

Este exemplar corresponde a redação final da tese  
defendida por *Carlos Alberto de Oliveira*  
*Fernandes* e aprovada pela Comissão  
Julgada em *14* / *05* / *1999*  
*Marcus Fabius*  
Orientador

**UNIVERSIDADE ESTADUAL DE CAMPINAS**  
**FACULDADE DE ENGENHARIA ELÉTRICA**  
**E DE COMPUTAÇÃO**

# **Planejamento da Produção da Manufatura por Fluxo em Redes com Restrições Adicionais**

Autor: **Carlos Alberto de Oliveira Fernandes**  
Orientador: **Marcus Fabius Henriques de Carvalho**  
Co-orientador: **Paulo Augusto Valente Ferreira**

**Banca examinadora:**

Marcus Fabius Henriques de Carvalho - FEEC/UNICAMP - Credenciado  
Reinaldo Morabito - UFSCAR  
Arthur José Vieira Porto - USP/São Carlos  
Akebo Yamakami - FEEC/UNICAMP  
Aurélio Ribeiro Leite de Oliveira - FEEC/UNICAMP

Dissertação de doutoramento apresentada à comissão de  
Pós Graduação da Faculdade de Engenharia Elétrica e de  
Computação, como requisito para a obtenção do título de  
DOUTOR em Engenharia Elétrica.

Campinas, maio de 1999  
S.P. - Brasil

9916673

UNICAMP

UNIDADE	BC
N.º CHAMADA:	01/38685
PREÇO	229,99
DATA	04/09/99
N.º CPD	

CM-00134263-9

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

F391p Fernandes, Carlos Alberto de Oliveira  
Planejamento da produção da manufatura por fluxo em  
redes com restrições adicionais. / Carlos Alberto de  
Oliveira Fernandes.--Campinas, SP: [s.n.], 1999.

Orientadores: Marcius Fabius Henriques de Carvalho,  
Paulo Augusto Valente Ferreira.

Tese (doutorado) - Universidade Estadual de  
Campinas, Faculdade de Engenharia Elétrica e de  
Computação.

1. Planejamento da produção. 2. Programação linear.  
3. Otimização matemática. I. Carvalho, Marcius Fabius  
Henriques de. II. Ferreira, Paulo Augusto Valente. III.  
Universidade Estadual de Campinas. Faculdade de  
Engenharia Elétrica e de Computação. IV. Título.

## **Resumo**

A alocação temporal dos recursos da produção num ambiente de manufatura multiestágio, multiperíodo, multiproduto, com demanda determinística é modelada como um problema de fluxo em redes com restrições adicionais. Assume-se que decisões tais como agrupamento de máquinas, escala de manutenção e capacidade de carregamento das máquinas e/ou ferramentas tenham sido tomadas num nível hierárquico superior, como por exemplo no nível de planejamento estratégico. O objetivo deste trabalho é propor algoritmos de otimização para o planejamento da produção que explorem a estrutura especial desta formulação. Os resultados esperados 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. Formulações multiobjetivo de problema e métodos eficientes de solução que levem em conta o grande porte e a natureza linear do problema são propostos. As principais contribuições da tese são ilustradas através de aplicações reais em ambientes de manufatura encontrados no parque industrial brasileiro.

## **Abstract**

The production planning of a multiproduct multiperiod multistage manufacturing system with deterministic demand is modeled as a network flow optimization problem with side constraints. This approach allows to explore the special structure of the problem during the solution process. It is assumed that decisions such as maintenance scheduling, machine and tools capacity, had been made by the strategic decision level. The optimization algorithm determines the optimum storage level, the demand supply level, backorders, and the temporal allocation of raw material. A multiobjective formulation associated to an efficient solution technique, taking into consideration the large size of the problem, is proposed. The main contributions of this work are illustrated through an application to an actual manufacturing environment from the brazilian industry.

*Palavras-chave:*

Planejamento da Produção, Sequenciamento, Programação Matemática, Fluxo em Redes, Otimização.

## **Agradecimentos**

A todos que colaboraram, de alguma forma, para a realização deste trabalho.

## Índice

Capítulo 1 - Introdução .....	01
Capítulo 2 - Modelagem de Linhas de Montagem por Fluxo em Redes .....	07
2.1 - Notação .....	08
2.1.1 - Índices .....	08
2.1.2 - Parâmetros de Entrada .....	08
2.1.3 - Variáveis de Decisão .....	09
2.2 - O Planejamento da Produção em uma Linha Serial Monoproduto .....	10
2.3 - O Planejamento da Produção em uma Linha de Montagem Monoproduto .....	16
2.4 - O Planejamento da Produção em uma Linha de Montagem Multiproduto .....	21
2.4.1 - Restrições de Grafo .....	22
2.4.2 - Restrições Adicionais .....	23
2.4.3 - Estrutura da Matriz de Restrições .....	24
2.4.4 - Matriz de Restrições - Formato Geral .....	29
2.5 - Função Objetivo .....	30
2.5.1 - Função Lucro .....	30
2.5.2 - Outras Funções Objetivos .....	31
2.6 - Formulação Geral .....	32

Capítulo 3 - Abordagem Mono-Objetivo .....	35
3.1 - Técnica de Decomposição Primal .....	36
3.1.1 - Algoritmo Netside .....	37
3.1.2 - Atualização dos Custos Reduzidos .....	38
3.1.3 - Atualização da Coluna da Variável $k$ .....	39
3.1.4 - Atualização da Inversa da Base de Trabalho $Q$ .....	40
3.2- Técnica de Penalidade .....	44
3.2.1 - Algoritmo de Penalidade Linear-Quadrática (PLQ) .....	46
3.2.2 - Técnica de Linearização por Decomposição Simplicial .....	46
3.2.2 - Parâmetros de Penalidade .....	53
3.3. - Aplicação .....	54
3.4. - Inicialização .....	61
Capítulo 4 - Abordagem Multiobjetivo .....	63
4.1 - Notação .....	63
4.2 - Formulação do Problema Multiobjetivo .....	64
4.2.1 - O Problema Ponderado .....	67
4.3 - Programação Alvo .....	68
4.4 - Aplicação .....	70
4.4.1 - Outros Resultados .....	77
4.5 - Programação Alvo Interativa .....	77
Capítulo 5 - Conclusões .....	79
Capítulo 6 - Referências Bibliográficas .....	83

## CAPÍTULO 1

### INTRODUÇÃO

Uma das mais ricas fontes de problemas de planejamento e controle da produção é encontrada no âmbito dos sistemas de manufatura. Enquanto que em algumas áreas de produção, como por exemplo em geração de energia e refino de petróleo (Dembo *et al.*, 1989), os problemas de produção são tipicamente *bem-estruturados* permitindo o emprego de técnicas de programação matemática, nos ambientes de manufatura os problemas são em geral *semi-estruturados*, dificultando o emprego de algoritmos de otimização. Por este fato, ênfase tem sido dada aos procedimentos conhecidos como *práticas industriais* (Gershwin *et al.*, 1986 e Rodammer e Write, 1988), que são basicamente sistemas heurísticos de tratamento de informação desenvolvidos para auxiliar gerentes no processo de tomada de decisão. Exemplos típicos destes sistemas são: o *Material Requirement Planning* (MRP) (Fox, 1984), a filosofia japonesa *Just-in-Time* (JIT) (Schonberger, 1984 e Golhar *et al.*, 1991), e o *Optimized Production Timetable* (OPT) (Jacobs, 1984).

No contexto acima, o MRP realiza uma importante função no planejamento dos sistemas de manufatura, estando sua utilidade e sucesso muito bem documentados na literatura. Pode-se dizer, a grosso modo, que o MRP partindo de uma demanda conhecida, em um primeiro estágio, determina as quantidades dos componentes que fazem parte de cada produto final. A seguir, utilizando tempos de processamento pré-determinados, o MRP calcula o prazo no qual o componente ou matéria prima deve ser processado ou encomendado, requerendo decisão humana somente para ordens de compra, ordens de fabricação e reação a variações em datas de entrega, ou outro evento aleatório qualquer. Em suma, o MRP caminha da demanda até a matéria prima estabelecendo o plano temporal de necessidades, que determina a utilização de máquinas, o plano de compras, etc.



Apesar do sucesso comercial do MRP, é possível destacar algumas de suas deficiências (Benton e Shin, 1998). Por exemplo, na determinação do plano temporal de necessidades não são levadas em conta a disponibilidade efetiva de matéria prima, os requisitos reais de capacidade produtiva, os custos de produção, e as incertezas tanto da demanda quanto do processo produtivo. Além disso, não considera as perdas por defeitos de fabricação, como também tem grande dificuldade em tratar sistemas de produção multiproduto multiperíodo capacitados, onde diferentes itens compartilham os mesmos recursos de produção. Resumindo, pode-se dizer que o MRP é simplesmente um viabilizador de cenários de produção definidos a partir dos pedidos de produção. Em outras palavras, para que a política temporal de recursos gerada pelo MRP possa ser aplicável é necessário que os diversos departamentos da empresa e seus fornecedores estejam altamente comprometidos com a mesma e ainda que o sistema seja totalmente confiável, o que nem sempre ocorre.

As limitações do MRP, mencionadas acima, têm motivado a pesquisa e o desenvolvimento de metodologias que não só possibilitam contemplar os pontos abordados pelo MRP, como também permitem estabelecer uma programação de capacidade otimizada do ponto de vista econômico.

A filosofia JIT controlada por *Kanban* poderia melhorar o desempenho do MRP, já que é uma forma de operar o sistema com realimentação discreta de informação. Neste tipo de sistema as informações acerca dos pedidos dos consumidores e suas previsões são processadas pelo último estágio de produção ou pelo estoque de produtos finais. Se a demanda não puder ser atendida diretamente pelo estoque disponível, quando da colocação do pedido, este estágio irá emitir um pedido de fabricação para o estágio imediatamente precedente, e assim sucessivamente até o primeiro estágio. Neste tipo de sistema a direção do fluxo de informação (ou ordens de fabricação) é sempre contrária à direção do fluxo de produção. Estes são sistemas seriais com múltiplas ordens de produção que podem ser estendidas a todos os fornecedores de matéria prima. O JIT apresenta uma desvantagem quanto à velocidade de resposta a flutuações de demanda, pois se caracteriza por um grande tempo de transmissão das mudanças, já que atua com realimentação discreta da informação transmitida estágio a estágio de produção. Devido a

esta característica, o JIT é adequado somente quando a taxa de produção é constante e torna-se desastroso quando ocorrem mudanças bruscas na demanda. Além disto, não visualiza as necessidades futuras e portanto não permite que o sistema produtivo se prepare adequadamente.

O OPT, um software complexo que faz o planejamento da produção, é visto como uma caixa preta, pois sua metodologia não está divulgada na literatura. Entretanto sabe-se que está baseado em um conjunto de nove regras (Jacobs, 1984) cujo objetivo é a programação das máquinas e da mão-de-obra para suprir os gargalos de produção de forma sincronizada, de modo a reduzir inventário e aumentar as taxas de produção. Segundo o OPT, o objetivo maior da empresa é realizar lucro. Partindo então desta premissa são estabelecidos os princípios heurísticos e as medidas de desempenho que determinam o planejamento da produção. Um dos problemas para a aplicação do OPT é a determinação do gargalo, pois muitas vezes a dinâmica do processo produtivo torna o gargalo móvel, dificultando enormemente a aplicação deste enfoque.

As tecnologias apresentadas acima, citadas como práticas industriais, foram formadas por experiências e intuições adquiridas pelos gerentes e operadores ao longo dos anos. Estas práticas vêem os sistemas produtivos do ponto de vista do cliente (demanda) e trabalham no sentido de maximizar a satisfação deste, entregando produtos de qualidade nas datas combinadas. Estas tecnologias se enquadram nas chamadas *técnicas avaliativas* de solução, que partem de um conjunto de decisões e prevêm ou avaliam o desempenho do sistema sob as condições pré-estabelecidas (Suri, 1985).

Para Edghill e Davis (1985), 50 % dos FMS's (*Flexible Manufacturing Systems*) são compostos por máquinas de comando numérico muito versáteis. Versatilidade proporciona graus de liberdade e ao mesmo tempo aumenta a complexidade das funções de planejamento. Devido ao elevado número de alternativas ou decisões candidatas, e das dificuldades de considerar as incertezas sobre a demanda e do processo produtivo, além da impossibilidade de compatibilizar os recursos de capacidade em sistemas multiprodutos, as técnicas avaliativas não são adequadas quando aplicadas isoladamente. Precisam ser complementadas por outros tipos de técnicas de planejamento, chamadas *técnicas gerativas*, que partem de certos critérios e restrições e extraem dentro do

universo de soluções um número limitado de soluções a serem analisadas pelas técnicas avaliativas. As técnicas gerativas são baseadas em métodos e técnicas de programação matemática.

Dentro do contexto acima, é importante observar que no nível de decisão onde atua o MRP, o processo de manufatura pode ser visualizado como um fluxo de partes ou itens (matéria prima, semi-acabados, componentes, etc.) sobre a linha de produção. Tais itens esperam em filas e passam por operações de fabricação e/ou montagem até se tornarem produtos e, portanto, prontos para serem absorvidos pelo mercado. Assim, tomando por base esta característica de fluxo do processo produtivo, é possível formular um problema de planejamento do tipo bem-estruturado (Carvalho *et al.*, 1992). Este problema não considera os custos e tempos de *setup* fundamentando-se em duas suposições (Ghosh e Gaimon, 1992, e Hindi, 1995). A primeira é que existe hoje uma tendência na direção de formas mais flexíveis de manufatura, com capacidade de se adaptarem rapidamente a alterações nos produtos, resultando em setups desprezíveis e que devem ser incluídos nos tempos de processamento. A segunda é que, ao se considerar setups, o problema se torna um problema de programação inteira-mista (Garfinkel e Nemhauser, 1972) onde o correspondente problema sem setups é repetidamente resolvido durante o processo de solução para se encontrar os limites inferiores e/ou superiores pelo método branch-and-bound por exemplo.

Assumindo esta simplificação, o problema de planejamento da produção pode ser formulado como um problema de Programação Linear (PL), técnica adotada neste trabalho. As soluções encontradas através desta técnica tornam-se agora metas a serem perseguidas pelas práticas industriais, o que sugere a integração das duas técnicas visando complementariedade. Por outro lado, este enfoque permite olhar para todo o horizonte de planejamento e com isto pode considerar as necessidades de produção a serem transferidas inter-períodos, causadas principalmente pela flutuação e sazonalidade da demanda.

Outras importantes restrições de produção podem ser consideradas, como por exemplo, a sincronização temporal de produção das várias sublinhas de montagem, as incertezas associadas ao processo de produção, e o compartilhamento de diferentes

produtos com os recursos de produção. Como exposto anteriormente, esta é uma técnica gerativa que parte de um conjunto de critérios e restrições e gera um conjunto de decisões. Estas decisões devem ser testadas utilizando-se uma representação mais detalhada do sistema de produção, o que é possível através das práticas industriais. A técnica gerativa, dirigida pelos critérios, e as práticas industriais, dirigidas pela demanda, são enfoques complementares (Carvalho *et al.*, 1998). Assim, a composição dos dois enfoques certamente enriquece o processo de resolução do problema.

Embora a formulação completa do problema de planejamento da produção seja um problema de PL, em aplicações práticas, o tamanho do problema torna ineficiente a utilização direta das técnicas de PL. Para que todos os requisitos descritos anteriormente sejam atendidos e para que o tempo de processamento não se torne proibitivo, o algoritmo de solução deve explorar a estrutura especial do problema.

Como será mostrado a seguir, o problema do planejamento da produção em uma linha serial monoproduto multiperíodo pode ser estruturado como um grafo (Gaimon, 1986, e Carvalho *et al.*, 1994). Um esquema mais complexo, onde uma linha de produção é caracterizada por uma árvore de produto, dá origem a um problema de fluxo em redes com restrições adicionais (Carvalho *et al.*, 1998a). Se o sistema possibilita o processamento de mais de um produto, então tem-se um problema multiestágio, multiperíodo, multiproduto de fluxo em redes com restrições adicionais (Carvalho *et al.*, 1999).

Algumas vantagens de modelos por fluxo em redes são: rápida solução que permite ao usuário (decisor) responder prontamente à dinâmica do mercado; ágil atualização do último planejamento; formulação gráfica de seus problemas, entre outras. A modelagem gráfica do problema pode ser extremamente poderosa para a formulação e refinamento do modelo, por não exigir do usuário conhecimentos de programação matemática.

Uma vantagem adicional do emprego de modelos de programação matemática é a possibilidade de consideração simultânea de vários critérios (objetivos) de planejamento. Problemas práticos de planejamento freqüentemente envolvem critérios de decisão

conflitantes (Norbis e MacGregor Smith, 1988 e Singh *et al.*, 1990), que através do enfoque adotado neste trabalho, podem ser adequadamente levados em conta através de Programação Linear Multiobjetivo (PLMO). Adota-se uma abordagem baseada em Programação Alvo (ou Programação por Metas) (Cohon, 1978 e Zeleny, 1982) tendo em vista que trabalhar com metas é uma prática comum a muitos problemas de produção que consideram múltiplos critérios para avaliação de desempenho (Dean *et al.*, 1990).

O capítulo 2 a seguir trata da modelagem do problema de planejamento da produção por fluxo em redes. No capítulo 3 será utilizada uma única função objetivo (abordagem mono-objetivo) na formulação do problema, enquanto que no capítulo 4 será adotada uma formulação multiobjetivo do problema. Para efeito de comparação entre as técnicas descritas nestes dois capítulos, serão utilizados dados de uma fábrica de molas do parque industrial brasileiro. Finalmente, no capítulo 5 são apresentadas as conclusões e perspectivas do trabalho.

## CAPÍTULO 2

### MODELAGEM DE LINHAS DE MONTAGEM POR FLUXO EM REDES

O processo de manufatura pode ser visto como um fluxo de itens (i.e., matéria-prima, produtos semi-acabados, etc) em uma rede de produção. No caminho pelo processo produtivo estes itens esperam em filas, passam por operações de fabricação e/ou montagem, até se tornarem produtos acabados para serem absorvidos pelo mercado. Tomando por base esta característica do processo produtivo, é possível formular um problema de planejamento do tipo bem-estruturado e assim considerar os recursos disponíveis ou previstos, quer de matéria-prima ou de equipamentos, além dos custos de produção e de matéria-prima, com o objetivo de atender a demanda de maneira ótima, dentro do horizonte de planejamento. De acordo com este enfoque, mostra-se a seguir que o problema do planejamento da produção em uma linha serial monoproduto multiperíodo pode ter uma estrutura de grafo.

Outras características relacionadas diretamente com o processo de manufatura também devem ser levadas em conta. Exemplo: se a linha em estudo é de montagem, então deve-se considerar a sincronização temporal entre os fluxos das várias sub-linhas que contribuem para a formação do produto final.

A linha de montagem é caracterizada por uma árvore de produto e quando expandida no tempo dá origem a um problema de fluxo em redes com restrições adicionais. Se o sistema possibilita o processamento de mais de um produto, então tem-se um problema multiestágio, multiperíodo, multiproduto de fluxo em redes com restrições adicionais.

Para que todos estes requisitos sejam atendidos e ainda assim o tempo de processamento em computador não se torne proibitivo (soluções rápidas possibilitam ao

decisor responder prontamente à dinâmica do mercado), o algoritmo de solução deve aproveitar a estrutura especial do problema.

Este capítulo trata da modelagem de um sistema de produção por fluxo em redes. Inicialmente é feita a modelagem de uma linha serial monoproduto. Em seguida a modelagem é estendida para considerar o caso de linhas de montagem mono e multiproduto. Por fim, são descritas as funções objetivo do problema.

## 2.1 - NOTAÇÃO

Esta seção estabelece a notação adotada neste capítulo.

### 2.1.1 - Índices:

$i$	índice relativo a estágio: $i=1,2,\dots,I$
$k$	índice relativo a produto: $k=1,2,\dots,K$
$\ell$	índice do ponto terminal de uma sub-linha da árvore de produto
$m_i$	índice de máquinas
$t$	índice relativo a tempo: $t=1,2,\dots,T$

### 2.1.2 - Parâmetros de entrada:

$c_a(i,t,k)$	custo unitário de armazenagem do produto $k$ no estágio $i$ e no tempo $t$
$c_b(t,k)$	custo unitário de atendimento à demanda em atraso do produto $k$ no tempo $t$
$c_m(t,k)$	custo unitário de aquisição de matéria prima do produto $k$ no tempo $t$
$c_p(i,t,k)$	custo unitário de produção do produto $k$ no estágio $i$ e no tempo $t$

$d(t)$	demanda de produtos acabados a ser atendida no tempo $t$
$d(t,k)$	demanda de produtos acabados do tipo $k$ a ser atendida no tempo $t$
$D$	demanda de produtos acabados no horizonte de planejamento
$D(k)$	demanda de produtos acabados $k$
$I$	número de estágios de produção
$J$	conjunto dos arcos associados a operações na máquina $m_i$
$K$	conjunto de produtos diferentes do sistema produtivo
$M$	disponibilidade total de matéria prima no horizonte de planejamento
$M(k)$	disponibilidade total de matéria prima para o produto $k$
$p(i,t,k)$	taxa de produção do estágio $i$ do produto $k$ no tempo $t$
$q(\ell)$	relação de acoplamento do ponto terminal $\ell$ de uma sub-linha
$q(\ell,k)$	relação de acoplamento do ponto terminal $\ell$ para o produto $k$
$r(t,k)$	retorno por unidade de demanda atendida do produto $k$ no tempo $t$
$T$	número de períodos do horizonte de planejamento
$TD(m_i,t)$	capacidade disponível para operação da máquina $m_i$ no tempo $t$

### 2.1.3 - Variáveis de Decisão:

$x(i,t)$	quantidade produzida no estágio $i$ e no tempo $t$
$x(0,t)$	quantidade de matéria prima utilizada no tempo $t$
$x(I + 1,t)$	nível de atendimento da demanda
$x(i,t,k)$	quantidade do produto $k$ produzida no estágio $i$ e no tempo $t$
$x(0,t,k)$	quantidade de matéria prima do produto $k$ utilizada no tempo $t$



$x(I + 1, t, k)$	nível de atendimento da demanda do produto $k$
$y(i, t)$	quantidade armazenada no estágio de produção $i$ entre os tempos $t$ e $t+1$
$y(i, 0)$	injeção de produtos semi-acabados recebidos de períodos de planejamentos anteriores para o estágio de produção $i$
$y(i, t, k)$	quantidade do produto $k$ armazenada no estágio de produção $i$ entre o tempo $t$ e o tempo $t+1$
$z(t)$	demanda em atraso atendida no tempo $t$
$z(t, k)$	demanda em atraso do produto $k$ atendida no tempo $t$

## 2.2 - O PLANEJAMENTO DA PRODUÇÃO EM UMA LINHA SERIAL MONOPRODUTO

Tendo como objetivo final a formulação do problema de planejamento da produção, nesta seção considera-se o caso mais simples de uma linha serial multiestágio, multiperíodo de um processo discreto de manufatura. A figura 2.1 ilustra uma linha com três estágios de produção, onde cada estágio está associado a uma máquina ou um grupo de máquinas que realizam transformações ou transporte com capacidades definidas à priori.

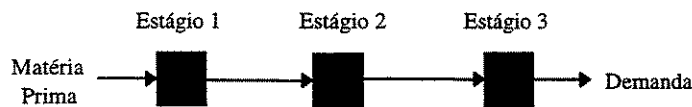


Figura 2.1: Linha de produção serial

A tarefa do planejamento é estabelecer a escala de produção, a partir de metas semanais definidas por um nível hierárquico superior (por exemplo, decisões geradas pelo programa mestre de produção (Fox, 1984)). Para isto deve desmembrar a demanda semanal em demandas diárias de acordo com as restrições de capacidade das máquinas,

de armazenagem e de oferta de matéria-prima, e considerar também os objetivos do planejamento.

As decisões a serem tomadas para cada dia são: a quantidade a ser produzida, a utilização de matéria-prima e o carregamento de cada máquina ou grupo de máquinas. A expansão da figura 2.1 para considerar um período de planejamento de cinco dias é mostrada na figura 2.2. Nesta expansão, o número de máquinas de cada grupo de máquinas pode ser diferente ao longo do período de planejamento. Assim, o grupo de máquinas 1 sofre decréscimo de capacidade nos períodos 4 e 5, enquanto que o grupo de máquinas 3 recebe uma nova máquina à partir do período 3.

