalonamento da produção.



Figura 1.1 Hierarquia de decisão



Uma das principais ferramentas de software disponíveis para automatização de atividades de planejamento no nível tático é o *Planejamento de Requisitos de Materiais* (MRP) (Slack, 1996). Esta é uma técnica *avaliativa* que, a partir de um conjunto de decisões, avalia ou prevê o desempenho do sistema sob determinadas condições. Uma deficiência do MRP reside no fato de que qualquer evento inesperado (por exemplo, uma quebra de máquina ou uma mudança nos pedidos) pode colocar em risco a validade do plano de produção corrente. Outro problema diz respeito à carência de considerações de capacidade na geração dos planos de produção. Estes inconvenientes podem resultar em repetidas (e exaustivas) simulações de diversos planos, até a obtenção de um plano factível com respeito aos recursos de produção (Carvalho, 1998).

Os problemas inerentes ao MRP, comentados acima, provocam a ruptura das atividades do processo de planejamento o que, no ambiente de manufatura extremamente dinâmico e competitivo dos dias de hoje, é indesejável. Uma solução é a integração de técnicas avaliativas, como o MRP, com técnicas *gerativas* (Carvalho, 1996). As técnicas gerativas, que se baseiam em métodos matemáticos de otimização, partem de critérios e restrições e geram um conjunto de decisões que, em geral, estão muito próximas de um plano factível para o MRP.

Este trabalho trata da aplicação da técnica gerativa como ferramenta do planejamento tático da produção e propõe um ambiente para modelagem que facilite a integração do decisor com a técnica através de recursos gráficos. O trabalho vem complementar desenvolvimentos anteriores em dois aspectos: o primeiro, a implementação e avaliação do desempenho de um algoritmo para, em conjunto com outros algoritmos dedicados à mesma finalidade, cobrir todas as possibilidades em termos de software de otimização. O segundo, o desenvolvimento de um ambiente de modelagem para disponibilizar, a usuários com pouca experiência em computador e em otimização, todos os desenvolvimentos até aqui realizados.

## **1.1 Técnica Gerativa de Solução**

Como comentado anteriormente, um dos enfoques para apoio a tomada de decisão nos processos de planejamento da manufatura é a utilização dos métodos de otimização. Entre eles, um enfoque muito atrativo é a modelagem do sistema produtivo como um problema de fluxo em redes com restrições adicionais (Carvalho, 1999). O grande apelo deste enfoque é olhar para o



ponto de vista da empresa (retorno esperado) além de procurar resolver o problema com uma visão da demanda ao longo do tempo, considerando assim as informações disponíveis para todo o horizonte de planejamento. Este fato permite:

- eliminar gargalos através do deslocamento no tempo, em alguns casos adiantando e em outros atrasando a produção para linhas que processem multiprodutos;
- analisar objetivos conflitantes e a demanda probabilística sob o ponto de vista do negócio, traduzidos no objetivo *lucro*.

O método de otimização parte de metas mensais estabelecidas por um nível hierárquico superior (por exemplo, decisões geradas pelo programa mestre da produção) e as desmembra em produções diárias, de forma a otimizar os custos, atendendo ao mesmo tempo às restrições de capacidade das máquinas, de armazenagem e de disponibilidade de matéria-prima. As decisões a serem tomadas em cada período envolvem a quantidade produzida, a utilização de matéria-prima, a carga de cada máquina, entre outras. A cada máquina é associado um custo de operação, que pode ser diferente para cada período de tempo. A mesma associação ocorre para as armazenagens e para a matéria-prima. Um caminho para modelar e resolver este tipo de problema passa pela observação de que uma linha de produção pode ser vista como o fluxo de peças, em trajetórias pré-definidas, através de máquinas e unidades de armazenagem, onde sofrem trabalho em cada estágio até se tornarem produto final. Nestas trajetórias as peças esperam em uma fila por condições de liberação e estão sujeitas a processos de montagem ou fabricação (Carvalho, 1992).

Um ponto importante a considerar é o processamento multi-item (ou multiproduto). Os produtos podem ter seqüências de fabricação idênticas ou diferentes, no entanto disputam o mesmo conjunto de máquinas, durante o processo de fabricação. Assim sendo é importante garantir que, dentro do horizonte de planejamento, o processamento de peças diferentes em uma mesma máquina não leve à violação de sua capacidade de produção. Esta restrição estende o problema de grafo a um problema multifluxo (Ahuja, 1993).

## **1.2 Técnicas de Decomposição**

Em geral, os problemas de planejamento da produção são de grande porte, devido principalmente à natureza temporal e à quantidade de produtos que compartilham os mesmos



recursos de produção. Logo, requisitos como armazenagem de variáveis e tempo de execução computacional são fatores essenciais na escolha de um método de solução. Um meio de satisfazer estes requisitos é aproveitar a estrutura particular do problema pela utilização de um método especializado para estruturas de rede. Uma abordagem possível é o *método de decomposição primal para fluxo em redes com restrições adicionais* (Kennington, 1980), implementado no algoritmo NETSIDE, código em Fortran (Kennington, 1990).

Um outro enfoque é tratar as restrições adicionais por técnicas de penalidade. Este enfoque reduz as restrições do problema apenas às restrições de conservação de fluxo (associadas às matrizes de incidência nó-arco do modelo de rede). Contudo a função objetivo passa a ser não linear e não “desacoplável” no espaço dos produtos. Uma forma de contornar este problema é através da decomposição simplicial. Este enfoque vem apresentando bons resultados através do software RETRA (Fernandes, 1999), utilizado aqui como um algoritmo de referência. A proposta do trabalho, com relação ao desenvolvimento de algoritmo de otimização, refere-se ao tratamento das matrizes de restrições pela técnica de decomposição primal.

### **1.3 Ambiente para Modelagem**

O enfoque de fluxo em redes permite resolver o problema de planejamento da produção pela representação do fluxo de produção. Assim sendo, a descrição de um problema real, em linguagem de computador, pode ser feita por ícones que identifiquem a rota de cada produto. O ambiente deve possibilitar definir parâmetros como capacidade ou taxa de produção, horas disponíveis para processamento, discretização, número de peças a serem montadas em cada centro de montagem, disponibilidade e tipo de matéria-prima requerida para a produção, demanda requerida para cada intervalo de discretização, etc.

A partir da determinação das rotas, o ambiente expande a representação do fluxo de produção ao longo do horizonte de planejamento e transforma esta representação em um *problema de otimização* para o computador. Este enfoque torna *transparente* ao usuário as equações matemáticas que descrevem o sistema físico, fator sempre relevante pois amplia a utilização da metodologia para usuários que possuem experiência no sistema físico mas que não têm intimidade com as representações matemáticas.



O Capítulo 2 trata da modelagem do problema de planejamento da produção por fluxo em redes. No Capítulo 3 serão apresentadas técnicas de solução para o modelo proposto. O Capítulo 4 descreve a interface gráfica desenvolvida para modelagem por computador. No Capítulo 5 mostra-se a aplicabilidade da Tecnologia de Grupo como técnica auxiliar ao problema de planejamento da produção discutido nos capítulos anteriores. Por fim, o Capítulo 6 apresenta as conclusões e sugere alternativas para futuros trabalhos.



## Capítulo 2

### Modelagem de Sistemas de Produção por Fluxo em Redes

Define-se *sistema* como uma entidade constituída de um conjunto de elementos que mantém relações e realizam ações. Num sistema de produção (ou manufatura) os elementos, classificados tradicionalmente em três grupos principais, são: 1) matéria-prima (*elementos de entrada*), 2) unidades processadoras (máquinas que modificam – por transformação, montagem ou transporte – atributos como forma e localização do produto) e produtos semi-acabados (*elementos de processos de transformação interna*) e 3) produtos finais (*elementos de saída*). Um dos objetivos do planejamento da produção é propor meios de atender a demanda pelos elementos do terceiro grupo, considerando as disponibilidades de elementos dos dois primeiros grupos. Outro objetivo importante, complementar ao anterior, é fazê-lo de maneira que o custo total da produção, relacionado com os custos individuais associados aos elementos do sistema, seja o menor possível. Tais características permitem, como será mostrado a seguir, modelar o sistema como um *problema de fluxo de custo mínimo* (Ahuja, 1993 ; Bazaraa, 1990).

O modelo para planejamento da produção apresentado neste capítulo constitui um problema multiestágio, multiperíodo, multiproduto (ou multi-item) de fluxo em redes com restrições adicionais (Carvalho, 1999). Este modelo aplica-se a sistemas de manufatura compostos por uma série de estágios que podem processar diversos tipos ou famílias de produtos (itens), para vários períodos de tempo. A necessidade de sincronização das linhas seriais nos pontos de montagem e as limitações para compartilhamento das máquinas por mais de um tipo de produto acrescentam as restrições adicionais complicadoras do problema.



## 2.1 Modelagem de Linha Serial Monoproduto

A Figura 2.1(a) ilustra uma linha de produção serial com quatro estágios, cada um associado a uma máquina ou um grupo de máquinas (com capacidades previamente definidas) que realizam transformações ou transporte. As metas periódicas, definidas por um nível hierárquico superior (por exemplo, pelo programa mestre de produção), são desmembradas para períodos menores, como mostra a Figura 2.1(b). A cada estágio associa-se um custo de operação, que pode ser diferente para cada período de tempo. Ao deixar um estágio, um item pode ir ao estágio seguinte ou ser armazenado para operação em períodos subsequentes, dependendo da capacidade de produção da linha e do custo de estocagem dos itens.

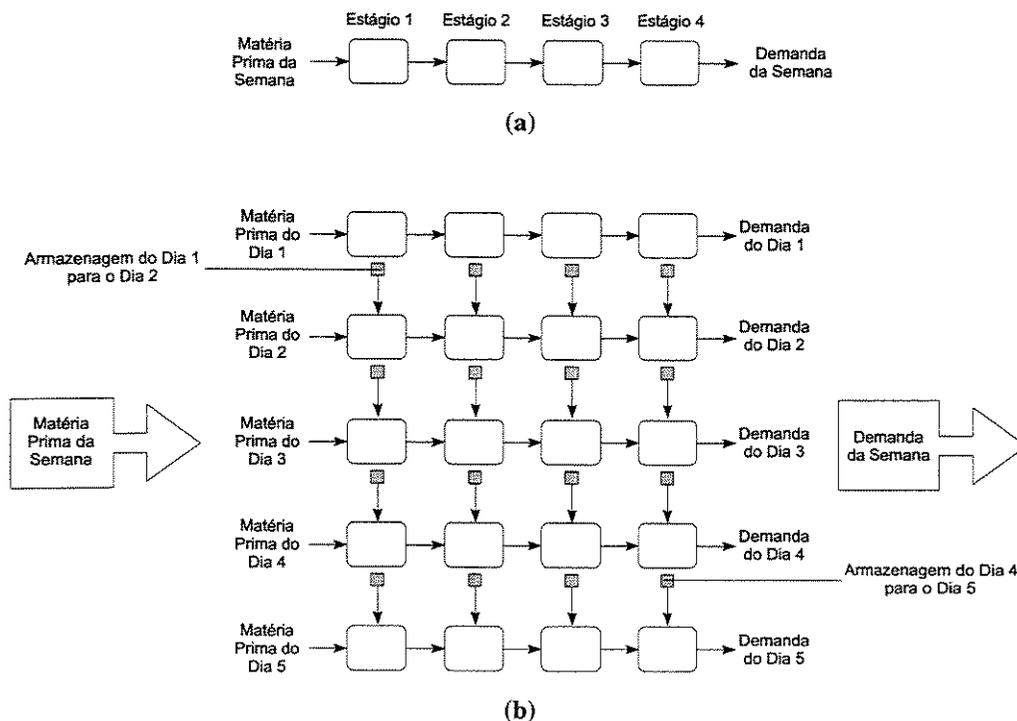


Figura 2.1 Esquema de linha de produção serial

O planejamento da produção deve determinar:

- a quantidade de itens que cada máquina deve produzir e em que período;
- a quantidade de itens semi-acabados a armazenar por estágio e em que período;
- o nível de utilização de matéria-prima por período;
- o nível de atendimento à demanda por período;
- a parcela da demanda atendida com atraso.



Um modelo matemático capaz de fornecer as informações acima, objetivando a redução do custo total de produção ao menor nível possível, é dado pelo seguinte programa linear (PL):

$$\text{Min } \sum_t \sum_i (cm_t m_t + c_{ii} x_{it} + ca_{ii} y_{it} + cb_t z_t + cd_t d_t)$$

sujeito a :  $\sum_{t=1}^T m_t = M$  (2.1)

$$x_{i-1,t} + y_{i,t-1} = x_{it} + y_{it} \quad t = 1, \dots, T; i = 1, \dots, I \quad (2.2)$$

$$x_{it} + y_{i+1,t-1} + z_t = d_t + y_{i+1,t} + z_{t-1} \quad t = 1, \dots, T \quad (2.3)$$

$$\sum_{t=1}^T d_t = D \quad (2.4)$$

$$m_t, x_{it}, y_{it}, z_t, d_t \geq 0 \quad \text{para } t \text{ e } i \text{ adequados}$$

Neste modelo, as variáveis de decisão são

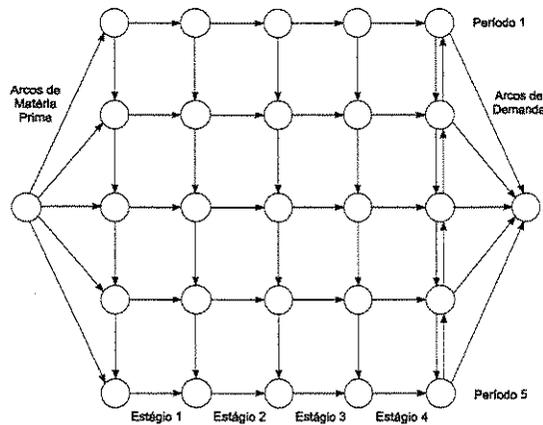
$m_t$	quantidade de matéria prima utilizada no período $t$
$x_{it}$	quantidade produzida no estágio $i$ , no período $t$
$y_{it}$	quantidade armazenada no estágio $i$ , entre os períodos $t$ e $t+1$
$z_t$	demanda em atraso atendida no período $t$
$d_t$	nível de atendimento da demanda do período $t$

Os parâmetros de entrada do modelo são

$I$	quantidade de estágios de produção
$T$	quantidade de períodos do horizonte de planejamento
$M$	disponibilidade total de matéria prima no horizonte de planejamento
$D$	demanda de produtos acabados no horizonte de planejamento
$cm_t$	custo unitário de aquisição de matéria prima no período $t$
$c_{ii}$	custo unitário de produção no estágio $i$ , período $t$
$ca_{ii}$	custo unitário de armazenagem no estágio $i$ , período $t$
$cb_t$	custo unitário de atendimento à demanda em atraso (backorder) no período $t$
$cd_t$	custo unitário de atendimento da demanda do período $t$

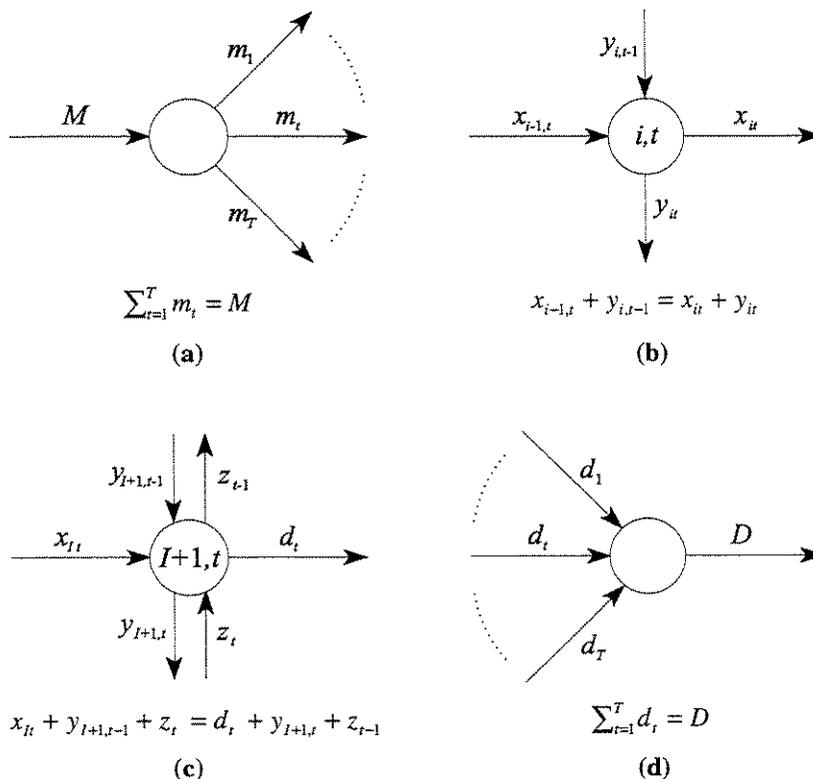
O PL sugerido acima caracteriza um problema particular de fluxo em redes conhecido como *problema de fluxo de custo mínimo* (Ahuja, 1993; Bazaraa, 1990). Este tipo de problema está associado a um grafo cujos nós representam, no contexto deste trabalho, pontos de decisão onde são definidos o quanto produzir e o quanto armazenar em determinado período. A Figura 2.2 reproduz o grafo correspondente à linha serial da Figura 2.1(b).





