



PLANEJAMENTO DA PRODUÇÃO DE MANUFATURA POR FLUXOS EM REDES

Marcus F. Carvalho

Fundação Centro Tecnológico Para Informática – CTI
Caixa Postal 6162 – Campinas – SP

C.A.O. Fernandes

Faculdade de Engenharia Mecânica
Universidade Estadual de Campinas – UNICAMP

Resumo

Este artigo apresenta o planejamento da manufatura no nível tático como um problema de fluxo em redes com restrições adicionais. Mostra que, por esta formulação, o modelo matemático possui uma estrutura muito esparsa e particular. O aproveitamento desta característica encoraja o desenvolvimento de algoritmos especiais dedicados a problemas de grande dimensão, como é o caso dos sistemas de manufatura multiprodutos, multiestágios, multiperíodos e, como consequência, permite uma representação mais detalhada do sistema físico. Uma comparação mostra o melhor desempenho destes algoritmos em relação a outros de caráter geral.

***Palavras-chave:* planejamento da produção, programação matemática, fluxo em redes, otimização.**

1. Introdução

A organização da produção e o transporte de produtos dos centros produtores (origens) aos consumidores finais (destinos) constituem-se em um cenário constante dos sistemas de manufatura, requerem estratégias de gerenciamento informatizado e ferramentas que apoiem a

tomada da melhor decisão. Os itens (peças, petróleo, água, veículo, informação, etc.) fluem pelos recursos de produção (máquinas, meio de transporte, linhas, etc.) sofrendo transformação antes de atingirem seus destinos, armazenagem ou centros de distribuição. A sincronização temporal dos recursos disponíveis para a produção com a necessidade do consumidor e

um maior *retorno* para a empresa são os objetivos a serem alcançados pelo decisor.

Na busca da solução alguns problemas aparecem. Por exemplo, uma pequena fábrica de molas, composta de 5 estágios de produção e 20 máquinas, pode ter uma centena de produtos diferentes a serem processados por diferentes linhas de produção e com datas de entrega diferentes. Assim, o problema de planejamento da produção de sistemas reais, nas diversas áreas possíveis, envolve um número muito grande de variáveis. Na busca de uma melhor solução deve-se representar este problema real por um modelo.

A tarefa crítica na escolha de um modelo é que o mesmo expresse um grau adequado de realismo em face aos objetivos e que produza uma solução eficiente. Para trazer o problema para uma dimensão razoável em computador, torna-se importante um balanceamento das variáveis realmente necessárias ao modelo com a dimensão do problema final a ser resolvido. Esta relação de compromisso é determinada pela capacidade do computador no qual se quer resolver o problema e, grande parte das vezes, pela estrutura particular que cada problema apresenta.

Nos problemas reais é sempre possível identificar uma estrutura particular e o aproveitamento desta resulta em dois benefícios:

- redução da dimensão com a conseqüente possibilidade de representação mais detalhada do problema original e,
- maior rapidez na solução o que facilita a análise de um número maior de cenários.

Utilizar *software* de propósito geral para a resolução do problema ou desenvolver algoritmo dedicado para tal, passa a ser uma questão relevante. A necessidade de rapidez de resposta com uma representação adequada do sistema físico incentiva a busca de uma ferramenta dedicada que possa melhorar o desempenho e a utilização de sistemas de auxílio a tomada de decisão (DEMBO *et al.*, 1989 e FERNANDES *et al.*, 1995).

Outro fator de incentivo a algoritmos específicos está no fato de que, embora a ênfase deste

artigo esteja em aplicações de manufatura, vários outros sistemas físicos podem ser representados por modelos matemáticos similares. Em particular, os problemas de fluxo de veículos por um sistema viário (FLORIN, 1986), fluxo de mensagem por um sistema de comunicação, energia elétrica pelos sistemas de transmissão (FRANCO *et al.*, 1994), recursos hídricos (ROSENTHAL, 1981 e CARVALHO, 1988), finança (MURVEY, 1987), planejamento da produção (GLOVER *et al.*, 1979) e outros podem ser modelados como problemas de fluxo em redes.

Os problemas acima citados são representados por modelos dinâmicos de fluxo em rede por possuírem uma estrutura que se repete ao longo do tempo. Os termos multiperíodo/multiestágio e escada tempo/fase são sinônimos do conceito *dinâmico* aqui adotado.

Em resumo, três aspectos são considerados na escolha deste modelo. Em primeiro lugar, problemas com estrutura de fluxo em rede permitem uma visão diagramática de um problema físico. Em segundo lugar, possibilitam a definição do problema por meio de estruturas gráficas, o que facilita a compreensão de especialistas ou não. Por fim, esta classe de problemas pode ser uma ordem de grandeza mais rápida que códigos de propósito geral (GLOVER *et al.*, 1979 e DEMBO *et al.*, 1989). Aspectos visuais da rede facilitam a interpretação dos resultados e contribuem para que o time de usuários se envolva ativamente no desenvolvimento do modelo independentemente da base em pesquisa operacional.

Este artigo trata do estudo das estruturas particulares de rede, dos problemas relacionados ao planejamento da produção baseado nas observações: grande área de aplicação, facilidade de entendimento, definição do modelo para usuários não especialistas e rapidez de solução. Compara o desempenho de alguns algoritmos adequados à solução destes problemas. Inicia pela descrição dos sistemas de produção. A seguir descreve a modelagem do sistema produtivo. Apresenta a técnica de penalidade

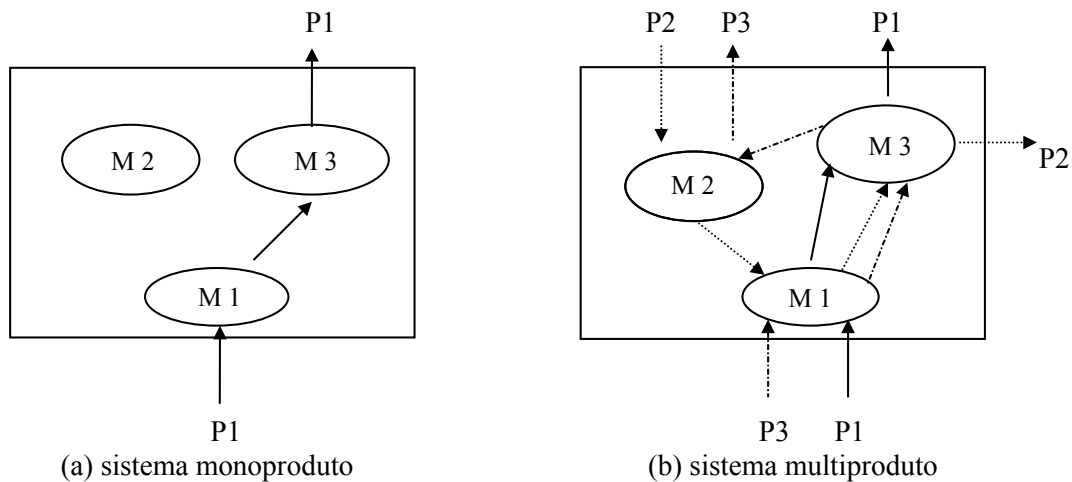


Figura 1 – Sistema de produção discreta.

como a melhor forma de resolver o problema e analisa o desempenho de algoritmos alternativos por meio de um exemplo em uma empresa.