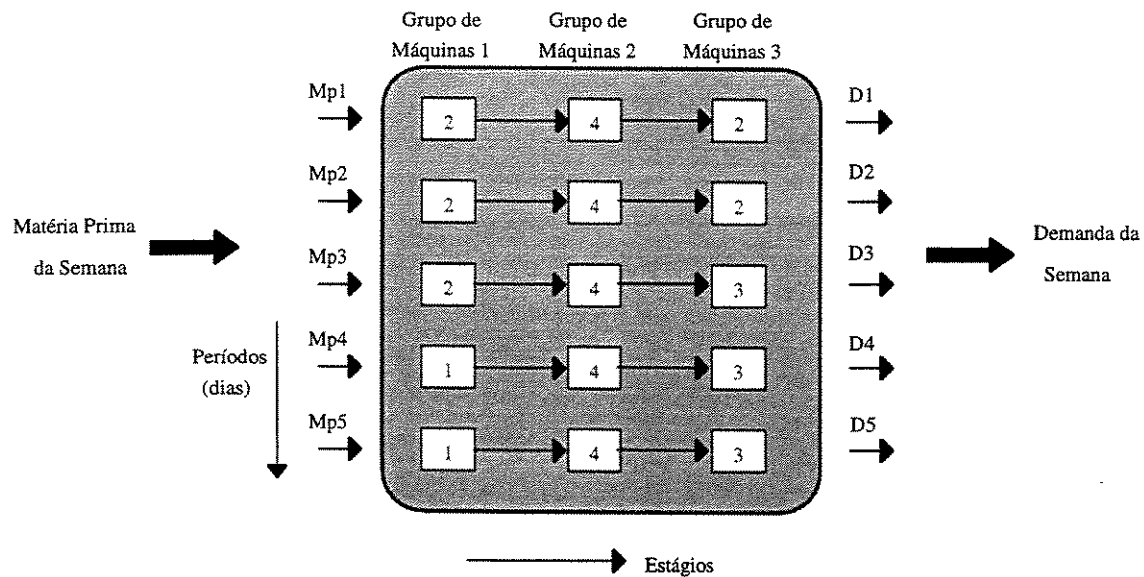


Figura 2.2: Linha de produção com três estágios e cinco períodos.

Cada estágio tem associado um custo de operação, que pode ser diferente para cada período de tempo. Ao deixar o estágio  $i$ , um item pode ir imediatamente ao estágio  $i+1$  ou ser armazenado para operação em períodos subsequentes. Esta decisão depende da capacidade de produção da linha e do custo de estocagem dos itens.

De maneira geral, o problema apresenta as seguintes características:

- o horizonte de planejamento é discretizado em períodos (hora, turno, dia),
- cada estágio de produção pode dispor de uma ou mais máquinas,
- a quantidade de máquinas pode variar ao longo dos períodos de planejamento,
- as capacidades de produção e de armazenagem são definidas à priori dentro do horizonte de planejamento, e
- os limites de oferta de matéria prima para atendimento à demanda são conhecidos dentro do horizonte de planejamento.

O planejamento deve gerar, de acordo com políticas de produção, armazenagem, aquisição de matéria prima, e/ou atendimento à demanda em atraso, um conjunto de decisões que determina:

- o quanto cada máquina deve produzir e em que período,
- o quanto armazenar por estágio e em que período,
- a escala de utilização de matéria prima por período,
- a escala de atendimento à demanda por período,
- a parcela da demanda atendida com atraso, e
- o lucro financeiro dentro do período de planejamento.

Um caminho para modelar e resolver este tipo de problema passa pela observação de que uma linha de produção como a ilustrada na figura 2.2, pode ser visualizada como um conjunto de itens fluindo através das máquinas e das unidades de armazenagem, sofrendo operações (fabricação ou transporte) em cada período de tempo, até se tornarem um produto final. Esta característica sugere a aplicação de técnicas de grafos para a modelagem do problema de planejamento da produção.

Os nós do grafo representam os pontos de decisão, onde são definidos o quanto produzir e o quanto armazenar num determinado período  $t$ . Cada um destes nós dá origem a uma restrição, derivada de uma equação de balanço como será mostrado a seguir.

O primeiro conjunto de restrições refere-se ao nó de oferta de matéria prima no horizonte de planeamento. A representação esquemática do nó é apresentada na figura 2.3. Os arcos que deixam o nó representam as possibilidades de utilização de matéria prima dentro do horizonte  $T$  de planeamento,  $M$  é a disponibilidade total de matéria prima no horizonte de planeamento e  $x(0,t)$ ,  $t=1,2,\dots,T$  a quantidade de matéria prima utilizada no período  $t$ .

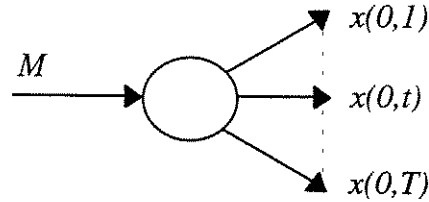


Figura 2.3: Representação esquemática para o balanço de oferta de matéria prima.

A equação de balanço deste nó é:

$$\sum_{t=1}^T x(0,t) = M \quad (2.1)$$

O segundo conjunto de restrições define uma relação de balanço nos pontos de decisão antes de cada máquina ou estágio de produção. Nestes pontos define-se para um estágio  $i$  no período  $t$ , o quanto produzir  $x(i,t)$  e o quanto armazenar  $y(i,t)$  para o próximo período, de acordo com o nível produzido  $x(i-1,t)$  no estágio anterior e a quantidade armazenada  $y(i,t-1)$  no período anterior. A representação esquemática deste nó para o estágio de produção  $i$  e período de tempo  $t$  está apresentada na figura 2.4.

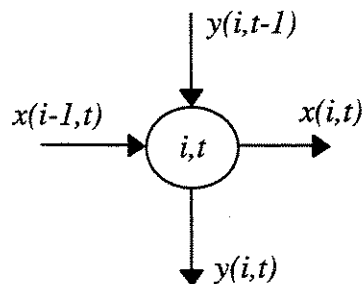


Figura 2.4 Representação esquemática para o ponto de decisão do estágio  $i$  e período  $t$

A equação de balanço deste nó é:

$$x(i-1,t)+y(i,t-1)=x(i,t)+y(i,t) \quad i=1,2,\dots,I, \text{ e } t=1,2,\dots,T. \quad (2.2)$$

No primeiro estágio a variável  $x(i-1,t)=x(0,t)$  é igual a quantidade de matéria prima distribuída para o período  $t$ . No primeiro período a variável  $y(i,t-1)=y(i,0)$  é a injeção de produtos semi-acabados recebidos de períodos de planejamentos anteriores. No período  $t=T$ , a variável  $y(i,T)$  representa a decisão de se produzir neste período produtos semi-acabados para períodos de planejamentos futuros.

O modelo deve considerar a possibilidade do sistema distribuir o atendimento da demanda total  $D$  entre os períodos de planejamento. Para isto, adiciona-se um terceiro conjunto de restrições, cuja representação esquemática para o período  $t$  está ilustrada na figura 2.5. A variável  $y(I+1,t)$  representa a quantidade de produtos acabados a ser armazenada para o próximo período. O estoque final do sistema pode ser positivo, significando que é vantajoso produzir em  $t$  para atender uma demanda futura. Por outro lado, o estoque final pode ser negativo, indicando deficiência de produção ou vantagem em adiar o atendimento à demanda. A decisão de atender uma demanda do presente no futuro é modelada pela variável  $z(t)$  que representa a quantidade de demanda em atraso (ordens de fabricação em atraso ou *backorder*) atendida no período  $t$ . No período  $t=T$ , o arco  $z(T)$  é ligado ao nó raiz com custo penalizado. O arco  $d(t)$  representa a demanda de produtos acabados a ser atendida no período  $t$ .

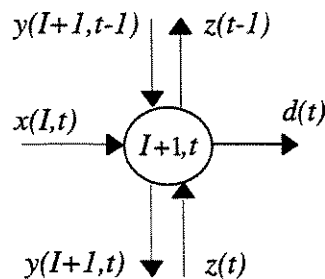


Figura 2.5: Representação esquemática para o balanço no último ponto de decisão.

A equação de balanço deste nó é:

$$x(I,t)+y(I+1,t-1)+z(t)=d(t)+y(I+1,t)+z(t-1) \quad t=1,2,\dots,T. \quad (2.3)$$

Na equação (2.3), a demanda  $d(t)$  de um período  $t$  é fixa (pré-estabelecida), significando a obrigatoriedade de atendimento. Outro enfoque é tratar a demanda como uma variável que pode flutuar dentro de certos limites. Esta situação reflete flexibilidade no perfil de atendimento à demanda no horizonte de planejamento ou objetivo do processo otimizador. Neste caso adiciona-se uma quarta e última restrição, que se refere ao balanço em relação à demanda atendida do horizonte de planejamento  $T$ . A representação esquemática deste nó é apresentada na figura 2.6.

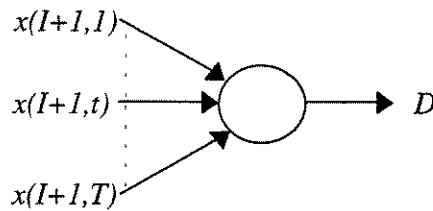


Figura 2.6: Representação esquemática para o balanço de atendimento à demanda  $D$ .

Esta representação estabelece que a soma dos níveis de atendimento da demanda de todos os períodos de tempo deve ser igual à demanda do horizonte de planejamento. A equação de balanço deste nó é:

$$\sum_{t=1}^T x(I+1,t)=D \quad (2.4)$$

A aplicação das representações esquemáticas para os diferentes tipos de nós do sistema da figura 2.2, resulta no grafo da figura 2.7. Cada nó, correspondente a um estágio de produção, possui dois arcos divergentes, um para representar o processamento do período e outro para representar a armazenagem do sistema. A exceção ocorre com os nós relativos ao suprimento de matéria prima e de balanço de demanda. Esta característica associada a estrutura esparsa apresentada pela matriz incidência nó-arco (como será visto nas próximas seções) é o que incentiva a pesquisa de algoritmos dedicados de solução na busca de melhor desempenho. Tirar proveito de estruturas particulares pode resultar em ganhos significativos de tempo e de memória em computadores possibilitando uma

representação mais detalhada do sistema produtivo e a análise de um maior número de cenários.

A modelagem acima aplica-se apenas a processos produtivos seriais simples e portanto deve ser estendida para contemplar problemas reais da indústria de manufatura moderna, como é o caso do problema de linhas de montagem tratado a seguir.

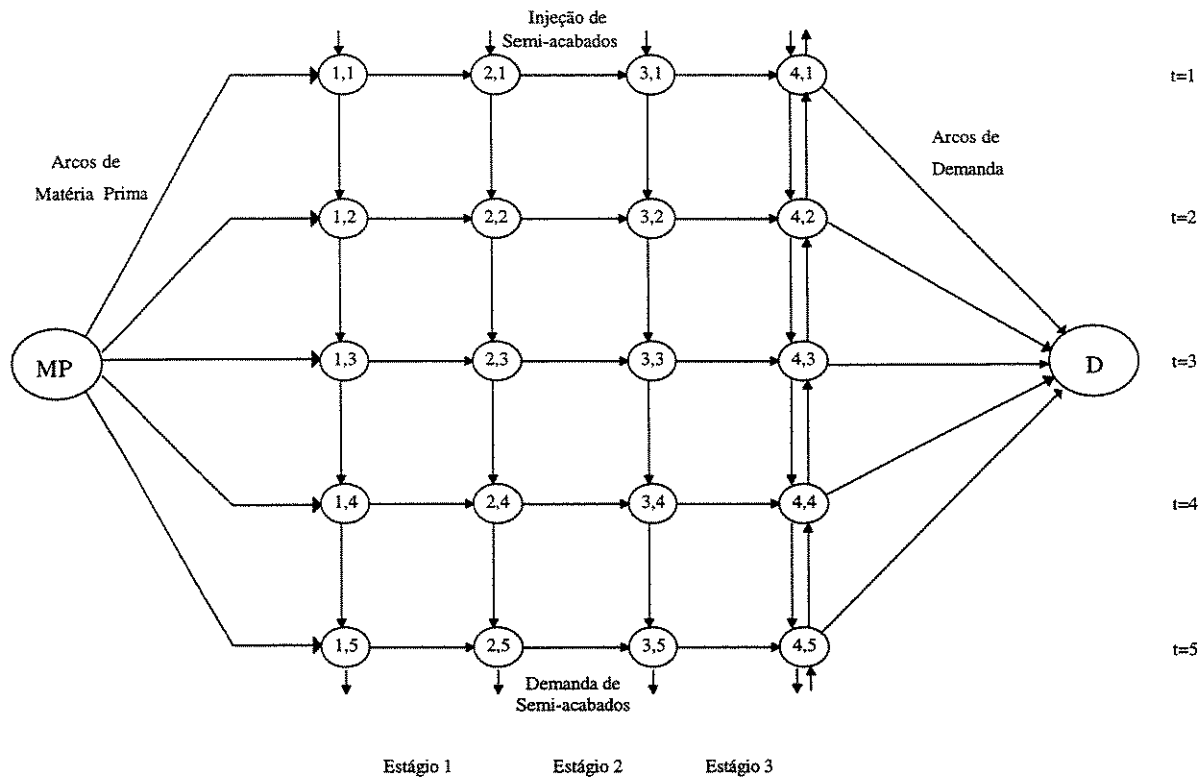


Figura 2.7 - Representação por grafo do sistema produtivo.

### 2.3 - O PLANEJAMENTO DA PRODUÇÃO EM UMA LINHA DE MONTAGEM MONOPRODUTO

Uma linha de montagem é definida pela árvore do produto, ou seja, pela seqüência de operações que o produto e seus itens devem percorrer até gerarem o produto final (Fernandes, 1995). O problema de uma linha de montagem composta de três sub-linhas é ilustrado na figura 2.8. Através das sub-linhas fluem os itens e os componentes

necessários à montagem do produto final, que são acoplados em pontos determinados da linha. Nesta figura, as linhas pontilhadas indicam os acoplamentos entre as sub-linhas nos seus respectivos pontos de montagem.

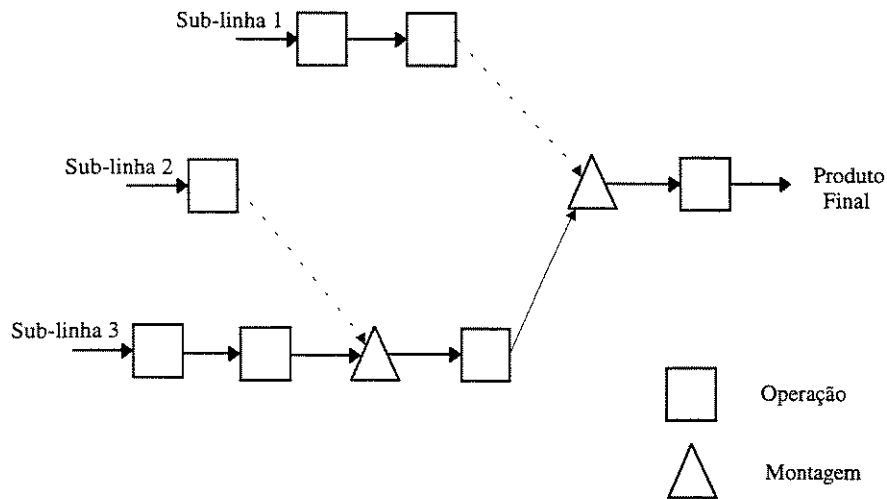


Figura 2.8: Representação de uma linha de montagem.

É necessário definir o número de componentes a serem montados em cada item do produto principal em cada ponto de montagem (quatro pneus por carro, por exemplo). Desta forma, uma das funções mais importantes do planejamento da produção é coordenar o fluxo entre as sub-linhas concorrentes visando eficiência na montagem dos produtos finais. Esta coordenação ou sincronização é necessária para prevenir a criação de estoques supérfluos entre estágios ou que a produção pare, devido a um fluxo não-balanceado entre produtos e componentes.

A figura 2.9 expande a linha de montagem da figura 2.8 para um horizonte com dois períodos de planejamento.

A representação por grafo da figura 2.9 é apresentada na figura 2.10. Nesta figura, os arcos contínuos representam as produções, os arcos tracejados as armazenagens e os arcos pontilhados indicam os pontos de acoplamento.



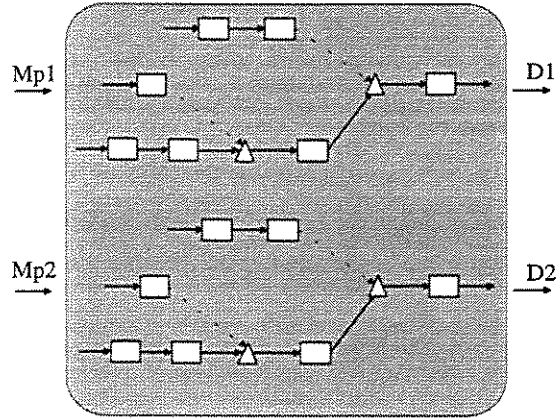


Figura 2.9: Linha de montagem com dois períodos.

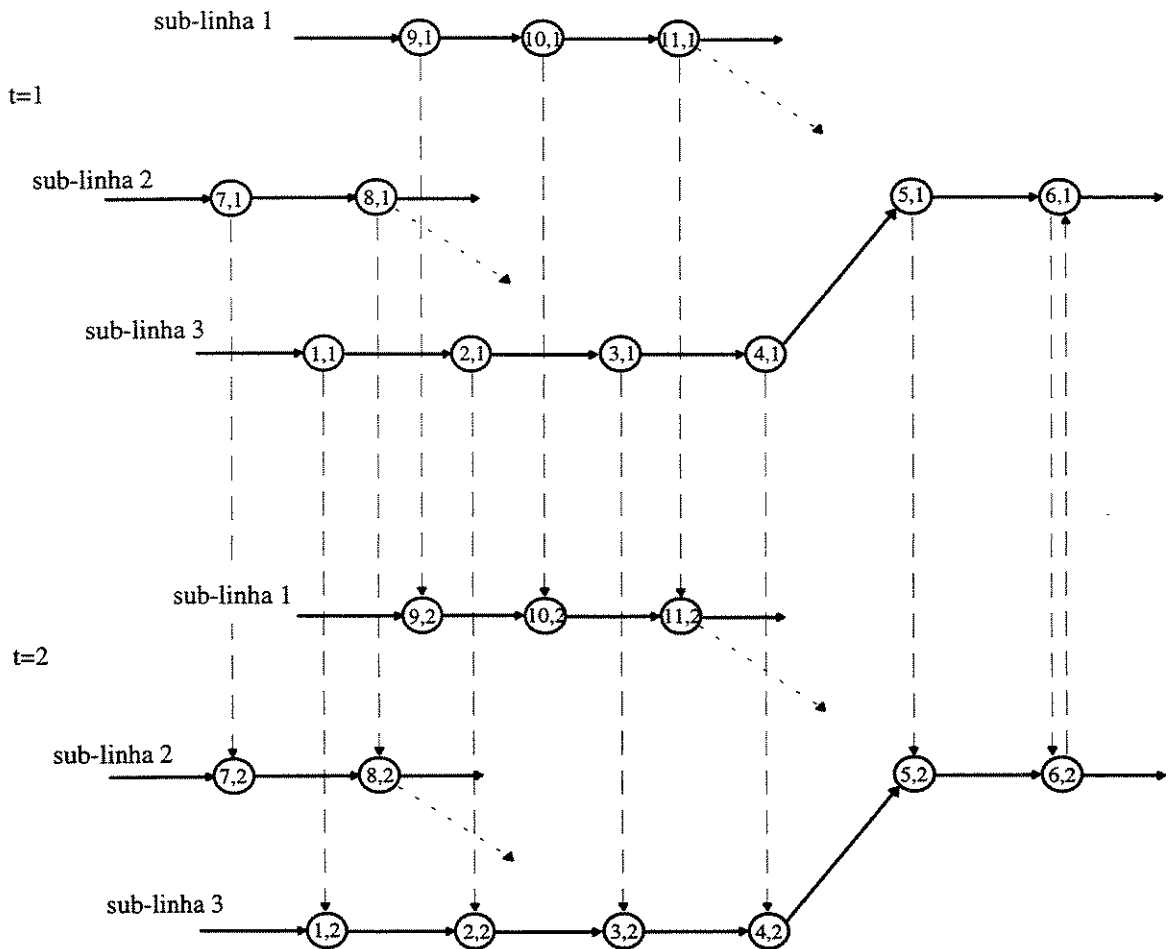


Figura 2.10 - Representação por grafo de uma linha de montagem.

Para se garantir o balanço de fluxo no ponto terminal  $\ell$  de uma sub-linha, deve-se considerar a seguinte equação de acoplamento:

$$q(\ell)x(i,t)-x(\ell,t)=0, \quad t=1,2,\dots,T \quad (2.5)$$

onde  $\ell \in \Omega$  é o conjunto dos pontos terminais e  $x(\ell,t)$  representa a quantidade de peças disponíveis no ponto  $\ell$  no tempo  $t$  a serem acopladas às  $x(i,t)$  peças no ponto de montagem ou acoplamento  $i$  no tempo  $t$ , a uma dada relação de acoplamento  $q(\ell)$ . Esta equação garante que o fluxo no ponto terminal  $\ell$  de cada sub-linha é suficiente para realizar as montagens previstas no ponto de acoplamento  $i$ . Este conjunto de restrições faz a sincronização da produção e cria um número de restrições adicionais igual ao número de pontos de acoplamento multiplicado pelo número de períodos de tempo.

Uma análise da estrutura da matriz de restrições para o problema é feita à partir do exemplo da figura 2.10. Uma característica importante deste problema é apresentar estrutura de grafos independentes para cada sub-linha de produção, ou seja, cada sub-linha forma grafos independentes no tempo, como é visto na figura 2.11.

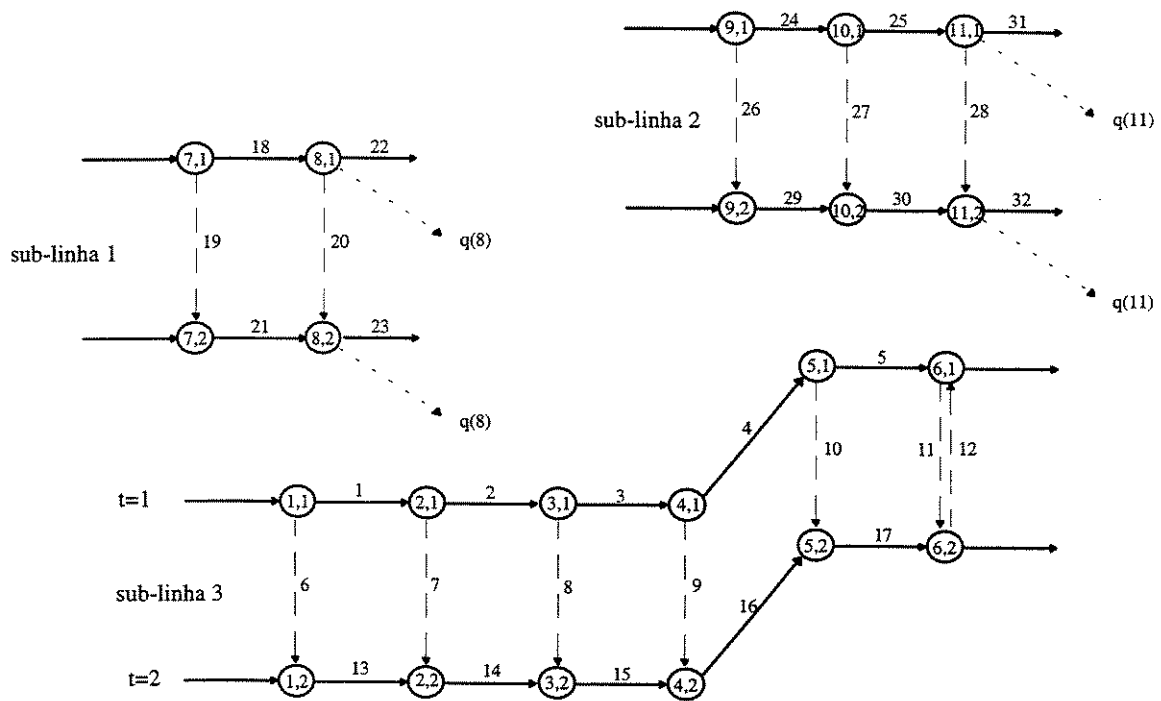


Figura 2.11: Grafos resultantes no espaço das sub-linhas.

Observe que existem três grafos independentes e a ligação entre os grafos é estabelecida pela equação (2.5) para cada um dos quatro acoplamentos. Por simplicidade não estão representados nesta figura os arcos de matéria prima, de demanda, e de injeção e demanda de produtos semi-acabados.