**Figura 2.2** Representação de linha serial por grafo

Deve-se notar que cada um dos nós de um grafo como o da Figura 2.2 tem relação direta com uma das restrições de conservação de fluxo dadas por (2.1), (2.2), (2.3) e (2.4), como ilustrado na Figura 2.3. No nó de oferta de matéria-prima o fluxo injetado (de valor  $M$ ) deve ser distribuído entre os arcos correspondentes às variáveis  $m_n$ , que saem do nó (Figura 2.3(a)). De maneira análoga, a soma dos fluxos dos arcos incidentes no nó de demanda (variáveis  $d_i$ ) deve ser igual ao fluxo, de valor  $D$ , do arco que sai deste nó (Figura 2.3(d)). O conjunto de restrições (2.2)



**Figura 2.3** Conservação de fluxo nos nós



garante que, nos nós que representam pontos de decisão, o fluxo “entrante”, dado pelo que foi produzido no estágio anterior ( $x_{i-1,t}$ ) mais o que foi armazenado no período anterior ( $y_{i,t-1}$ ), é igual à produção do estágio atual ( $x_{it}$ ) mais o que será armazenado ( $y_{it}$ ) neste estágio para processamento no período subsequente, de acordo com a Figura 2.3(b). Por fim, o grupo de restrições (2.3) modela o balanço dos fluxos dos nós relativos ao último ponto de decisão (Figura 2.3(c)). Nestes nós, a variável  $y_{i+1,t}$  representa a quantidade de produtos acabados a ser armazenada para o próximo período. Se esta variável assume valor positivo, significa que é vantajoso produzir em  $t$  para atender uma demanda futura. Por outro lado, um valor negativo de  $y_{i+1,t}$  indica deficiência de produção ou vantagem financeira em adiar o atendimento à demanda. Devido às restrições de não-negatividade (que impedem a atribuição de valores negativos para  $y_{i+1,t}$ ), torna-se necessário introduzir a variável  $z_t$ . Esta variável, que representa a quantidade de demanda em atraso (ordem de fabricação em atraso ou *backorder*) atendida no período  $t$ , modela a decisão de atender uma demanda do presente no futuro. O arco associado à variável  $d_t$  representa a demanda de produtos acabados a ser atendida no período  $t$ .

Há ainda alguns casos particulares a serem analisados. Para  $i = 1$ , ou seja, no primeiro estágio, a variável  $x_{i,t}$  é igual à quantidade de matéria-prima distribuída para o período  $t$ , isto é,  $m_t$ . Em  $t = 1$  (primeiro período do horizonte de planejamento) a variável  $y_{i,t-1}$  geralmente assume o valor nulo mas, em outro caso, pode modelar a injeção de produtos semi-acabados provenientes de períodos de planejamento anteriores. No período  $t = T$  (último período do horizonte de planejamento), a variável  $y_{it}$  representa a decisão de se produzir neste período produtos semi-acabados para períodos de planejamentos futuros.

## 2.2 Modelagem de Linha Serial Multiproduto

No âmbito da manufatura moderna, um sistema de produção deve ter flexibilidade para fabricação de diversos tipos de produtos, que passam a compartilhar os mesmos recursos de produção (máquinas). O modelo estendido para considerar o caso multiproduto, multiestágio e multiperíodo é dado por



$$\text{Min } \sum_p \sum_t \sum_i (cm_{ip}m_{ip} + c_{iip}x_{iip} + ca_{iip}y_{iip} + cb_{ip}z_{ip} + cd_{ip}d_{ip})$$

sujeito a :

$$\sum_{t=1}^T m_{ip} = M_p \quad p = 1, \dots, P \quad (2.5)$$

$$x_{i-1,t,p} + y_{i,t-1,p} = x_{iip} + y_{iip} \quad i = 1, \dots, I; t = 1, \dots, T; p = 1, \dots, P \quad (2.6)$$

$$x_{itp} + y_{i+1,t-1,p} + z_{ip} = d_{ip} + y_{i+1,t,p} + z_{i-1,p} \quad t = 1, \dots, T; p = 1, \dots, P \quad (2.7)$$

$$\sum_{t=1}^T d_{ip} = D_p \quad p = 1, \dots, P \quad (2.8)$$

$$\sum_{p=1}^P \frac{x_{iip}}{r_{iip}} \leq C_{ii} \quad i = 1, \dots, I; t = 1, \dots, T \quad (2.9)$$

$m_{ip}, x_{iip}, y_{iip}, z_{ip}, d_{ip} \geq 0$  para  $p, t,$  e  $i$  adequados

onde as variáveis de decisão são

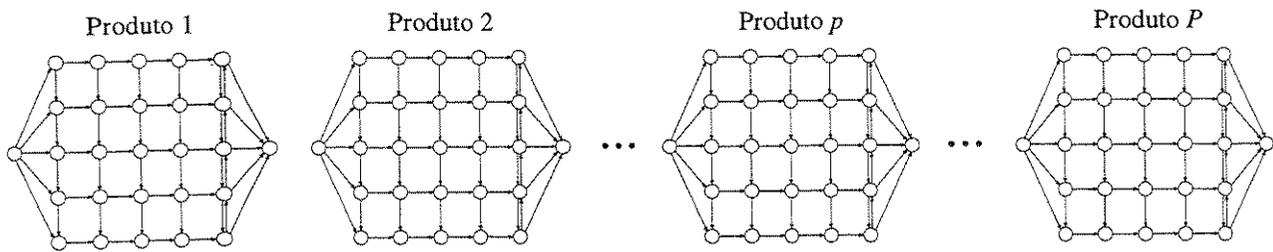
$m_{ip}$	quantidade de matéria prima utilizada no período $t$ , para o produto $p$
$x_{iip}$	quantidade produzida do produto $p$ , no estágio $i$ , no período $t$
$y_{iip}$	quantidade do produto $p$ armazenada no estágio $i$ , entre os períodos $t$ e $t+1$
$z_{ip}$	demanda em atraso do produto $p$ , atendida no período $t$
$d_{ip}$	nível de atendimento da demanda do período $t$ , para o produto $p$

e os parâmetros de entrada são

$P$	quantidade de tipos de produtos
$I$	quantidade de estágios de produção
$T$	quantidade de períodos do horizonte de planejamento
$M_p$	disponibilidade total de matéria prima para o produto $p$ no horizonte de planejamento
$D_p$	demanda de produtos acabados do tipo $p$ no horizonte de planejamento
$C_{ii}$	capacidade mútua do estágio $i$ , no período $t$
$r_{iip}$	taxa de produção do estágio $i$ , para o produto $p$ , no período $t$
$cm_{ip}$	custo unitário de aquisição de matéria prima para o produto $p$ , no período $t$
$c_{iip}$	custo unitário de produção do item $p$ , no estágio $i$ , período $t$
$ca_{iip}$	custo unitário de armazenagem do produto $p$ , no estágio $i$ , período $t$
$cb_{ip}$	custo unitário de atendimento à demanda em atraso (backorder) do produto $p$ , no período $t$
$cd_{ip}$	custo unitário de atendimento da demanda do período $t$ , para o produto $p$

As restrições dadas por (2.5), (2.6), (2.7) e (2.8) têm a mesma forma das restrições (2.1), (2.2), (2.3) e (2.4), diferindo apenas pela consideração dos  $P$  tipos de produtos. Agora, cada um dos produtos tem associado a si um grafo como o da Figura 2.2, sendo que o problema, como um todo, está associado a um conjunto de  $P$  grafos, como mostra a Figura 2.4.

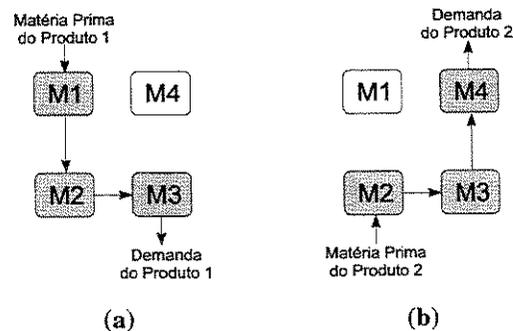




**Figura 2.4** Representação do problema multiproducto por grafo

A característica nova introduzida diz respeito às *restrições adicionais de capacidade mútua* dadas pelo grupo de inequações (2.9). O papel destas restrições é garantir que as capacidades de produção das máquinas não sejam violadas já que, no contexto de multiprodução, uma máquina pode processar produtos de tipos diferentes. O parâmetro  $C_{it}$  define a capacidade horária disponível no estágio  $i$  para processamento mútuo de produtos no período  $t$ .  $C_{it}$  pode ser ajustado para modelagem de quebras de máquina, paradas para setup, absentismo etc. A taxa  $r_{itp}$  expressa a quantidade de itens  $p$  que podem ser produzidos por unidade de tempo no estágio  $i$ , período  $t$ . Na prática, define-se o parâmetro  $r_{itp}$  como sendo um múltiplo da capacidade\* do estágio  $i$  para o produto  $p$ , no período  $t$ . Dessa forma, o quociente  $x_{itp}/r_{itp}$  representa a fração (nunca é superior a 1) do recurso  $i$  requerida pelo produto  $p$  no período  $t$ . O parâmetro  $C_{it}$  é geralmente definido como 1, significando que o estágio  $i$  está 100% disponível no período  $t$ .

O enfoque de fluxo em redes permite que os produtos compartilhem os recursos de produção total ou parcialmente. A Figura 2.5 ilustra o compartilhamento parcial de uma célula de manufatura por dois produtos diferentes.



**Figura 2.5** Compartilhamento parcial em célula de manufatura

\* Os parâmetros de capacidade foram omitidos no modelo para simplificar a notação.



Enquanto o Produto 1 segue a seqüência determinada pelas máquinas M1, M2 e M3 (Figura 2.5(a)), o Produto 2 segue uma seqüência diferente, determinada pelas máquinas M2, M3 e M4 (Figura 2.5(b)). No caso, apenas as máquinas M2 e M3 são compartilhadas. A Figura 2.6 mostra os grafos que seriam gerados para o sistema da Figura 2.5 para um horizonte de planejamento discretizado em três períodos de tempo. Deve-se notar que as demandas podem ser diversas para cada período e para cada tipo de produto.

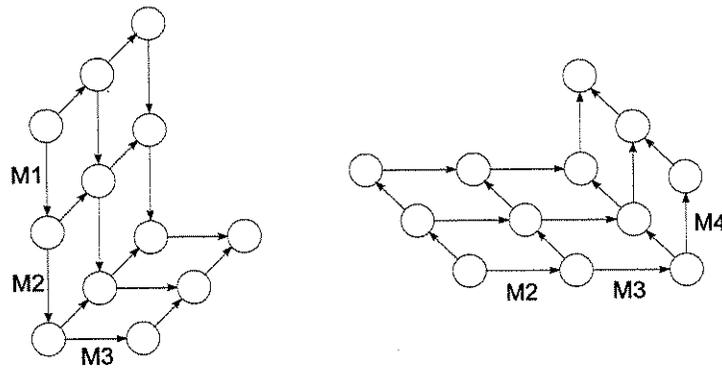


Figura 2.6 Representação por grafo de seqüências de produção diferentes

A matriz relativa às restrições (2.5), (2.6), (2.7), (2.8) e (2.9) tem o formato da Figura 2.7(a), para o caso em que os  $P$  produtos compartilham totalmente a linha de produção.  $A_p$  é a matriz de incidência nó-arco associada ao grafo relacionado com o produto  $p$  e  $S_p$  é uma matriz esparsa, contendo no máximo um elemento não nulo por linha, associada aos coeficientes das variáveis relacionadas com o produto  $p$ , nas restrições de capacidade mútua (2.9).

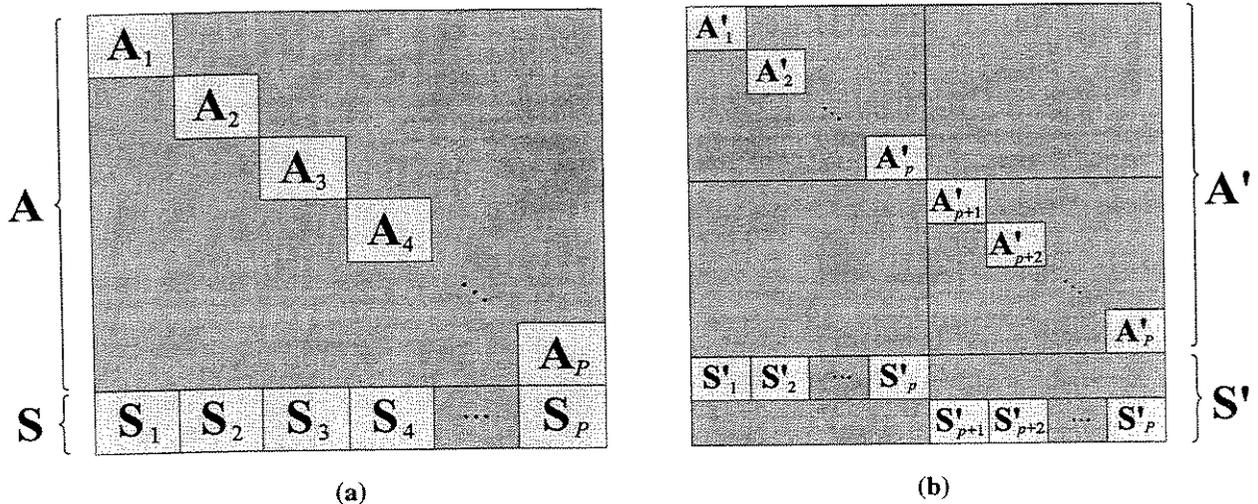


Figura 2.7 Estrutura da matriz de restrições



A matriz da Figura 2.7(b) ilustra uma situação de compartilhamento parcial da linha. No caso, o grupo de máquinas compartilhado pelos produtos de 1 a  $p$  é disjunto do grupo de máquinas que os produtos de  $p+1$  a  $P$  compartilham. Esta estrutura disjunta (que permite dividir o problema global em subproblemas a serem resolvidos separadamente), bem como o formato de *blocos diagonais* da matriz  $A$  e as *esparsidades localizadas* da matriz  $S$ , são características especiais importantes na busca de um algoritmo eficiente para a resolução do problema.