2. Modelagem do Sistema de Produção Discreta

Um sistema de manufatura é composto por itens a serem processados em centros de produção. Estes centros podem ser compostos por máquinas que modificam características do produto (por transformação ou montagem) e/ou transportadoras que modificam a localização do produto. O item flui pelo sistema produtivo sofrendo transformações, sendo transportado segundo uma seqüência tecnológica predeterminada, fazendo parte de montagens até se tornar produto final. A Figura 1.a mostra um sistema com três centros de produção e um roteiro para a fabricação do produto P1. A capacidade máxima de produção deste sistema é igual a menor capacidade entre as capacidades máximas de todos os estágios de produção. Este seria um sistema de fácil gestão caso considerasse um único produto, demanda conhecida e determinística, além de disponibilidade infinita de matéria-prima.

A extensão para o processamento de vários produtos por uma célula faz com que um recurso (por exemplo: máquina, transportador, robô) seja

compartilhado por mais de um produto (Figura 1.b). Agora a determinação da capacidade de produção vai depender de vários fatores, entre eles: o mix (proporcionalidade) entre produtos, seqüência de produção, taxa de compartilhamento de recursos e o perfil temporal da demanda de cada produto (CARVALHO *et al.*, 1998). Assim, o problema de planejamento evolui para a gestão temporal de recursos compartilhados. Para que esta gestão venha a ser realizada com eficácia, torna-se necessário a introdução de alguns conceitos e metodologias de apoio à tomada de decisão. A seguir será apresentada uma modelagem do problema de planejamento tático evoluindo do caso monoproduto para o caso multiproduto.

2.1 Caso Monoproduto

A Figura 1.a ilustra um sistema de produção com dois estágios alocados para processamento de um único produto. Esta figura é repetida na Figura 2. O objetivo do planejamento tático é, a partir de metas estabelecidas pelo nível estratégico, desmembrar a demanda (mensal/semanal) e a disponibilidade de matéria-prima, em produções (semanal/diárias), de forma a otimizar critérios (normalmente traduzidos em custos), atendendo, ao mesmo tempo, as restrições de capacidade de produção, de armazenagem e de suprimento de

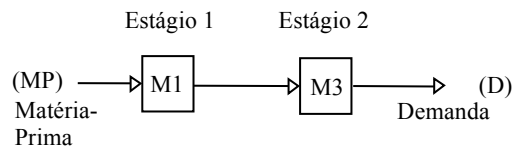


Figura 2 – Representação do sistema físico.

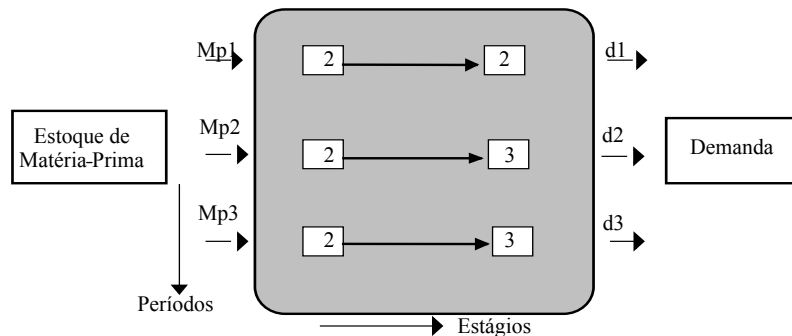


Figura 3 – Expansão para três períodos.

matéria-prima. A expansão da Figura 2, para considerar três períodos de tempo, é mostrada na Figura 3. Cada estágio tem associado um custo de transporte, processamento ou transformação, que pode ser diferente para cada período de tempo. Também estão associadas capacidades de produção. Por exemplo: no período 1, o estágio 2 (grupo de máquinas M3) tem capacidade de produção de 2 unidades passando para 3 unidades no período 2. Também são associados custos e capacidade às armazenagens.

As decisões a serem tomadas para cada período podem envolver: quantidade a ser produzida, carregamento de máquina, tempo de processamento, nível de armazenagem, entre outras. Em cada estágio de produção deve-se tomar a decisão do quanto armazenar e do quanto processar. O processo de tomada de decisão se repete a cada período e antecipações de produção podem ser implementadas visando atendimento de demandas futuras por questões econômicas ou para evitar gargalos. Isto cria, para o problema, uma dependência temporal expressa pela armazenagem interperíodos. Para se desenvolver um sistema de apoio à tomada de decisão é adotada a modelagem representada pelas Figuras 2 e 3.

2.2 Formulação Matemática

Se um item deixa um estágio i , ele pode seguir para processamento no estágio $i+1$ ou ser armazenado para processamento em períodos subsequentes. Representando, na Figura 3, os caminhos que um item pode seguir por arcos, resulta o grafo apresentado na Figura 4. A armazenagem de matéria-prima é representada por MP (deste nó saem os arcos que *abastecem* os períodos) e a demanda total do sistema a ser suprida pela contribuição de cada período de tempo por D.

Aos arcos que deixam o nó de matéria-prima estão associados custos (custo da matéria-prima) e capacidade (capacidade de compra da empresa, capacidade de transporte no período, capacidade do fornecedor). Os fluxos que passam através dos arcos representam as decisões. Por exemplo, o fluxo $x(i,t)$ que passa pelo arco e_{it} , representa a decisão do quanto processar no estágio de produção i e período t . Já o fluxo $y(i,t)$, que passa pelo arco a_{it} , representa a quantidade de produto a ser armazenada em a_{it} para processamento futuro. O atendimento à ordem em atraso é modelado por b_{it-1} . Finalmente a soma dos