As equações de acoplamento para este exemplo são:

$$\begin{aligned}
 q(8)x(3,1) - x(8,1) &= 0 \\
 q(8)x(3,2) - x(8,2) &= 0 \\
 q(11)x(5,1) - x(11,1) &= 0 \\
 q(11)x(5,2) - x(11,2) &= 0
 \end{aligned}$$

A figura 2.12 ilustra a estrutura da matriz de restrições resultante. As variáveis  $x(3,1)$ ,  $x(3,2)$ ,  $x(5,1)$ ,  $x(5,2)$ ,  $x(8,1)$ ,  $x(8,2)$ ,  $x(11,1)$  e  $x(11,2)$  correspondem respectivamente as linhas 3, 15, 5, 17, 22, 23, 31 e 32 da matriz de restrição.

	sub - linha 3																	sub - linha 1					sub - linha 2												
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32			
1,1	1					1																													
2,1	-1	1					1																												
3,1		-1	1					1																											
4,1			-1	1					1																										
5,1				-1	1					1																									
6,1					-1						1	-1																							
1,2						-1							1																						
2,2							-1						-1	1																					
3,2								-1						-1	1																				
4,2									-1						-1	1																			
5,2										-1						-1	1																		
6,2											-1	1				-1																			
7,1																		1	1																
8,1																		-1		1		1													
7,2																			-1		1														
8,2																				-1	-1		1												
9,1																								1			1								
10,1																								-1	1			1							
11,1																									-1			1				1			
9,2																										-1			1						
10,2																											-1		-1	1					
11,2																												-1		-1		1			
1			$q(8)$																				-1												
2														$q(8)$									-1												
3				$q(11)$																											-1				
4															$q(11)$																	-1			

Figura 2.12: Estrutura da matriz de restrições.

## 2.4 - O PLANEJAMENTO DA PRODUÇÃO EM UMA LINHA DE MONTAGEM MULTIPRODUTO

Nos processos produtivos modernos, um sistema de produção deve considerar a possibilidade de fabricação de diferentes itens, atendendo assim a diversos tipos de demanda. A extensão do problema de linha de montagem para mais de um produto, resulta em um problema multiproduto, multiestágio e multiperíodo, onde vários produtos compartilham os mesmos recursos de produção. A figura 2.13 ilustra o sistema da figura 2.9, agora para processamento simultâneo de  $K$  diferentes produtos. Note que a árvore de cada produto pode ser diferente. A figura 2.14 é a representação por grafo da figura 2.13.

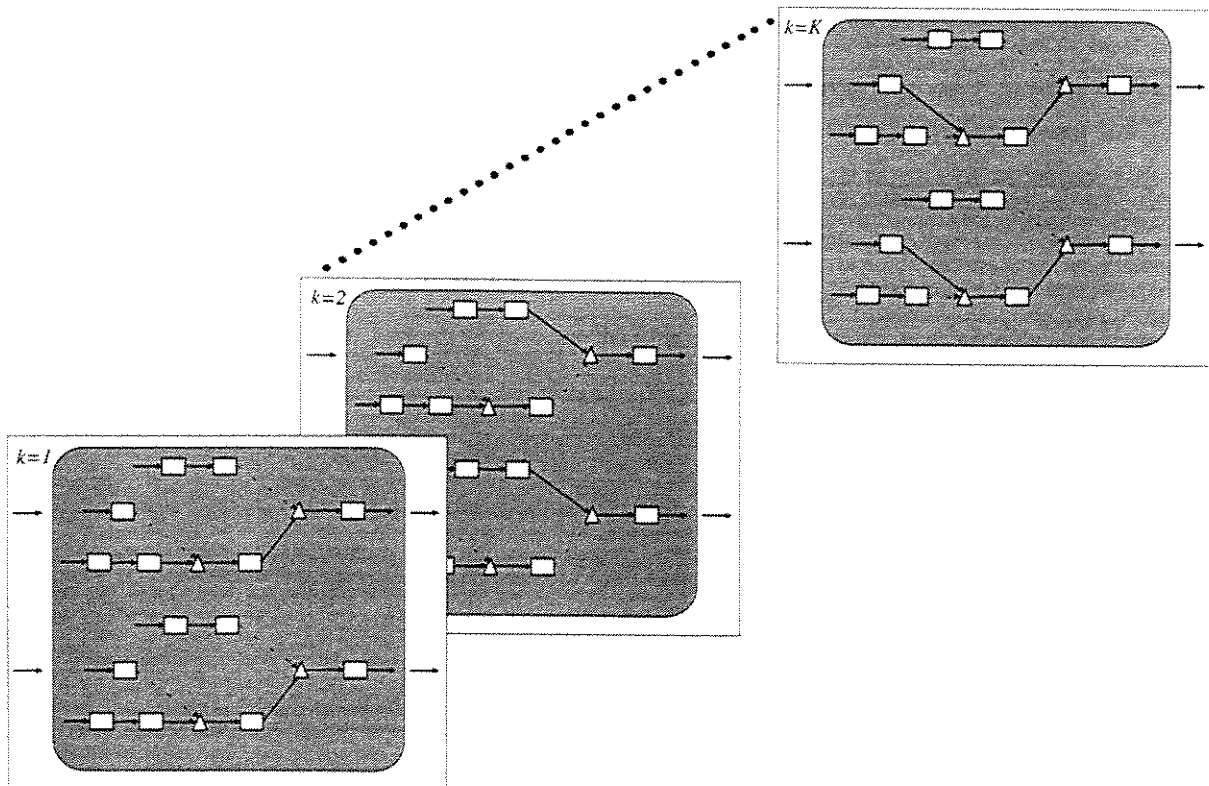


Figura 2.13: Sistema com  $K$  produtos e dois períodos.

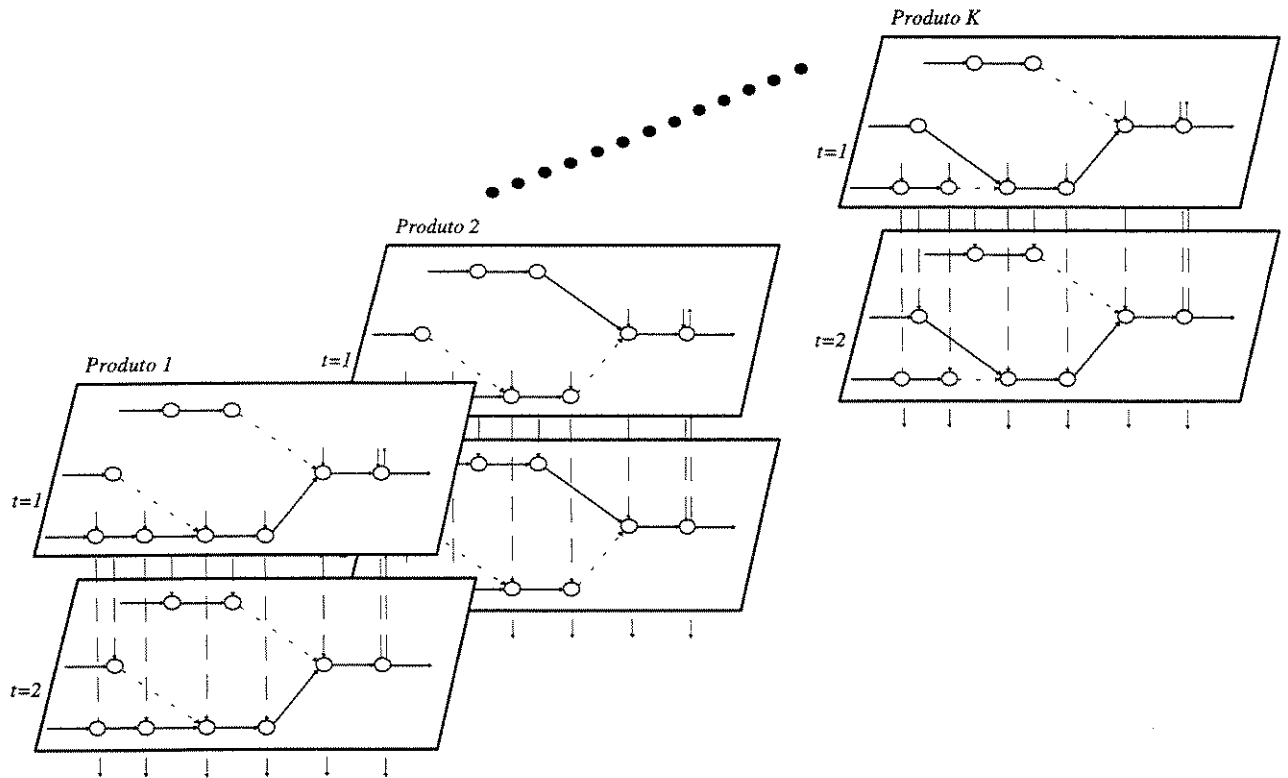


Figura 2.14: Representação por grafo da linha de montagem multiproduto.

A seguir a modelagem anterior é estendida para o processamento de  $K$  produtos, com árvores de produtos idênticas ou não. Uma característica importante é que o modelo matemático deste problema é representado por dois grupos principais de restrições: restrições de grafo e restrições adicionais (Carvalho *et al.*, 1992; Fernandes *et al.*, 1995).

#### 2.4.1 - Restrições de grafo

As equações (2.1), (2.2), (2.3) e (2.4) considerando agora a produção de  $K$  diferentes produtos passam a ser representadas por (2.6), (2.7), (2.8) e (2.9), respectivamente. Além disso, todas as variáveis aqui envolvidas estão sujeitas a limites inferior ou superior, valores que são definidos pelas capacidades mínimas e máximas de cada arco do grafo. Estes limitantes são expressos pelas equações (2.10) a (2.13).

$$\sum_{t=1}^T x(0,t,k)=M(k), \quad k=1,2,\dots,K \quad (2.6)$$

$$x(i-1,t,k)+y(i,t-1,k)=x(i,t,k)+y(i,t,k), \quad i=1,2,\dots,I, t=1,2,\dots,T, k=1,2,\dots,K \quad (2.7)$$

$$x(I,t,k)+y(I+1,t-1,k)+z(t,k)=d(t,k)+y(I+1,t,k)+z(t-1,k), \quad t=1,2,\dots,T, k=1,2,\dots,K \quad (2.8)$$

$$\sum_{t=1}^T x(I+1,t,k)=D(k), \quad k=1,2,\dots,K \quad (2.9)$$

$$\underline{x}(i,t,k) \leq x(i,t,k) \leq \bar{x}(i,t,k), \quad i=1,2,\dots,I, t=1,2,\dots,T, k=1,2,\dots,K \quad (2.10)$$

$$\underline{y}(i,t,k) \leq y(i,t,k) \leq \bar{y}(i,t,k), \quad i=1,2,\dots,I, t=1,2,\dots,T, k=1,2,\dots,K \quad (2.11)$$

$$\underline{d}(t,k) \leq x(I+1,t,k) \leq \bar{d}(t,k), \quad t=1,2,\dots,T, k=1,2,\dots,K \quad (2.12)$$

$$\underline{z}(t,k) \leq z(t,k) \leq \bar{z}(t,k), \quad t=1,2,\dots,T, k=1,2,\dots,K \quad (2.13)$$

#### 2.4.2 - Restrições adicionais

As equações que garantem o balanceamento de fluxo no ponto terminal de uma linha  $\ell$  para cada produto  $k$  são expressas por

$$q(\ell,k)x(i,t,k)-x(\ell,t,k)=0, \quad t=1,2,\dots,T, k=1,2,\dots,K \quad (2.14)$$

Na representação da figura 2.14, os produtos podem ter árvores de produtos idênticas ou não, e compartilhar o mesmo conjunto de máquinas durante o horizonte de planejamento.

Uma máquina  $m_i$  pode processar produtos diferentes. Assim, é importante garantir que a produção de peças diferentes em uma mesma máquina não leve à violação de sua capacidade horária de produção. Para assegurar o atendimento a esta restrição ao se considerar o problema multiproduto, adiciona-se mais uma restrição ao problema, do tipo:

$$\sum_{j \in J} x(j,t,k)/p(j,t,k) \leq TD(m_i,t), \quad t=1,2,\dots,T \text{ e } k=1,2,\dots,K \quad (2.15)$$

onde  $J$  representa o conjunto dos arcos associados a operações na máquina  $m_i$ ,  $p(j,t,k)$  é a taxa de produção que expressa o número de itens do produto  $k$  produzidos por unidade de tempo, e  $TD(m_i,t)$  é a fração do período  $t$  na qual a máquina  $m_i$  fica disponível para operação no período  $t$ .

### 2.4.3 - Estrutura da matriz de restrições

Da mesma forma que no caso monoproduto, a estrutura da matriz de restrições é analisada através de um exemplo. O exemplo da figura 2.10 é estendido para a produção de dois produtos com árvores diferentes, como ilustrado na figura 2.15.

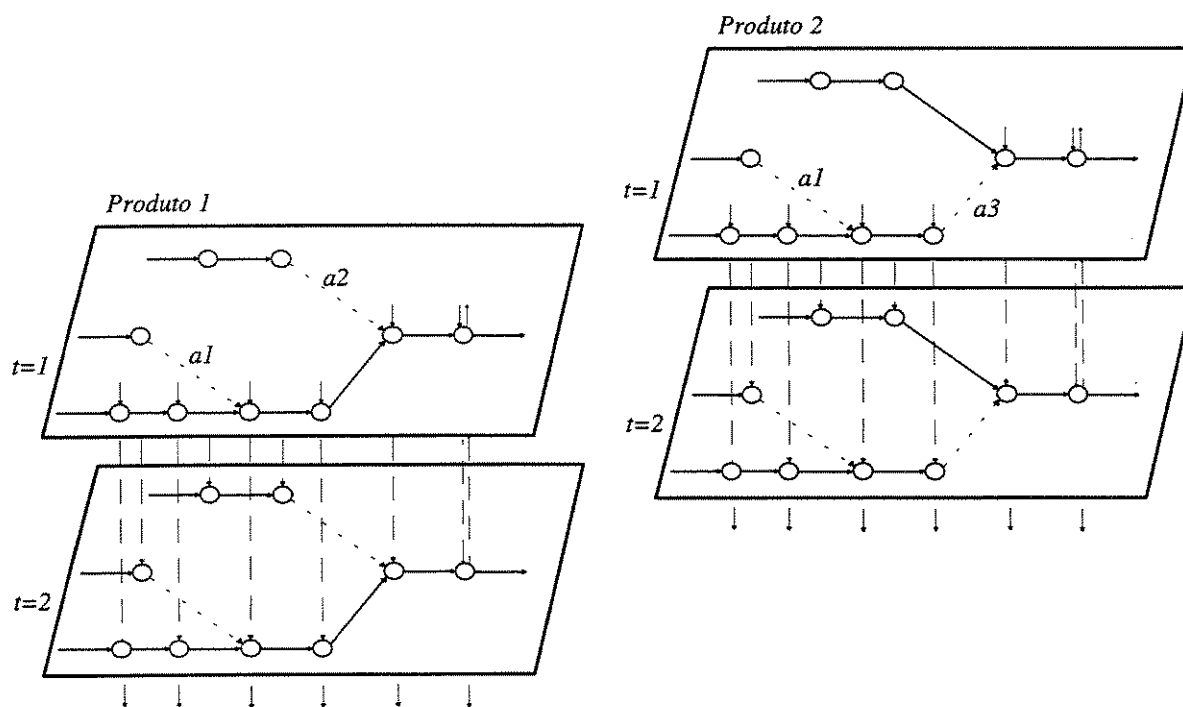
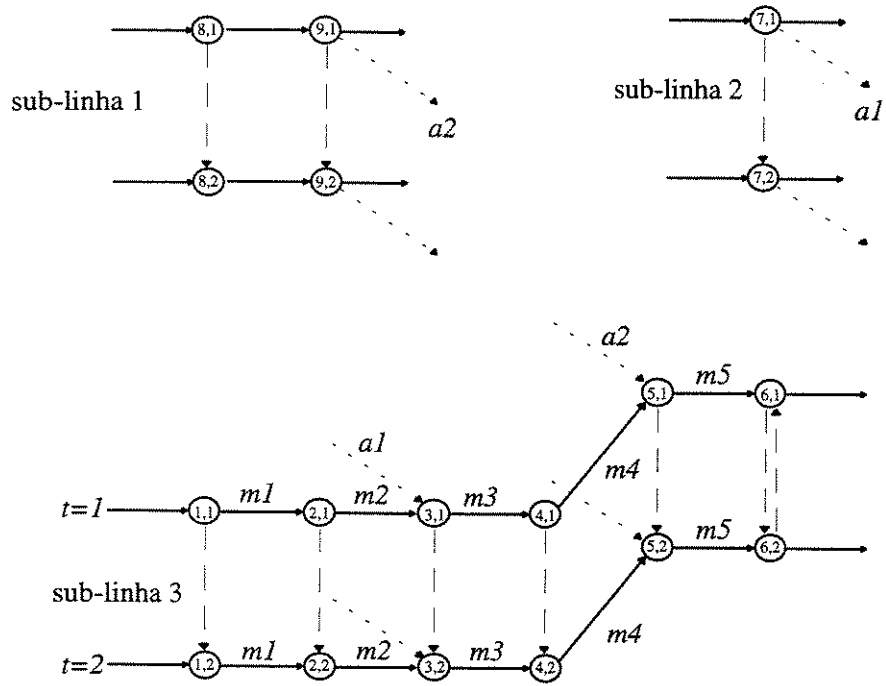


Figura 2.15: Representação por grafo do sistema produtivo.

Para cada um dos dois produtos existem 3 sub-linhas, gerando portanto 6 grafos independentes, como ilustrado na figura 2.16.



produto 1

produto 2

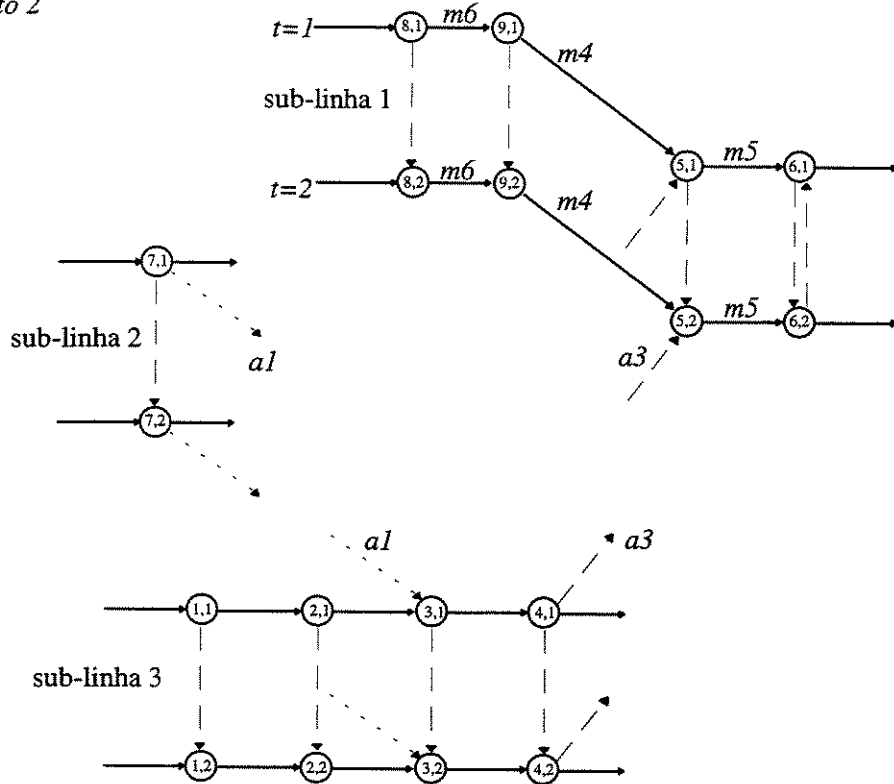


Figura 2.16: Grafos resultantes da separação no tempo.



A coordenação entre os grafos é estabelecida, em cada um dos pontos de acoplamento  $a_1$ ,  $a_2$  e  $a_3$ , por 8 equações, uma equação para cada acoplamento, em cada período de tempo e para cada produto. Ainda nesta figura são apresentadas as seqüências de produção da sub-linha 3 do produto 1 e da sub-linha 1 do produto 2. Note que as máquinas  $m_4$  e  $m_5$  são compartilhadas pelos dois produtos, embora os produtos tenham árvores diferentes. Isto resulta em mais 4 equações, uma para cada máquina e para cada período de tempo. As equações para cada um dos dois pontos de acoplamento do exemplo são:

$$\begin{aligned}
 q(7,1)x(3,1,1) - x(7,1,1) &= 0 \\
 q(7,1)x(3,2,1) - x(7,2,1) &= 0 \\
 q(9,1)x(5,1,1) - x(9,1,1) &= 0 \\
 q(9,1)x(5,2,1) - x(9,2,1) &= 0 \\
 q(7,2)x(3,1,2) - x(7,1,2) &= 0 \\
 q(7,2)x(3,2,2) - x(7,2,2) &= 0 \\
 q(4,2)x(5,1,2) - x(4,1,2) &= 0 \\
 q(4,2)x(5,2,2) - x(4,2,2) &= 0
 \end{aligned}$$

As equações de capacidade mútua das máquinas  $m_4$  e  $m_5$  para os dois períodos de tempo são:

$$\begin{aligned}
 \frac{x(4,1,1)}{p(4,1,1)} + \frac{x(9,1,2)}{p(9,1,2)} &\leq TD(m_4,1) \\
 \frac{x(5,1,1)}{p(5,1,1)} + \frac{x(5,1,2)}{p(5,1,2)} &\leq TD(m_5,1) \\
 \frac{x(4,2,1)}{p(4,2,1)} + \frac{x(9,2,2)}{p(9,2,2)} &\leq TD(m_4,2) \\
 \frac{x(5,2,1)}{p(5,2,1)} + \frac{x(5,2,2)}{p(5,2,2)} &\leq TD(m_5,2)
 \end{aligned}$$

A matriz de restrições do exemplo pode ser expressa como

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \\ \hline Q_1 & 0 \\ 0 & Q_2 \\ \hline E_1 & E_2 \end{bmatrix} \quad (2.16)$$

onde







## 2.5 - FUNÇÃO OBJETIVO

Existem inúmeras técnicas para a solução de problemas de planejamento da produção que empregam uma função objetivo única, sendo a mais comum a minimização dos custos de produção (Lewis *et al.*, 1992). Tal formulação permite ao planejador considerar implicitamente objetivos conflitantes manipulando-os através dos coeficientes da função custo. Por outro lado, diversas outras funções objetivo expressas em grandezas diferentes das de custo, são largamente utilizadas na literatura (Norbis, 1988 e Trzaskalik, 1990). Esta seção trata da modelagem das funções objetivos para o problema de planejamento da manufatura.

### 2.5.1 - FUNÇÃO LUCRO

No nível tático de planejamento, o objetivo de uma empresa de manufatura é a maximização do lucro, que é definido como a diferença entre o retorno monetário devido ao atendimento à demanda e os custos de produção (compra de matéria prima, produção, armazenagem, demanda em atraso). O lucro pode ser expresso pela seguinte função:

$$f(d, x, y, z) = f_r(d) - f_m(x) - f_p(x) - f_a(y) - f_b(z) \quad (2.18)$$

O primeiro termo de (2.18) contempla o retorno referente ao nível de atendimento da demanda

$$f_r(d) = \sum_{t=1}^T \sum_{k=1}^K r(t,k)x(I+1,t,k) \quad (2.19)$$

Embora o objetivo principal de um processo produtivo seja atender a demanda no horizonte de planejamento, este pode ser expresso de duas formas: o atendimento das demandas fixadas para todos os períodos de planejamento ( $\underline{d}(t,k) = \bar{d}(t,k)$ ,  $t=1,2,\dots,T$ ,  $k=1,2,\dots,K$ ) ou buscar um atendimento ótimo supondo-se que os níveis de atendimento possam flutuar entre limitantes inferior e superior para a demanda. O primeiro caso refere-se a demanda determinística e a política da empresa é totalmente

voltada ao cliente. Neste caso deve-se atender completamente a demanda com lucro ou prejuízo. Para o segundo caso, o valor do retorno  $r(t)$  deve assumir o valor real de venda do produto. O nível máximo de atendimento da demanda é dependente dos custos de produção.