## Capítulo 3

### Algoritmo para Resolução de Problemas com Restrições Adicionais

O modelo de fluxo em redes com restrições adicionais descrito no Capítulo 2 pode ser escrito na forma matricial

$$\begin{array}{ll} \text{Min} & \mathbf{cx} \\ \text{s.a.} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{Sx} \leq \mathbf{r} \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \end{array} \quad (3.1)$$

onde  $\mathbf{A}$  é a matriz de incidência nó-arco relacionada com as restrições de conservação de fluxo (também conhecidas como restrições de balanço da rede ou *restrições de grafo*) e  $\mathbf{S}$  é a matriz associada aos coeficientes das variáveis das restrições adicionais (também conhecidas como *side constraints*). Estas matrizes possuem uma estrutura particular que deve ser explorada pelo algoritmo de solução.

Existem diversas abordagens para resolução deste tipo de problema, das quais serão tratadas neste capítulo:

- *método de decomposição primal* (Kennington, 1980), implementado no algoritmo NETSIDE (Kennington, 1990);
- *método de decomposição simplicial*, implementado no algoritmo RETRA (Fernandes, 1999).



O método de decomposição primal, que será apresentado na seção 3.1, foi a base para a implementação do algoritmo Side Constraints (SC) (uma especialização do algoritmo NETSIDE para resolução de problemas de fluxo em redes sujeitos às restrições adicionais de capacidade mútua apresentadas no Capítulo 2) apresentado na seção 3.2 a seguir. Na seção 3.3 são apresentadas técnicas de inicialização que dão maior agilidade ao algoritmo. A seção 3.4 propõe um método híbrido, baseado nos algoritmos RETRA e SC, como alternativa para resolução do problema. Finalmente, na seção 3.5 é feita uma análise do desempenho do SC, comparando-o a outros algoritmos aplicáveis ao problema em estudo.

### 3.1 Método de Decomposição Primal (Kennington, 1980)

O método de decomposição primal é uma especialização do método primal simplex para problemas de fluxo em redes com restrições adicionais genéricas. O método parte da seguinte definição de uma base para a matriz de restrições:

$$\bar{\mathbf{B}} = \begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{D} & \mathbf{F} \end{bmatrix} \quad (3.2)$$

onde  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{D}$  e  $\mathbf{F}$  são as partições que compõem a base. A matriz  $\mathbf{B}$  é composta por colunas de  $\mathbf{A}$  incorporadas à base. A matriz  $\mathbf{C}$  possui uma estrutura particular pois é composta por colunas de  $\mathbf{A}$  ou por colunas de zeros. No primeiro caso as colunas referem-se aos arcos da rede e portanto têm dois elementos diferentes de zero por coluna. No segundo caso são colunas de zeros referentes às variáveis de folga. As matrizes  $\mathbf{D}$  e  $\mathbf{F}$  resultam da definição das matrizes  $\mathbf{B}$  e  $\mathbf{C}$ .

As variáveis associadas a  $\mathbf{B}$  são chamadas de **variáveis chave**. As variáveis restantes, associadas a  $\mathbf{C}$ , são chamadas de **variáveis não-chave**. A inversa da base é dada por

$$\bar{\mathbf{B}}^{-1} = \begin{bmatrix} \mathbf{B}^{-1} + \mathbf{B}^{-1}\mathbf{C}\mathbf{Q}^{-1}\mathbf{D}\mathbf{B}^{-1} & -\mathbf{B}^{-1}\mathbf{C}\mathbf{Q}^{-1} \\ -\mathbf{Q}^{-1}\mathbf{D}\mathbf{B}^{-1} & \mathbf{Q}^{-1} \end{bmatrix} \quad (3.3)$$

onde

$$\mathbf{Q} = \mathbf{F} - \mathbf{D}\mathbf{B}^{-1}\mathbf{C} \quad (3.4)$$

é chamada *matriz de trabalho*.



A atualização da inversa da base pode ser feita a partir de elementos de  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{D}$  e  $\mathbf{Q}^{-1}$ . O problema consiste em trabalhar eficientemente com estes elementos. A seguir é apresentado o algoritmo NETSIDE, uma implementação em código Fortran do método de decomposição primal.

### 3.1.1 Algoritmo NETSIDE

- Passo 1:* Particione as variáveis em básicas chave, básicas não-chave e não-básicas.  
*Passo 2:* Determine  $\mathbf{Q}$  e  $\mathbf{Q}^{-1}$ .  
*Passo 3:* Atualize os custos reduzidos.  
*Passo 4:* Escolha a variável não básica  $k$ , candidata a entrar na base. Se não existem variáveis candidatas, FIM.  
*Passo 5:* Atualize a coluna da variável  $k$ .  
*Passo 6:* Determine valor de acréscimo das variáveis ( $\Delta$ ) e a variável que deixa a base.  
*Passo 7:* Atualize as variáveis.  
*Passo 8:* Atualize  $\mathbf{Q}^{-1}$  e obtenha a nova partição das variáveis. Volte ao Passo 3.

Para enfatizar como a estrutura de grafo do problema pode ser aproveitada, os principais passos do algoritmo, referentes à atualização dos custos reduzidos (Passo 3), atualização da coluna candidata a entrar na base (Passo 5) e atualização da inversa da matriz de trabalho (Passo 8), são apresentados a seguir em detalhes.

*Atualização dos Custos Reduzidos.* A estrutura da base pode ser usada para calcular eficientemente as variáveis duais utilizadas no algoritmo primal simplex. De acordo com a partição usada em (3.2), tem-se

$$\begin{aligned} \pi = \mathbf{c}^B \mathbf{B}^{-1} &\Rightarrow [\pi^1 \quad \pi^2] = [\mathbf{c}^1 \quad \mathbf{c}^2] \begin{bmatrix} \mathbf{B}^{-1} + \mathbf{B}^{-1} \mathbf{C} \mathbf{Q}^{-1} \mathbf{D} \mathbf{B}^{-1} & -\mathbf{B}^{-1} \mathbf{C} \mathbf{Q}^{-1} \\ -\mathbf{Q}^{-1} \mathbf{D} \mathbf{B}^{-1} & \mathbf{Q}^{-1} \end{bmatrix} \\ &= [(\mathbf{c}^1 + \mathbf{c}^1 \mathbf{B}^{-1} \mathbf{C} \mathbf{Q}^{-1} \mathbf{D} - \mathbf{c}^2 \mathbf{Q}^{-1} \mathbf{D}) \mathbf{B}^{-1} \quad (\mathbf{c}^2 - \mathbf{c}^1 \mathbf{B}^{-1} \mathbf{C}) \mathbf{Q}^{-1}] \end{aligned}$$

As operações envolvendo  $\mathbf{B}^{-1}$  são eficientemente realizadas como em um problema de fluxo em redes que utiliza um algoritmo específico para fluxo em redes (Jensen, 1980). Para explorar este fato, o vetor  $\pi$  pode ser calculado pela seguinte seqüência de passos:



- Passo 3.1: Faça  $\gamma^1 = \mathbf{c}^1 \mathbf{B}^{-1}$   
 Passo 3.2: Faça  $\gamma^2 = \mathbf{c}^1 + (\gamma^1 \mathbf{C} - \mathbf{c}^2) \mathbf{Q}^{-1} \mathbf{D}$   
 Passo 3.3: Faça  $\pi^1 = \gamma^2 \mathbf{B}^{-1}$   
 Passo 3.4: Faça  $\pi^2 = (\mathbf{c}^2 - \gamma^1 \mathbf{C}) \mathbf{Q}^{-1}$

Nos passos 3.1 e 3.3, a estrutura de árvore da matriz  $\mathbf{B}$  é explorada. Devido às suas estruturas particulares, as matrizes  $\mathbf{C}$  e  $\mathbf{D}$  não precisam ser armazenadas explicitamente, pois seus elementos não nulos estão colocados em posições muito bem definidas. Este fato permite que as operações envolvendo essas duas matrizes sejam realizadas de forma rápida, explorando portanto a estrutura particular do problema.

*Atualização da Coluna da Variável  $k$ .* Utilizando a partição adotada em (3.2) e definindo-se  $\mathbf{y}$  e  $\overline{\mathbf{A}}(k)$  adequadamente, tem-se

$$\mathbf{y} = \overline{\mathbf{B}}^{-1} \overline{\mathbf{A}}(k) \Rightarrow \begin{bmatrix} \mathbf{y}^1 \\ \mathbf{y}^2 \end{bmatrix} = \begin{bmatrix} \mathbf{B}^{-1} + \mathbf{B}^{-1} \mathbf{C} \mathbf{Q}^{-1} \mathbf{D} \mathbf{B}^{-1} & -\mathbf{B}^{-1} \mathbf{C} \mathbf{Q}^{-1} \\ -\mathbf{Q}^{-1} \mathbf{D} \mathbf{B}^{-1} & \mathbf{Q}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{A}(k) \\ \mathbf{S}(k) \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{B}^{-1} (\mathbf{A}(k) + \mathbf{C} \mathbf{Q}^{-1} \mathbf{D} \mathbf{B}^{-1} \mathbf{A}(k) - \mathbf{C} \mathbf{Q}^{-1} \mathbf{S}(k)) \\ \mathbf{Q}^{-1} (\mathbf{S}(k) - \mathbf{D} \mathbf{B}^{-1} \mathbf{A}(k)) \end{bmatrix}$$

Esta atualização da coluna não básica pode ser calculada através dos seguintes passos:

- Passo 5.1: Faça  $\gamma^1 = \mathbf{B}^{-1} \mathbf{A}(k)$   
 Passo 5.2: Faça  $\gamma^2 = \mathbf{A}(k) + \mathbf{C} \mathbf{Q}^{-1} (\mathbf{D} \gamma^1 - \mathbf{S}(k))$   
 Passo 5.3: Faça  $\mathbf{y}^1 = \mathbf{B}^{-1} \gamma^2$   
 Passo 5.4: Faça  $\mathbf{y}^2 = \mathbf{Q}^{-1} (\mathbf{S}(k) - \mathbf{D} \gamma^1)$

O passo 5.1 determina o ciclo que a variável  $k$ , candidata a entrar na base, “fecha” na árvore, enquanto o passo 5.3, que é efetuado de forma semelhante ao 5.1, determina a partição chave do vetor  $\mathbf{y}$ . As demais operações envolvendo  $\mathbf{C}$  e  $\mathbf{D}$  são realizadas de forma semelhante à atualização dos custos reduzidos.

*Atualização da Inversa da Matriz de Trabalho.* Mostra-se que

$$\mathbf{Q}^{-1}_{i+1} = (\mathbf{E}_4 - \mathbf{E}_3 \mathbf{B}^{-1} \mathbf{C}) \mathbf{Q}^{-1}_i$$



onde  $E_4$  e  $E_3$  são partições de  $E$ , uma *matriz coluna elementar* ou *matriz de permutação* dada, em concordância com a partição da base (3.2), por

$$E = \begin{bmatrix} E_1 & E_2 \\ E_3 & E_4 \end{bmatrix}$$

Se a coluna que deixa a base é não-chave, então  $E_3 B^{-1} C$  resulta em uma matriz nula, o que implica em  $Q^{-1}_{i+1} = E_4 Q^{-1}_i$ , onde  $E_4$  é uma matriz coluna elementar. Se a coluna que deixa a base é chave, então tal coluna corresponde a um arco que, se for removido, deve ser substituído por outro arco pertencente ao conjunto de variáveis não-chave, para que a estrutura de grafo seja mantida. Seja  $l$  o índice da coluna relativa ao arco que deixa a base e  $\beta^l$  a  $l$ -ésima linha de  $B^{-1}$ . Defina-se  $\gamma = \beta^l C$ . Há dois casos possíveis:

- *Caso 1:  $\gamma \neq 0$ .* Neste caso, prova-se que, para  $\gamma_j \neq 0$ , a  $j$ -ésima coluna de  $C$  pode ser trocada com a  $l$ -ésima coluna de  $B$  e, com  $E_4 - E_3 B^{-1} C = E_5$ , tem-se  $Q^{-1}_{i+1} = E_5 Q^{-1}_i$  onde

$$E_5 = \begin{bmatrix} \mathbf{I} & & & \\ & & & \\ & & -\beta^l C & \\ & & & \mathbf{I} \end{bmatrix}$$

Após esta transformação, a coluna que deixa a base passa a ser não-chave e pode-se aplicar  $Q^{-1}_{i+1} = E_4 Q^{-1}_i$  como anteriormente.

- *Caso 2:  $\gamma = 0$ .* Neste caso é feito um pivoteamento direto e  $Q^{-1}_{i+1} = Q^{-1}_i$

### 3.2 Algoritmo SC

O algoritmo *Side Constraints* (SC), implementado em linguagem C, é uma especialização do método de decomposição primal para o problema de fluxo em redes sujeito às restrições adicionais de capacidade mútua associadas à matriz  $S$  em (3.1). Para resolver o problema dado por (3.1) através do SC é necessário reescrevê-lo na forma

$$\begin{aligned} \text{Min} \quad & \mathbf{cx} + \mathbf{dz} \\ \text{s.a.} \quad & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{Sx} + \mathbf{Pz} = \mathbf{r} \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}, \quad \mathbf{0} \leq \mathbf{z} \leq \mathbf{v} \end{aligned} \tag{3.5}$$



onde  $\mathbf{P}$  é a matriz dos coeficientes das variáveis de folga dadas pelo vetor  $\mathbf{z}$  e  $\mathbf{d}$  é o vetor dos custos associados a estas mesmas variáveis. As subseções 3.2.1 a 3.2.8 apresentam a estrutura do SC, tomando por base os oito passos do algoritmo NETSIDE.

### 3.2.1 Partição de Variáveis

As variáveis são particionadas inicialmente em básicas e não-básicas, de acordo com a solução obtida pela heurística de inicialização (vide seção 3.3). Por sua vez, as variáveis básicas são particionadas em **chaves** e **não-chaves**, de acordo com a partição da base dada por (3.2). No SC adotou-se a estratégia de escolher sempre as variáveis de folga como variáveis não-chave iniciais pois, como será mostrado em seguida, isto facilita o cálculo da matriz de trabalho  $\mathbf{Q}$ .

### 3.2.2 Determinação de $\mathbf{Q}$ e $\mathbf{Q}^{-1}$

Com a partição de variáveis proposta em 3.2.1, a base assume a seguinte configuração inicial

$$\left[ \begin{array}{c|c} \mathbf{B} & \\ \hline \mathbf{D} & \mathbf{I} \end{array} \right] \quad (3.6)$$

onde a matriz  $\mathbf{C}$ , associada às variáveis não-chave, é nula e a matriz identidade  $\mathbf{I}$  corresponde à matriz  $\mathbf{F}$  em (3.2). Assim, segundo a equação (3.4), tem-se  $\mathbf{Q} = \mathbf{I} - \mathbf{DB}^{-1}\mathbf{0} = \mathbf{I}$  (o símbolo  $\mathbf{0}$  representa aqui a matriz nula) o que implica  $\mathbf{Q}^{-1} = \mathbf{I}$ . Este resultado tem importância porque evita a utilização de rotina para inversão da matriz  $\mathbf{Q}$  e a conseqüente perda de eficiência do algoritmo.

### 3.2.3 Atualização dos Custos Reduzidos

Todos os elementos das matrizes envolvidas no cálculo do vetor  $\pi$ , exceção feita à matriz  $\mathbf{Q}^{-1}$ , podem ser obtidos indiretamente, a partir dos parâmetros do modelo (sabe-se que as matrizes  $\mathbf{B}$  e  $\mathbf{C}$  estão associadas ao grafo que representa a linha de produção e que a matriz  $\mathbf{D}$  tem relação com os parâmetros de carga horária e capacidade dos estágios). Portanto, é desnecessário, e também desinteressante do ponto de vista de desempenho, armazenar em memória tais matrizes.