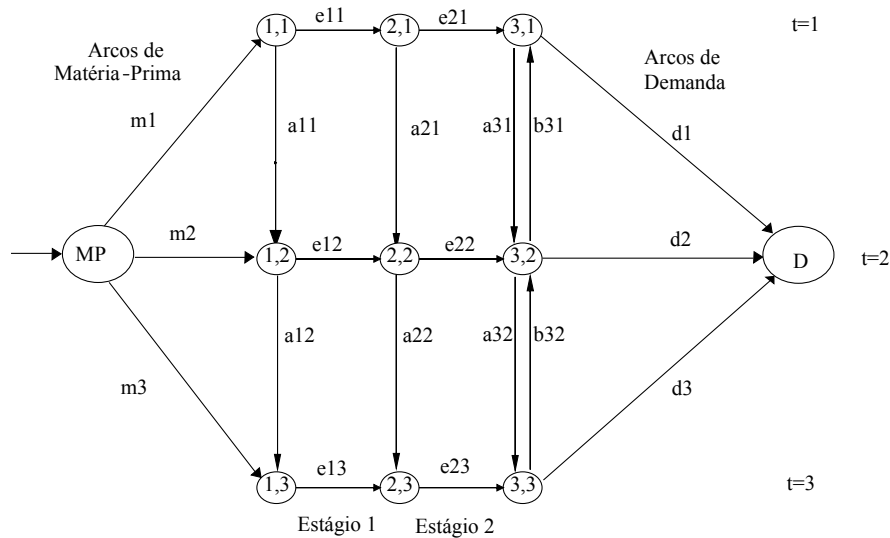


Figura 4 – Grafo referente à Figura 3.

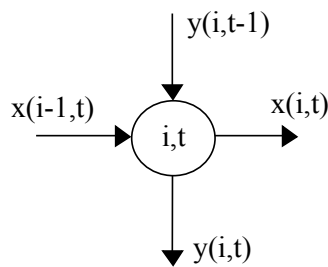


Figura 5 – Representação esquemática para o ponto de decisão do estágio i e período t .

fluxos nos arcos $d1$, $d2$ e $d3$ deve ser suficiente para atender a demanda total D .

Os pontos de decisão antes de cada estágio i no período t define uma relação de balanço entre 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 5, e a equação de balanço (1) a seguir.

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

No estágio $i=1$, a variável $x(i-1,t) = x(0,t)$ é igual a quantidade de matéria-prima para o

período t . No período $t=1$, a variável $y(i,t-1) = y(i,0)$ é a injeção de produtos semi-acabados recebidos de períodos de planejamentos anteriores.

Os arcos do grafo acima são direcionados e pode-se definir uma matriz de incidência que relacione os nós aos arcos direcionados, chamada matriz incidência nó-arco. Esta matriz unimodal tem elemento 1 quando o arco deixa o nó e elemento -1 quando o arco chega ao nó. A esparsidade desta matriz e a unimodalidade incentivam a busca de métodos específicos de solução para este problema (JENSEN, 1987).

O grafo da Figura 4, descrito pela matriz de incidência nó-arco, possui uma estrutura muito especial (veja Tabela 1). Cada nó, correspondente a um estágio de produção, possui dois arcos

Tabela 1 – Matriz incidência nó-arco.

	m1	m2	m3	e11	e12	e13	e21	e22	e23	a11	a12	a21	a22	a31	a32	b31	b32	d1	d2	d3
MP	1	1	1																	
1,1	-1			1						1										
1,2		-1			1					-1	1									
1,3			-1			1					-1									
2,1				-1			1					1								
2,2					-1			1				-1	1							
2,3						-1			1				-1							
3,1							-1							1		-1		1		
3,2								-1						-1	1	1	-1		1	
3,3									-1						-1		1			1
D																		-1	-1	-1

divergentes, um para representar a capacidade de processamento do período e outro para representar a capacidade de 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 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 possibilitando uma representação mais detalhada do sistema produtivo e a análise de um maior número de cenários.

2.3 Problema Monoproduto

De um modo geral, o problema monoproduto pode ser formulado, na forma matricial, como segue:

$$\begin{aligned} & \text{Max } f(x) \\ & \text{s.a } Ax = b \\ & \quad x \leq X \end{aligned} \quad (2)$$

onde $f(\cdot)$ é a função retorno líquido, A é a matriz de incidência nó-arco para o horizonte de planejamento, x o conjunto de variáveis que representam os fluxos nos arcos do gráfico e b o conjunto de disponibilidades de matérias a serem processadas ou demandas a serem atendidas. A função de retorno líquido é composta dos custos

de produção, custos de matéria-prima, custos de armazenagem, penalidades por atraso e do retorno pela venda dos itens. A modelagem acima aplica-se a processos produtivos seriais simples para um único produto. Portanto, para atender a problemas reais da indústria de manufatura, deve ser estendida aos sistemas multiprodutos e problemas mais complexos como é o caso do problema de linhas de montagem.

2.4 Caso Linha de Montagem

Uma linha de montagem é definida pela seqüência de operações (aqui chamada árvore de produto) que o produto e seus itens devem percorrer até se tornarem o produto acabado. O problema de uma linha de montagem composta de três sublinhas é ilustrado na Figura 6. Através das sublinhas fluem os itens e os componentes necessários à montagem do produto final. Montagens ou acoplamentos ocorrem em pontos determinados da linha. Nesta figura, as linhas pontilhadas indicam os acoplamentos entre as sublinhas nos seus respectivos pontos de montagem e torna-se necessário definir o número de componentes a serem montados em cada ponto de montagem (ex.: quatro pneus por carro). Desta forma, uma função entre as mais importantes do planejamento da produção é coordenar o fluxo entre as sublinhas concorrentes visando eficiência na montagem dos produtos finais. Esta

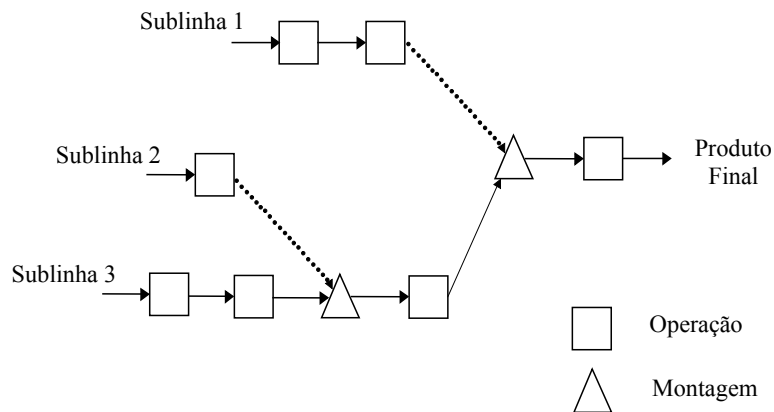


Figura 6 – Representação de uma linha de montagem.