O segundo termo de (2.18) expressa o custo de aquisição da matéria prima

$$f_m(x) = \sum_{t=1}^T \sum_{k=1}^K c_m(t,k)x(0,t,k) \quad (2.20)$$

O terceiro termo de (2.18) é relativo aos custos de produção, que envolvem custos de processamento das peças pelas máquinas ou centros produtivos

$$f_p(x) = \sum_{i=1}^I \sum_{t=1}^T \sum_{k=1}^K c_p(i,t,k)x(i,t,k) \quad (2.21)$$

O quarto termo de (2.18) contempla custos de armazenagem

$$f_a(y) = \sum_{i=1}^{I+1} \sum_{t=1}^T \sum_{k=1}^K c_a(i,t,k)y(i,t,k) \quad (2.22)$$

O quinto e último termo de (2.18) denota o custo ou penalização pelo atendimento a ordens de fabricação em atraso (*backorder*)

$$f_b(z) = \sum_{t=1}^{T-1} \sum_{k=1}^K c_b(t,k)z(t,k) \quad (2.23)$$

## 2.5.2 - OUTRAS FUNÇÕES OBJETIVO

De maneira geral, técnicas de minimização de uma função custo, como em (2.18), falham em considerar outros importantes objetivos nos processos de planejamento. Além disso, considerando-se que os coeficientes dessa função custo podem ser de difícil determinação prática (Jaaskelainen, 1969), outros objetivos expressos em grandezas

diferentes são mais facilmente definidos. Outras funções comumente encontradas na literatura são:

- minimização do número de peças armazenadas (*work-in-process inventories*) - este objetivo é de grande utilização e importância, pois proporciona ao planejador a consideração do limite físico de armazenagem dentro do ambiente de manufatura pela função objetivo:

$$f_w(y) = \sum_{i=1}^{I+1} \sum_{t=1}^T \sum_{k=1}^K y(i,t,k) \quad (2.24)$$

- minimização do tempo total de produção (*makespan*) - minimização do tempo necessário para se completar uma peça. Nesta aplicação, a diferença entre dois tempos consecutivos  $\Delta t$  é constante e tomada como unitária:

$$f_{mk}(y) = f_w(y)Dt = \sum_{i=1}^{I+1} \sum_{t=1}^T \sum_{k=1}^K y(i,t,k)\Delta t \quad (2.25)$$

- minimização do número de peças de *backorder* - minimização da quantidade de peças atendidas em atraso:

$$f_{bk}(z) = \sum_{t=1}^{T-1} \sum_{k=1}^K z(t,k) \quad (2.26)$$

Essas funções objetivos podem ser consideradas por mono ou multicritério. Ambos enfoques serão discutidos neste trabalho.

## 2.6 - FORMULAÇÃO GERAL

A formulação geral do problema multiobjetivo de fluxo em rede com restrições adicionais é a seguinte:

$$\begin{aligned} & \text{Min } [f_1(x) \ f_2(x) \ \dots \ f_{no}(x)]^T \\ & \text{s.a.: } \quad Ax = b \\ & \quad \quad Qx = 0 \\ & \quad \quad Ex \leq TD \\ & \quad \quad \underline{x} \leq x \leq \bar{x} \end{aligned} \quad (2.27)$$

onde  $b$  e  $TD$  são os coeficientes do lado direito das restrições.

Embora a formulação proposta seja a de um problema de programação linear (PL), o tamanho do problema para aplicações a sistemas reais torna ineficiente a utilização direta do algoritmo simplex. Isto se deve principalmente à natureza temporal do problema de planejamento da produção e à quantidade de produtos que compartilham o mesmo sistema produtivo. Portanto, requisitos como armazenamento de variáveis e tempo de execução computacional são fatores essenciais na escolha de um método de solução.

Com a utilização de apenas uma função objetivo, e sem as restrições lineares adicionais, equações (2.14) e (2.15), a estrutura resulta em um problema de fluxo em redes puro, para o qual existem muitos algoritmos eficientes de solução. A presença das restrições lineares adicionais impede o uso direto de técnicas de grafo.





## CAPÍTULO 3

### ABORDAGEM MONO-OBJETIVO

O problema de planejamento da produção descrito no capítulo anterior é um problema linear de fluxo em redes com restrições adicionais e pode ser genericamente expresso como:

$$\begin{aligned}
 & \text{Min } c^T x \\
 & \text{s.a: } Ax = b \\
 & \quad Sx \leq d \\
 & \quad \underline{x} \leq x \leq \bar{x}
 \end{aligned} \tag{3.1}$$

onde  $x \in R^n$  são variáveis que representam o fluxo nos arcos,  $A$  é a matriz  $n_1 \times n$  de incidência nó-arco ( $n_1 \leq n$ ),  $S$  é a matriz  $s \times n$  das restrições lineares adicionais (restrições de acoplamento  $Q$  e de processamento mútuo  $E$ ),  $c \in R^n$  são os coeficientes da função linear a ser minimizada,  $b \in R^{n_1}$  e  $d \in R^s$  são os coeficientes do lado direito das restrições, e  $\underline{x}$ ,  $\bar{x}$  são os limites inferior e superior de  $x$ , respectivamente. Neste capítulo adota-se uma abordagem mono-objetivo para o problema, isto é, uma única função objetivo  $c: R^n \rightarrow R$ . Existem diversas formas de resolução de problemas deste tipo, entre as quais encontra-se a utilização de pacotes comerciais para a resolução de problemas de programação matemática, como por exemplo o GAMS (Brooke e Meeraus, 1992). Uma possível estratégia para resolução de (3.1) seria o uso de relaxação das restrições adicionais. Entre as técnicas que utilizam este conceito encontram-se a relaxação lagrangeana (Ahuja *et al.*, 1993). Outras formas são algoritmos específicos como a técnica de decomposição primal (*side-constraint*) (Kennington e Helgason, 1980) e a utilização de funções de penalidade (Zenios *et al.*, 1995), que procuram aproveitar a estrutura especial do problema em busca de melhor desempenho computacional.

Inicialmente são apresentadas as técnicas de decomposição primal e de penalidade. A seguir é feita uma comparação de desempenho das técnicas citadas acima através de uma aplicação real em um ambiente de manufatura encontrado no parque industrial brasileiro.

### 3.1 - TÉCNICA DE DECOMPOSIÇÃO PRIMAL

Nesta seção é apresentada a técnica de decomposição primal, específica para a resolução de problemas de fluxo em redes com restrições adicionais. Inicialmente, define-se a base  $\bar{B}$  de (3.1), que pode ser escrita como

$$\bar{B} = \left[ \begin{array}{c|c} B & C \\ \hline D & F \end{array} \right] \quad (3.2)$$

onde  $B (n_1 \times n_1)$ ,  $C (n_1 \times s)$ ,  $D (s \times n_1)$  e  $F (s \times s)$  são submatrizes que compõem a base  $\bar{B}$ . A submatriz  $B$  é composta por colunas de  $A$  incorporadas à base  $\bar{B}$ . A submatriz  $C$  possui uma estrutura particular pois é composta por colunas de  $A$  ou por colunas de zeros. No primeiro caso as colunas se referem 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 submatrizes  $D$  e  $F$  resultam da definição das submatrizes  $B$  e  $C$ .

As variáveis associadas a  $B$  são chamadas de variáveis chaves. As variáveis restantes, associadas a  $C$ , são chamadas de variáveis não-chaves. A inversa da matriz  $\bar{B}$  é dada por

$$\bar{B}^{-1} = \left[ \begin{array}{c|c} B^{-1} + B^{-1}CQ^{-1}DB^{-1} & -B^{-1}CQ^{-1} \\ \hline -Q^{-1}DB^{-1} & Q^{-1} \end{array} \right] \quad (3.3)$$

onde a matriz de trabalho  $Q$  é definida como

$$Q = F - DB^{-1}C. \quad (3.4)$$

A atualização da inversa  $\bar{B}^{-1}$  pode ser obtida à partir de  $B^{-1}$ ,  $Q^{-1}$  e das matrizes  $C$  e  $D$ . O problema consiste em gerar, armazenar e trabalhar eficientemente com os elementos de  $B^{-1}$  e  $Q^{-1}$ . A seguir é apresentado um algoritmo para a resolução de problemas de fluxo em redes com restrições adicionais, que aproveita a estrutura de grafo do problema e utiliza como base de trabalho a matriz  $Q$ . Esta base de trabalho será atualizada a cada iteração evitando assim sua inversão após cada pivoteamento. Estas particularidades são aproveitadas pelo Algoritmo NETSIDE (Kennington e Helgason, 1980).

### 3.1.1 - ALGORITMO NETSIDE

- Passo 1: Partição de variáveis em básicas chaves, básicas não-chaves e não-básicas.
- Passo 2: Determine  $Q$  e  $Q^{-1}$ .
- Passo 3: Atualize os custos reduzidos.
- Passo 4: Escolha a variável não-básica  $k$ , candidata a base. Se não existem variáveis candidatas, FIM.
- Passo 5: Atualize a coluna da variável  $k$ .
- Passo 6: Determine o tamanho do passo e a variável que deixa a base.
- Passo 7: Atualize as variáveis.
- Passo 8: Atualize  $Q^{-1}$  e 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 NETSIDE são apresentados a seguir em detalhes, com atenção especial aos passos referentes à atualização dos custos reduzidos (passo 3), atualização da coluna candidata a entrar na base (passo 5) e atualização da inversa da base de trabalho (passo 8).

### 3.1.2 - ATUALIZAÇÃO DOS CUSTOS REDUZIDOS

Esta seção mostra como a estrutura da base  $\bar{B}$  pode ser usada para calcular eficientemente as variáveis duais  $\pi = c^B \bar{B}^{-1}$  utilizadas no algoritmo primal simplex. De acordo com a partição usada em (3.2), tem-se:

$$\begin{aligned} \pi &= [\pi_1 \mid \pi_2] \\ c^B &= [c^1 \mid c^2] \\ [\pi_1 \mid \pi_2] &= [c^1 \mid c^2] \begin{bmatrix} B^{-1} + B^{-1}CQ^{-1}DB^{-1} & \mid & -B^{-1}CQ^{-1} \\ \hline -Q^{-1}DB^{-1} & & Q^{-1} \end{bmatrix} \\ &= [(c^1 + (c^1 B^{-1}C - c^2)Q^{-1}D)B^{-1} \mid (c^2 - c^1 B^{-1}C)Q^{-1}] \end{aligned}$$

As operações envolvendo  $B^{-1}$  são eficientemente realizadas como em um problema de fluxo em redes que utiliza um algoritmo de marca (Jensen e Barnes, 1980; Kennington e Helgason, 1980). Para explorar este fato, o custo reduzido pode ser calculado pela seguinte seqüência de passos:

- Passo 1: Faça  $\gamma^1 = c^1 B^{-1}$ .
- Passo 2: Faça  $\gamma^2 = c^1 + (\gamma^1 C - c^2)Q^{-1}D$ .
- Passo 3: Faça  $\pi^1 = \gamma^2 B^{-1}$ .
- Passo 4: Faça  $\pi^2 = (c^2 - \gamma^1 C)Q^{-1}$ .

Nos passos 1 e 3, a estrutura de árvore da matriz  $B$  é explorada. Devido as suas estruturas particulares, as matrizes  $C$  e  $D$  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.

### 3.1.3 - ATUALIZAÇÃO DA COLUNA DA VARIÁVEL $k$

Esta seção mostra como aproveitar a estrutura da base  $\bar{B}$  para atualizar eficientemente  $y = \bar{B}^{-1}\bar{A}(k)$ , que é a coluna da variável  $k$  não-básica candidata a entrar na base. Utilizando a partição adotada em (3.2) e definindo-se  $y$  e  $\bar{A}(k)$  como

$$y = \begin{bmatrix} y^1 \\ \dots \\ y^2 \end{bmatrix}, \quad \bar{A}(k) = \begin{bmatrix} A(k) \\ \dots \\ S(k) \end{bmatrix}$$

tem-se

$$\begin{aligned} \begin{bmatrix} y^1 \\ \dots \\ y^2 \end{bmatrix} &= \begin{bmatrix} B^{-1} + B^{-1}CQ^{-1}DB^{-1} & \dots & -B^{-1}CQ^{-1} \\ \dots & \dots & \dots \\ -Q^{-1}DB^{-1} & \dots & Q^{-1} \end{bmatrix} \begin{bmatrix} A(k) \\ \dots \\ S(k) \end{bmatrix} \\ &= \begin{bmatrix} B^{-1}(A(k) + CQ^{-1}(DB^{-1}A(k) - S(k))) \\ \dots \\ Q^{-1}(S(k) - DB^{-1}A(k)) \end{bmatrix}. \end{aligned}$$

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

- Passo 1: Faça  $\gamma^1 = B^{-1}A(k)$ .
- Passo 2: Faça  $\gamma^2 = A(k) + CQ^{-1}(D\gamma^1 - S(k))$ .
- Passo 3: Faça  $y^1 = B^{-1}\gamma^2$ .
- Passo 4: Faça  $y^2 = Q^{-1}(S(k) - D\gamma^1)$ .

O passo 1 determina o ciclo que a variável  $k$ , candidata à base, fecha na árvore, enquanto que o passo 3 é efetuado de forma semelhante ao passo 1. As demais operações envolvendo  $C$  e  $D$  são realizadas de forma semelhante à atualização dos custos reduzidos.

### 3.1.4 - ATUALIZAÇÃO DA INVERSA DA BASE DE TRABALHO $Q$

A forma utilizada para a atualização da inversa da base de trabalho  $Q$  é apresentada a seguir. De acordo com a matriz  $\bar{B}$ , define-se uma matriz de transformação  $L$  como:

$$L = \left[ \begin{array}{c|c} I & -B^{-1}C \\ \hline 0 & I \end{array} \right]$$

Pós-multiplicando-se  $\bar{B}$  por  $L$  tem-se

$$\bar{B}L = \left[ \begin{array}{c|c} B & 0 \\ \hline D & F - DB^{-1}C \end{array} \right] = \left[ \begin{array}{c|c} B & 0 \\ \hline D & Q \end{array} \right] \quad (3.5)$$

Sejam  $\bar{B}_i$ ,  $L_i$  e  $Q_i$  respectivamente as matrizes da base, de transformação e de trabalho na  $i$ -ésima iteração. Deseja-se determinar uma expressão para a atualização de  $Q_{i+1}^{-1}$  a partir de  $Q_i^{-1}$ . Se

$$\beta_i = \bar{B}_i L_i$$

então

$$\bar{B}_i^{-1} = L_i \beta_i^{-1}. \quad (3.6)$$

Da mesma forma

$$\beta_{i+1} = \bar{B}_{i+1} L_{i+1}$$

e

$$\beta_{i+1}^{-1} = L_{i+1}^{-1} \bar{B}_{i+1}^{-1}. \quad (3.7)$$

Para manter a partição adotada em (3.2), antes de cada operação de pivoteamento é necessário trocar uma coluna não-chave com uma coluna chave, que está deixando a base. Neste caso, são necessárias duas alterações na base de trabalho para realizar a atualização. Mesmo que  $\bar{B}_i$  e  $\bar{B}_{i+1}$  sejam diferentes em apenas duas colunas, as matrizes

$Q_i^{-1}$  e  $Q_{i+1}^{-1}$  serão, em geral, completamente diferentes. Por esta razão, a inversa de  $\bar{B}_{i+1}$  será dada por

$$\bar{B}_{i+1}^{-1} = E \bar{B}_i^{-1} \quad (3.8)$$

onde  $E$  é uma matriz de transformação, podendo ser matriz-coluna elementar ou matriz de permutação.

Se  $E$  for particionada de forma compatível com a partição de variáveis chaves e não-chaves, resulta

$$E = \left[ \begin{array}{c|c} E_1 & E_2 \\ \hline E_3 & E_4 \end{array} \right] \quad (3.9)$$

Substituindo-se (3.8) em (3.7), tem-se

$$\beta_{i+1}^{-1} = L_{i+1}^{-1} E \bar{B}_i^{-1} \quad (3.10)$$

e substituindo-se (3.6) em (3.10), obtém-se então

$$\beta_{i+1}^{-1} = L_{i+1}^{-1} E L_i \beta_i^{-1}. \quad (3.11)$$

Adotando-se a mesma partição, e lembrando que

$$\beta^{-1} = \begin{bmatrix} B & 0 \\ D & Q \end{bmatrix}^{-1} = \begin{bmatrix} B^{-1} & 0 \\ -Q^{-1}DB^{-1} & Q^{-1} \end{bmatrix},$$

tem-se

$$\beta_{i+1}^{-1} = \left[ \begin{array}{c|c} B_{i+1}^{-1} & 0 \\ \hline -Q_{i+1}^{-1}D_{i+1}B_{i+1}^{-1} & Q_{i+1}^{-1} \end{array} \right] \quad (3.12)$$

$$\beta_i^{-1} = \left[ \begin{array}{c|c} B_i^{-1} & 0 \\ \hline -Q_i^{-1}D_iB_i^{-1} & Q_i^{-1} \end{array} \right]; L_{i+1} = \left[ \begin{array}{c|c} I & -B_{i+1}^{-1}C_{i+1} \\ \hline 0 & I \end{array} \right]; L_i = \left[ \begin{array}{c|c} I & -B_i^{-1}C_i \\ \hline 0 & I \end{array} \right]. \quad (3.13)$$

A inversa de  $L_{i+1}$  é



$$L_{i+1}^{-1} = \left[ \begin{array}{c|c} I & B_{i+1}^{-1}C_{i+1} \\ \hline 0 & I \end{array} \right]$$

Assim

$$\beta_{i+1}^{-1} = \left[ \begin{array}{c|c} B_{i+1} & 0 \\ \hline -Q_{i+1}^{-1}D_{i+1}B_{i+1}^{-1} & Q_{i+1}^{-1} \end{array} \right] = \left[ \begin{array}{c|c} I & B_{i+1}^{-1}C_{i+1} \\ \hline 0 & I \end{array} \right] \left[ \begin{array}{c|c} E_1 & E_2 \\ \hline E_3 & E_4 \end{array} \right] \left[ \begin{array}{c|c} I & -B_i^{-1}C_i \\ \hline 0 & I \end{array} \right] \left[ \begin{array}{c|c} B_i & 0 \\ \hline -Q_i^{-1}D_iB_i^{-1} & Q_i^{-1} \end{array} \right]$$

$$\beta_{i+1}^{-1} = \left[ \begin{array}{c|c} E_1 + B_{i+1}^{-1}C_{i+1}E_3 & E_2 + B_{i+1}^{-1}C_{i+1}E_4 \\ \hline E_3 & E_4 \end{array} \right] \left[ \begin{array}{c|c} B_i + B_i^{-1}C_iQ_i^{-1}D_iB_i^{-1} & -B_i^{-1}C_iQ_i^{-1} \\ \hline -Q_i^{-1}D_iB_i^{-1} & Q_i^{-1} \end{array} \right]$$

e após as manipulações matemáticas, tem-se que a base de trabalho para o  $(i+1)$ -ésimo pivoteamento pode ser calculada à partir de

$$Q_{i+1}^{-1} = (E_4 - E_3B_i^{-1}C_i)Q_i^{-1}.$$

Dois possíveis casos de atualização são avaliados a seguir:

- *Caso 1:* A coluna que deixa a base é não-chave. Para este caso, a matriz de transformação é da forma

$$E = \left[ \begin{array}{c|c} I & E_2 \\ \hline 0 & E_4 \end{array} \right]$$

Assim, a atualização da inversa da base de trabalho se reduz a:  $Q_{i+1}^{-1} = E_4Q_i^{-1}$ , onde  $E_4$  é uma matriz-coluna elementar.

- *Caso 2:* A coluna que deixa a base é uma coluna chave, correspondendo portanto a um arco da árvore. Se este arco for removido da árvore, outro arco deve se incorporar à árvore para manter a estrutura de  $\bar{B}$ . Então o primeiro passo é verificar se existe uma coluna básica pertencente ao conjunto de variáveis não-chaves que possa assumir o lugar da coluna chave candidata a deixar a base, como ilustrado na figura 3.1 abaixo.

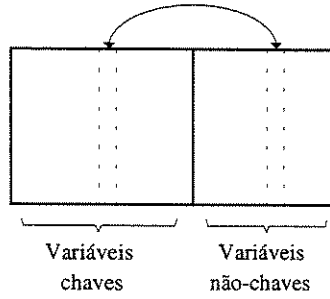


Figura 3.1: Troca de uma variável chave por uma variável não-chave.

Suponha que o arco que deixa a base corresponde à  $\ell$ -ésima coluna de  $B$ . Seja  $B^\ell$  igual a  $\ell$ -ésima linha de  $B^{-1}$  e seja  $\gamma = B^\ell C$ . Então, se  $\gamma_j \neq 0$ , a  $j$ -ésima coluna de  $C$  pode ser trocada com a  $\ell$ -ésima coluna de  $B$ , e a matriz resultante será não-singular. Dois sub-casos são agora considerados:

- *Sub-caso 2a:*  $\gamma_j \neq 0$ . Considere a seguinte matriz de permutação

$$E = \begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix} \begin{array}{l} \leftarrow \text{linha } \ell \\ \\ \\ \leftarrow \text{linha } n_1 + j \end{array}$$

$\uparrow \qquad \qquad \uparrow$   
*coluna*  $\ell$       *coluna*  $n_1 + j$

Então

$$E_3 = \begin{bmatrix} \vdots & \vdots & \vdots \\ \vdots & 1 & \vdots \\ \vdots & \vdots & \vdots \end{bmatrix} \leftarrow \text{linha } j \quad e \quad E_4 = \begin{bmatrix} I & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & I \end{bmatrix}$$

$\uparrow \qquad \qquad \qquad \uparrow$   
*coluna*  $\ell$                       *coluna*  $j$

Dessa forma,  $E_4 - E_3 B^{-1} C$  se reduz a

$$E_5 = E_4 - E_3 B^{-1} C = \begin{bmatrix} I & 0 & 0 \\ -B^j C & & \\ 0 & 0 & I \end{bmatrix}, \text{ pois } E_3 B^{-1} C = \begin{bmatrix} 0 & 0 & 0 \\ B^j C & & \\ 0 & 0 & 0 \end{bmatrix}$$

e portanto  $Q_{i+1}^{-1} = E_5 Q_i^{-1}$ . Agora a coluna que deixa a base se tornou uma coluna não-chave e o caso 1 pode ser aplicado.

- *Sub-caso 2b*:  $\gamma = 0$ . Neste caso, a matriz  $Q_{i+1}^{-1}$  será igual a matriz  $Q_i^{-1}$  e a variável que entra na base se torna variável chave.

### 3.2- TÉCNICA DE PENALIDADE

Nesta seção, as restrições do problema (3.1) são convenientemente manipuladas com objetivo de transformar um problema de fluxo em redes com restrições adicionais em um problema de fluxo em redes puro, para o qual existem eficientes algoritmos de resolução.

No problema (3.1), as matrizes de restrições definem dois conjuntos:

$$\begin{aligned} X &= \{x | Ax = b, \underline{x} \leq x \leq \bar{x}\}, e \\ \Omega &= \{x | x \in X, Sx \leq d\}. \end{aligned} \tag{3.14}$$

O conjunto  $X$  representa as soluções para o problema relaxando-se as restrições lineares adicionais, enquanto que  $\Omega$  considera as soluções de  $X$  que respeitam as restrições lineares adicionais.

Três etapas importantes são cumpridas durante a implementação desta técnica de penalidade, como discutido a seguir. Considerando as restrições adicionais através de uma técnica de penalidade, o problema pode ser resolvido como um problema de fluxo em redes puro (1ª etapa). Uma possível penalidade linear para a função objetivo teria a forma

$$p_j(\alpha_j) = \max\{0, \alpha_j\}, \tag{3.15}$$

onde  $\alpha_j = S_j x - d_j$ , e  $j = 1, \dots, s$ . Todavia, a função penalidade (3.15) apresenta problemas de descontinuidade. Para evitar as dificuldades de não-diferenciabilidade, Zenios (1995) sugere o uso de uma penalidade suave (2ª etapa). Tendo em vista (3.15), uma função penalidade linear-quadrática suave pode ser definida como (veja figura 3.2)

$$P(\varepsilon, \alpha) = \begin{cases} 0, & \text{se } \alpha \leq 0 \\ \alpha^2 / 2\varepsilon, & \text{se } 0 \leq \alpha \leq \varepsilon \\ \alpha - \varepsilon / 2, & \text{se } \alpha \geq \varepsilon \end{cases} \quad (3.16)$$

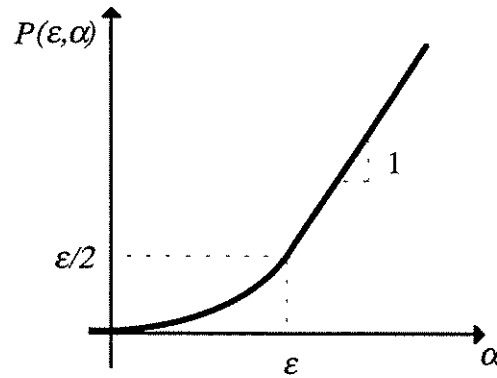


Figura 3.2: Função de penalidade  $P(\varepsilon, \alpha)$

A função objetivo modificada para o problema (3.1) será

$$c^T x + \mu \sum_{j=1}^s P(\varepsilon, \alpha_j)$$

onde:  $\mu$  e  $\varepsilon$  são números reais positivos e  $\alpha_j = S_j x - d_j$ , para  $j = 1, \dots, s$ , e o problema (3.1) se torna:

$$\text{Min}_{x \in X} c^T x + \mu \sum_{j=1}^s P(\varepsilon, \alpha_j). \quad (3.17)$$

A seguir é apresentado um algoritmo para solução de problemas em que a função de penalidade  $P(\varepsilon, \alpha)$  é usada para eliminar restrições adicionais. Zenios *et al.* (1995) apresenta detalhes sobre a convergência do algoritmo.