A estrutura de grafo da matriz  $\mathbf{B}$  permite que os cálculos envolvendo **pós-multiplicação** de um vetor (de dimensão compatível) pela matriz  $\mathbf{B}^{-1}$  sejam realizados através de rotinas específicas para obtenção dos **potenciais** dos nós da árvore relativa à base  $\mathbf{B}$  (Jensen, 1980). O SC utiliza este tipo de rotina para calcular os vetores  $\gamma^1$  e  $\pi^1$  (passos 3.1 e 3.3 do NETSIDE, respectivamente). Apenas os cálculos envolvendo  $\mathbf{Q}^{-1}$  são realizados utilizando-se álgebra matricial, já que  $\mathbf{Q}^{-1}$  precisa ser armazenada em forma de matriz (vide subseção 3.2.8). O vetor de potenciais  $\pi$  é utilizado para a obtenção dos custos reduzidos associados às variáveis não básicas.

#### *3.2.4 Escolha da Variável $k$ Candidata a Entrar na Base*

Os custos reduzidos calculados na etapa anterior são utilizados na escolha da variável  $k$  candidata a entrar na base. A técnica de escolha obedece a **critérios de otimalidade** determinados pelo método primal simplex (Kennington, 1980).

#### *3.2.5 Atualização da Coluna Correspondente à Variável $k$*

Também aqui a estrutura de grafo da matriz  $\mathbf{B}$  pode ser explorada. O SC utiliza rotinas eficientes propostas por Jensen (1980) para operações de atualização de vetores colunas (tais operações, presentes nos passos 5.1 e 5.3 do NETSIDE, correspondem à **pré-multiplicação** das referidas colunas pela matriz  $\mathbf{B}^{-1}$ ). O vetor  $\mathbf{y}$  resultante será usado na etapa seguinte.

#### *3.2.6 Determinação de $\Delta$ e da Variável que Deixa a Base*

Nesta etapa, que corresponde ao **teste da razão** do método primal simplex, calcula-se, a partir do vetor  $\mathbf{y}$ , o valor  $\Delta$  de correção das variáveis e determina-se a variável que deixará a base.

#### *3.2.7 Atualização das Variáveis*

As variáveis básicas chave, básicas não-chave e não-básicas que necessitam de atualização recebem novos valores, adicionando-se ou subtraindo-se  $\Delta$ , de acordo com a técnica simplex.



### 3.2.8 Atualização de $Q^{-1}$ e da Partição das Variáveis

Devido às peculiaridades de  $Q^{-1}$ , o SC utiliza uma estrutura de dados especial para armazenagem e recuperação eficientes de elementos em matrizes esparsas (Lucchesi, 1997 ; Chang, 1992 ; Watkins, 1991).

## 3.3 Inicialização

O algoritmo SC e o método primal Simplex possuem, naturalmente, diversas características em comum, já que o primeiro é uma especialização do segundo. Uma destas características é a necessidade de obtenção de uma *solução básica factível* (SBF) para iniciar o algoritmo (Bazaraa, 1990). Foi visto anteriormente que, para problemas de fluxo em redes com restrições adicionais, qualquer solução, incluindo a SBF inicial, deve satisfazer dois tipos de restrições: 1) restrições formadoras da matriz de incidência nó-arco (*restrições de grafo*) e 2) *restrições adicionais* complicadoras do problema (*side constraints*). Vale lembrar que apenas a factibilidade da solução, desvinculada do caráter básico, é insuficiente. Também não é suficiente identificar uma base adequada se a solução correspondente for infactível.

Pode-se conseguir uma SBF inicial para o SC através de uma variante do método do *M-grande* da programação linear (Ahuja, 1993 ; Bazaraa, 1990). Este método, quando adaptado para modelos de fluxo em redes, utiliza arcos artificiais para compor a base. No início, apenas os arcos artificiais serão “carregados” com algum fluxo, de maneira que, com o ajuste dos valores das variáveis de folga presentes nas restrições adicionais, a factibilidade da solução estará garantida. A Figura 3.1 mostra (a) o modelo de fluxo em redes para um sistema de produção com 4 estágios e 2 produtos, para 2 períodos de tempo e (b) a *árvore inicial* utilizada pelo método do *M-grande*. Note-se que esta árvore caracteriza uma solução degenerada para o modelo de rede, uma vez que todos os arcos, com exceção dos artificiais conectados à raiz, estão em seus limites inferiores (com fluxo nulo).

Apesar de sua aplicabilidade, a técnica do *M-grande* tem-se mostrado ineficiente para o modelo proposto neste trabalho, pois, somente para a operação de “pré-otimização” (esta operação, correspondente ao “carregamento” do sistema, será explicada na subseção 3.3.1), é



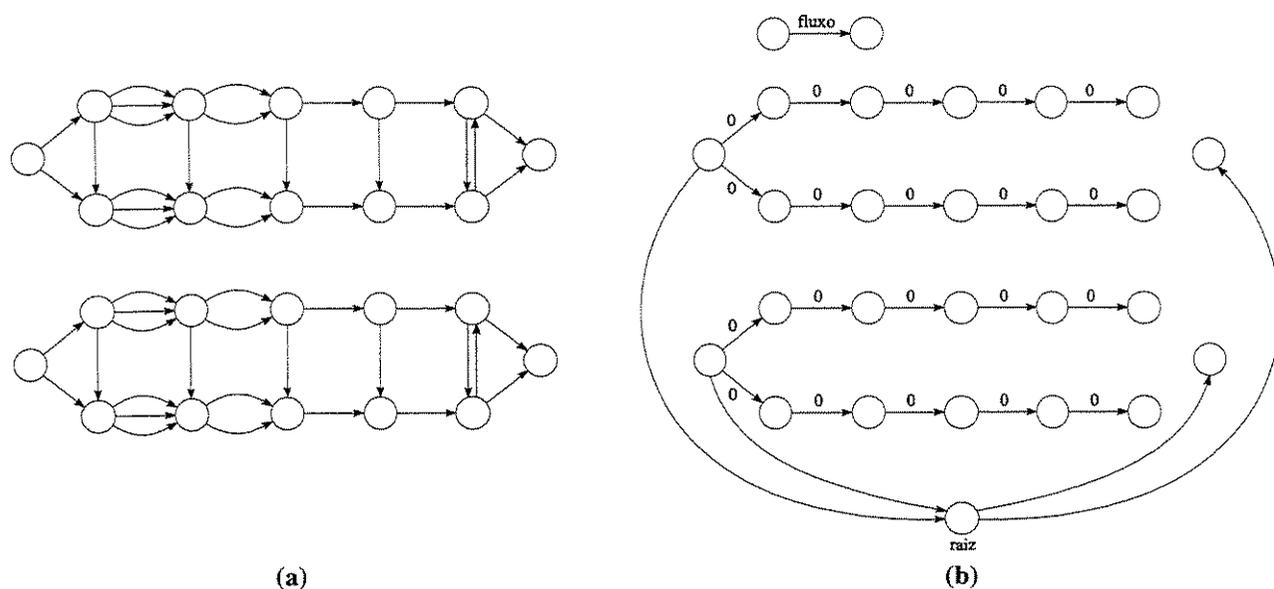


Figura 3.1 Modelo de fluxo em redes e árvore básica

necessário um número de iterações (do algoritmo SC) da ordem da quantidade de produtos envolvidos no sistema modelado. Em problemas práticos, de grande dimensão, esta quantidade de iterações leva a um aumento significativo do tempo de resolução do algoritmo, um inconveniente que pode ser contornado através da proposição de *heurísticas* que façam uso das particularidades do modelo e da experiência do programador da produção.

Uma heurística de inicialização deve tirar proveito do fato de que, em modelos como o da Figura 3.1(a), a solução ótima tende a concentrar-se nos arcos de produção, o que sugere que a solução inicial deve ser direcionada para uma configuração semelhante à da Figura 3.1(b), porém com atribuição de fluxos convenientes aos arcos, de forma a aproximar-se do resultado ótimo. Nas próximas duas subseções serão discutidos os algoritmos heurísticos implementados para a inicialização do SC.

### 3.3.1 Heurística 1

A primeira heurística de inicialização está representada no pseudo-código mostrado a seguir. Na representação, deve-se entender o termo *linha* como uma sequência de arcos começando em um nó de suprimento e terminando em um nó de demanda (ver Figura 3.2 no exemplo apresentado mais adiante).



```

procedure Inicializa
begin
  for each linha Lin do
    Dem ← DeterminaDemandaParaLinha(Lin);
    CarregaLinha(Lin , Dem);
  endFor;

  repeat
    ExisteRestricaoViolada ← VerificaRestricoes();
    if ExisteRestricaoViolada then
      Lin ← IdentificaLinhaDeViolacao();
      DescarregaLinhaDeViolacao(Lin);
    endIf;
  until ExisteRestricaoViolada = False
end;

```

**Etapa 1:**  
 determinação dos arcos e fluxos  
 da árvore inicial (para satisfazer as  
 restrições de grafo)

**Etapa 2:**  
 Identificação e correção de  
 violações (para satisfazer  
 as restrições adicionais)

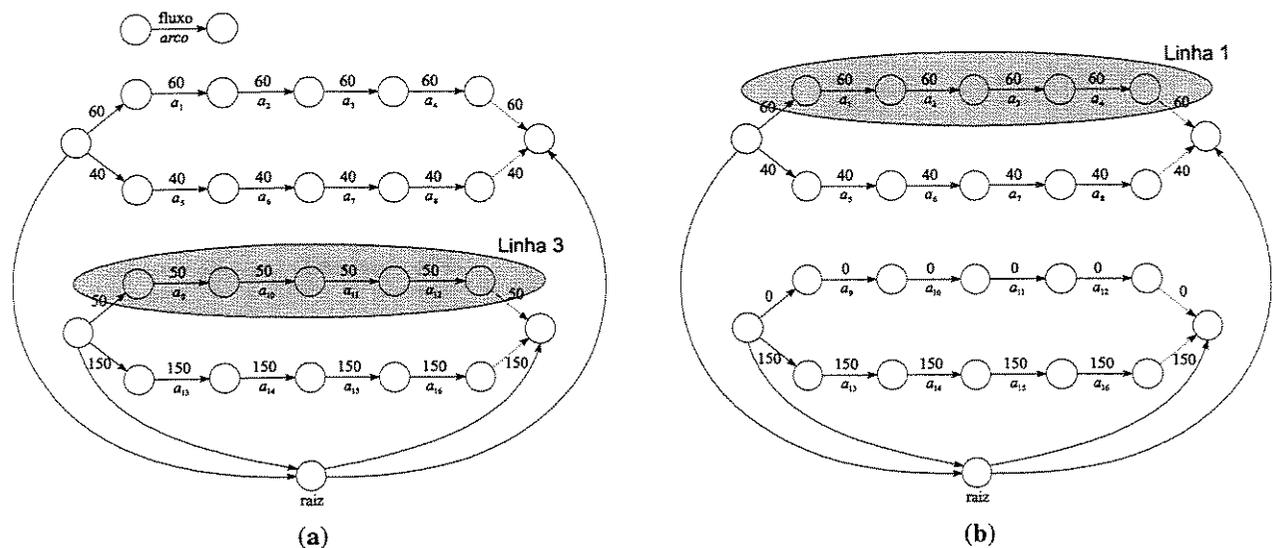
A primeira parte do procedimento *Inicializa* faz a configuração da árvore básica, “carregando” (procedimento *CarregaLinha*) os arcos de cada linha com valores de fluxo adequados à demanda (função *DeterminaDemandaParaLinha*) requerida pela linha. Esta etapa gera uma solução que satisfaz as *restrições de grafo* mas que pode, eventualmente, não satisfazer as *restrições adicionais* relativas à capacidade de utilização mútua das máquinas. Por isso, torna-se necessária uma segunda etapa que, repetidamente, verifica (função *VerificaRestricoes*) se existe (ou não) alguma restrição de capacidade violada e, em caso positivo, atribui fluxos nulos aos arcos da linha responsável pela violação (a função *IdentificaLinhaDeViolacao* retorna o identificador da linha que provocou a violação; esta linha será então “descarregada” pelo procedimento *DescarregaLinhaDeViolacao*). Quando uma linha que está provocando violação de uma restrição é descarregada, a violação “diminui” (mostra-se que a violação é mensurável, portanto, nesse sentido, pode-se falar em diminuição da mesma). Este fato garante que a repetição do processo de verificação de restrições e descarregamento de linhas leva à anulação de todas as eventuais violações, e, conseqüentemente, à obtenção de uma solução factível (no pior caso, todas as linhas serão descarregadas e, portanto, não poderá haver nenhuma violação nas restrições adicionais).

### Exemplo

A Figura 3.2 ilustra o funcionamento da heurística de inicialização descrita. As linhas já estão carregadas convenientemente, segundo as demandas correspondentes (notar que os arcos de demanda, pontilhados, também estão carregados mas não fazem parte da árvore, a fim de evitar



ciclos que descaracterizariam a solução básica). Como  $a_1 + a_9 = a_2 + a_{10} = a_3 + a_{11} = a_4 + a_{12} = 110$  (de acordo com os fluxos da Figura 3.2(a)), as restrições adicionais de capacidade mútua (Figura 3.2(c)) relativas às máquinas 1 e 2, no período 1, estão satisfeitas, mas as restrições relativas às máquinas 3 e 4, no mesmo período, estão sendo violadas. A primeira violação é causada pela variável  $a_{11}$  pois, ao percorrer a restrição, a heurística só percebe a violação após somar  $a_{11}$  a  $a_3$ . Assim, a linha que contém o arco  $a_{11}$  (linha 3, indicada na Figura 3.2(a)) será descarregada e a terceira restrição não mais estará violada já que, após o descarregamento (Figura 3.2(b)), tem-se  $a_3 + a_{11} = a_3 + 0 = 60 < Cap3 = 100$  onde  $Cap3$  é a capacidade mútua da máquina 3, no primeiro período. Deve-se notar que as restrições 1 e 2 analisadas anteriormente continuam satisfeitas, fato que é relevante para a economia de operações de busca no algoritmo.



	Produto 1		Produto 2	Capacidade Mútua
Período 1	$a_1$	+	$a_9$	$\leq 120$ ⚡ Máquina 1
	$a_2$	+	$a_{10}$	$\leq 150$ ⚡ Máquina 2
	$a_3$	+	$a_{11}$	$\leq 100$ ⚡ Máquina 3
	$a_4$	+	$a_{12}$	$\leq 50$ ⚡ Máquina 4
Período 2	$a_5$	+	$a_{13}$	$\leq 200$ ⚡ Máquina 1
	$a_6$	+	$a_{14}$	$\leq 210$ ⚡ Máquina 2
	$a_7$	+	$a_{15}$	$\leq 190$ ⚡ Máquina 3
	$a_8$	+	$a_{16}$	$\leq 300$ ⚡ Máquina 4

(c)

Figura 3.2 Base e restrições adicionais do modelo de fluxo em redes de uma pequena linha de produção



O próximo passo é corrigir a restrição 4, que continua não satisfeita. A linha 1 (indicada na Figura 3.2(b)), que contém o arco correspondente à variável  $a_4$ , deve ser descarregada, o que elimina todas as violações relacionadas com as quatro primeiras restrições adicionais. Como as restrições do período 2 já estão satisfeitas, a solução atual é factível e pode ser usada para inicializar o SC.

O gráfico da Figura 3.3 mostra, para problemas com 90, 100 e 110 produtos, os percentuais de coincidência e não coincidência entre soluções ótimas e soluções heurísticas, considerando-se todos os arcos associados às máquinas indicadas. Note-se que, mesmo no caso em que houve a maior discrepância, aproximadamente 80% dos fluxos obtidos pela heurística são iguais aos fluxos ótimos, o que pode ser considerado um bom índice de eficiência para uma heurística.