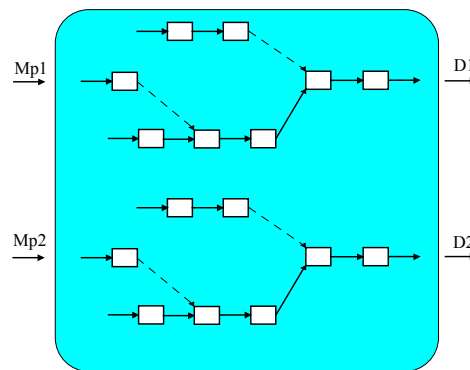


Figura 7 – Linha de montagem para dois períodos.

coordenação ou sincronização é necessária para prevenir a criação de estoques desnecessários entre estágios ou que a produção pare, devido a um fluxo desbalanceado entre produtos e componentes.

A Figura 7 expande a representação da linha de montagem da Figura 6. Esta estrutura mostra os pontos de decisão para o planejamento da produção, para um horizonte de dois períodos de planejamento.

A representação por grafo da Figura 7 é apresentada na Figura 8. Nesta figura, os arcos em linha cheia representam as produções, os arcos em tracejado as armazenagens e os arcos em pontilhado indicam os pontos de montagem ou acoplamento de itens.

Para se garantir o balanço de fluxo no ponto terminal \hat{h} de uma sublinha, deve-se considerar a

seguinte equação de acoplamento:

$$q(\hat{h}) * x(i, t) - x(\hat{h}, t) = 0, \quad t = 1, 2, \dots, T \quad (3)$$

onde $\hat{h} \in \Omega$ é o conjunto dos pontos terminais e $x(\hat{h}, t)$ representa a quantidade de peças do ponto \hat{h} no tempo t , a serem montadas às $x(i, t)$ peças no ponto de montagem ou acoplamento i no tempo t , a uma dada relação de acoplamento $q(\hat{h})$. Esta equação garante que o fluxo do ponto terminal \hat{h} de cada sublinha seja suficiente para a quantidade necessária de montagens a serem realizadas no ponto de montagem 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. Portanto, para o exemplo acima existirão quatro restrições adicionais.

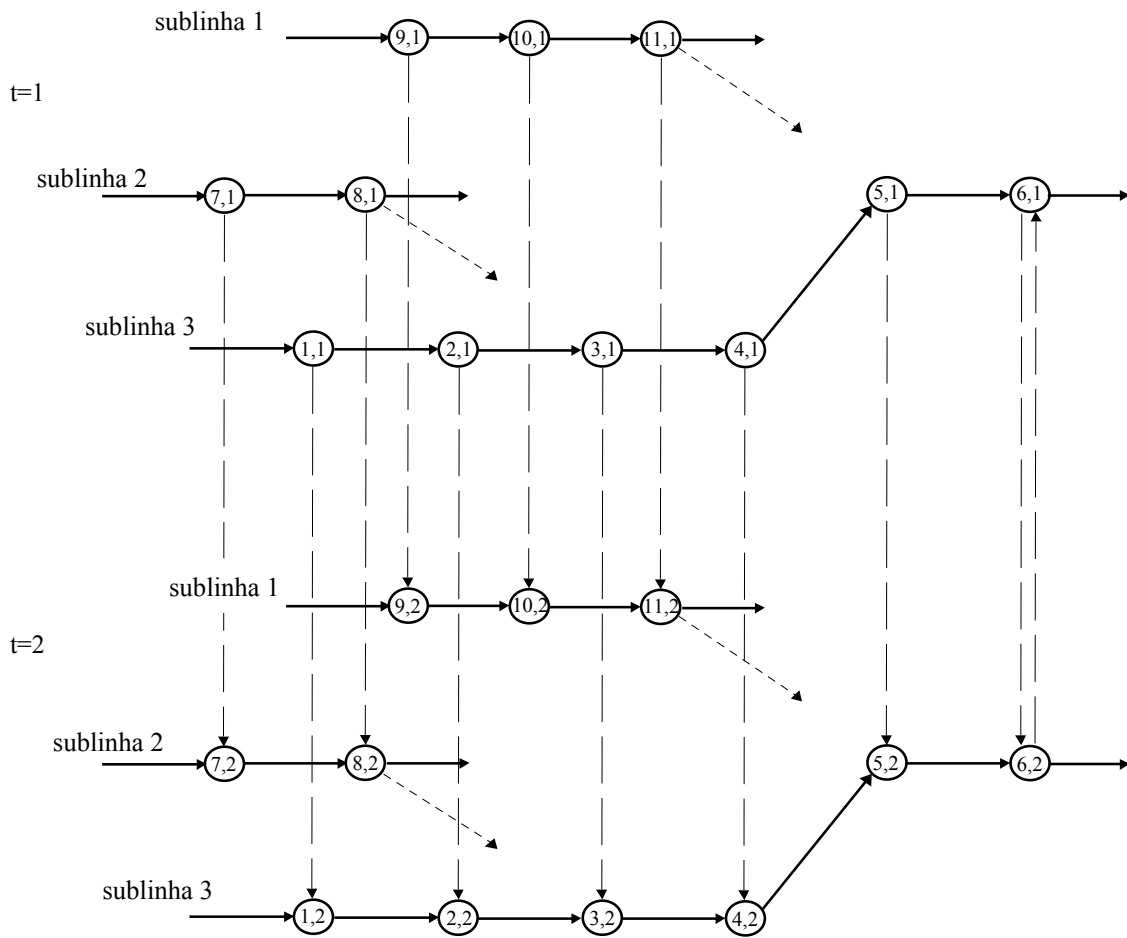


Figura 8 – Representação por grafo do sistema produtivo.

Uma análise da estrutura da matriz de restrições para o problema é feita a partir do exemplo da Figura 8. Uma característica importante deste problema é apresentar uma estrutura de grafos independentes para cada sublinha de produção, ou seja, cada sublinha forma no tempo grafos independentes, como é visto na Figura 9. A coordenação entre os grafos é estabelecida por (3) para cada relação de acoplamento $q(\bar{h})$ entre as sublinhas.

As seguintes equações de acoplamento para este exemplo são:

$$\begin{aligned}
 q(8)x(3,1) - x(8,1) &\leq 0 \\
 q(8)x(3,2) - x(8,2) &\leq 0 \\
 q(11)x(5,1) - x(11,1) &\leq 0 \\
 q(11)x(5,2) - x(11,2) &\leq 0
 \end{aligned}
 \tag{4}$$

A matriz de restrições resultante tem a estrutura apresentada na Figura 10. Note ainda que o número de restrições adicionais é pequeno comparado à dimensão da matriz de restrições. Em um processo produtivo real poucas delas estarão ativas e normalmente um planejador experiente sabe onde são os pontos problemas de seu sistema. Somente estes precisam ser introduzidos como restrição.