### 3.2.1 - ALGORITMO DE PENALIDADE LINEAR-QUADRÁTICA (PLQ)

Passo 0: (Inicialização) Faça  $k \leftarrow 0$  e seja  $x^0$  a solução ótima do problema relaxado

$$\begin{aligned} & \text{Min } c^T x \\ & \text{s.a: } Ax = b \\ & \quad \underline{x} \leq x \leq \bar{x} \end{aligned} \tag{3.18}$$

Escolha  $\mu^0 > 0, \varepsilon^0 > 0$ .

Passo 1: Se  $x^k \in \Omega$ , fim. Senão, faça  $k \leftarrow k + 1$  e atualize os parâmetros de penalidade  $\mu$  e  $\varepsilon$ .

Passo 2: Usando a violação  $\alpha_j = S_j x - d_j$  para todo  $j = 1, \dots, s$ , resolva (3.17). Retorne ao Passo 1.

Note que no passo 1 sempre vai existir uma solução  $x^k \in \Omega$  pois alguns produtos podem não ser produzidos. O maior esforço computacional deste algoritmo é a resolução do problema (3.17) no Passo 2, por ser uma função objetivo não linear e não separável. Como a matriz de incidência nó-arco  $A$  é bloco diagonal por produtos (veja 2.17), é cumprida a terceira e última etapa: resolver o problema (3.17) utilizando uma técnica de linearização, tal como o algoritmo de linearização por Decomposição Simplicial (descrito detalhadamente na próxima seção), pois desta forma se teria  $K$  problemas separáveis e independentes por produto em cada resolução do problema (3.17). A figura 3.3 apresenta o fluxograma do algoritmo de penalidade linear-quadrática (PLQ), e na seção 3.2.3 é discutida a inicialização e atualização dos parâmetros de penalidade.

### 3.2.2 - TÉCNICA DE LINEARIZAÇÃO POR DECOMPOSIÇÃO SIMPLICIAL

Nesta seção é apresentado o algoritmo de decomposição simplicial (DS), proposto por Von Hohenbalken (1977), e que tem despertado muito interesse durante esta última década. Este interesse é motivado por sua simplicidade, robustez, e habilidade em obter

boas aproximações para problemas não-lineares de grande porte (Mulvey *et al.*, 1990). O algoritmo interage entre a resolução de um sub-problema linearizado e de um problema mestre (*master problem*) não-linear com restrições simples de não-negatividade.

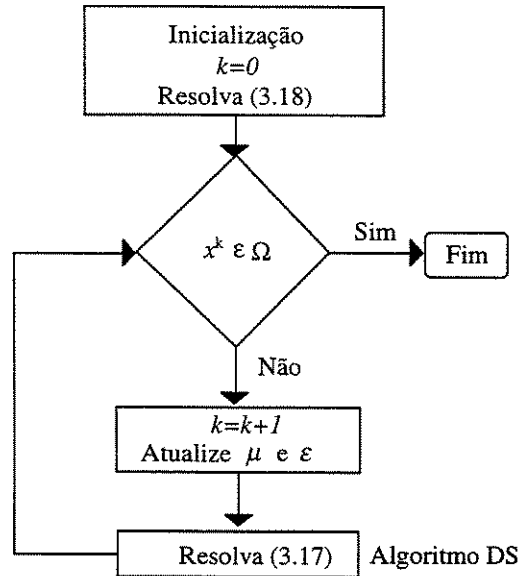


Figura 3.3: Fluxograma do algoritmo PLQ.

Considere o seguinte problema de programação não-linear:

$$\text{Min}_{x \in X} f(x)$$

onde  $X \equiv \{x \in \mathbb{R}^n \mid Ax \leq b, \underline{x} \leq x \leq \bar{x}\}$ ,  $A$  é uma matrix  $n_1 \times n$ ,  $b$  é um vetor em  $\mathbb{R}^{n_1}$ , e  $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$  é uma função convexa não-linear e continuamente diferenciável. Inicialmente, assume-se que a região de factibilidade, denotada por  $X$ , é fechada. Com esta suposição, qualquer elemento de  $X$  pode ser escrito como uma combinação convexa de seus pontos extremos.

A seguinte notação será utilizada:

- $Y^k = \{y^1, y^2, \dots, y^s\}$  é o conjunto dos vértices gerados de  $X$  na  $k$ -ésima iteração, de cardinalidade  $s$ . Os elementos de  $Y^k$  são linearmente independentes, e definem um simplex de dimensão  $(s-1)$ ;

- $D^k = \{y^1 - y^s, y^2 - y^s, \dots, y^{s-1} - y^s\}$  é a base linear relativa a  $Y^k$ ;
- $w^k = \{w_1, w_2, \dots, w_{s-1}\}$  é o vetor de pesos relativos ao conjunto  $D^k$ .

O subproblema linearizado é definido como

$$\text{Min}_{y \in X} \nabla f(x^k)^T y \quad (3.19)$$

onde  $x^k$  é a solução na  $k$ -ésima iteração e  $\nabla f(x^k)$  o gradiente de  $f(x)$  em  $x = x^k$ . Cada iteração do algoritmo gera um ponto  $y^k \in X$  tal que seu conjunto  $Y^k = \{y^1, y^2, \dots, y^s\}$  contenha  $s$  elementos linearmente independentes. Dada esta base  $Y^k$ , a base relativa  $D^k = \{y^1 - y^s, y^2 - y^s, \dots, y^{s-1} - y^s\}$  é utilizada para se definir o seguinte problema mestre:

$$\text{Min}_w f(D^k w + y^s),$$

onde  $D^k w + y^s = x^k$  e  $w$  é o vetor dos pesos relativos ao conjunto  $D^k$ . Se uma solução do problema mestre resulta em uma componente  $w_i = 0$ , a coluna do conjunto  $D^k$  associada a este  $w_i$  é retirada do conjunto. A solução  $x^k$  de cada iteração será portanto uma combinação positiva dos elementos pertencentes do conjunto  $D^k$ . Além disso,  $x^k$  satisfaz  $f(x^k) < f(x^{k-1})$  e será utilizado na próxima iteração do subproblema linearizado. Este procedimento termina quando a solução  $y^*$  de uma iteração do subproblema linearizado satisfaz o critério de otimalidade  $\nabla f(x^k)^T (y^* - x^k) \geq 0$ . O algoritmo de decomposição simplicial é apresentado a seguir.

**Passo 0 (Inicialização):** Seja  $x^1 \in X$  uma solução factível. Faça:  $Y^1 = \{x^1\}$ ;  $k = 1$  e  $s = 1$ .

**Passo 1 (Subproblema linearizado):** Determine a solução ótima  $y^*$  para o problema:

$$\begin{aligned}
& \text{Min } \nabla f(x^k)^T y \\
& \text{s.a: } Ay = b \\
& \quad \underline{x} \leq y \leq \bar{x}
\end{aligned} \tag{3.20}$$

Se  $\nabla f(x^k)^T (y^* - x^k) \geq 0$ , FIM;  $x^k$  é a solução ótima.

Senão, faça  $Y^k \leftarrow Y^k \cup \{y^*\}$ ,  $s = s + 1$ ,  $y^s = y^*$  e determine  $D^k$ .

Passo 2 (Problema mestre não-linear): Determine a solução ótima  $w^*$  do problema:

$$\text{Min } f(D^k w + y^s) \tag{3.21}$$

onde  $w_i \geq 0$ ,  $i = 1, 2, \dots, s-1$ .

A seguir, faça  $w_s = 1 - \sum_{i=1}^{s-1} w_i^* \geq 0$

$$Y^k \leftarrow Y^k - \{y^j\}, s = s - 1, \forall w_j = 0, j = 1, \dots, s$$

$$x^{k+1} = \sum_{i=1}^s w_i y^i, e k = k + 1.$$

Retorne ao Passo 1.

A figura 3.4 apresenta o fluxograma do algoritmo de decomposição simplicial. O Passo 1 deste algoritmo utiliza uma aproximação linear para o problema não-linear, e pode ser facilmente resolvido por pacotes de programação linear. O problema mestre não-linear do Passo 2 é de tamanho muito menor do que o problema original e possui uma estrutura muito mais simples podendo ser resolvido por um pacote de programação não-linear, ou pelo procedimento descrito a seguir.

Utilizando-se a regra de diferenciação em cadeia para a função objetivo do problema mestre, tem-se

$$\begin{aligned}
\nabla_w f(D^k w + y^s) &= \nabla_x f(D^k w + y^s) D^k \\
\nabla_w^2 f(D^k w + y^s) &= (D^k)^T \nabla_x^2 f(D^k w + y^s) D^k
\end{aligned}$$



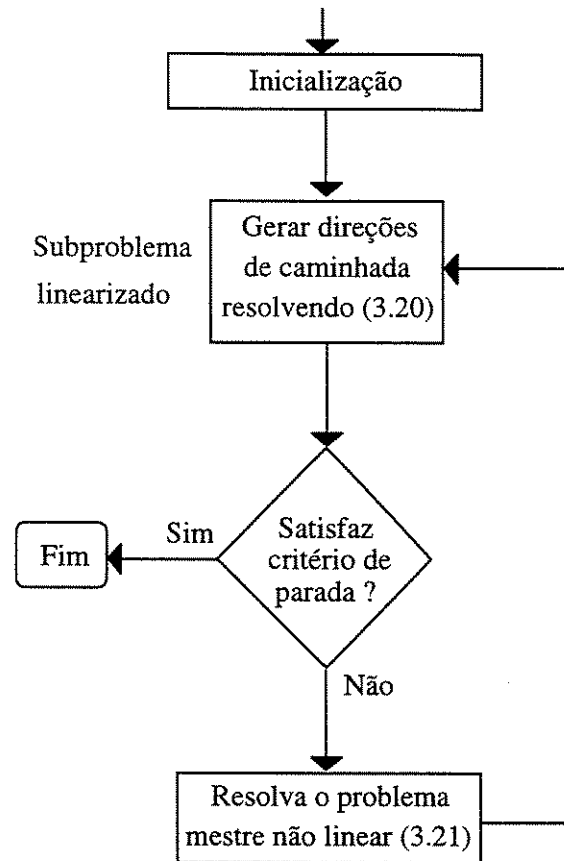


Figura 3.4: Fluxograma do algoritmo DS.

Estas relações permitem a projeção do gradiente e da Hessiana da função objetivo no espaço do simplex definido por  $D^k$ . O procedimento para a resolução do problema mestre envolve a utilização de uma matriz de correção  $C$  (Von Hohenbalken, 1977). Esta matriz pode ser definida de várias maneiras, como discutido adiante, resultando em um número de algoritmos bem conhecidos de otimização irrestrita. O procedimento para a resolução deste problema é dado a seguir.

1. cálculo da matriz de correção:  $C^k = D^k M^k D^k$ .
2. cálculo da direção  $p^k$ :  $C^k p^k = -D^k \nabla_x f(x^k)$ .
3. busca unidimensional:  $\text{Min}_{0 \leq \alpha \leq \alpha_{MAX}} f(D^k(w^k + \alpha p^k) + y^s)$ , onde  $\alpha_{MAX}$  é o passo máximo neste simplex.

4. atualização de  $w^k$ :  $w^k \leftarrow w^k + \alpha p^k$ .

A escolha da matriz  $M$  pode afetar significativamente o desempenho do procedimento de resolução do problema mestre. A seguir são dadas três alternativas para se definir  $M$  (Luenberger, (1984) e Zenios, 1995).

- algoritmo de Gradiente Projetado:  $M \equiv I$
- algoritmo de Newton:  $M \equiv \nabla_x^2 f(D^k w^k + y^s)$  (iteração corrente)
- algoritmo Quasi-Newton:  $M \equiv \nabla_x^2 f(D^{k-1} w^{k-1} + y^s)$  (iteração anterior)

Nos testes realizados na próxima seção, foi escolhido utilizar  $M \equiv I$ .

Apesar deste problema mestre ser de dimensão muito menor do que o problema original, o tamanho do conjunto  $Y^k$  pode complicar a resolução do problema. Para contornar esta dificuldade, uma versão restrita deste método foi proposta em Hearn *et al.* (1987) e Mulvey *et al.* (1990). Tal método propõe um truncamento no tamanho do problema mestre por um parâmetro  $r$ , que limita o número de pontos do conjunto  $Y^k$ . Um valor razoável para este truncamento e utilizado nos experimentos a seguir é  $r = 20$ . Quando  $r = 1$ , o método se reduz ao algoritmo de Frank e Wolfe, e quando  $r$  é irrestrito obtém-se o algoritmo DS descrito neste seção.

Como exemplo ilustrativo, considere o problema onde

$$f(x) = (x_1 - 14)^2 + (x_2 - 15)^2$$

$$\begin{bmatrix} 2 & -5 \\ 2 & -1 \\ -3 & 1 \\ -1 & 10 \\ 2 & 6 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 10 \\ 18 \\ 6 \\ 118 \\ 102 \\ 0 \\ 0 \end{bmatrix}$$

A tabela 3.1 resume as iterações do algoritmo de decomposição simplicial necessárias para a resolução do exemplo. Na figura 3.5 a seguir, pode-se ver a região de factibilidade e a sequência dos pontos gerados pelo algoritmo.

Tabela 3.1: Iterações do exemplo.

$k$	$x$	$y^*$	$Y$	$w$	$\nabla f(x^k)^T(y^* - x^k)$	$f$
1	(2,12)	(15,12)	[(2,12); (15,12)]	(1/13; 12/13)	-312,00	153,0
2	(14,12)	(12,13)	[(2,12); (15,12); (12,13)]	(0; 2/3; 1/3)	-6,00	9,0
3	(14,37/3)	(12,13)	[(15,12); (12,13)]	(2/5; 3/5)	-3,55	7,1
4	(13,2,12,6)	(15,12)	-----		0,00	6,4

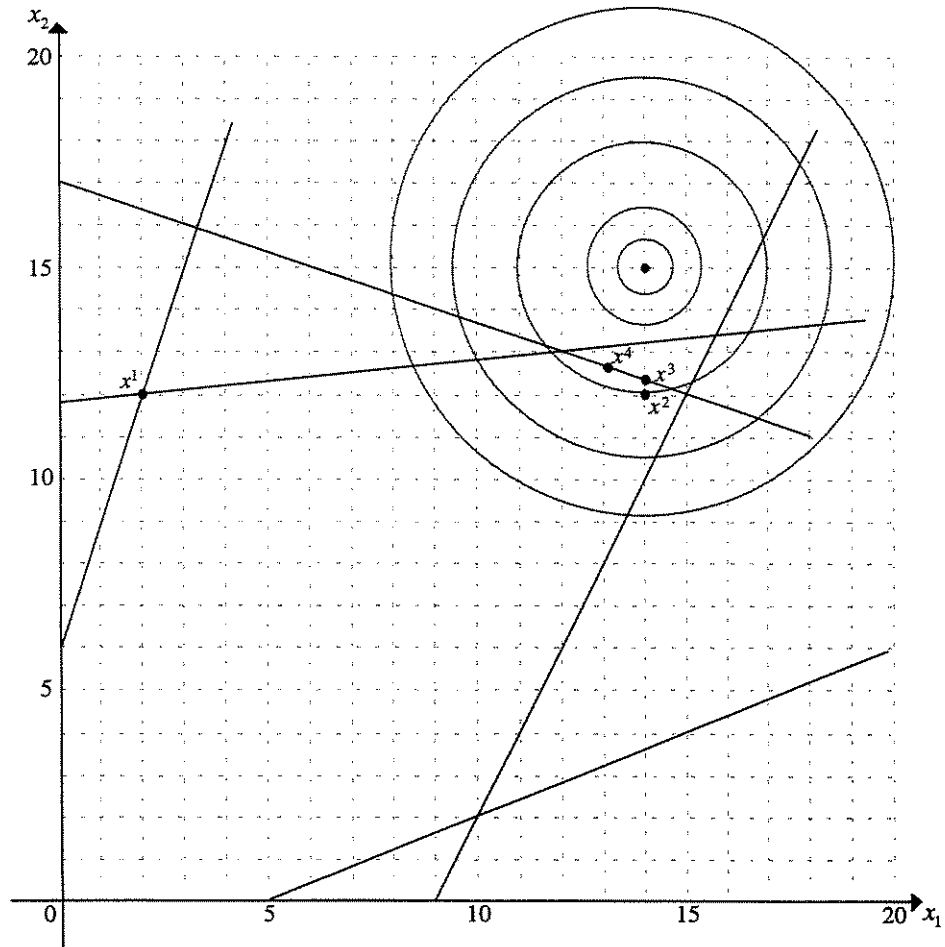


Figura 3.5: Sequência dos pontos gerados

Na iteração  $k = 1$ , o conjunto  $Y$  é formado por dois pontos extremos:  $y^1 = (2, 12)$ , solução inicial, e  $y^2 = (15, 12)$ , solução do sub-problema linearizado. A solução do problema mestre não-linear  $w^1 = (1/13; 12/13)$  resulta no ponto  $x^2 = (14, 12)$ . Na iteração  $k = 2$ , a solução do sub-problema linearizado resulta no ponto  $y^3 = (12, 13)$ , e o conjunto  $Y$  passa a ter três elementos. A nova solução do problema mestre não-linear  $w^2 = (0; 2/3; 1/3)$  faz com que o ponto  $y^1 = (2, 12)$  seja retirado do conjunto  $Y$  (note que  $w^1 = 0$ ). A partir de  $w^2$  e dos dois pontos de  $Y$ , encontra-se o novo ponto  $x^3 = (14, 37/3)$ , e assim sucessivamente.

### 3.2.2 - PARÂMETROS DE PENALIDADE

Neste seção é discutido o esquema de inicialização e atualização dos parâmetros de penalidade  $\mu$  e  $\varepsilon$ . Suponha que  $x^k \in X$ ,  $\mu^k$ ,  $\varepsilon^k$  e  $a^k = Sx^k - d$  são conhecidos na  $k$ -ésima iteração do algoritmo PLQ. A atualização dos parâmetros de penalidade  $\mu$  e  $\varepsilon$  é feita considerando-se dois casos:

- Caso 1: Se  $0 < \alpha_j \leq \varepsilon$  para todo  $j = 1, \dots, s$ , (solução  $\varepsilon$ -factível), há uma indicação de que o parâmetro  $\mu$  foi adequado na última iteração e então o valor de  $\varepsilon$  deve ser reduzido de acordo com

$$\varepsilon^{k+1} = \max(\varepsilon_{\min}, \eta \varepsilon^k)$$

onde  $\varepsilon_{\min}$  é a tolerância mínima para  $\varepsilon$  e  $\eta \in (0, 1]$ .

- Caso 2: Se  $\alpha_j > \varepsilon$  para algum  $j = 1, \dots, s$ , há uma indicação de que o parâmetro  $\mu$  deve ser aumentado. Uma maneira possível é aumentar proporcionalmente ao grau de infactibilidade. Considera-se a seguinte equação:

$$\mu^{k+1} = \frac{\mu^k}{\eta \varepsilon^k} \max(\alpha_j).$$

Uma estratégia razoável para a inicialização de  $\mu$  (estabelecida depois de algumas experimentações práticas), é utilizar o maior valor dos coeficientes da função objetivo. A inicialização de  $\varepsilon$  é obtida da solução  $x^0$  do problema relaxado, escolhendo-se um valor igual a uma fração da máxima violação das restrições adicionais. Em todos os testes realizados na próxima seção, foram tomados os seguintes valores:  $\mu = 0,5$  e  $\varepsilon_{min} = 10^{-2}$ .

No modelo proposto na seção anterior, ao se considerar as restrições de acoplamento (2.14) e de produção simultânea (2.15) através de uma função de penalidade, a função objetivo (2.18) torna-se

$$\Phi_{\mu,\varepsilon}(\cdot) = f(\cdot) + \mu_1 \sum_{j=1}^{s_1} \phi_q(\varepsilon_1, \alpha_1) + \mu_2 \sum_{j=1}^{s_2} \phi_p(\varepsilon_2, \alpha_2) \quad (3.22)$$

onde  $\mu_1$ ,  $\mu_2$ ,  $\varepsilon_1$  e  $\varepsilon_2$  são parâmetros de penalidades,  $s_1$  é o número de restrições de acoplamento com  $\alpha_1$  dado por:

$$\alpha_1 = x(\ell, t, k) - q(\ell, k)x(i, t, k), \quad t = 1, 2, \dots, T, \quad k = 1, 2, \dots, K$$

e  $s_2$  é o número de restrições de capacidade mútua com  $\alpha_2$  dado por:

$$\alpha_2 = \sum_{i \in J} x(i, t, k) / p(i, t, k) - TD(m, t), \quad t = 1, 2, \dots, T, \quad k = 1, 2, \dots, K.$$

### 3.3. - APLICAÇÃO

Para a aplicação serão utilizados dados de uma empresa de molas, encontrada no parque industrial brasileiro, que tem um cliente preferencial. Esta empresa trabalha como uma célula autônoma e deve suprir os requisitos de demanda estabelecidos por seu cliente preferencial. O caso aqui estudado considera 73 diferentes tipos de produtos que podem ser agregados em 68 famílias de peças de acordo com a sequência e/ou características de produção. A linha mais extensa tem 5 estágios de produção e o objetivo é estabelecer as necessidades de produção para cada semana, bem como o estoque inter-semanas. A

função objetivo utilizada reúne os custos de armazenagem, de atendimento à demanda em atraso, e de produção. O problema de planejamento consiste em resolver

$$\text{Min} \left\{ \sum_{i=1}^{I+1} \sum_{t=1}^T c_a(i,t,k)y(i,t,k) + \sum_{t=0}^{T-1} c_b(t,k)z(t,k) + \sum_{i=1}^I \sum_{t=1}^T c_p(i,t,k)x(i,t,k) \right\} \quad (3.23)$$

sujeito as restrições discutidas previamente no capítulo anterior.

Com a finalidade de demonstrar a capacidade do enfoque, o tipo de solução obtida e comparar o desempenho das técnicas citadas na seção anterior, alguns exemplos serão apresentados. Inicialmente tem-se um exemplo simplificado (ex02) com apenas dois produtos (PROD00087 e PROD00088) a serem produzidos em uma linha com 4 estágios de produção. O período de planejamento é de duas semanas, perfazendo um total de 40 horas por semana. Os testes foram realizados em um microcomputador Pentium 200 Mhz.

Na figura 3.6, que ilustra a representação por grafo deste exemplo, são apresentadas as demandas de cada produto para duas semanas do planejamento. Os arcos em paralelo, dentro do mesmo estágio de produção, representam máquinas alternativas para execução da tarefa. As taxas e custos de produção para cada máquina em seu respectivo estágio de produção, além da disponibilidade de máquina para cada semana, são fornecidos na tabela 3.2. O custo de armazenagem e a penalidade por atendimento à demanda em atraso de qualquer produto são assumidos constantes e iguais a \$ 1,00.

A figura 3.7 e a tabela 3.3 mostram os resultados obtidos pela técnica das penalidades. Note que devido à limitação da capacidade de produção (na tabela 3.3, o estágio dois de produção, composto pelas máquinas M15 e M19, está com capacidade saturada) houve atendimento de demanda em atraso para os dois produtos. Ou seja, a semana dois produziu 3063 peças do produto PROD00087 e 1178 peças do produto PROD00088 para atender a demanda da primeira semana.