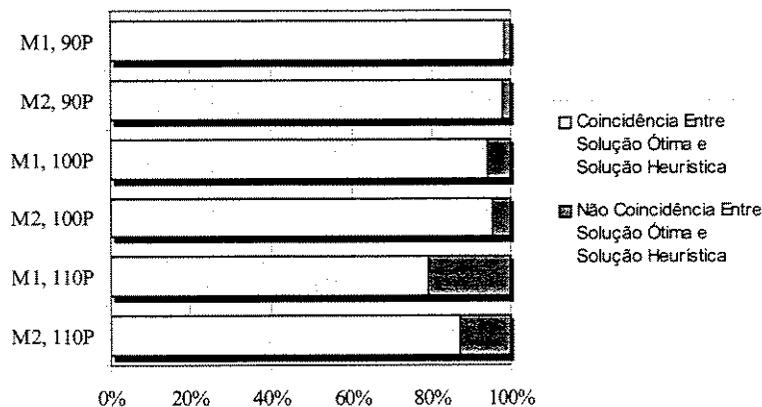


Figura 3.3 Comparação entre soluções heurísticas e soluções ótimas

### 3.3.2 Heurística 2

A segunda heurística de inicialização parte da seguinte idéia: carregar as linhas como anteriormente e então corrigir as restrições adicionais através da introdução de variáveis artificiais com valores convenientes que, subtraídos na restrição, eliminam eventuais violações.

Foi visto no Capítulo 2 que toda restrição adicional de capacidade mútua para uma determinada máquina  $Maq$  tem a forma

$$\sum_{i \in I} x_i + x_{Maq}^f = C_{Maq}$$



onde, para um período fixado,  $I$  é o conjunto de índices para os produtos (itens) processados pela máquina  $Maq$ ,  $x_i$  é a variável que representa a quantidade do produto de índice  $i$  processada pela máquina  $Maq$ ,  $x_{Maq}^f$  é a variável de folga que ajusta a restrição associada à máquina  $Maq$ , e  $C_{Maq}$  é a capacidade total disponível na máquina  $Maq$  para processamento mútuo dos produtos indexados em  $I$ . A estratégia é transformar toda restrição violada em uma nova restrição do tipo

$$\sum_{i \in I} x_i + x_{Maq}^f - x_{Maq}^a = C_{Maq}$$

onde  $x_{Maq}^a$  é uma variável artificial à qual deve-se atribuir inicialmente o valor da violação. As variáveis artificiais devem estar presentes na base, que terá a forma

$$\left( \begin{array}{c|c} \mathbf{B} & \mathbf{0} \\ \hline \mathbf{D} & \pm \mathbf{I} \end{array} \right) \begin{array}{l} \text{Coeficientes das Variáveis das Restrições de Grafo} \\ \text{Coeficientes das Variáveis das Restrições Adicionais} \end{array}$$

onde (vide seção 3.1)  $\mathbf{B}$  é a partição da matriz de incidência nó-arco correspondente à árvore básica,  $\mathbf{D}$  é a matriz dos coeficientes das variáveis básicas associadas às máquinas nas restrições adicionais, e  $\pm \mathbf{I}$  é uma matriz quadrada cujos elementos da diagonal principal podem ser  $+1$  (coeficiente das **variáveis de folga** colocadas na base no caso de não haver violação de restrições) ou  $-1$  (coeficiente das **variáveis artificiais** colocadas na base no caso de haver violação de restrições) e todos os outros elementos são nulos. Feitas estas transformações, pode-se resolver o problema diretamente através do SC.

Claramente, o problema sujeito à estas novas restrições é diferente do original. Porém, existe uma relação estrita entre os dois problemas, o que permite o uso desta técnica. Assim como no método do M-grande, associa-se um custo alto às variáveis artificiais, de forma a incentivar a função objetivo a levá-las ao valor nulo. Se, na solução ótima do problema modificado, todas as variáveis artificiais estiverem no nível zero, pode-se dizer seguramente que esta solução, desconsiderando-se as variáveis artificiais introduzidas, é ótima também para o problema original. Se, contudo, alguma variável artificial for não nula, significa que o algoritmo não conseguiu eliminar todas as violações existentes e o problema original é infactível.



*Exemplo.* No problema relativo à árvore da Figura 3.2(a) e às restrições adicionais da Figura 3.2(c) da seção anterior, existem violações nas restrições correspondentes às máquinas 3 e 4 no primeiro período. A heurística de inicialização atribui os respectivos valores das violações às variáveis artificiais, que são introduzidas com coeficiente  $-1$  nas restrições, como mostrado na Figura 3.4.

	Produto 1		Produto 2		Variáveis de Folga e Artificiais	Capacidade Mútua
Período 1	$a_1$	+	$a_9$	+	$x_{1,1}^f$	= 120 ; $x_{1,1}^f = 120 - a_1 - a_9 = 10$
	$a_2$	+	$a_{10}$	+	$x_{2,1}^f$	= 150 ; $x_{2,1}^f = 150 - a_2 - a_{10} = 40$
	$a_3$	+	$a_{11}$	⊖	$x_{3,1}^a$	= 100 ; $x_{3,1}^a = a_3 + a_{11} - 100 = 10$
	$a_4$	+	$a_{12}$	⊖	$x_{4,1}^a$	= 50 ; $x_{4,1}^a = a_4 + a_{12} - 50 = 60$
Período 2	$a_5$	+	$a_{13}$	+	$x_{1,2}^f$	= 200 ; $x_{1,2}^f = 200 - a_5 - a_{13} = 10$
	$a_6$	+	$a_{14}$	+	$x_{2,2}^f$	= 210 ; $x_{2,2}^f = 210 - a_6 - a_{14} = 20$
	$a_7$	+	$a_{15}$	+	$x_{3,2}^f$	= 190 ; $x_{3,2}^f = 190 - a_7 - a_{15} = 0$
	$a_8$	+	$a_{16}$	+	$x_{4,2}^f$	= 300 ; $x_{4,2}^f = 300 - a_8 - a_{16} = 110$

**Figura 3.4** Restrições adicionais modificadas

### 3.4 Algoritmo Híbrido RETRA-SC

Dentre diversas abordagens utilizadas atualmente para resolução do tipo de problema discutido neste trabalho, incluindo decomposição de Dantzig-Wolfe (Ahuja, 1993), relaxação de Lagrange (Ahuja, 1993), pontos interiores (Fujisawa, 1997) e outras, o algoritmo RETRA (Fernandes, 1999) tem-se mostrado uma das mais eficientes. Entretanto, tal eficiência depende da escolha correta de determinados parâmetros relacionados com a técnica de penalidade na qual se baseia o algoritmo.

O gráfico da Figura 3.5 demonstra a influência do parâmetro de penalidade  $\mu$  no comportamento do RETRA. Nota-se que pequenas variações nos valores considerados adequados para este parâmetro podem provocar uma piora sensível no desempenho do algoritmo.

Para determinar adequadamente os parâmetros de penalidade, é necessário conhecer a estrutura do algoritmo e ter experiência acerca do sistema de manufatura modelado. Portanto, o RETRA pode tornar-se pouco confiável quando utilizado por usuários não-especialistas. A



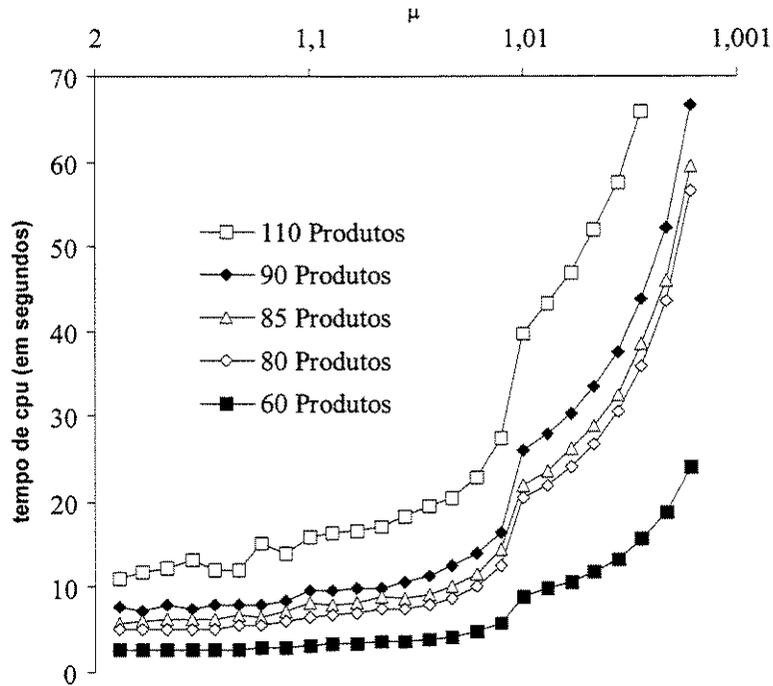


Figura 3.5 Alteração do tempo de execução do RETRA devido à variação do parâmetro  $\mu$

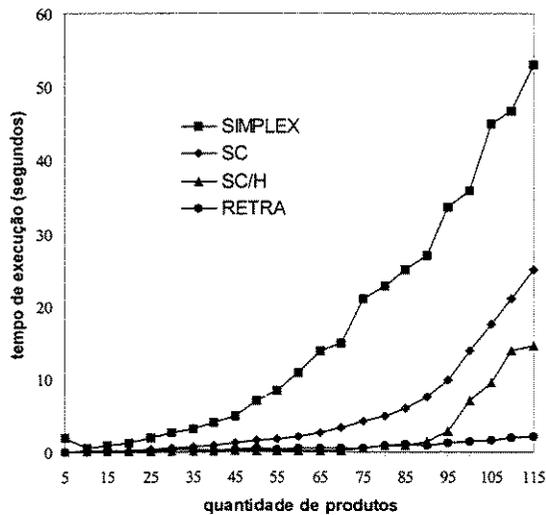
proposta é “rodar” o RETRA durante algumas iterações, resolvendo o problema parcialmente, e então “trocar” para o SC, obtendo-se a solução ótima. Deve ser estudada uma heurística que determine um número adequado de iterações do RETRA antes da mudança para o SC. É preciso também atentar para o fato de que a solução parcial fornecida pelo RETRA (supondo-se que a mudança de algoritmo será feita antes de se atingir a solução ótima) é infactível com relação às restrições adicionais de capacidade (esta é uma característica do RETRA explicada com maiores detalhes em Fernandes (1999)), sendo necessário utilizar uma heurística de factibilização (possivelmente a Heurística 2 apresentada na seção 3.3.2) por ocasião da troca de solução para o SC. Com a conjunção entre RETRA e SC espera-se conseguir um *método híbrido* que una confiabilidade a velocidade.

### 3.5 Resultados

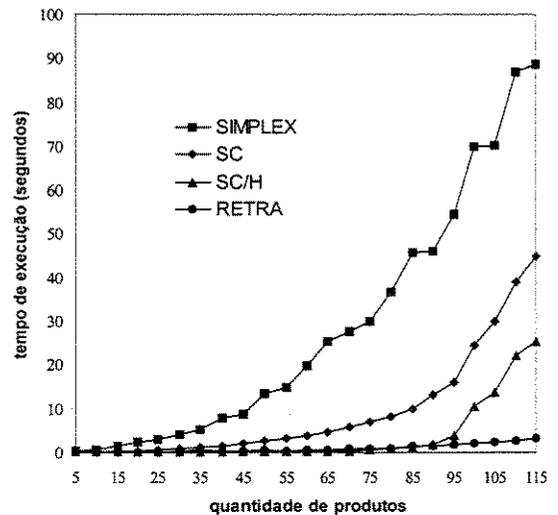
O gráfico da Figura 3.6 compara os desempenhos dos algoritmos Simplex (sem especialização), RETRA e SC, para problemas de planejamento de produção considerando de 5 até 115 variedades de produtos, processados em linhas seriais com 5 estágios, para horizontes



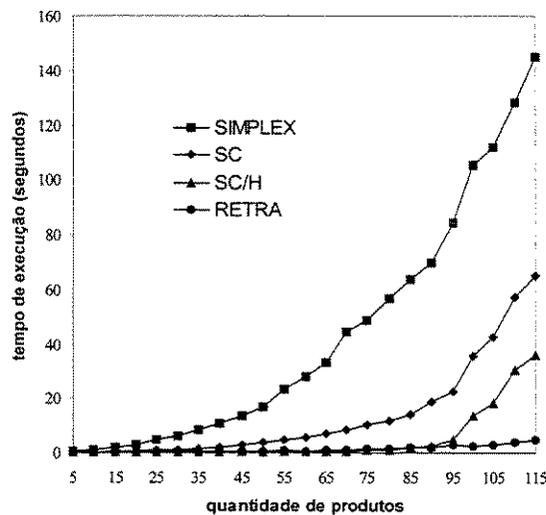
discretizados em 8 (Figura 3.6(a)), 10 (Figura 3.6(b)) e 12 (Figura 3.6(c)) períodos de tempo\*. Nestes gráficos, a curva identificada como SC/H diz respeito aos resultados obtidos para o SC utilizando inicialização através de heurística, enquanto a curva que se refere aos resultados do algoritmo sem utilização de heurística é identificada pela legenda SC.



(a)



(b)



(c)

Figura 3.6 Comparação dos desempenhos dos algoritmos

\* Os dados de entrada utilizados fazem parte de projeto de planejamento da produção coordenado pela Fundação Centro Tecnológico para Informática (Campinas, SP), que envolveu empresa do parque industrial brasileiro. Os testes foram realizados em PC Pentium 300 MHz, com 32 MB de memória RAM.



Segundo Fernandes (1999), o crescimento “quase-linear” observado no RETRA se deve ao fato de que a adição de um novo produto  $p$  ao problema resulta na adição de uma matriz  $A_p$  e em pequena modificação da função objetivo para a consideração das restrições adicionais. O tempo computacional adicional é praticamente o tempo para resolução do novo grafo associado à matriz  $A_p$  introduzida no sistema. No caso de sistemas de manufatura, os grafos relacionados com cada produto têm dimensões praticamente iguais, o que também justifica o eficiente comportamento do RETRA.

Com relação ao SC, no caso em que não é utilizada heurística de inicialização observa-se um padrão de comportamento exponencial, porém suave. No caso em que a heurística de inicialização é utilizada, nota-se piora brusca de desempenho a partir do problema com 95 itens, devido à queda de eficiência da heurística a partir desta quantidade de produtos. Em ambos os casos, o aumento da quantidade de produtos conduz a uma matriz de restrições adicionais de dimensão significativamente grande e, conseqüentemente, a uma matriz de trabalho mais difícil de tratar, o que explica a curva de crescimento acentuada.

O comportamento exponencial do algoritmo Simplex sem especialização é previsto pela teoria de programação linear (Bazaraa, 1990).

Deve-se observar que embora o RETRA seja, até o momento, um dos métodos de resolução mais eficientes para o problema estudado, nem sempre ele fornece soluções precisas, chegando mesmo a não convergir em determinadas situações (estes aspectos foram discutidos na seção 3.4). O SC, além de ser um método relativamente rápido, se comparado a algoritmos de uso geral baseados em Simplex, é também robusto com relação às soluções que fornece e não apresenta problemas de convergência, o que o torna um solver confiável.



## Capítulo 4

### Interface para Modelagem

O enfoque de fluxo em redes apresentado neste trabalho permite resolver o problema de planejamento da produção pela representação do fluxo de produção. Assim sendo, a descrição de um problema real, em linguagem de computador, pode ser feita por ícones que identifiquem a rota de cada produto. A partir da determinação das rotas (ou seqüências de produção), a representação do sistema físico é traduzida para um algoritmo de otimização (SC ou RETRA, no contexto deste trabalho) que resolve o problema. Este capítulo apresenta uma interface computacional que permite ao usuário modelar sistemas de manufatura de forma intuitiva, através de uma linguagem visual, sem que seja necessário ao usuário conhecer a formulação matemática que existe “por trás” da representação simbólica.

#### 4.1 Características Gerais da Interface

A interface gráfica possui recursos para

- representação de máquinas ou conjuntos de máquinas;
- representação de seqüências de produção, através de conexões entre máquinas;
- tratamento de dados relativos a capacidade de produção, discretização do horizonte de planejamento, disponibilidade e tipo de matéria-prima requerida para a produção, demanda requerida para cada intervalo de discretização etc.



A interface inicial foi concebida em linguagem Visual C++, mas posteriormente reprojeta em linguagem Visual Basic (VB) para Windows. Isto foi feito porque pretende-se utilizar a interface em projetos futuros, como ferramenta de integração entre diversas metodologias de gestão da produção já implementadas em VB. A adoção desta linguagem deve facilitar a integração entre pessoas e sistemas envolvidos nas futuras implementações.