2.5 Problema Multiproduto

Se produtos diferentes fossem processados por células inteiramente diferentes bastaria repetir a equação (2) para cada conjunto produto/célula. Mas a característica principal de um sistema de produção está no compartilhamento de

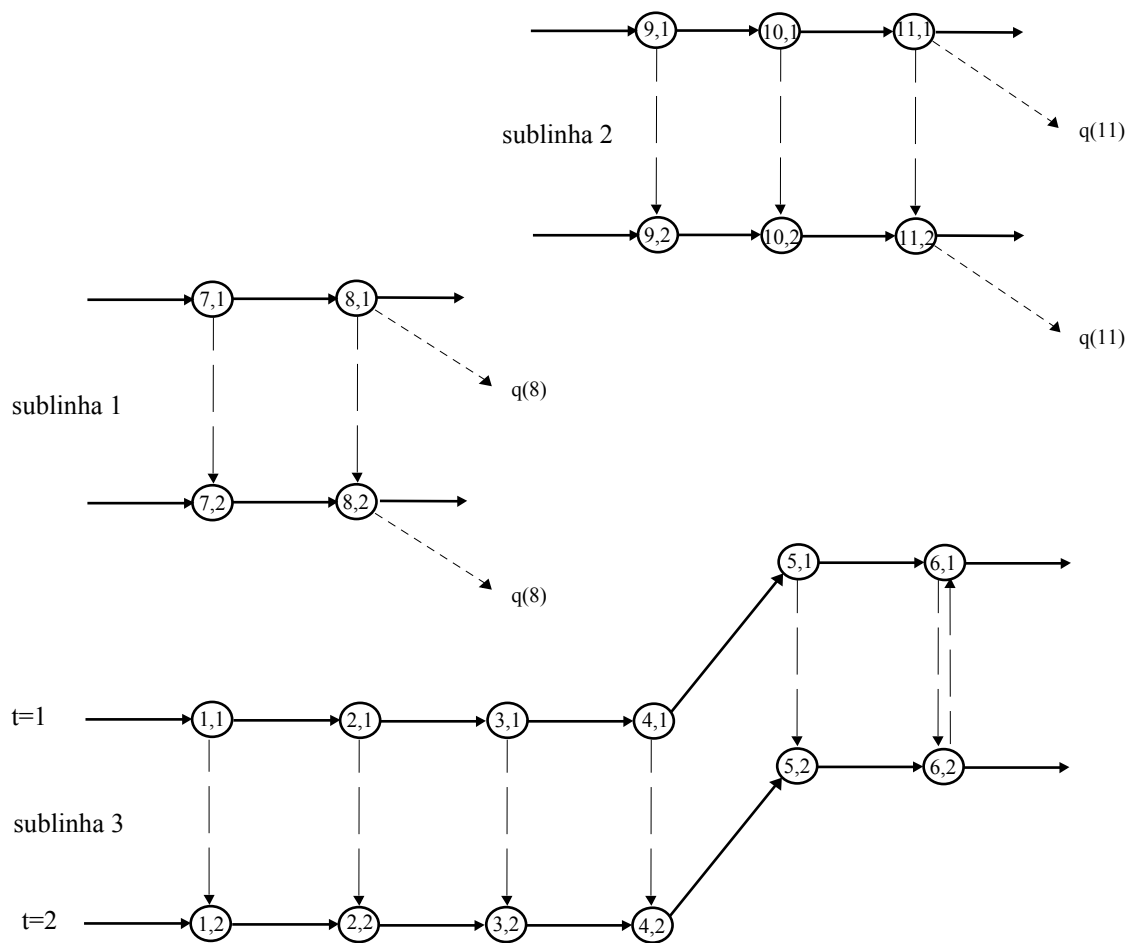


Figura 9 – Grafos resultantes da separação em sublinhas.

recursos com produtos diferentes. Este compartilhamento leva a um menor custo de instalação, a um maior coeficiente de produtividade para as máquinas, e portanto a melhor utilização do sistema produtivo. A Figura 11 ilustra o sistema da Figura 6, agora para processamento simultâneo de K diferentes produtos. Note que a árvore de cada produto pode ser diferente.

A seguir, a modelagem anterior é estendida para processamento de K produtos, com árvore de produtos idênticas ou não. A equação de balanço para o caso multiproduto passa a ser:

$$x(i-1,t,k) + y(i,t-1,k) = x(i,t,k) + y(i,t,k) \quad (5)$$

onde $x(i,t,k)$ é a produção do período t no estágio i , produto k ; $y(i,t,k)$ é a quantidade armazenada do produto k , no estágio i e período t .

O balanceamento de fluxo no ponto de montagem \bar{h} é garantido pela seguinte equação de balanço:

$$x(\bar{h}, t, k) = q(\bar{h}, k) * x(i, t, k), \quad t = 1, 2, \dots, T \quad (6)$$

onde $\bar{h} \in \Omega$ é o conjunto dos pontos de montagem ou acoplamento e $x(\bar{h}, t, k)$ representa a quantidade de peças necessárias, da linha \bar{h} no tempo t , a serem acopladas às $x(i, t, k)$ peças processadas no estágio i e tempo t , a uma dada relação de acoplamento $q(\bar{h}, k)$.

Estes produtos podem ter árvores de produto idênticas ou diferentes. Assim sendo, é importante garantir que, dentro do horizonte de planejamento, o processamento de peças diferentes em uma mesma máquina não leve a violação de capacidade de produção.

	sublinha 3																	sublinha 1						sublinha 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	1			
11,2																											-1	-1	1	1		
1		q(8)																														
2											q(8)																					
3			q(11)																												-1	
4														q(11)																	-1	

Figura 10 – Estrutura da matriz de restrições.

Para garantir que a soma de alocações de produção de itens individuais não ultrapassa a capacidade de um recurso, é escrita as equações 7 (KENNINGTON, 1980).

$$\sum_{i \in I(j)} x(i, t, k) \leq X(j, t)$$

$$\sum_{i \in I(j)} y(i, t, k) \leq Y(j, t) \quad (7)$$

onde $X(j,t)$ é a capacidade máxima do recurso j no tempo t , e $x(i,t,k)$ é quantidade a ser produzida do item k pelo recurso j no tempo t ; $y(j,t,k)$ é a quantidade a ser armazenada, com capacidade máxima $Y(j,t)$, que antecede o recurso j e no tempo t . O conjunto $I(j)$ contém os tipos de peças que a serem processadas pela máquina j .

A representação matricial para a Figura 11, considerando o caso multiproduto, pode ser escrita como:

$$A = \begin{bmatrix} A_1 & & \\ & A_2 & \\ & & A_3 \\ \hline Q_1 & Q_2 & Q_3 \\ \hline N_1 & N_2 & N_3 \end{bmatrix} \quad (8)$$

onde A_i são matrizes incidência nó-arco e Q_i são as matrizes que definem as relações de montagem ou acoplamento entre as linhas concorrentes. As submatrizes N_i representam as restrições de capacidade mútua de transformação. A matriz 'A', resultante para o caso multiproduto, continua possuindo uma estrutura muito particular que deve ser aproveitada pela técnica de solução.