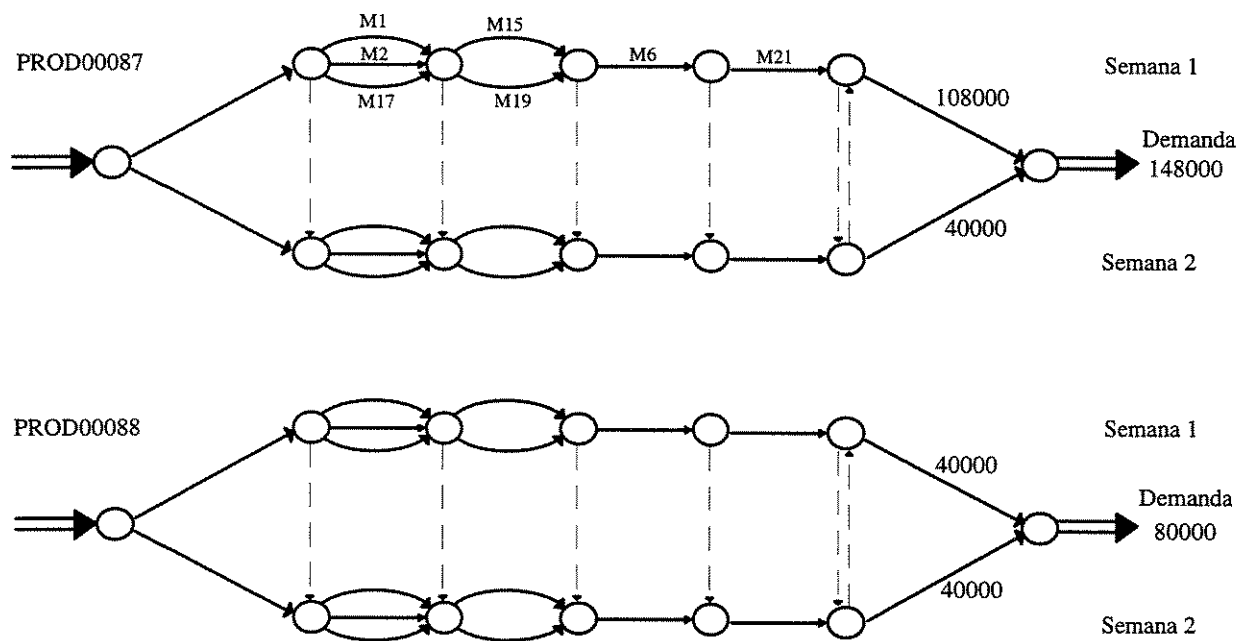


Figura 3.6: Representação por grafo do exemplo.

Tabela 3.2: Dados de produção.

estágio	máquina	PROD00087		PROD00088		disponibilidade de máquina (% por semana)
		taxa de produção (peças/semana)	custo de produção (\$/peça)	taxa de produção (peças/semana)	custo de produção (\$/peça)	
1	M1	144000	1,00	144000	1,00	100
1	M2	48000	1,00	48000	1,00	100
1	M17	48000	1,00	48000	1,00	100
2	M15	96000	1,00	80000	1,00	100
2	M19	48000	1,00	48000	1,00	100
3	M6	400000	1,00	600000	1,00	100
4	M21	320000	1,00	192000	1,00	100

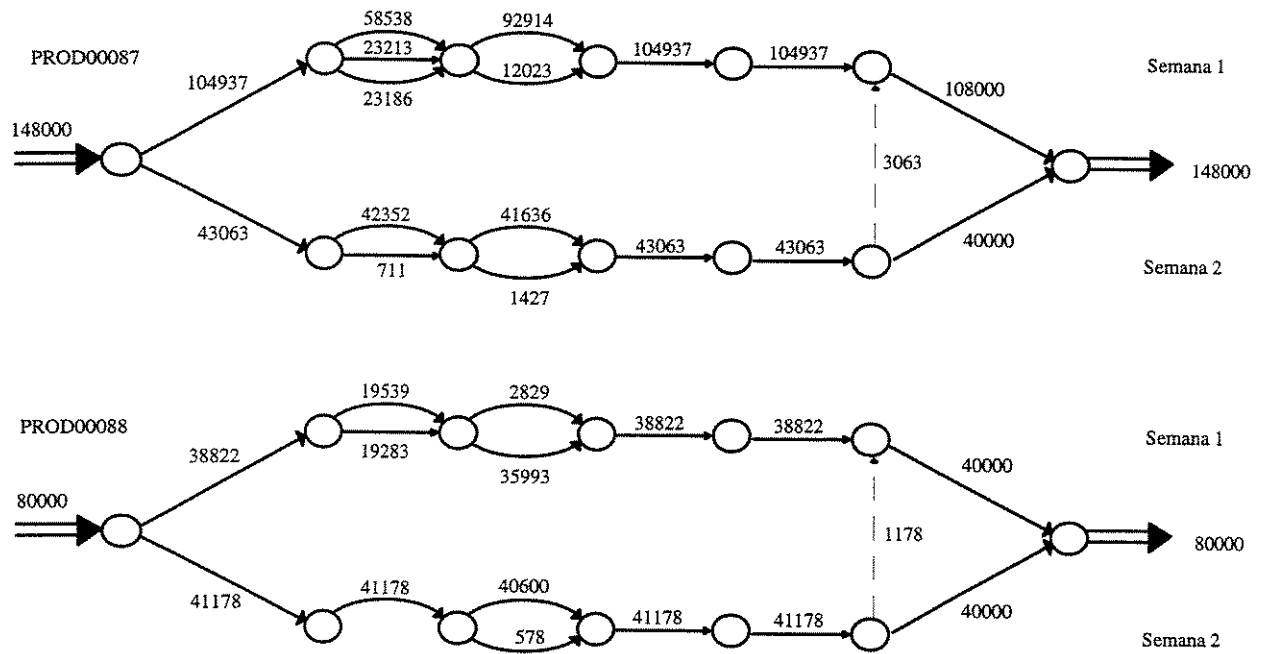


Figura 3.7: Resultados do exemplo (ex02).

Tabela 3.3: Resultados do exemplo.

	máquina	Utilização das máquinas (%)	
		Semana 1	Semana 2
FO = 916241	M1	54,22	58,01
$\mu = 240058,368$	M2	88,53	1,48
$\varepsilon = 0,0020616$	M17	48,30	0,00
200 iterações	M15	100,32	94,12
21 passos de penalidade	M19	100,03	4,18
CPU = 0,16 seg.	M6	32,70	17,63
	M21	53,01	34,90

Outra importante observação é que a técnica adotada não produz necessariamente valores inteiros. Entretanto, como os níveis de produção são valores consideravelmente elevados, uma aproximação desses valores reais por valores inteiros não implica em erro significativo. Este exemplo mostra que um importante resultado obtido com a aplicação



da programação matemática em sistemas reais de manufatura é a determinação da transferência da produção inter-períodos. No caso geral, a antecipação da produção para atender demanda futura pode acontecer por decisão econômica (custos de compra de matéria prima ou produção menores) ou de limitação de capacidade. Outra possibilidade seria o atendimento de demanda em atraso devido a insuficiência de capacidade de produção ou disponibilidade limitada de matéria prima.

Para efeito de comparação, além das técnicas aqui estudadas, técnica de decomposição primal (NETSIDE) e técnica de penalidade (aqui denominada RETRA REde de TRANsportes), utilizou-se também o GAMS/OSL, um pacote comercial para a resolução de problemas de programação matemática.

O exemplo que considera um sistema de produção real (ex68) com todos os produtos foi analisado para o planejamento de duas semanas. Este exemplo tem 609 nós e 1405 arcos. Os resultados estão apresentados na tabela 3.4. Para o RETRA foram necessários 11 passos de penalidade com um total de 16 iterações de decomposição simplicial, com um número máximo 3 de soluções armazenadas no algoritmo DS. A técnica de penalidade (RETRA) requer somente operações sobre grafos enquanto que as outras exigem a inversão e/ou atualização de matrizes. Pelo algoritmo RETRA, o número de iterações em grafos é muito maior, mas com custo computacional significativamente menor. A técnica de decomposição primal (NETSIDE) é um misto de iteração de grafo com operações matriciais e para o problema com linhas curtas, baixa densidade de restrições adicionais e grande esparsidade, caso típico de manufatura, apresenta bom comportamento.

Tabela 3.4: Resultados do exemplo completo.

Exemplo	Técnica	Função objetivo	Iterações	Tempo CPU (seg)
ex68	RETRA	1.871.699,00	2574	2,69
	NETSIDE	1.871.699,00	430	5,33
	GAMS	1.871.699,00	509	22,33

Ainda com a finalidade de comparar o desempenho das técnicas citadas, outros cenários foram estudados. A figura 3.8 ilustra os valores da função objetivo obtidos pelo RETRA para o exemplo ex68 e diferentes fatores de demanda (um multiplicador que se aplica a um determinado perfil de demanda e serve para avaliar o comportamento do sistema em diferentes situações de carga). A curva apresenta um ponto de quebra para o fator de demanda de 1,3. Isto significa a ocorrência de gargalos (saturação) relacionada a alguma(s) linha(s) de produção. Note que outras linhas encontram-se ainda em estado que possibilita o aumento de produção.

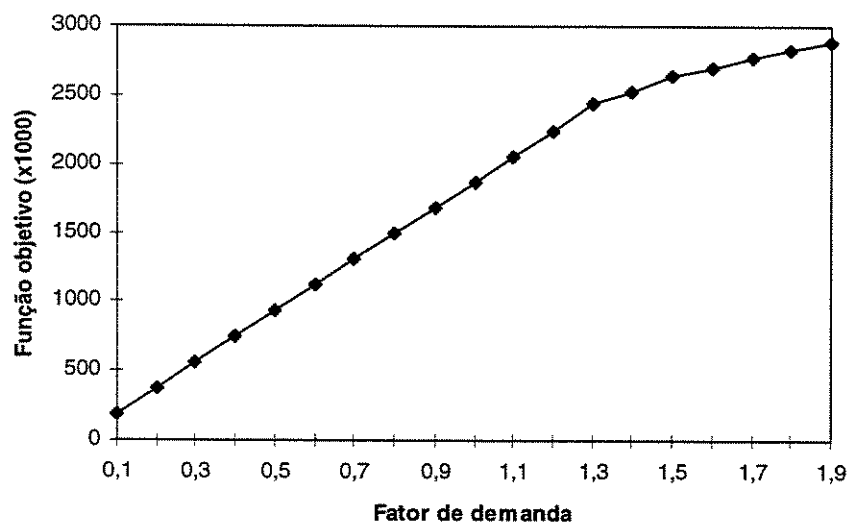


Figura 3.8: Função objetivo em função da demanda.

A figura 3.9 exibe o comportamento dos três algoritmos em função da variação nos fatores de demanda. O algoritmo RETRA apresenta o melhor desempenho. Há um aumento significativo do tempo de CPU para o sistema sobrecarregado, característica dos algoritmos de penalidade. Contudo níveis de saturação elevados devem ser raros em aplicações práticas, pois os gerentes conhecem as capacidades dos sistemas e evitam programações deste tipo. O algoritmo NETSIDE é bem mais rápido que o GAMS, e para os três algoritmos, o tempo de CPU não varia significativamente com o aumento da demanda antes da saturação.

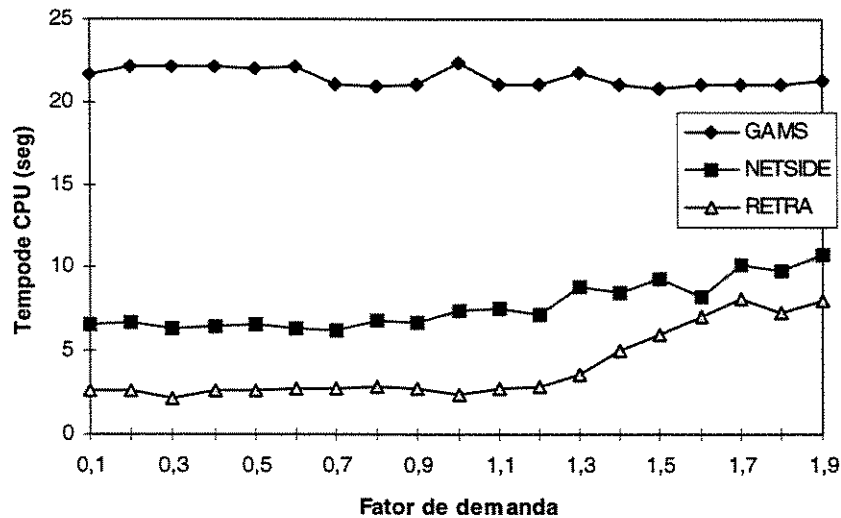


Figura 3.9: Desempenho em função da demanda.

A figura 3.10 estuda o efeito da variação do número de produtos. Com o aumento do número de produtos, o GAMS tem um aumento exponencial do tempo de CPU, como esperado para um pacote de programação linear. O NETSIDE e o RETRA apresentam um comportamento de aumento praticamente linear com o número de produtos. Com relação ao RETRA, a justificativa para o aumento *quase-linear*, se deve ao fato de que a adição de um novo produto  $k$  resulta na adição de uma nova sub-matriz  $A_k$  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  $k$  introduzido no sistema. No caso de sistemas de manufatura, os grafos de cada produto  $k$  têm dimensão razoavelmente iguais, o que também justifica este comportamento quase-linear. Esta é uma grande vantagem do algoritmo de penalidade quando utilizado em sistemas reais de manufatura com grande número de itens.

O mesmo não ocorre com o algoritmo de decomposição primal, pois um sistema de produção com um número grande de produtos e estágios conduz a uma matriz  $Q$  (equação 3.4) de dimensão significativamente grande. A inversão desta matriz consumiria tempo substancial de computação. Portanto o algoritmo de decomposição primal tem

desempenho ruim para redes de produção com muitos estágios, por aumentar significativamente o número de restrições adicionais.

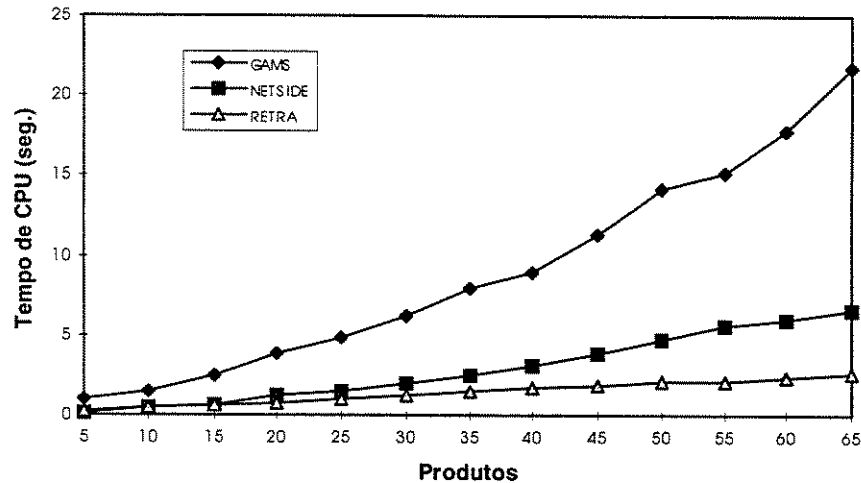


Figura 3.10: Desempenho em função do número de produtos.

### 3.4. - INICIALIZAÇÃO

Na resolução de problemas práticos pode-se tirar vantagem de características particulares e de experiência. No algoritmo de penalidade com inicialização baseada no método do M-grande da programação linear, verifica-se que a solução inicial tende a se concentrar nos arcos de produção e ocasionalmente em alguns dos arcos de armazenagem. Dessa forma pode-se sugerir um direcionamento da solução inicial e assim obter uma inicialização melhor, por heurística, agilizando o processo de obtenção da solução ótima do problema. Escolhendo-se então uma solução inicial onde apenas os arcos de produção formam a árvore ou base inicial, tem-se os resultados na tabela 3.5. Note que o direcionamento da solução efetivamente reduziu o número de iterações do algoritmo RETRA e conseqüentemente o tempo de CPU.

Tabela 3.5: Resultados com o direcionamento da solução.

Exemplo	Inicialização	FO	ITER	T. CPU (seg)
	M-grande	1.871.699	2574	2,69
ex68	Solução direcionada	1.871.699	609	1,86

## CAPÍTULO 4

### ABORDAGEM MULTIOBJETIVO

Este capítulo considera uma abordagem multiobjetivo do problema de planejamento da produção apresentado em (2.28). Inicia com a notação utilizada nesta seção. A seguir é apresentada a formulação e algumas considerações gerais sobre problemas multiobjetivos e uma breve descrição das técnicas de solução de problemas deste tipo que têm sido efetivamente utilizadas em aplicações reais descritas na literatura. Em seguida é proposta uma técnica de Programação Alvo que explore adequadamente a estrutura especial do problema de planejamento da produção. Por fim, uma aplicação a um exemplo real é discutida.

#### 4.1 - NOTAÇÃO

$x \in R^n$	vetor das variáveis de decisão
$X \subseteq R^n$	região de factibilidade
$Y \subseteq R^m$	conjunto de soluções eficientes
$m$	número de objetivos
$n$	número de variáveis
$s$	número de restrições adicionais
$A$	matrix de incidência nó-arco
$S$	matrix $s \times n$ das restrições lineares adicionais
$b$	coeficientes do lado direito das restrições

$M_a$	metas relativas às restrições adicionais
$M_f$	metas relativas aos $m$ objetivos
$\phi_0$	função linear dos desvios $\delta^- \geq 0$ e $\delta^+ \geq 0$ de cada uma das $s$ restrições adicionais
$\phi_i, i = 1, \dots, m$	funções lineares dos desvios $d^- \geq 0$ e $d^+ \geq 0$ associados aos objetivos
$d^-; d^+$	desvios associados aos objetivos ( $d^- \geq 0$ e $d^+ \geq 0$ )
$\delta^-; \delta^+$	desvios de cada uma das $s$ restrições adicionais ( $\delta^- \geq 0$ e $\delta^+ \geq 0$ )
$U: R^m \rightarrow R$	função de preferência (ou utilidade)
$w \in R^m$	pesos associados pelo decisor aos objetivos

#### 4.2 - FORMULAÇÃO DO PROBLEMA MULTIOBJETIVO

Um problema geral multiobjetivo pode ser formulado como

$$\underset{x \in X}{\text{Min}} f(x) \quad (4.1)$$

onde  $x \in R^n$  representa o vetor das variáveis de decisão,  $X \subseteq R^n$  a região de factibilidade e  $f(x) = [f_1(x) \ f_2(x) \ \dots \ f_m(x)]^T$  o vetor das funções objetivos ( $m \geq 2$ ).

Quando os objetivos são conflitantes, o que ocorre na maioria das aplicações de interesse prático, uma solução ótima para (4.1) no sentido tradicional é impossível de ser encontrada. Problemas multiobjetivos são caracterizados pelo conceito de solução eficiente, também referenciada como solução Pareto-ótima, não-dominada ou não-inferior. Uma solução  $x^* \in X$  é eficiente se não existe qualquer outra solução  $x \in X$  tal que  $f(x) \leq f(x^*)$  e  $f(x) \neq f(x^*)$  (Ferreira e Machado, 1996). Em palavras, uma solução é eficiente quando qualquer decréscimo no valor de um objetivo implica no acréscimo de pelo menos algum outro objetivo, como se pode ver na figura 4.1. Um

problema multiobjetivo pode apresentar infinitas soluções eficientes. Na figura 4.1,  $X^*$  é o conjunto de soluções eficientes relativo às duas funções. Em geral, a solução do problema é escolhida dentre o conjunto de soluções eficientes através de critérios subjetivos (ou preferências) de um decisor.

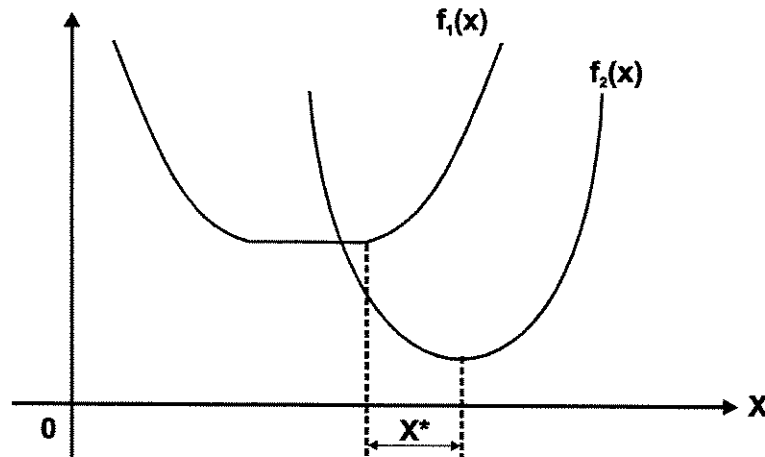


Figura 4.1: Interpretação de soluções eficientes.

Inúmeras técnicas de programação multiobjetivo têm sido desenvolvidas (Cohon, 1978 e Hwang e Masud, 1979). Cada uma destas técnicas requer uma indicação explícita ou implícita das preferências do decisor, e são classificadas como:

- *a-priori* - o decisor indica suas preferências com relação aos objetivos antes que, de fato, o problema seja resolvido;
- *a-posteriori* - o decisor seleciona a solução de sua preferência após a geração de um subconjunto representativo de soluções eficientes; e
- *iterativos* - o decisor interage com o modelo de otimização avaliando um número limitado de soluções até convergir para a solução do problema.

O uso de ponderações (técnica *a-posteriori*) é tradicional em programação multiobjetivo e foi citado nos trabalhos de Taboun e Ulger (1992) e Das e Sarin (1994) aplicados a problemas de planejamento da produção. A idéia desta técnica é variar parametricamente os coeficientes que ponderam as funções objetivos e assim obter



soluções eficientes do problema. Associando-se pesos (ou preferências) aos objetivos obtém-se um problema com uma única função objetivo, como será visto na seção 4.2.1.

Outra técnica a-posteriori foi utilizada no trabalho de Gallagher e Saleh (1994). Neste trabalho, para um problema de transporte linear multiobjetivo, em vez de se gerar o conjunto de soluções eficientes  $X^*$ , constrói-se um conjunto  $Y$  das suas projeções no espaço dos objetivos. Dado que o número de objetivos é tipicamente muito menor do que o número de variáveis do problema, o conjunto  $Y$  tem significativamente menos faces e pontos extremos do que o conjunto  $X^*$  (Dauer, 1987). Além disto o decisor é influenciado grandemente por considerações formuladas no espaço dos objetivos (custos, número de peças, etc) em vez de considerações no espaço das variáveis de decisão. Portanto uma análise no espaço dos objetivos é potencialmente mais fácil e pode proporcionar melhores resultados do que uma análise no espaço das variáveis de decisão.

Uma das técnicas a-priori, a Programação Alvo, tem sido efetivamente utilizada em problemas de planejamento de produção. Lee e Jung (1989), Dean *et al.* (1990) e Kumar *et al.* (1991) utilizaram esta técnica em problemas de planejamento da produção de um FMS (*Flexible Manufacturing System*) e Galbraith e Miller (1989) em um problema de sequenciamento da produção para uma linha de montagem de circuitos impressos. Na seção 4.3, a Programação Alvo será utilizada para a resolução do problema de planejamento da produção formulado no capítulo 2.

Como técnica interativa podem-se citar os trabalhos de Malakooti (1991) e Lewis (1992) aplicados a problemas de planejamento da produção. Na seção 4.5 será feita uma abordagem interativa utilizando a programação alvo.

Outras estratégias ou abordagens foram utilizadas em vários trabalhos, como está resumido a seguir.

- Métodos Heurísticos:

- \* Norbis e MacGregor Smith (1988) - heurística de níveis de prioridades para o problema de sequenciamento da produção;
- \* Ogryczak (1989) - problema de transporte multiobjetivo;

- \* Ramesh e Cary (1989) - *jobshop scheduling*;
- \* Malakooti e Deviprasad (1989) - planejamento da manufatura (corte de metais).
- Métodos Hierárquicos:
  - \* Trzaskalik (1990) - planejamento da produção;
  - \* Wu e Tabucanon (1990) - gerenciamento da produção - sistema de suporte a produção.
- Métodos Min-Max:
  - \* Singh *et al.* (1990) - sequenciamento da produção para FMS;
  - \* Kumar *et al.* (1990) - problemas de carregamento em FMS.
- Métodos de Decomposição:
  - \* Yang *et al.* (1988) - planejamento da produção em uma refinaria de petróleo;
  - \* Malakooti (1989) - sistemas integrados de manufatura;
  - \* Mohanty e Singh (1992) - planejamento da produção em um sistema de manufatura de aço.

#### 4.2.1 - O PROBLEMA PONDERADO

O problema ponderado associado a (4.1) é formulado como

$$\text{Min}_{x \in X} \sum_{i=1}^m w_i f_i(x) \quad (4.2)$$

onde  $w_i \geq 0$ ;  $\sum_{i=1}^m w_i = 1$ .