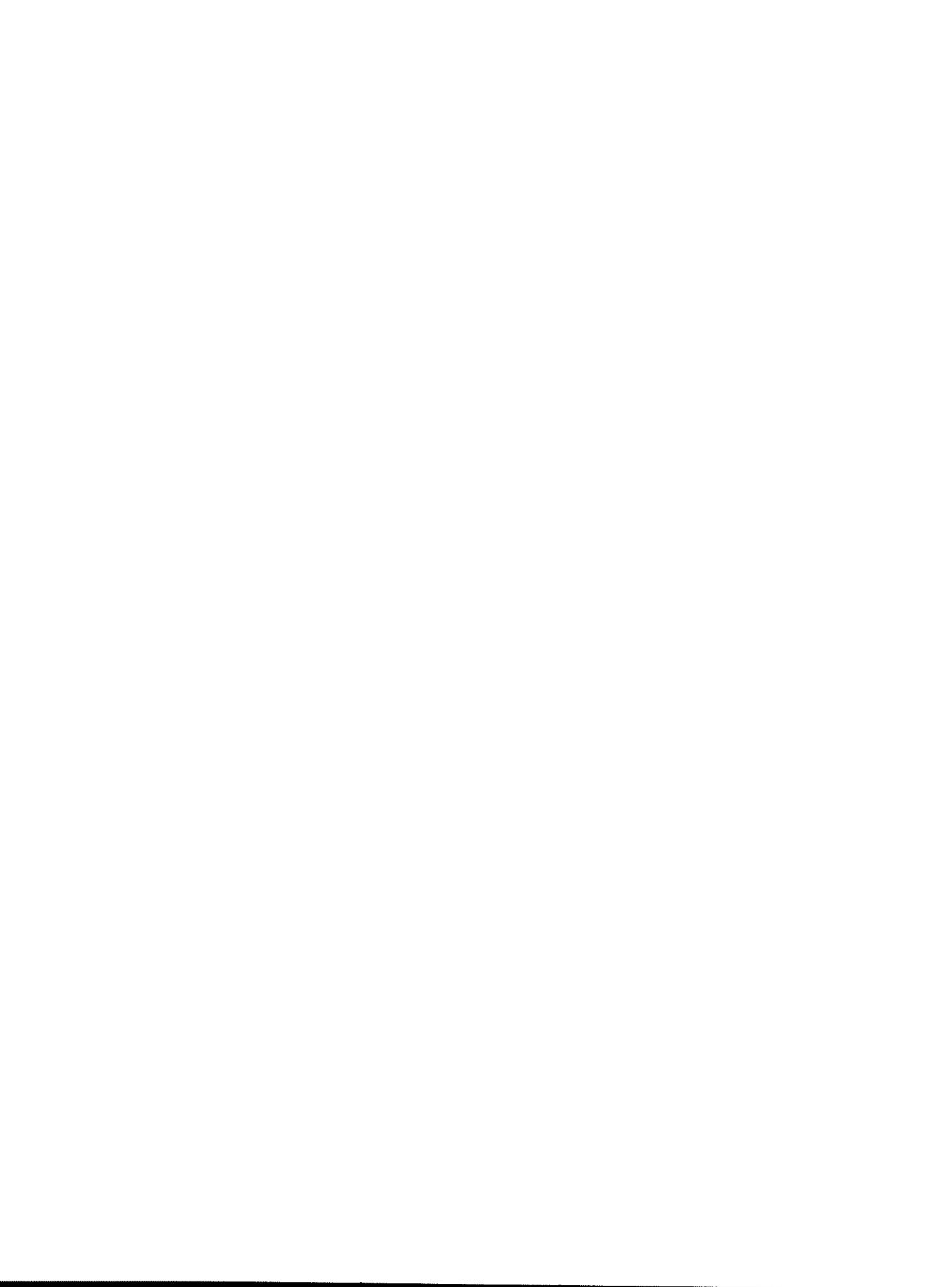
A interface dispõe de ferramentas para composição gráfica do modelo de sistema de manufatura (inserção de objetos gráficos representando máquinas, inserção de conexões entre as “máquinas” definindo as seqüências de produção) e recursos para armazenagem, recuperação e edição de dados externos armazenados em banco de dados. As seções seguintes detalham os desenvolvimentos realizados.

#### **4.2 Definição do Layout de Fábrica**

A Figura 4.1 mostra a janela para definição de layout de fábrica. Existem ferramentas para inserção, identificação e deleção de “máquinas”, permitindo que o usuário represente graficamente o sistema físico correspondente à linha de produção a ser analisada.

Após definir-se a localização física das máquinas no modelo, estas devem ser conectadas, determinando-se as possíveis rotas que os produtos poderão percorrer. É importante, no momento da inserção das conexões, especificar os sentidos das mesmas, os quais restringirão o movimento dos produtos. Por exemplo, no layout da Figura 4.1, um dado produto poderia percorrer o trajeto constituído pelas máquinas M8, M9, M10 e M11, nesta ordem, mas não o contrário, devido à imposição de sentido único entre as duas últimas máquinas.

Em muitos sistemas de produção existe a necessidade de permitir processamentos paralelos, ou seja, possibilitar que lotes de produtos, em determinados estágios da produção, sejam processados em mais de uma máquina, desde que tais máquinas realizem o mesmo tipo de transformação. No exemplo da Figura 4.1, as máquinas M1 e M2 são, na realidade, conjuntos de 3 e 2 máquinas paralelas, respectivamente, conforme indicado na **janela de dados do modo de exibição**.



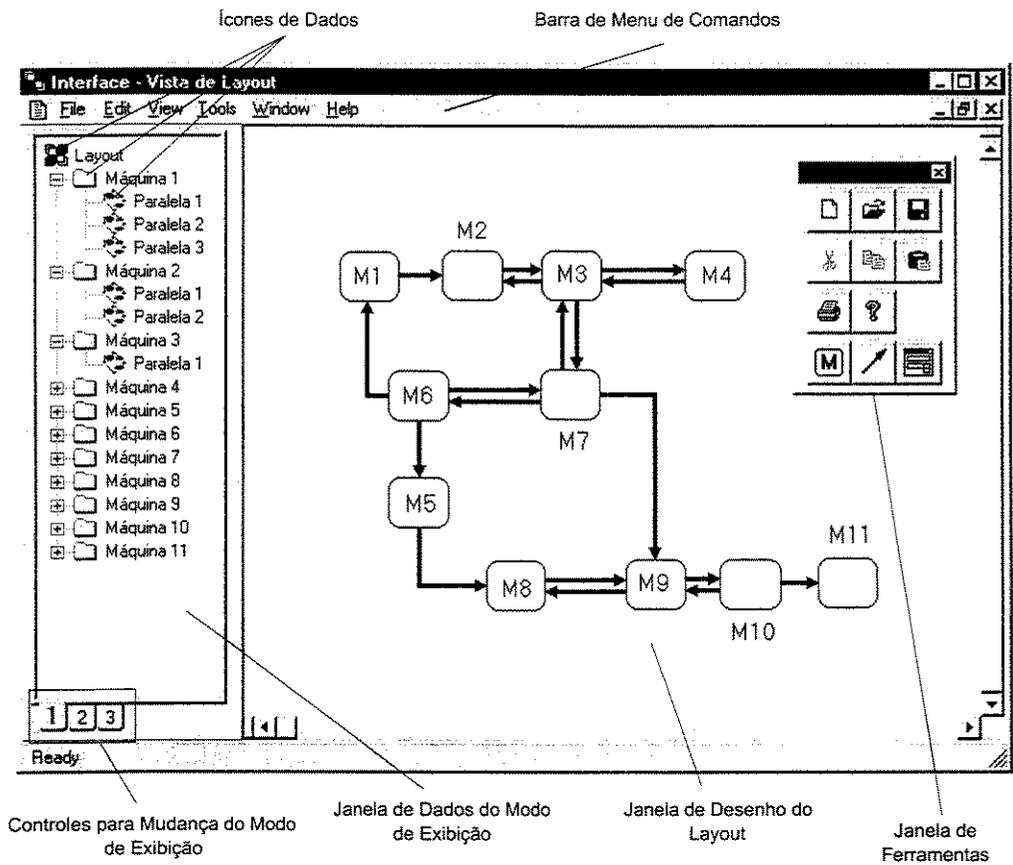


Figura 4.1 Tela para definição do layout de fábrica

### 4.3 Definição de Seqüências de Produção

A partir do layout de fábrica deve-se determinar as seqüências de produção. Partindo-se da idéia de que o sistema de manufatura a ser modelado é flexível, os recursos de produção poderão ser compartilhados por diversas variedades de produtos, cada qual podendo obedecer a uma **seqüência** particular de transformações ou montagens.

As Figuras 4.2 e 4.3 mostram exemplos de definição de seqüências de produção. Enquanto o **Produto 1** deve ser processado, em ordem, pelas máquinas M6, M7, M3 e M4 (Figura 4.2), o **Produto 2** segue a seqüência determinada pelas máquinas M1, M2, M3, M7, M9, M10 e M11 (Figura 4.3). Neste caso, as máquinas M3 e M7 seriam compartilhadas por componentes dos dois produtos. A exibição das diferentes seqüências é alternada através da seleção dos **ícones de dados** correspondentes aos produtos, na **janela de dados**.



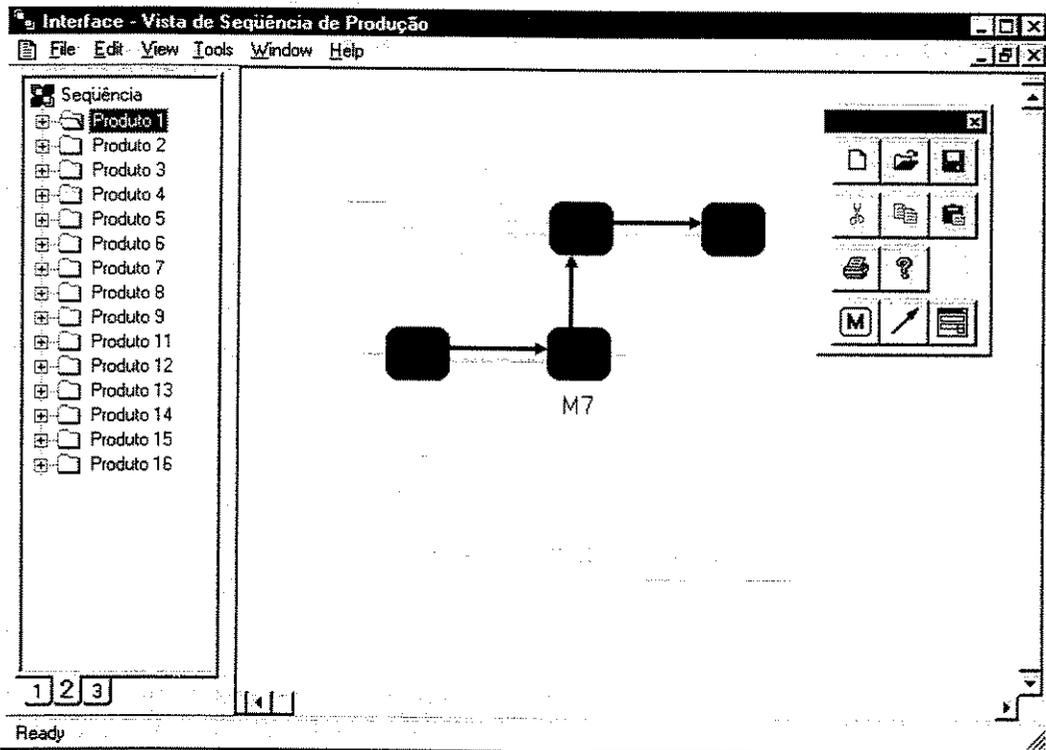


Figura 4.2 Definição de seqüência para o produto 1, a partir do layout da fábrica

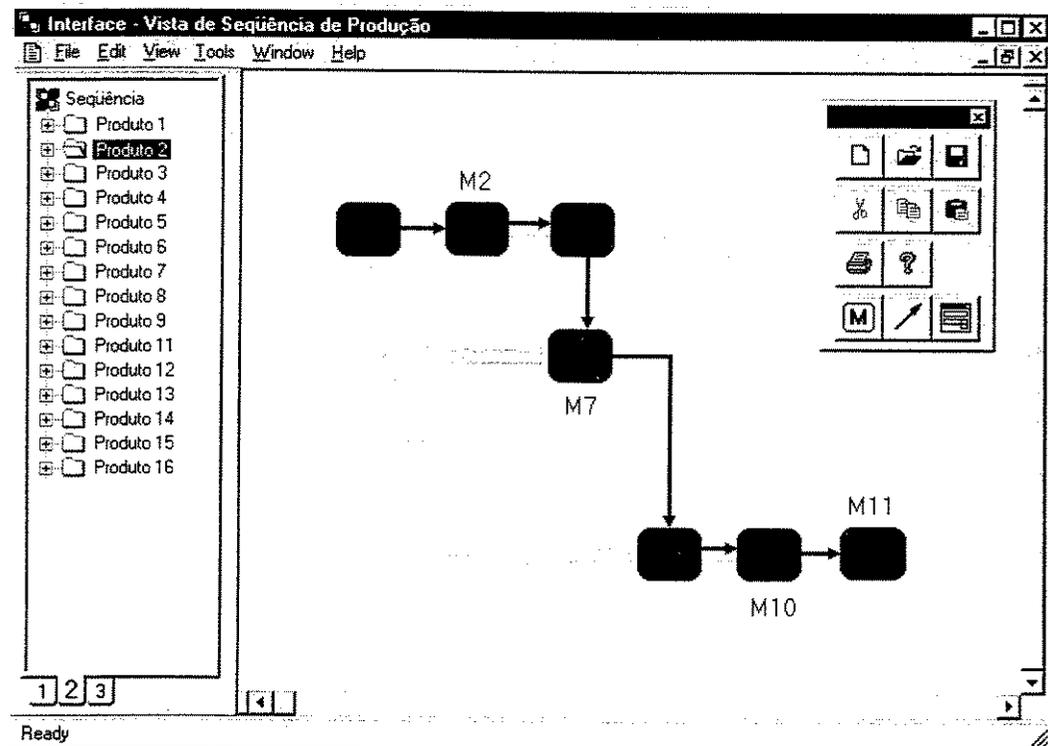


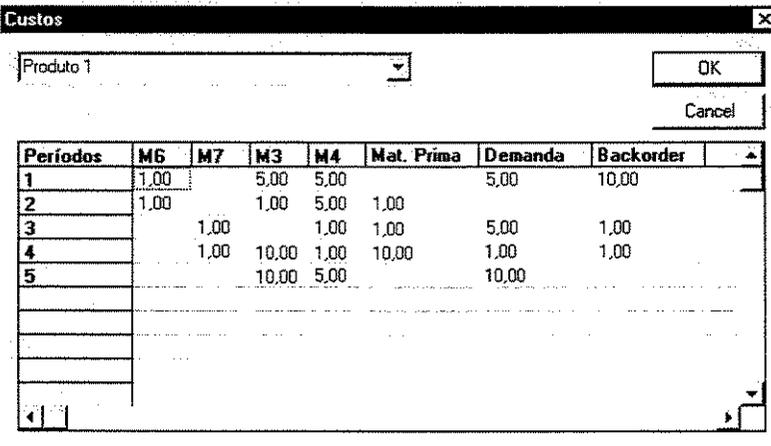
Figura 4.3 Definição de seqüência para o produto 2



#### 4.4 Tratamento de Dados

Uma vez definidas as seqüências de produção, deve-se definir os parâmetros que irão controlar o modelo. Tais parâmetros são, basicamente, a quantidade de períodos do horizonte de planejamento (discretização), os custos associados aos recursos de produção, as capacidades associadas aos recursos de produção, a quantidade disponível de matéria-prima para cada produto e a demanda requerida para cada produto.