2.6 Caracterização do Problema

De uma maneira geral, o problema de planeamento da produção apresenta as seguintes características:

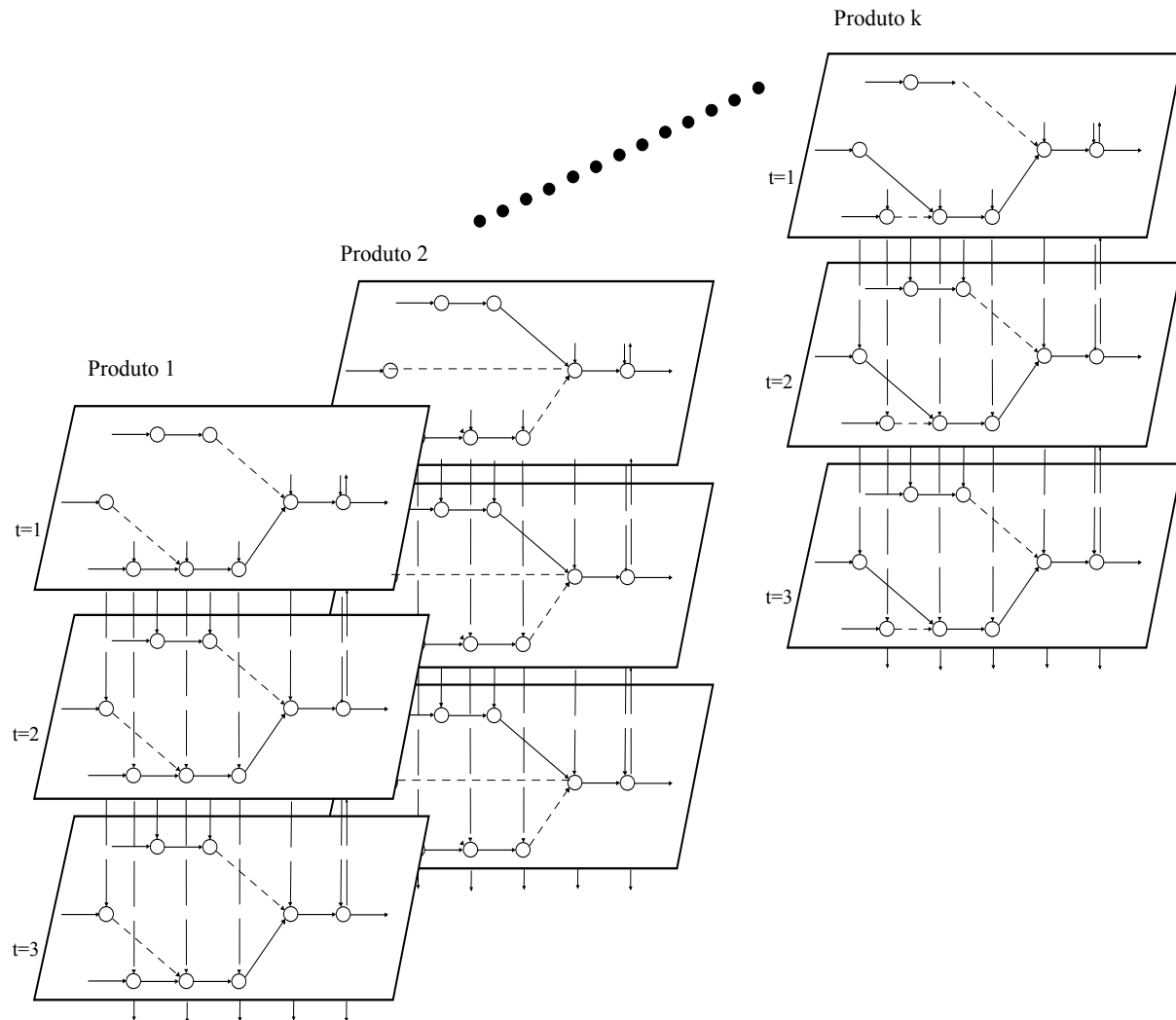


Figura 11 – Sistema multiproduto multiestágio.

- o horizonte de planejamento é discretizado em períodos (hora, turno, dia);
- cada estágio de produção pode conter um ou mais grupos de máquinas;
- são conhecidas as capacidades de produção e de armazenagem de matéria-prima para atendimento à demanda, dentro do horizonte de planejamento; e
- a capacidade de cada grupo de máquinas pode ser diferente ao longo do período de planejamento.

Segundo os custos unitários de matéria-prima, de operação, de estocagem e de atrasos na entrega dos produtos e com o retorno por

unidade de demanda atendida, o planejamento deve gerar 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 quantidade de demanda atendida com atraso; e
- o retorno dentro do período de planejamento.

3. 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 maximização do retorno líquido. Tal formulação permite ao planejador considerar implicitamente objetivos conflitantes manipulando-os pelos coeficientes da função custo. Por outro lado, diversas outras funções objetivo expressas em grandezas diferentes de custo são largamente utilizadas na literatura (NORBIS, 1988 e TRZASKALIK, 1990). Dentre elas pode-se citar:

- minimização dos custos de produção (compra de matéria-prima, produção, armazenagem, demanda em atraso) (LEWIS *et al.*, 1992),
- minimização do número de peças armazenadas (*work-in-process inventories*),
- minimização do tempo total de produção (*makespan*),
- minimização do número de peças de *backorder*, etc.

A consideração de todas estas funções leva a um problema multiobjetivo (FERNANDES, 1999).

4. Formulação Geral

A formulação geral do problema multiproducto, multiestágio, multiperíodo como um problema de fluxo em rede com restrições adicionais é a seguinte:

$$\begin{aligned} & \text{Max } f(x) \\ & \text{s.a.: } A x \leq b \\ & \quad Sx \leq d \\ & \quad \underline{x} \leq x \leq \bar{x} \end{aligned} \quad (9)$$

onde $f(x)$ representa a função objetivo, A a matriz de incidência nó-arco, S as restrições adicionais de capacidade e de acoplamento, b e d os coeficientes do lado direito das restrições. Embora a formulação completa apresentada seja um problema de programação linear (PL), em aplicações práticas o tamanho do problema torna ineficiente a utilização direta do algoritmo

simplex. Isto, principalmente pela natureza temporal do problema e pela 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.

A presença das restrições lineares adicionais impede o uso direto de técnicas de grafo. A seguir mostra-se como transformar o modelo acima em um problema de fluxo puro em redes.