No capítulo 3, a função objetivo em (3.23) pode ser interpretada como uma soma ponderada de funções com pesos  $w^1 = w^2 = w^3 = 1/3$ . A função em (3.23) representa os custos de armazenagem, de atendimento à demanda em atraso, e de produção. A

utilização do método de ponderações apresenta como desvantagem a necessidade de se associar custos às funções objetivos, os quais podem ser difíceis de estimar na prática (Lewis *et al.*, 1992).

#### 4.3 - PROGRAMAÇÃO ALVO

A Programação Alvo (PA) fornece soluções relativamente precisas para problemas reais de manufatura consumindo tempos pequenos de CPU (Dean *et al.*, 1990). A idéia básica desta técnica é procurar minimizar desvios em relação a metas pré-estabelecidas para os objetivos, o que a classifica como uma técnica de indicação de preferências a-priori. Uma formulação multiobjetivo do problema de planejamento da produção descrito no capítulo 2 seria

$$\begin{aligned}
 & \text{Min}[f_1(x) \ f_2(x) \dots f_m(x)]^T \\
 \text{s.a.:} \quad & Ax = b \\
 & Sx \leq M_a \\
 & \underline{x} \leq x \leq \bar{x}
 \end{aligned} \tag{4.3}$$

onde:  $x \in R^n$  são variáveis de decisão que representam o fluxo nos arcos,  $A$  é a matrix de incidência nó-arco,  $S$  é a matrix  $s \times n$  das restrições lineares adicionais, e  $b$  e  $M_a$  são os coeficientes do lado direito das restrições.

Em aplicações reais da manufatura procura-se satisfazer a demanda de acordo com as limitações dos recursos de produção. Uma saída natural seria portanto fazer com que as restrições de grafo  $Ax = b$  sejam satisfeitas inicialmente priorizando a factibilização das restrições adicionais  $Sx \leq M_a$  e a seguir minimizar desvios de cada uma das funções  $f_1(x), f_2(x), \dots, f_m(x)$  em relação a metas pré-estabelecidas. A representação clássica da programação alvo para este problema é

$$\begin{aligned}
& \text{Min Lex } \{ \phi_0(\delta^+, \delta^-), \phi_1(d_1^+, d_1^-), \dots, \phi_m(d_m^+, d_m^-) \} \\
& \text{s.a.: } Ax = b \\
& \quad Sx + \delta^- - \delta^+ = M_a \\
& \quad f(x) + d^- - d^+ = M_f \\
& \quad \underline{x} \leq x \leq \bar{x} \\
& \quad \delta^-, \delta^+, d^-, d^+ \geq 0; \delta^- \delta^+ = 0; d^- d^+ = 0
\end{aligned} \tag{4.4}$$

onde  $M_a$  é o vetor das metas relativas às restrições adicionais,  $M_f$  é o vetor das metas relativas aos  $m$  objetivos,  $\phi_0$  é uma função linear dos desvios  $\delta^- \geq 0$  e  $\delta^+ \geq 0$  de cada uma das  $s$  restrições adicionais e  $\phi_i, i = 1, \dots, m$  são funções lineares dos desvios  $d^- \geq 0$  e  $d^+ \geq 0$  associados aos objetivos. A minimização indicada é do tipo lexicográfica, isto é, minimiza-se sequencialmente as funções de desvios sujeitas a que os valores das funções de desvios precedentes não sejam alterados.

O método simplex pode ser utilizado de maneira extremamente eficiente para resolver problemas de PA. Entretanto, como os objetivos não possuem uma estrutura particular, ao serem incorporados como restrições em modelos de PA, o problema perde a estrutura de grafo, mesmo que não existam restrições adicionais (de acoplamento e de capacidade mútua das máquinas). Um pacote genérico de programação linear não explora adequadamente a estrutura particular do problema, o que é fundamental para que o problema possa ser resolvido eficientemente através de uma técnica de otimização multiobjetivo como a programação alvo. Em princípio, a técnica de penalidades discutida no capítulo anterior também poderia ser utilizada para implementar o modelo de programação alvo. Entretanto, uma característica fundamental da programação alvo é a minimização sequencial de funções desvio lineares a partir do *tableaux* que representa todas as soluções minimizantes das funções de desvio anteriores. A escolha de um algoritmo para a resolução de problemas de grafo com restrições adicionais, como o NETSIDE (Kennington e Whisman, 1990), é bastante natural neste contexto. Além disso, como o número de objetivos é pequeno (tipicamente,  $2 \leq m \leq 5$ , Dauer (1987)), o uso de programação alvo não aumenta significativamente o número de restrições adicionais.

#### 4.4 - APLICAÇÃO

Com a finalidade de ilustrar o procedimento proposto na seção anterior serão utilizados os dados do exemplo descrito na seção 3.3 (veja figura 3.6 e tabela 3.2). O exemplo foi rodado em um microcomputador Pentium 200 MHz. A ordem dos objetivos será a seguinte:

- $\phi_0$ : minimização das violações das restrições adicionais (facilização);
- $\phi_1$ : minimização dos níveis de armazenagem (y);
- $\phi_2$ : minimização dos custos de produção e de aquisição da matéria prima.

O conjunto de objetivos  $f(x)$  é:

$$f_1(x)=f_a(y)=\sum_{i=1}^{I+1}\sum_{t=1}^T y(i,t,k)$$

$$f_2(x)=f_p(x)+f_m(x)=\sum_{i=1}^I\sum_{t=1}^T c_p(i,t,k)x(i,t,k)+\sum_{t=1}^T c_m(t,k)x(0,t,k)$$

As funções  $\phi_0$ ,  $\phi_1$  e  $\phi_2$  são definidas como:

$$\phi_0(\delta^-, \delta^+) = \sum_{j=1}^i \delta_j^+$$

$$\phi_1(d_1^-, d_1^+) = d_1^- + d_1^+$$

$$\phi_2(d_2^-, d_2^+) = d_2^- + d_2^+$$

Os valores das metas para cada objetivo é:

$$\phi_0: M_0 = M_a = 0;$$

$$\phi_1: M_1 = M_{f_1} = 0;$$

$$\phi_2: M_2 = M_{f_2} = 912000;$$

A tabela 4.1 apresenta os resultados para o exemplo e os níveis de demanda atendida em atraso (backorder). As figuras 4.2, 4.3, e 4.4 ilustram os resultados da minimização de cada objetivo. A solução do problema foi alcançada em 0,27 segundos de CPU.

Tabela 4.1: Resultados da minimização dos objetivos.

	factibilização	armazenagem	custo	backorder	
Interação 1	0	40000	956000	4000	57 iterações
Interação 2	0	0	1032000	40000	17 iterações
Interação 3	0	0	996000	4000	19 iterações
Metas	0	0	912000	----	

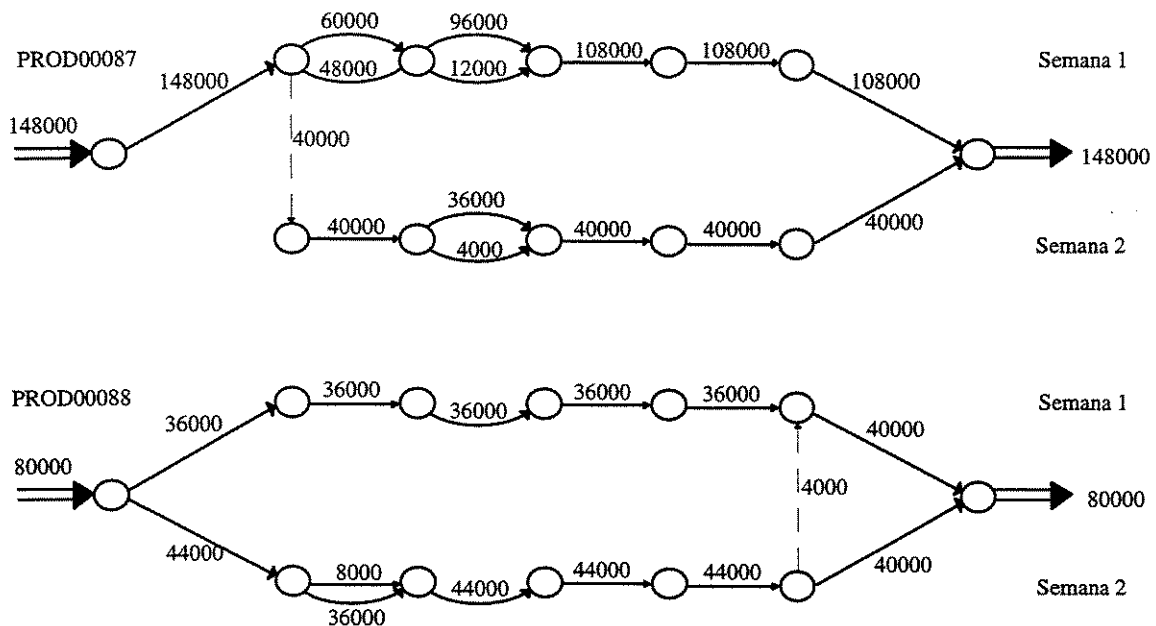


Figura 4.2: Resultados para o primeiro objetivo (factibilização).

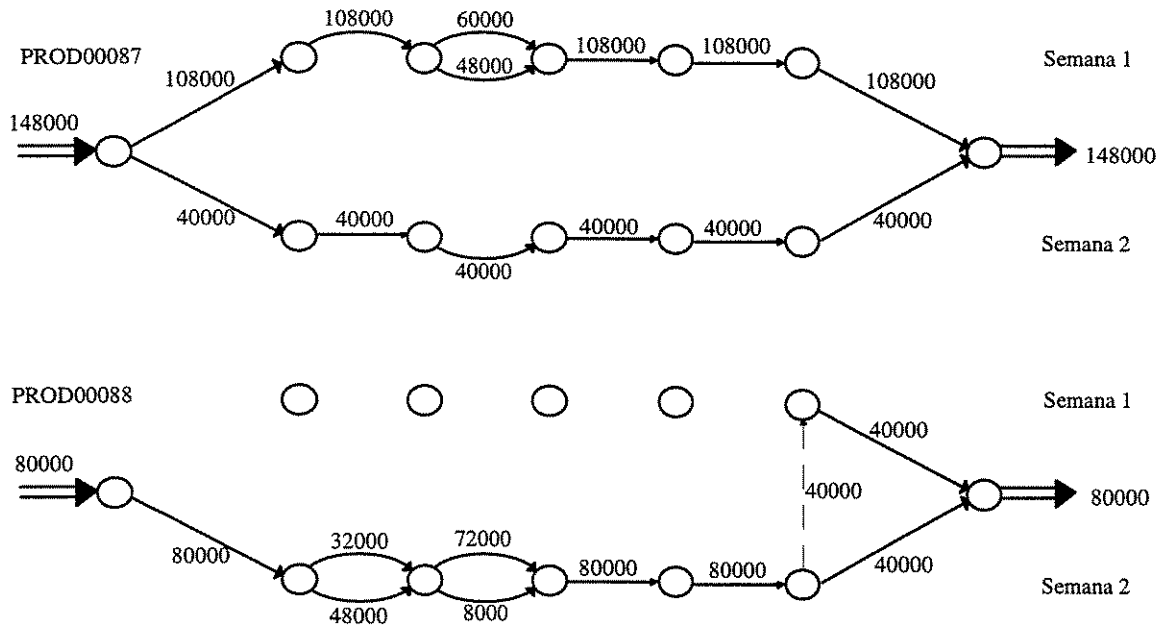


Figura 4.3: Resultados para o segundo objetivo (níveis de armazenagem).

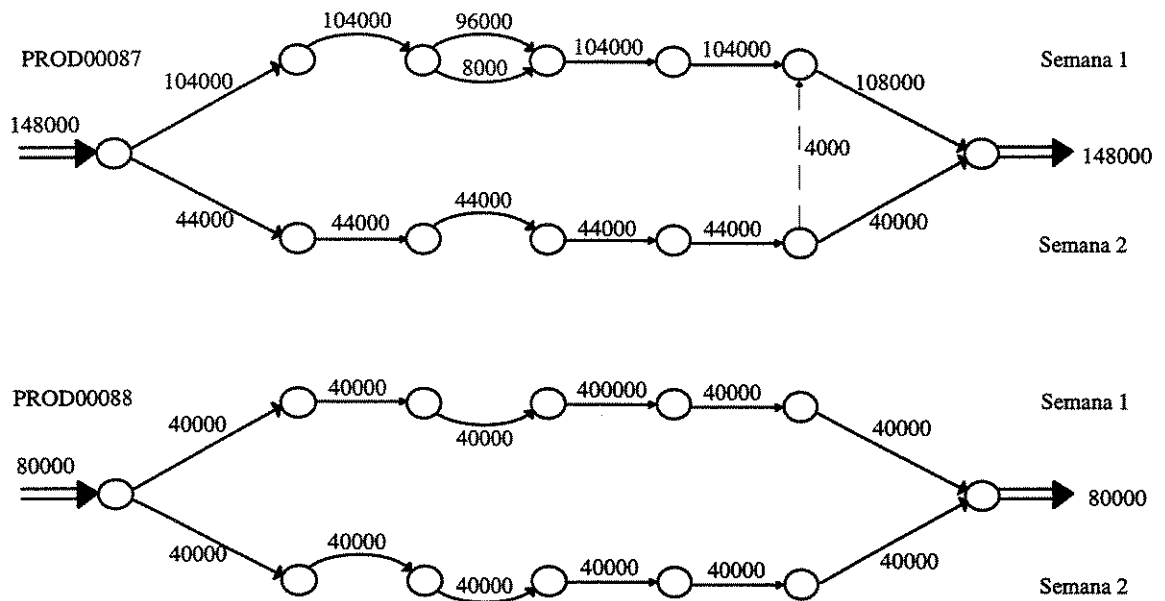


Figura 4.4: Resultados para o terceiro objetivo (custos de produção e de aquisição da matéria prima).

É importante observar o comportamento dos níveis de demanda atendida em atraso (backorder) durante as interações. Note que na factibilização, havia um nível de 4000 peças do produto PROD00088, nível este que passou para 40000 peças na interação de minimização dos níveis de armazenagem, e que a seguir passou a 4000 peças do produto PROD00087 após a minimização dos custos de produção e de aquisição de matéria prima.

A prioridade adotada para os objetivos é agora alterada para se verificar a potencialidade e a flexibilidade desta aplicação via PA. Após a factibilização inicial (veja figura 4.2), a prioridade dos objetivos será a seguinte: primeiro a minimização dos custos de produção e aquisição da matéria prima e a seguir a minimização dos níveis de armazenagem. Os resultados são apresentados nas figuras 4.5 e 4.6 e tabela 4.2 a seguir.

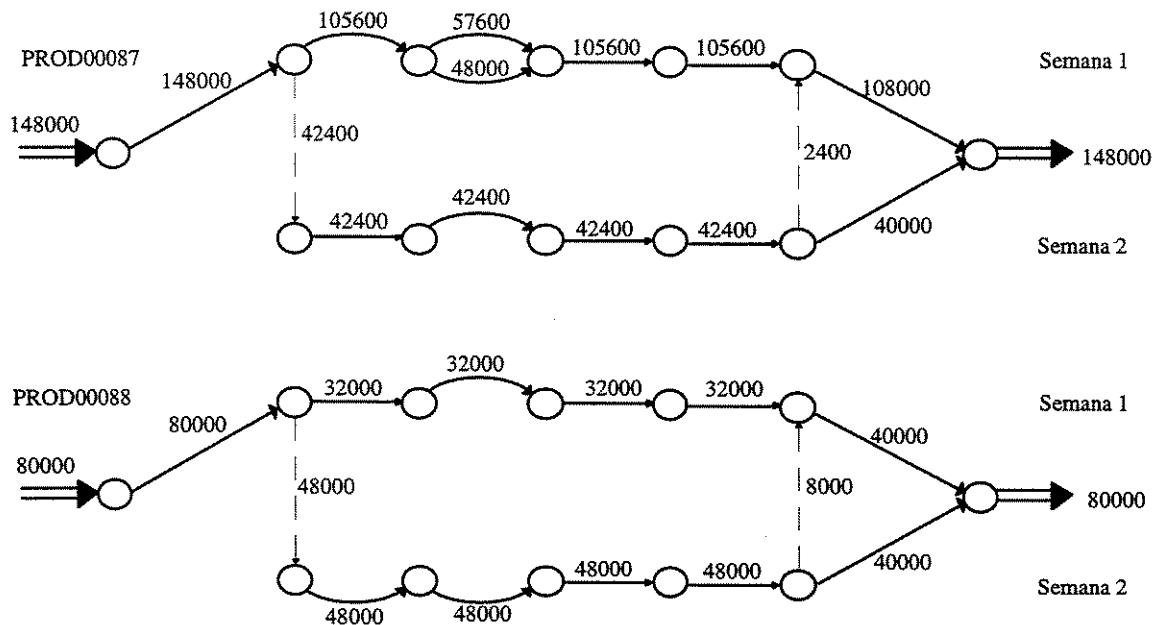


Figura 4.5: Resultados para o segundo objetivo (custos de produção e de aquisição de matéria prima).



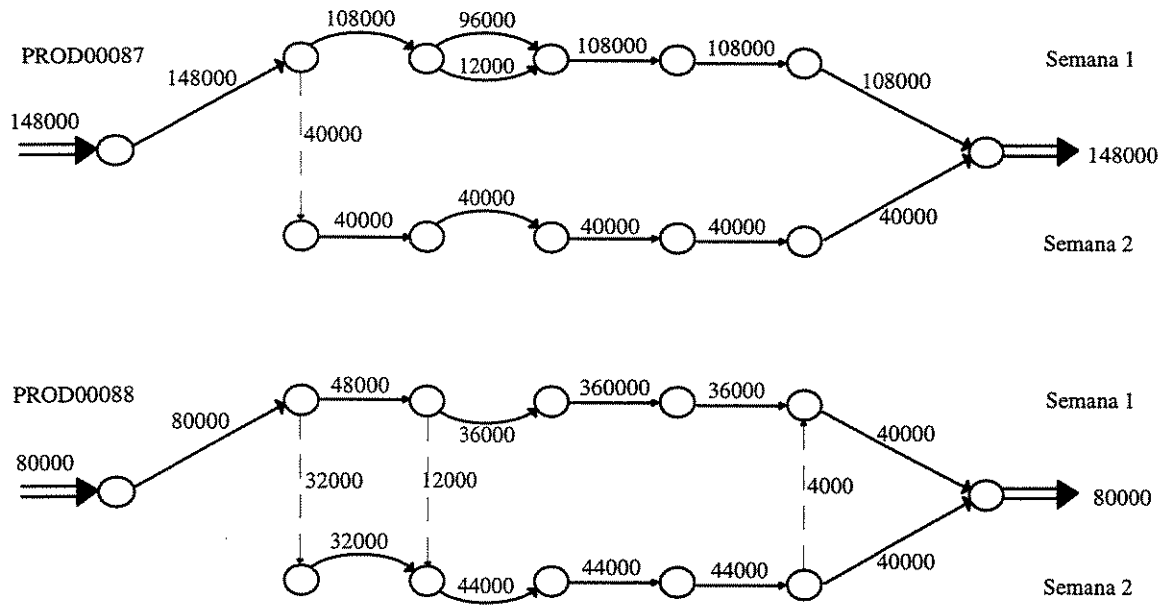


Figura 4.6: Resultados para o terceiro objetivo (níveis de armazenagem).

Tabela 4.2: Resultados da minimização dos objetivos.

	factibilização	custo	armazenagem	backorder	
Interação 1	0	956000	40000	4000	57 iterações
Interação 2	0	912000	90400	10400	25 iterações
Interação 3	0	912000	84000	4000	18 iterações
Metas	0	912000	0	---	

Nesta segunda aplicação, o nível de demanda atendida em atraso (backorder) durante as interações apresentou o seguinte comportamento: na interação inicial (factibilização) havia um nível de 4000 peças do produto PROD00088, nível este que passou, após a interação de minimização dos custos de produção e de aquisição de matéria prima, para 2400 peças do produto PROD00087 e de 8000 peças do produto PROD00088, e que por fim passou a 4000 peças do produto PROD00088 após a minimização dos níveis de armazenagem.

Comparando-se os resultados das tabelas 4.1 e 4.2, do ponto de vista das diferentes políticas (preferências) em relação à armazenagem e ao custo, o decisor tem duas situações: a primeira onde, com uma ausência de armazenagem, o custo de produção e aquisição de matéria prima é de \$ 996.000 e a segunda com um custo de \$ 912.000 o nível de armazenagem é de 84.000 peças. O decisor pode então, por considerar alto o nível de armazenagem, tentar uma nova solução intermediária, indagando qual seria o custo de estabelecer por exemplo uma meta de armazenagem de 64.000 peças. Os resultados deste novo cenário estão ilustrados nas figuras 4.7, 4.8 e 4.9 e tabela 4.3 a seguir. Para esta meta de armazenagem, o custo de produção e aquisição de matéria prima resulta em \$ 936.000.

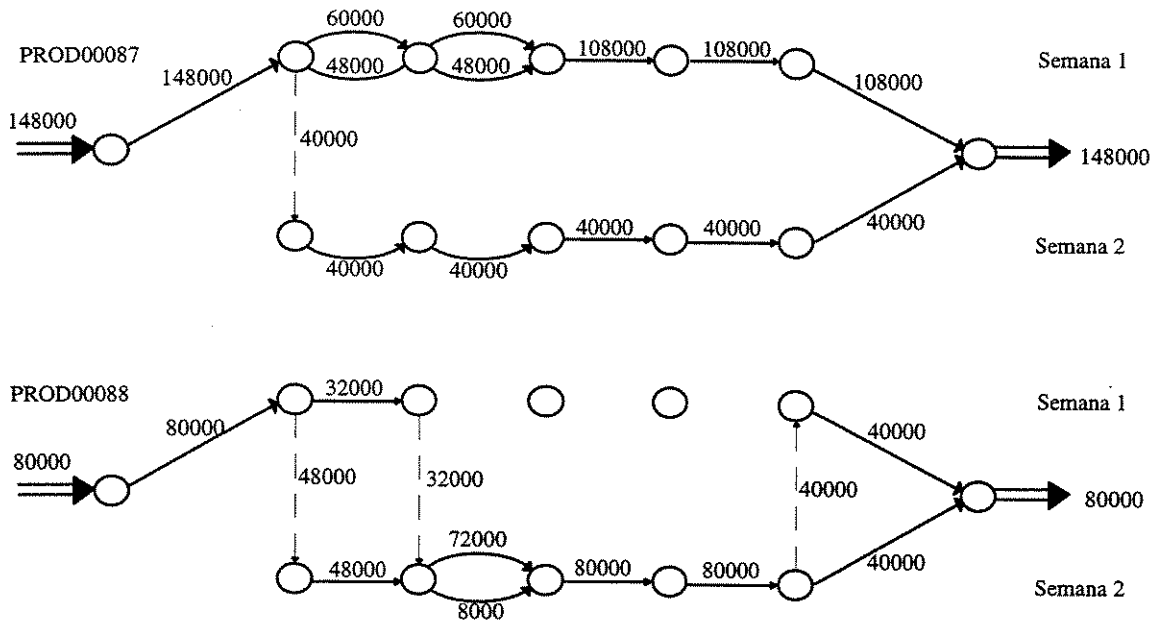


Figura 4.7: Resultados para o primeiro objetivo (factibilização).

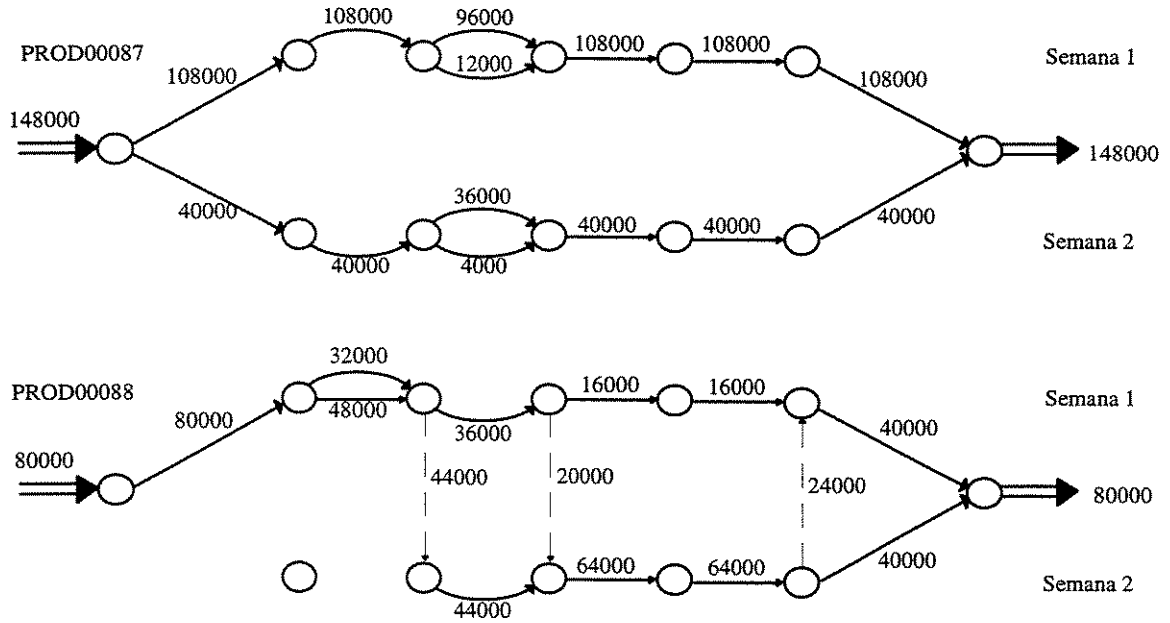


Figura 4.8: Resultados para o segundo objetivo (níveis de armazenagem).