A Figura 4.4 mostra um exemplo de tela para definição de custos. Trata-se de uma **tabela dinâmica** em que as colunas relativas às máquinas são geradas segundo a seqüência definida para o produto. No exemplo, o usuário determinou a discretização do horizonte de planejamento em cinco períodos, gerando as linhas da tabela. O custo de produção do Produto 1 na máquina M6 é \$1,00 tanto para o período 1 como para o período 2. No período 5, a máquina M4 processa o mesmo produto ao custo de \$5,00. Além dos custos de produção, nesta tabela também definem-se custos de aquisição de matéria-prima para cada período, custos de atendimento da demanda de cada período e custos de atendimento da demanda em atraso entre períodos. Também existe uma tabela similar para determinação dos custos de armazenagem entre períodos, com uma coluna para cada máquina.



Períodos	M6	M7	M3	M4	Mat. Prima	Demanda	Backorder
1	1,00		5,00	5,00		5,00	10,00
2	1,00		1,00	5,00	1,00		
3		1,00		1,00	1,00	5,00	1,00
4		1,00	10,00	1,00	10,00	1,00	1,00
5			10,00	5,00		10,00	

Figura 4.4 Tela para definição de custos

A tela da Figura 4.5 é análoga à tela para definição de custos, porém aplica-se às capacidades. No exemplo são definidas as capacidades de produção das máquinas presentes na



seqüência relativa ao Produto 2. A máquina M1 tem capacidade para processar 10.000 unidades do Produto 2 no primeiro período, contudo pode processar até 15.000 unidades do mesmo item no período 5. As colunas correspondentes a matéria-prima, demanda e backorder estão ocultas, mas também fazem parte da tabela.

Períodos	M1	M2	M3	M7	M9	M1
1	10000	20000			10000	
2		15000	20000	15000		
3			15000	20000		
4					20000	
5	15000		10000		10000	

Figura 4.5 Tela para definição de capacidades

Para definição da capacidade mútua das máquinas utiliza-se a janela da Figura 4.6. Nesta janela aparecem, nas colunas, todas as máquinas inseridas pelo usuário no layout da fábrica. Dessa forma, pode-se definir a capacidade percentual disponível de cada máquina, em cada período. No exemplo, todas as máquinas em todos os períodos estão trabalhando com capacidade total, com exceção da máquina 5, que tem 75% de disponibilidade no quarto período.

Períodos	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	0.75	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1

Figura 4.6 Tela para definição de capacidades mútuas



O último conjunto de dados a ser especificado diz respeito às disponibilidades de matéria-prima e requerimentos de demanda para cada produto, no horizonte de planejamento. Estes parâmetros são definidos na janela mostrada na Figura 4.7. No exemplo, a demanda requerida para o Produto 2, a ser distribuída para os cinco períodos de discretização, é igual à quantidade de matéria-prima disponível, ou seja, 7.000 unidades.

The image shows a dialog box titled "Matéria Prima e Demanda". At the top left, there is a dropdown menu currently showing "Produto 2". To the right of the dropdown are two buttons: "OK" and "Cancel". Below the dropdown, there are two columns of input fields. The first column is labeled "Quantidade de Matéria Prima Disponível" and contains a text box with the value "7000". The second column is labeled "Demanda Requerida" and also contains a text box with the value "7000".

Figura 4.7 Tela para definição de matéria-prima e demanda



## Capítulo 5

### Aplicação de Tecnologia de Grupo

O mercado consumidor globalizado exige produtos diversificados, de alta qualidade, a custos reduzidos. Neste contexto, as indústrias devem buscar vantagens competitivas através da utilização de tecnologias que suportem a manufatura em lotes.

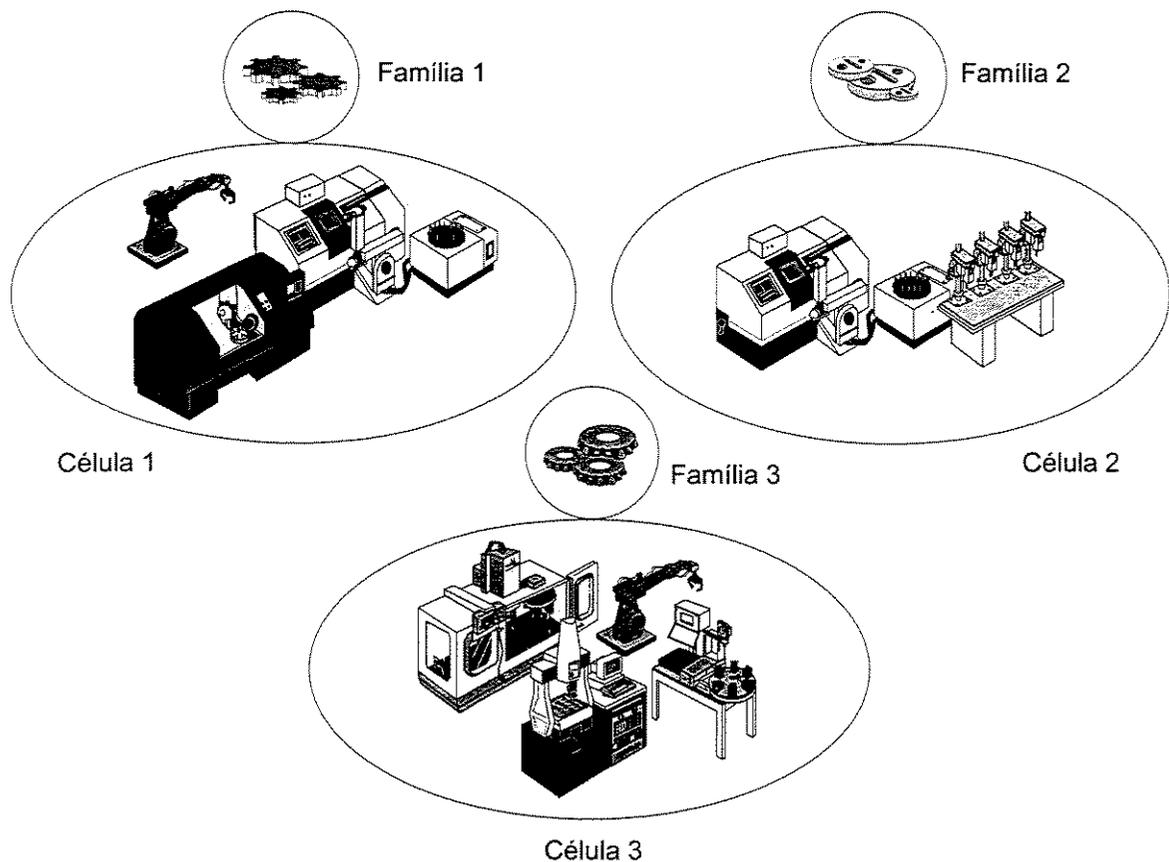
A manufatura em lotes localiza-se entre dois tipos contrastantes de processos produtivos: o *job shop* e o *flow shop*. No *job shop* a diversidade de produtos é grande, porém o volume de produção é baixo. O *flow shop*, ao contrário, destina-se à produção em larga escala, com pouca diversificação. O problema da manufatura em lotes consiste em aliar a flexibilidade do *job shop* à produtividade do *flow shop*. Tal combinação provoca aumento dos tempos de preparação de máquinas (tempos de *setup*) e, conseqüentemente, elevação dos custos de produção.

As dificuldades da manufatura em lotes, que estão relacionadas com o ambiente de multiprodução (diversos produtos compartilhando total ou parcialmente um mesmo conjunto de máquinas), podem ser superadas por meio da Tecnologia de Grupo (TG). Este capítulo apresenta fundamentos da TG e discute como a aplicação dos métodos de TG pode contribuir para a eficiência dos algoritmos de solução do problema de planejamento da produção de sistemas multiproduto.



## 5.1 Conceitos Fundamentais de Tecnologia de Grupo

A Tecnologia de Grupo é uma filosofia de fabricação que procura explorar as similaridades dos produtos (ou das peças que constituem os produtos) em um processo de manufatura. As peças que possuem alguma semelhança – seja geométrica, dimensional ou de operação (peças que requerem operações e recursos semelhantes para fabricação) – podem ser agrupadas em *famílias*. A cada família de peças pode-se associar um grupo de máquinas, denominado *célula*, capaz de executar todas (ou quase todas) as operações necessárias à fabricação das peças que fazem parte daquela família. A Figura 5.1 mostra famílias de peças e grupos de máquinas correspondentes, formando diferentes *células de manufatura*.



**Figura 5.1** Células de manufatura

A organização da fábrica em células de manufatura traz benefícios como

- redução dos tempos de preparação das máquinas (tempos de *setup*);
- melhoria e simplificação das atividades de planejamento e controle da produção;



- redução de ferramental;
- melhoria do fluxo de materiais;
- redução de inventário em processo.

Enfim, o uso da TG proporciona redução de custos e aumento de produtividade (minimização dos tempos não-produtivos e conseqüente redução do tempo total de produção (*lead time*)), o que se traduz em vantagem competitiva e maiores lucros.

## 5.2 Técnicas para Obtenção de Células de Manufatura

As técnicas de agrupamento da TG partem do princípio de decompor o sistema de manufatura em subsistemas (células de manufatura) mais fáceis de gerenciar que o sistema como um todo.

A representação do sistema de manufatura – utilizada pelos algoritmos de agrupamento – é feita através da *matriz binária de operações*. A Figura 5.2(a)\* mostra um exemplo deste tipo de matriz, para um sistema com 10 máquinas e 20 peças. As colunas da matriz representam planos de processo para as peças, indicando quais máquinas (relacionadas às linhas) são requeridas para realizar as operações necessárias a cada peça. A Figura 5.2(b) mostra a matriz rearranjada, na qual identificam-se claramente quatro possíveis células de manufatura.

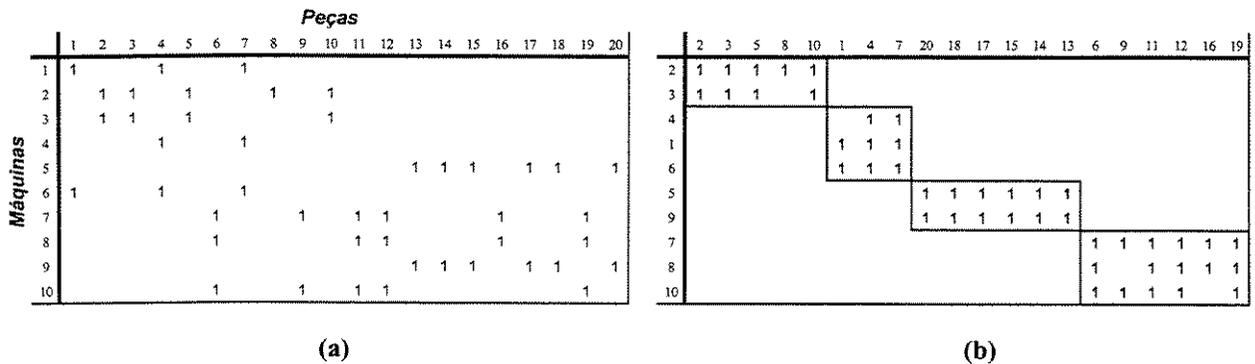


Figura 5.2 Matriz binária de operações (a) original e (b) rearranjada

\* As matrizes de exemplo deste capítulo foram adaptadas de Rodrigues (1997). Os elementos nulos das matrizes foram substituídos por espaços vazios, a fim de melhorar a visualização.



O exemplo da Figura 5.2 caracteriza o caso ideal, em que é possível definir células de manufatura completamente independentes entre si. Em problemas práticos, no entanto, é provável que, após o agrupamento, ocorram *movimentos inter-células*, com algumas peças requerendo processamento em máquinas (as chamadas *máquinas-gargalo*) localizadas fora das células para as quais tais peças foram alocadas. A eficiência do algoritmo de agrupamento, portanto, está relacionada à sua capacidade de minimização do movimento inter-células. A Figura 5.3 mostra (a) uma matriz binária representando um plano de processo de um sistema de manufatura com 22 peças e 11 máquinas e (b) a mesma matriz depois de submetida a um algoritmo de agrupamento (detalhes do algoritmo encontram-se em Rodrigues, 1997) que procura determinar uma configuração na qual a transferência de peças de uma célula para outra seja mínima.

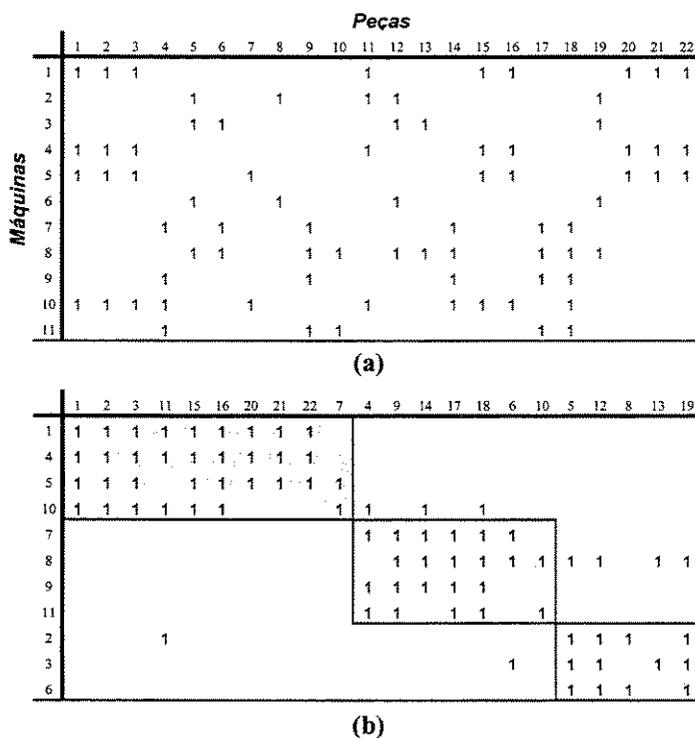


Figura 5.3 Agrupamento com movimentos inter-células

Existem alternativas para eliminação de movimentos inter-células, caso o custo de movimentação seja relativamente elevado. São elas:

- duplicação de máquina;
- subcontratação de peça;
- remoção de máquina;
- remoção de peça.



A adoção de uma destas estratégias (ou combinação delas) depende obviamente de fatores econômicos, que variam de um caso para outro. Tomando por base o exemplo da Figura 5.3(b), pode-se optar por duplicar as máquinas 2, 3, 8 e 10, obtendo-se a configuração representada pela Figura 5.4(a). Se, contudo, o custo de aquisição das novas máquinas for muito alto, pode-se pensar em “terceirizar” a fabricação das peças 6 e 11, ou mesmo removê-las do processo, se isto parecer viável economicamente. Desta forma, apenas as máquinas-gargalo 8 e 10, que são as responsáveis pelo maior número de movimentos inter-células, precisariam ser duplicadas (ver Figura 5.4(b)).

		Peças																						
		1	2	3	11	15	16	20	21	22	7	4	9	14	17	18	6	10	5	12	8	13	19	
Máquinas	1	1	1	1	1	1	1	1	1	1														
	4	1	1	1	1	1	1	1	1	1														
	5	1	1	1		1	1	1	1	1	1													
	10	1	1	1	1	1	1	1	1	1	1													
	2																							
	7											1	1	1	1	1	1	1						
	8												1	1	1	1	1	1	1					
	9											1	1	1	1	1								
	11											1	1		1	1	1		1					
	10											1	1		1	1	1							
	3																							
	2																							
	3																							
	6																							
8																								

		Peças																					
		1	2	3	15	16	20	21	22	7	4	9	14	17	18	10	5	12	8	13	19		
Máquinas	1	1	1	1	1	1	1	1	1	1													
	4	1	1	1	1	1	1	1	1	1													
	5	1	1	1	1	1	1	1	1	1	1												
	10	1	1	1	1	1	1	1	1	1	1												
	7											1	1	1	1	1	1						
	8												1	1	1	1	1	1					
	9											1	1	1	1	1							
	11											1	1		1	1	1						
	10											1	1		1	1	1						
	2																						
	3																						
	6																						
	8																						

Figura 5.4 Alternativas para eliminação de movimentos inter-células

### 5.3 Influência sobre o Algoritmo de Otimização

Nos capítulos anteriores, mostrou-se que o problema de planejamento da produção de sistemas multiproduto, multiestágio e multiperíodo possui uma estrutura particular que sugere o particionamento do problema global em problemas menores, que possam ser resolvidos independentemente pelo algoritmo de otimização.