5. Técnica de Solução

O problema de otimização descrito na seção anterior pode ser resolvido por técnicas gerais de otimização ou por códigos baseados em técnicas especializadas de fluxo em rede. As técnicas gerais de otimização utilizadas por códigos genéricos estão baseadas nas teorias gerais do algoritmo Simplex ou Simplex Convexo (LUENBERGUER, 1973), para problemas de otimização de grande dimensão. O procedimento seguido por estas técnicas envolve inversão de matrizes completas, operações matriciais e vetoriais genéricas que normalmente são dispendiosas num processo iterativo para problemas de grande dimensão.

Em geral, os problemas reais são de grande porte, devido principalmente à natureza temporal e à quantidade de produtos que compartilham o mesmo sistema produtivo. Logo, requisitos como armazenamento de variáveis e tempo de execução computacional são fatores essenciais na escolha de um método de solução. Para tanto, um meio de satisfazer estes requisitos é aproveitar a estrutura especial do problema. Um método que pode tirar proveito deste tipo de estrutura é o GUB (*Generalized Upper Bounding*) especializado para estruturas de rede. Este método é também conhecido como algoritmo primal para fluxo em redes com restrições adicionais. O algoritmo empregado é o NETSIDE (KENNINGTON 1990), código em Fortran para resolver problemas de fluxo em redes com restrições adicionais e que possui bom desempenho quando o número

de restrições adicionais é menor que o número de nós do grafo, caso específico do problema de planejamento da produção aqui formulado.

A seguir é apresentado um procedimento que permite resolver um problema de fluxo em redes com restrições adicionais de maneira eficiente. Este procedimento considera as restrições adicionais por técnicas de penalidade e é chamado de redes de transporte (RETRA).

5.1 Técnica de Penalidade

No problema (9), as matrizes de restrições definem dois conjuntos:

$$\begin{aligned} X &= \{x \mid Ax = b, \underline{x} \leq x \leq \bar{x}\} \\ \Omega &= \{x \mid x \in X, Sx \leq d\} \end{aligned} \quad (10)$$

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

Considerando as restrições lineares adicionais por uma função de penalidade, o problema pode ser resolvido como um problema de fluxo puro em redes. Uma penalidade linear a ser adicionada à função objetivo tem a forma:

$$\begin{aligned} p_j(y_j) &= \max\{0, y_j\} \\ \text{onde } y_j &= S_j x - d_j, e_j = 1, \dots, s. \end{aligned} \quad (11)$$

Todavia, a utilização da função de penalidade apresenta um problema de descontinuidade. Para evitar as dificuldades de não-diferenciabilidade, ZENIOS *et al.* (1995) sugerem suavizar a penalidade exata. Para a expressão exata (11), a função suave de penalidade linear-quadrática é definida como:

$$\phi(\varepsilon, y) = \begin{cases} 0 & , \text{ se } y \leq 0 \\ y^2 / 2\varepsilon & , \text{ se } 0 \leq y \leq \varepsilon \\ y - \varepsilon / 2 & , \text{ se } y \geq \varepsilon \end{cases} \quad (12)$$

A Figura 12 ilustra a função de penalidade $\phi(\varepsilon, y)$.

A função objetivo modificada para o problema será:

$$\Phi(x) = f(x) + \mu \sum_{j=1}^s \phi(\varepsilon, y_j) \quad (13)$$

onde: μ e ε são números reais positivos e $y_j = S_j x - d_j$, para $j = 1, \dots, s$.

A seguir é apresentado um algoritmo para solução de problemas que utilizam a função de penalidade para eliminar as restrições lineares adicionais. Para maiores detalhes e considerações sobre a inicialização e atualização de μ e ε além da convergência do algoritmo veja ZENIOS *et al.* (1995).

5.2 Algoritmo de Penalidade Linear-Quadrática (PLQ)

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

$$\begin{aligned} \text{Min } & \Phi(x) \\ \text{s.a: } & Ax = b \\ & \underline{x} \leq x \leq \bar{x} \end{aligned}$$

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

Passo 1: Se $x^k \in \Omega$, fim. Se não, faça $k \leftarrow k + 1$ e atualize os parâmetros de penalidade μ e ε .

Passo 2: Usando a violação $y_j = S_j x - d_j$, para $j = 1, \dots, s$, minimize (13). Retorne ao Passo 1.

O maior esforço computacional deste algoritmo é a minimização do problema (13) no Passo 2, por ter função objetivo não-linear e não-separável. Por isto será utilizado um algoritmo de linearização por decomposição simplicial (DS), proposto por Von HOHENBALKEN (1977), e que tem recebido muito interesse durante a última década. Este interesse é motivado por sua simplicidade, robustez, e habilidade em obter boas aproximações para problemas

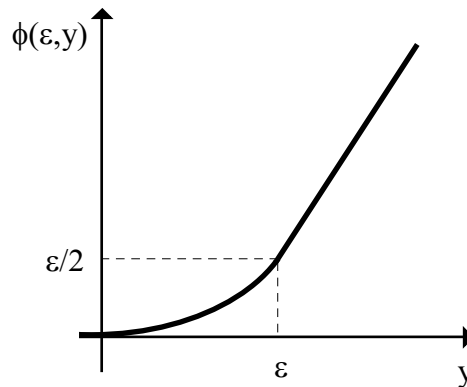


Figura 12 – A Função de penalidade $\Phi(\epsilon, t)$.

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

O subproblema linearizado é definido como:

$$\text{Min}_{y \in X} \nabla \Phi(x^k) \cdot y$$

onde x^k é a solução na k -ésima iteração. 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 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:

$$\text{Min}_w \Phi(D^k w + y^s), \text{ onde: } D^k w + y^s = x^k.$$

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 $\Phi(x^k) < \Phi(x^{k-1})$, e será utilizada na próxima iteração do subproblema linearizado. Este procedimento termina quando a solução y^* de uma

iteração do subproblema linearizado satisfaz o critério de otimalidade $\nabla \Phi(x^k)(y^* - x^k) \geq 0$.

5.3 Algoritmo de Decomposição Simplicial

Passo 0 (Inicialização): seja $x^1 \in X$ uma solução factível. Faça:

$$Y^1 = \{x^1\}, v_1 = 1, V^1 = \{v_1\}, k = 1.$$

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

$$\text{Min}_{y \in X} \nabla \Phi(x^k) \cdot y$$

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

Se não, faça: $Y^k \leftarrow Y^k \cup \{y^*\}$.