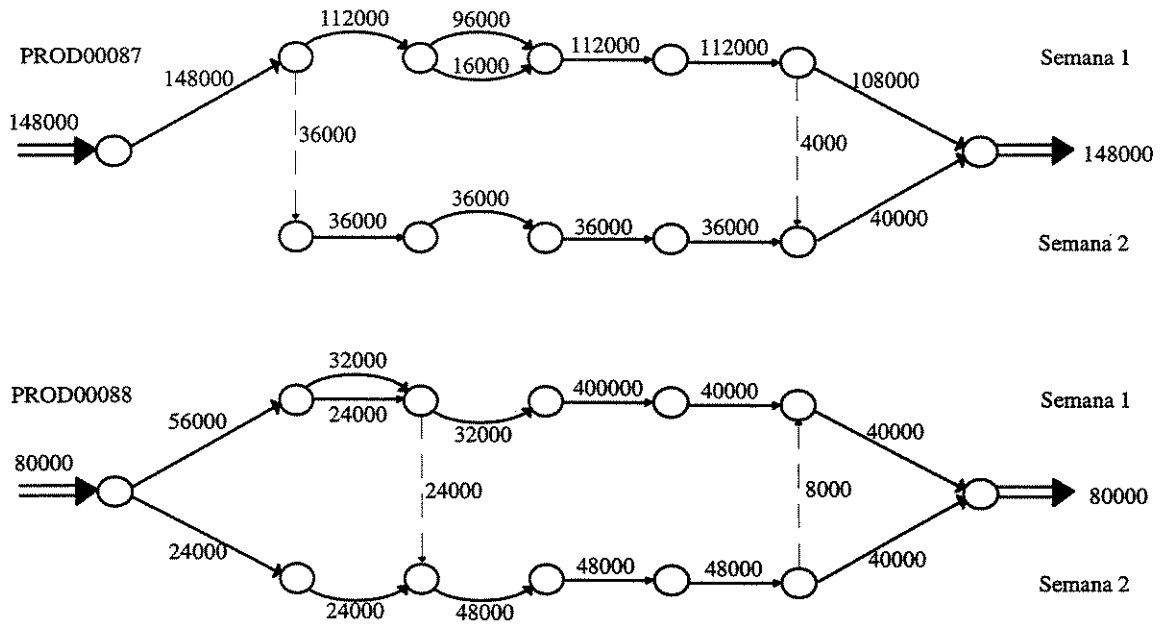


Figura 4.9: Resultados para o terceiro objetivo (custos de produção e de aquisição de matéria prima).

Tabela 4.3: Resultados da minimização dos objetivos.

	factibilização	armazenagem	custo	backorder	
Interação 1	0	120000	912000	40000	54 iterações
Interação 2	0	64000	952000	24000	31 iterações
Interação 3	0	64000	936000	8000	38 iterações
Metas	0	64000	912000	----	

#### 4.4.1 - OUTROS RESULTADOS

No capítulo 3, alguns testes foram realizados com a finalidade de se comparar as técnicas estudadas. Nestes cenários foram estudados o comportamento das técnicas em função de variação nos fatores de demanda com relação à demanda original. A figura 3.10 é aqui reproduzida com a inclusão dos resultados obtidos com a técnica de programação alvo. A figura 4.10 mostra o desempenho dos algoritmos em função do número de produtos. Embora esta seja uma técnica interativa, utilizando portanto o algoritmo NETSIDE tantas vezes quantas forem as funções objetivo, o tempo de CPU não aumenta significativamente para a obtenção de uma solução de compromisso. Isto se explica pelo fato do algoritmo NETSIDE poder utilizar a base ótima de cada interação como ponto de partida para a próxima interação.

#### 4.5 - PROGRAMAÇÃO ALVO INTERATIVA

A Programação Alvo é uma técnica a-priori e requer portanto que o decisor indique suas preferências com relação aos objetivos antes que de fato o problema seja resolvido. Como o decisor não sabe com antecedência as conseqüências dessas

preferências, tal decisão pode gerar soluções incompatíveis com os níveis de suas aspirações. Com o objetivo de contornar este problema, um algoritmo que estabeleça uma interação com o decisor torna-se necessário. Contudo, este procedimento obriga que o decisor forneça alguma informação a respeito de sua função de preferências com relação aos objetivos. Uma representação explícita da função de preferências (ou utilidade)  $U: R^m \rightarrow R$  não é necessária; pode-se assumir apenas que o decisor é capaz de fornecer informações locais à respeito de suas preferências com relação aos objetivos. Supõe-se ainda que  $U$  é crescente com relação a cada componente de  $f(x) = [f_1(x) \ f_2(x) \ \dots \ f_m(x)]^T$ . Portanto, o problema a ser resolvido seria

$$\underset{x \in X}{\text{Min}} U(f(x)) \quad (4.5)$$

Algoritmos iterativos de programação alvo são discutidos em detalhes em Dyer (1972) e são perfeitamente compatíveis com os modelos de otimização para o problema de planejamento da manufatura proposto neste trabalho. A implementação de algoritmos iterativos de PA é uma extensão natural ao trabalho realizado.

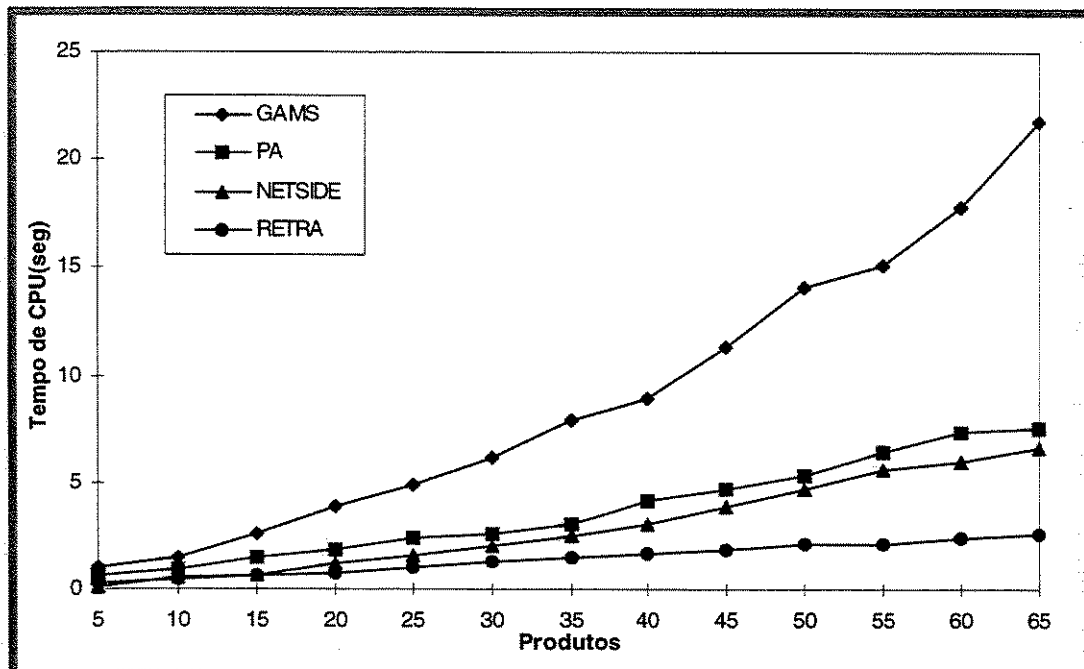


Figura 4.10: Desempenho em função do número de produtos.

além de mostrar o comportamento *quase-linear* do tempo de CPU com relação a variação do número de produtos. Este comportamento comprova a grande vantagem do algoritmo de penalidade quando utilizado em sistemas reais de manufatura com grande número de itens.

Formulações multiobjetivos e métodos eficientes de solução que levam em conta o porte, a natureza linear e a estrutura especial do problema foram discutidos. Foi então apresentada a técnica de programação alvo que tem sido eficientemente utilizada em problemas de planejamento de produção. Para a consideração de multiobjetivos através de um enfoque elegante, foi escolhido o algoritmo NETSIDE para a resolução de problemas de grafo com restrições adicionais. Os resultados obtidos nos exemplos mostraram que esta técnica agrega flexibilidade à análise do problema e não aumenta significativamente o tempo de CPU necessário para a obtenção de uma solução de compromisso.

As principais contribuições do trabalho foram ilustradas através de aplicações reais em um ambiente de manufatura encontrado no parque industrial brasileiro.

O enfoque apresentado neste trabalho, para tornar-se realista em sistemas de manufatura, deve ser integrado às práticas industriais tipo MRP. Este enfoque integra métodos gerativos de solução como elementos estabelecadores de metas a serem perseguidas pelas práticas industriais.

Como sugestões de futuros trabalhos, pode-se mencionar:

- o desenvolvimento de algoritmos ainda mais rápidos, que contornem as dificuldades encontradas nos métodos tradicionais de penalidades, como por exemplo, a inicialização e atualização de parâmetros de penalidades,
- a utilização de ícones e janelas para a descrição de problemas reais no computador, onde o decisor monta a árvore de produto utilizando recursos gráficos como ilustrado por exemplo na figura 2.13. Para cada produto, define-se as taxas de processamento por máquinas, custos, capacidades e o número de períodos que quer estudar. A modelagem gráfica torna transparente aos usuários as equações matemáticas que descreveriam o sistema físico, fator sempre relevante,

## CAPÍTULO 5

### CONCLUSÕES

Este trabalho apresentou um novo modelo para o planejamento da produção de sistemas de manufatura multiestágio, multiperíodo e multiproduto ao incorporar equações adicionais que consideram os acoplamentos de componentes necessários à montagem do produto final. A abordagem de Ahuja *et al.* (1993) limita-se a um problema multiproduto, enquanto que Drexl e Kimms, (1997) abordam problemas de programação inteira. Este trabalho apresentou um enfoque novo que é a consideração das linhas de montagem onde o aproveitamento da estrutura do problema não pode mais utilizar o algoritmo particular sugerido em Kennington and Helgason, (1980). Assim tornou-se necessária a adoção de um algoritmo para problemas com restrições adicionais com estrutura bem definidas e de grande esparcidade que não sejam específicos para multifluxo.

A formulação geral resultou em um problema de fluxo em redes com restrições adicionais, onde são esperados como resultados os níveis ótimos de estoque, de atendimento à demanda, de carregamento das máquinas, de pedidos em atraso e a necessidade temporal de matéria prima. A estrutura especial da formulação foi explorada pelos métodos de solução.

Inicialmente uma abordagem mono-objetivo do problema foi apresentada. Foi descrita a técnica de decomposição primal, específica para a resolução de problemas de fluxo em redes com restrições adicionais. A seguir, mostrou-se que se as restrições adicionais fossem tratadas por técnicas de penalidade, o modelo resultante seria um problema de fluxo em rede com função objetivo quadrática. A técnica de decomposição simplicial surgiu então como uma forma eficiente de solução de problemas deste tipo. Dois exemplos comparativos envolvendo o algoritmo de decomposição primal e o software GAMS mostraram a grande vantagem do algoritmo de penalidades RETRA,

- a busca de uma integração entre as práticas industriais e os modelos de otimização, visando complementariedade das duas técnicas. As soluções encontradas através dos modelos de otimização tornam-se agora metas a serem perseguidas pelas práticas industriais,
- o desenvolvimento de um algoritmo NETSIDE específico que aproveite ainda mais a estrutura especial do problema, evitando-se inversões de matrizes,
- o desenvolvimento de um procedimento de análise no espaço dos objetivos, por ser de entendimento mais fácil pelo decisor e por eventualmente proporcionar melhores resultados em relação àqueles baseados no espaço das variáveis de decisão,
- a utilização de processamento paralelo para resolver os K problemas separáveis e independentes por produto,
- o desenvolvimento de um algoritmo de grafo generalizado (Jensen e Barnes, 1980) visando modelar sistemas mais complexos com estrutura de fluxo em redes com ganho (positivo ou negativo). Como exemplo de aplicações onde se utilizam ganhos, tem-se: aplicações financeiras, manufatura com consideração de perdas, entre outras,
- a utilização de outras técnicas de programação multiobjetivo que explorem as vantagens da técnica de penalidades,
- a implementação de algoritmos baseados em programação multiobjetivo interativa, como o algoritmo interativo de programação alvo.





## CAPÍTULO 6

## REFERÊNCIAS BIBLIOGRÁFICAS

- AHUJA, Ravindra K., Thomas L. Magnanti e James B. Orlin (1993), "Network Flows: Theory, Algorithms, and Applications", Prentice-Hall.
- BENTON, W. C. e H. Shin (1998), "Manufacturing planning and control: The evolution of MRP and JIT integration", *European Journal of Operational Research*, 110, 411-440.
- BROOKE, A., D. Kendrick e A. Meeraus (1992), "GAMS - A User's Guide, Realease 2.25", Boyd & Fraser Publishing Company, Massachusetts.
- CARVALHO, M.F.H.; C.A.O. Fernandes e O.S. Silva Filho (1992), "Manufacturing Planning by Network Flow Model With Side Constraint", IFAC Workshop on CIM in Process and Manufacturing Industries, Helsinki, Finlândia.
- CARVALHO, M.F.H.; C.A.O. Fernandes; O.S. Silva Filho e E.J. Varga Filho (1994), "OPERA - Optimal Production Environment for Resource Allocation", *Production Management Methods (B-19)*, IFIP Transactions, Elsevier Science B.V. (North Holland).
- CARVALHO, M.F.; R.B.B. Hadad e C.A.O. Fernandes (1996), "Optimisation Method as Complementary Tool for Industrial Practices", *CAD/CAM Robotics and Factories of the Future*, London.
- CARVALHO, M.F.H.; O.S. Silva Filho e C.A.O. Fernandes (1998), "O Planejamento da Manufatura - Práticas Industriais e Métodos de Otimização", *Gestão e Produção*, vol. 5, nº 1, 34-59, *Revista do Departamento de Engenharia de Produção*, Universidade Federal de São Carlos.

- CARVALHO, M.F.; C.A.O. Fernandes; P.A.V. Ferreira & M.T. Takahashi (1998a), "Assembly Production Planning by Network Flow Approach", 9<sup>TH</sup> Symposium on Information Control in Manufacturing - INCON'98, Nance - Metz, France.
- CARVALHO, M.F.; C.A.O. Fernandes e P.A.V. Ferreira (1999), "Multiproduct Multistage Production Scheduling (MMPS) for Manufacturing Systems", to appear in Production Planning & Control, London, UK.
- CARVALHO, M.F. e C.A.O. Fernandes (1999a), "Planejamento da Produção de Manufatura por Fluxos em Redes", Gestão e Produção, vol. 6, nº 1, 30-50, Revista do Departamento de Engenharia de Produção, Universidade Federal de São Carlos.
- COHON, J. L. (1978), "Multiobjective Programming and Planning", New York: Academic Press.
- DAS, S. K. e S. C. Sarin (1994), "An integrated approach to solving the master aggregate scheduling problem", International Journal of Production Economics, vol: 34 Iss: 2, 167-177.
- DAUER, Jerald P. (1987), "Analysis of the Objective Space in Multiple Objective Linear Programming", Journal of Mathematical Analysis and Applications, 126, 579-593.
- DEAN, Burton V., Yung-Mok Yu e Marc. J. Schniederjans (1990), "A Goal Programming Approach to Production Planning for Flexible Manufacturing Systems", Journal of Engineering and Technology Management, 6, 207-220.
- DEMBO, R.S., J.M. Mulvey e S.A. Zenios (1989), "Large-scale Nonlinear Network Models and Their Applications", Operations Research, vol 37, nº 3, 353-372.
- DREXL, A. e A. Kimms (1997), "Lot sizing and scheduling - Survey and extensions", European Journal of Operational Research, 99, 221-235.
- DYER, James S. (1972), "Interactive Goal Programming", Management Science, vol. 19, nº 1, 62-70.
- EDGHILL, J.S. e A. Davis (1985), "Flexible Manufacturing Systems - The Myth and Reality", I. J. of Adv. Manufacturing Techn., 1/1, 37-54.

- FERNANDES, C. A. O., M. F. H. Carvalho, P. A. V. Ferreira e O. S. S. Filho (1995), "Multicommodity Manufacturing Flow Problems by a Penalty Function Approach", IFAC Large Scale Systems, Pergamon, London, UK.
- FERREIRA, P. A. V. e M. E. S. Machado (1996), "Solving Multiple-Objective Problems in the Objective Space", Journal of Optimization Theory and Applications 89, 659-680.
- FOX, K.A. (1984), "MRP-II Providing a Natural Hub for Computer Integrated Manufacturing System", Ind. Eng., Oct, 44-50.
- GAIMON, Cheryl, (1986), "Optimal inventory, backlogging and machine loading in a serial, multi-stage, multi-period production environment", Int. J. Prod. Res., vol. 24, n° 3, 647-662.
- GALBRAITH, Lissa e William Miller (1989), "A comparison of goal programming and simulation model results for a multiobjective multiperiod multiproduct manufacturing scheduling problem", Computers & Industrial Engineering, vol: 17, 366-371.
- GALLAGHER, R.J. e A.S. Ossana (1994), "Constructing the set of efficient objective space in linear multiple objective transportation problem", European Journal of Operational Research, 73, 150-163.
- GARFINKEL, Robert S. e George L. Nemhauser, (1977), "Integer Programming", John Wiley & Sons, New York.
- GERSHWIN, S.B. e R.S. Hildebrant (1986), "A Control Perspectives on Systems", IEEE Control System Magazine, vol 6, n° 2, 3-15.
- GHOSH, S. e C. Gaimon (1992), "Routing Flexibility and Production Scheduling in a Flexible Manufacturing System", European Journal of Operational Research, 60, 344-364.
- GOLHAR, D.Y. e C. L. Stamm (1991), "The Just In Time Philosophy: A Literature Review", Int. J. Prod. Res., vol. 29, n° 4, 657-676.
- HINDI, K. S. (1995), "Algorithms for Capacited, Multi-Item Lot Sizing Without Set-ups", Journal of the Operational Research Society, 46, 465-472.

- HWANG, C. e A. S. Masud (1979), "Multiple Objective Decision Making: Methods and Applications", New York: Springer-Verlag.
- HEARN, D. W., S. Lawphongpanich e J. A. Ventura (1987), "Restricted Simplicial Decomposition: Computations and Extensions", *Mathematical Programming Study*, 31, 99-118.
- JAASKELAINEN, V. (1969), "A goal programming model for aggregate production planning", *Swedish Journal of Economics*, 2, 14-29.
- JACOBS, F.R. (1984), "OPT Uncovered Many Production Planning and Scheduling Concepts can be Applied with or without the Software", *Ind. Eng.* 32-41.
- JENSEN, Paul A. e J. Wesley Barnes (1980), "Network Flow Programming", John Wiley & Sons, New York.
- KENNINGTON, J. L. e R. V. Helgason (1980), "Algorithms for Network Programming", John Wiley & Sons, New York.
- KENNINGTON, J L. e A. Whisman (1990), "Netside User's Guide", Department of Computer Science and Engineering, Southern Methodist University, Dallas.
- KUMAR, P.; N.K. Tewari e N. Singh (1990), "Joint consideration of grouping and loading problems in a flexible manufacturing system", *International Journal of Production Research*, vol: 28 Iss: 7, 1345-1356.
- KUMAR, P.; N. Singh e N. K. Tewari (1991), "A nonlinear goal programming model for multistage, multiobjective decision problems with application to grouping and loading problem in a flexible manufacturing system", *European Journal of Operational Research*, vol. 53 Iss: 2, 166-171.
- LEE, S.M. e H.-J Jung (1989), "A multi-objective production planning model in a flexible manufacturing environment", *International Journal of Production Research*, vol. 27, Iss: 11, 1981-1992.
- LEWIS, Holly S.; James R. Sweigart e Robert E. Markland (1992), "Master Scheduling in Assemble-To-Order Environments: A Capacited Multiobjective Lot-Sizing Model", *Decision Sciences*, vol. 23, 21-43.

- LUENBERGER, David G., (1984), "Linear and Nonlinear Programming", Addison-Wesley Publishing Company.
- MALAKOOTI, B. (1989), "A hierarchical, multi-objective approach to the analysis, design, and selection of computer-integrated manufacturing systems", Robotics and Computer-Integrated Manufacturing vol. 6 Iss: 1, 83-97.
- MALAKOOTI, B. e J. Deviprasad, (1989), "An interactive multiple criteria approach for parameter selection in metal cutting", Operations Research, vol. 37, Iss: 5, 805-818.
- MALAKOOTI, B. (1991), "A multiple criteria decision making approach for the assembly line balancing problem", International Journal of Production Research, vol.29, n° 10, 1979-2001.
- MOHANTY, R.P. e R. Singh (1992), "A hierarchical production planning approach for a steel manufacturing system", International Journal of Operations & Production Management, vol. 12, Iss: 5, 69-78.
- MULVEY, John M., Stavros A. Zenios e David P. Ahlfeld (1990), "Simplicial Decomposition for Convex Generalized Networks", Journal of Information & Optimization Sciences, vol. 11, n° 2, 359-387.
- NORBIS, M.I. e J. MacGregor Smith (1988), "A multiobjective, multi-level heuristic for dynamic resource constrained scheduling problems", European Journal of Operational Research, vol.33, n° 1, 30-41.
- OGRYCZAK, W.; K. Studzinski e K. Zorychta (1989), "A solver for the multi-objective transshipment problem with facility location", European Journal of Operational Research, vol.43, n° 1, 53-64.
- RAMESH, R. e J.M. Cary (1989), "Multicriteria jobshop scheduling", Computers & Industrial Engineering, vol. 17, 597-602.
- RODAMMER, F.A. e K.P. Write (1988), "A Recent Survey of Production Scheduling", IEEE Trans. on Systems, Man a Cybernetics, vol 18, n° 6, 841-851.
- SCHONBERGER, R.J. (1984), "Just in Time Production System. Replacing Complexity Sith Simplicity in Manufacturing Management". Ind.Eng. Oct., 52-63.

- SILVA FILHO; O.S.; C.A. Passos; M.F.H. Carvalho e C.A.O. Fernandes (1992), "Sistemas Flexíveis de Manufatura - Avaliação de desempenho via Redes de Filas Fechadas", XXV SOBRAPO, SBPO, Campinas, SP.
- SINGH, N.; Y.P. Aneja e S.P. Rana (1990), "Multiobjective modelling and analysis of process planning in a manufacturing system", *International Journal of Systems Science*, vol. 21, nº 4, 621-630.
- SURI, R. (1985), "An Overview of Evaluative Models for Flexible Manufacturing Systems", *Annals of Oper. Res.*.
- TABOUN, S.M. e T. Ulger (1992), "Multi-objective modelling of operation-allocation problem in flexible manufacturing systems", *Computers & Industrial Engineering*, vol. 23, 295-299.
- TRZASKALIK, T. (1990), "Multi-objective, multi-period planning for a manufacturing plant", *Engineering Costs and Production Economics*, vol. 20, Iss: 2, 113-120.
- Von HOHENBALKEN, Balder (1977), "Simplicial Decomposition in Nonlinear Programming Algorithms", *Mathematical Programming*, 13, 49-68.
- Wu, De-Shih e M.T. Tabucanon (1990), "Multiple criteria decision support system for production management", *Engineering Costs and Production Economics*, vol. 20, Iss: 2, 203-212.
- YANG, Jian-Bo; Chen Chen e Zhong-Jun Zhang (1988), "The interactive decomposition method for multiobjective linear programming and its applications", *Information and Decision Technologies*, vol. 14, Iss: 4, 275-288.
- ZELNY, Milas (1982), "Multiple Criteria Decision Making", McGraw Hill, New York.
- ZENIOS, S. A., M. Ç. Pinar e R. S. Dembo (1995), "A Smooth Penalty Function Algorithm for Network-Structured Problems", *European Journal of Operational Research*, 83, 220-236.