Neste sentido, existe um forte apelo para a utilização da TG como técnica auxiliar à otimização. Através dos algoritmos de agrupamento, pode-se determinar – com base nas seqüências de produção – famílias de produtos a serem processadas por grupos distintos de máquinas. A partir das informações de grupo, algoritmos como SC e RETRA (ver Capítulo 3),



que atualmente resolvem o problema de planejamento da produção globalmente, poderiam passar a tratar subproblemas de menores dimensões, o que pode significar uma economia substancial nos tempos de execução.

O fator que mais fortemente motiva o uso da TG como técnica auxiliar aos algoritmos de otimização é a *dinâmica da demanda* ao longo do tempo. A variação da demanda determina a entrada e saída de itens do processo produtivo, levando a um constante estudo da combinação de configurações de grupos de máquinas e famílias de peças.

	<i>Semana 1</i>	<i>Semana 2</i>	<i>Semana 3</i>	<i>Semana 4</i>
PROD00015	48.000	34.000	34.000	20.400
PROD00016	38.200	14.600	33.000	5.400
PROD00017		500	1.000	1.700
PROD00020	1.800	4.600	1.500	2.000
PROD00021	1.000	4.100	1.100	
PROD00022			1.000	
PROD00023	1.700	900	2.000	500
PROD00024		1.800		
PROD00025				500
PROD00028		6.000	6.000	
PROD00064	66.200	29.100	60.900	36.500
PROD00065	34.000	18.300	34.000	20.400
PROD00067	32.200	10.800	26.900	16.100
PROD00070		3.000	5.400	
PROD00074				
PROD00075		6.000		10.800
PROD00076				12.800
PROD00077				1.000
PROD00080		900		
PROD00083		500		
PROD00087	500		1.100	500
PROD00088				
PROD00090	1.000			700
PROD00091		3.600	1.500	2.000
PROD00092				500
PROD00093	1.200	900	900	
PROD00094				1.000
PROD00095	1.000	4.100	1.100	
PROD00096	1.800	1.000		
PROD00100		100	1.300	

X Há demanda (em quantidade X) no período       Não há demanda no período

**Tabela 5.1** Alteração do “mix” de produção devido à variação temporal da demanda

A Tabela 5.1, obtida a partir dos mesmos dados de entrada utilizados nos testes dos algoritmos de otimização do Capítulo 3, mostra as alterações do “mix” e da quantidade de produção causadas pela dinâmica da demanda, ao longo de um período de 4 semanas. Nota-se



que existem produtos, como 087 e 090, que são requeridos na primeira semana do mês, mas não na semana seguinte, devendo, portanto, ser momentaneamente retirados do processo produtivo. Por outro lado, diversos itens que não são produzidos na primeira semana (exemplos: 017, 024, 070, 100, entre outros) passam a ser na segunda, o que caracteriza inserção no sistema de fabricação.

Supondo que a matriz da Figura 5.5(a) representa o agrupamento relativo à primeira semana, verifica-se ocorrência de movimentos inter-células relacionados aos produtos 087 e 090. A ausência (determinada pela demanda) destes itens na segunda semana (Figura 5.5(b)) elimina a transferência entre células (deve-se notar que os produtos, em destaque, que entraram na segunda semana puderam ser inseridos completamente nas células já existentes, sem ocasionar movimento intercelular), o que, teoricamente, permite que cada um dos três módulos do problema associado à matriz da Figura 5.5(b) seja resolvido independentemente pelo algoritmo de otimização (haveriam três problemas de grafos a serem resolvidos independentemente).

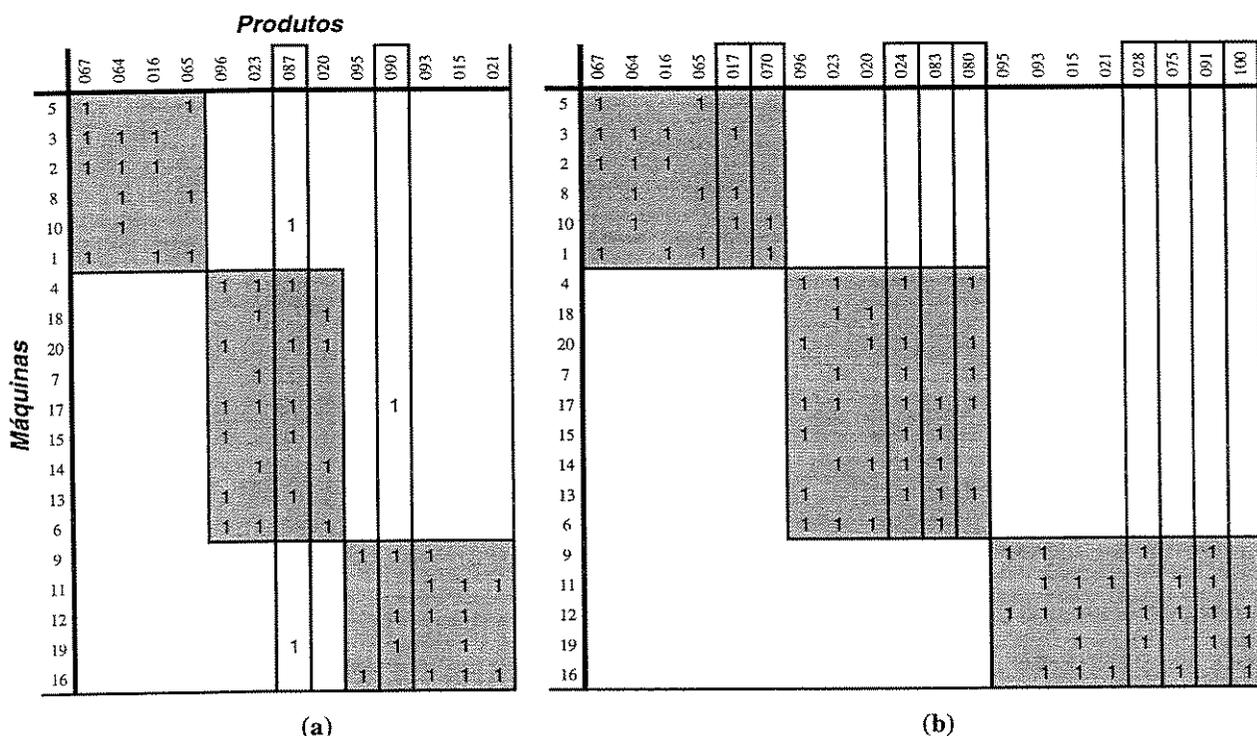


Figura 5.5 Agrupamentos associados às semanas 1 e 2



Em problemas reais, o grau de desacoplamento, ou seja, o quão separável é o problema, vai depender não somente do comportamento da demanda ao longo do tempo mas também do método de agrupamento aplicado. Como não existe método ideal para todas as situações (toda técnica tem limitações), é necessário aplicar diferentes métodos a um mesmo problema, de forma a garantir, através de diversas alternativas de solução geradas, maior flexibilidade ao processo de tomada de decisão.



## Capítulo 6

### Conclusões e Sugestões para Próximos Trabalhos

Neste trabalho foi apresentado um algoritmo de otimização dedicado à resolução do problema de planejamento da produção de sistemas de manufatura multiestágio, multiperíodo e multiproduto. Este algoritmo aproveita a estrutura particular do problema, o qual incorpora restrições adicionais que consideram as capacidades de produção mútua dos recursos de fabricação. Os resultados fornecidos pelo algoritmo, que utiliza uma formulação de fluxo em redes com restrições adicionais, são os níveis ótimos de estoque, de atendimento à demanda, de carregamento das máquinas, de pedidos em atraso e a necessidade temporal de matéria-prima.

Comparações de desempenho envolvendo o método Simplex “puro” (sem especialização para fluxo em redes) e o algoritmo RETRA mostraram que o algoritmo SC implementado é eficiente e confiável. Os algoritmos foram aplicados em sistemas de manufatura provenientes de fábricas do parque industrial brasileiro.

Para prover uma possível integração das práticas industriais do tipo MRP com os métodos de otimização, foi desenvolvido um ambiente computacional dotado de interface gráfica amigável, que permite ao usuário não especialista modelar sistemas de produção de maneira relativamente rápida e intuitiva.

Foi sugerida a aplicação da Tecnologia de Grupo ao problema de planejamento da produção discutido, como forma de tornar mais eficiente o algoritmo de otimização.



Como sugestões para trabalhos futuros, pode-se mencionar:

- desenvolvimento de algoritmos ainda mais rápidos, que contornem as dificuldades da técnica de decomposição primal, como operações robustas envolvendo matrizes esparsas;
- busca de um ambiente integrado utilizando softwares de MRP e a interface para modelagem desenvolvida;
- efetiva utilização de métodos de Tecnologia de Grupo em conjunto com o algoritmo de otimização implementado.



## Referências Bibliográficas

- Agostinho, O. L. *Sistemas de Manufatura*. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1997
- Ahuja, R. K. , Magnanti, T. L. , Orlin, J. B. *Network Flows: Theory, Algorithms, and Applications*. New Jersey: Prentice-Hall, 1993, 846 p.
- Amorim, Joni de Almeida. *Enfoque Iterativo para Agrupamento de Máquinas e Layout*. Instituto de Automação, Fundação Centro Tecnológico para Informática, 1999
- Bazaraa, M. S. , Jarvis, J. J. , Sherali, H. D. *Linear Programming and Network Flows*. New York: John Wiley & Sons, 1990, 684 p.
- Bentley, J. L. , Kernighan, B. W. A System for Algorithm Animation Tutorial and User Manual. Computing Science Report, AT&T Bell Laboratories, August, 1991
- Bresciani Filho, E. , D'Ottaviano, I. M. L. *Conceitos Básicos de Sistêmica*. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1998
- Carvalho, M. F. H. , Fernandes, C. A. O. , Silva Filho, O. S. Manufacturing Planning by Network Flow Model with Side Constraints. *IFAC Workshop on CIM in Process and Manufacturing Industries*, Helsinki, 1992
- Carvalho, M. F. H. , Fernandes, C. A. O. , Silva Filho, O. S. O Planejamento da Manufatura – Práticas Industriais e Métodos de Otimização. *Gestão e Produção* (revista do departamento de engenharia de produção da Universidade Federal de São Carlos), v.5, n.1, p.34-59, 1998



- Carvalho, M. F. H. , Fernandes, C. A. O. Planejamento da Produção de Manufatura por Fluxos em Redes. *Gestão e Produção* (revista do departamento de engenharia de produção da Universidade Federal de São Carlos), v.6, n.1, p.30-50, 1999
- Carvalho, M. F. H. , Hadad, R. B. B. , Fernandes, C. A. O. Optimisation Methods as Complementary Tool for Industrial Practices. *CAD/CAM Robotics and Factories of The Future*, London, 1996
- Chang, S. F. , McCormick, S. T. A Hierarchical Algorithm for Making Sparse Matrices Sparser. *Mathematical Programming*, v.56, n.1, p.1-30, 1992
- Dembo, R. S. , Mulvey, J. M. , Zenios, S. A. Large-Scale Nonlinear Network Models and Their Applications. *Operations Research*, v.37, n.3, p.353-372, May-June 1989
- Eisenbeis, C. , Jalby, W. , Windheiser, D. , Bodin, F. A Strategy for Array Management in Local Memory. *Mathematical Programming*, v.63, n.3, p.331-370, 1994
- Fernandes, Carlos Alberto de Oliveira. *Planejamento da Produção da Manufatura por Fluxo em Redes com Restrições Adicionais*. São Paulo: Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 1999. 88 p. Tese (Doutorado)
- Ferris, M. C. , Horn, J. D. Partitioning Mathematical Programs for Parallel Solution. *Mathematical Programming*, v.80, n.1, p.35-61, 1998
- Fujisawa, K. , Kojima, M. , Nakata, K. Exploiting Sparsity in Primal-Dual Interior-Point Methods for Semidefinite Programming. *Mathematical Programming*, v.79, n.1-3, p.235-253, 1997
- Gaimon, C. Optimal Inventory, Backlogging and Machine Loading in a Serial, Multi-Stage, Multi-Period Production Environment. *International Journal of Production Research*, v.24, n.3, p.647-662, 1986
- Ghosh, S. , Gaimon, C. Routing Flexibility and Production Scheduling in a Flexible Manufacturing System. *European Journal of Operational Research*, n.60, p.344-364, 1992



- Jensen, P. A. , Barnes, J. W. *Network Flow Programming*. New York: John Wiley & Sons, 1980, 408 p.
- Kennington, J. L. , Helgason, R. V. *Algorithms for Network Programming*. New York: John Wiley & Sons, 1980, 291 p.
- Kennington, J. L. , Whisman, A. *Netside User's Guide*. Department of Computer Science and Engineering, Southern Methodist University, Dallas, 1990
- Lin, S. , Lin, C. A Computationally Efficient Method for Nonlinear Multicommodity Network Flow Problems. *Networks*, v.29, p.225-244, 1997
- Lucchesi, C. L. , Kowaltowski, T. *Estruturas de Dados e Técnicas de Programação*. Instituto de Computação, Universidade Estadual de Campinas, Março 1997
- Marín, A. Restricted Simplicial Decomposition with Side Constraints. *Networks*, v.26, p.199-215, 1995
- McBride, R. D. Solving Embedded Generalized Network Problems. *European Journal of Operational Research*, n.21, p.82-92, 1985
- Özdamar, L. , Bozyel, M. A. , Birbil, S. I. A Hierarchical Decision Support System for Production Planning (with Case Study). *European Journal of Operational Research*, n.104, p.403-422, 1998
- Park, T. Y. , Han, K. H. , Choi, B. K. An Object-Oriented Modelling Framework for Automated Manufacturing System. *International Journal of Computer Integrated Manufacturing*, v.10, n.5, p.324-334, 1997
- Parsaei, H. R. , Kolli, S. , Hanley, T. R. *Manufacturing Decision Support Systems*. London: Chapman & Hall, 1997, 302 p.
- Rodrigues, Elder de Oliveira. *Problema de Especificação de Células de Manufatura na Presença de Restrições de Projeto – Uma Abordagem Baseada em Tabu Search*. Espírito Santo: Programa de Pós Graduação em Engenharia Elétrica, Universidade Federal do Espírito Santo, 1997. 100 p. Dissertação (Mestrado)



- Sandbothe, R. A. Backordering in a Multiperiod Transshipment Network. *Networks*, v.23, p.623-630, 1993
- Singh, N. , Rajamani, D. *Cellular Manufacturing Systems – Design, Planning and Control*. London: Chapman & Hall, 1996, 277 p.
- Slack, N. , Chambers, S. , Harland, C. , Harrison, A. , Johnston, R. *Administração da Produção*. São Paulo: Atlas, 1996, Cap. 14: MRP, p.442-472
- Soroush, H. , Mirchandani, P. B. The Stochastic Multicommodity Flow Problem. *Networks*, v.20, p.121-155, 1990
- Watkins, D. S. *Fundamentals of Matrix Computations*. New York: John Wiley & Sons, 1991, 449 p.
- Zahorik, A. , Thomas, L. J. , Trigeiro, W. W. Network Programming Models for Production Scheduling in Multi-Stage, Multi-Item Capacited Systems. *Management Science*, v.30, n.3, p.308-316, March 1984

UNICAMP  
BIBLIOTECA CENTRAL  
SEÇÃO CIRCULANTE