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

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

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

A seguir, faça:

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

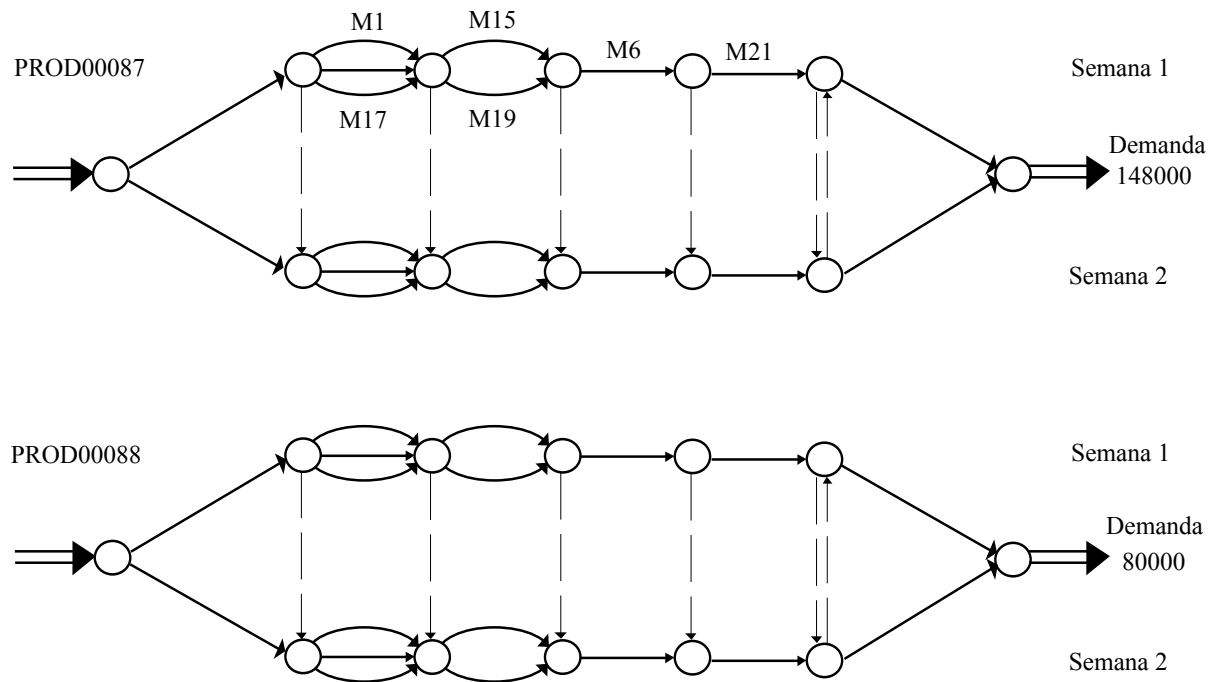


Figura 13 – Representação por grafo do exemplo.

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

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

Retorne ao Passo 1.

O Passo 1 deste algoritmo utiliza uma aproximação linear para o problema não-linear, e pode ser facilmente resolvido por algoritmos de fluxo em rede. O problema mestre não-linear do Passo 2 é de tamanho muito menor que o problema original, além de ter uma estrutura muito mais simples, mas pode ser mal condicionado. Para maiores detalhes, veja LUENBERGER (1984). Apesar deste problema mestre ser de dimensão muito menor do que o problema original, o tamanho do conjunto Y^k pode tornar o problema não realístico. 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 . Quando $r=1$, o método é o algoritmo de Frank e Wolfe, e quando r é irrestrito torna-se o método de decomposição simplicial, aqui descrito.

6. Aplicação

Dois exemplos serão utilizados na análise do método de solução proposto na seção anterior. Os testes foram realizados em um microcomputador Pentium 200 Mhz. Inicia-se um exemplo simplificado (ex02) com apenas dois produtos (PROD00087 e PROD00088) e uma linha com 4 estágios de produção. O período de planejamento é para duas semanas, e tem como objetivo analisar a necessidade de armazenagem interperíodos.

Na Figura 13, que ilustra a representação por grafo deste exemplo, são apresentadas as demandas de cada produto em cada semana do planejamento. Arcos em paralelo, dentro do mesmo estágio de produção, representam máquinas alternativas para execução da tarefa.

Tabela 2 – Dados de produção.

estágio	máquina	PROD00087		PROD00088		disponibilidade de máquina (% semana)
		taxa de produção (peça/semana)	custo de produção (\$/peça)	taxa de produção (peça/semana)	custo de produção (\$/peça)	
1	1	144000	1,00	144000	1,00	100
1	2	48000	1,00	48000	1,00	100
1	17	48000	1,00	48000	1,00	100
2	15	96000	1,00	80000	1,00	100
2	19	48000	1,00	48000	1,00	100
3	6	400000	1,00	600000	1,00	100
4	21	320000	1,00	192000	1,00	100

As taxas e custos de produção para cada máquina, além da disponibilidade de máquina para cada semana, são fornecidos na Tabela 2. O custo de armazenagem e da penalidade por atendimento à demanda em atraso de qualquer produto, é constante e igual a \$ 1,00.

A Figura 14 e a Tabela 3 mostram os resultados obtidos pela técnica de penalidade. Note que devido à limitação da capacidade de produção (o estágio dois, composto pelas máquinas 15 e 19, está com capacidade saturada) houve demanda atendida em atraso para os dois produtos. Ou seja, a semana dois produziu para atender demanda da primeira semana. Outra importante observação é que os resultados mostram níveis de produção nas máquinas como valores não inteiros. Como os níveis de produção são valores consideravelmente grandes, uma aproximação de valores reais por números inteiros não resulta em grande erro. Este exemplo mostra que um importante resultado obtido com a aplicação da programação matemática em sistemas reais de manufatura é a determinação da produção interperíodos, ou seja, antecipação da produção para atender demanda futura por decisão econômica (custos de compra de matéria-prima ou produção menores) ou produção para atendimento de demanda em atraso devido a insuficiência de capacidade ou disponibilidade limitada de matéria prima.

Para efeito de comparação além da técnica sugerida neste artigo foram utilizados outros enfoques de solução: GAMS (BROOKE *et al.*, 1992) e Técnica de Decomposição Primal, implementada no algoritmo NETSIDE (KENNINGTON, 1990).

O exemplo que considera um sistema de produção real (ex68) com todos os produtos foi analisado para se obter a solução ótima para o planejamento de duas semanas. Este exemplo tem 609 nós e 1405 arcos. Os resultados estão apresentados na Tabela 4. A técnica de penalidade (RETRA) possibilita haver somente iterações de grafo enquanto as outras exigem a inversão ou atualização de matrizes. Assim, o número de iterações em grafo é muito maior mas com custo 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.

Ainda com a finalidade de comparar o desempenho das técnicas citadas, outros cenários foram estudados. A Figura 15 ilustra os valores da função objetivo obtidos pelo RETRA para o exemplo ex68 e diferentes fatores de demanda. A curva apresenta um ponto de quebra para o fator de demanda de 1,3. Isto significa a ocorrência de gargalos (saturação) relacionada a

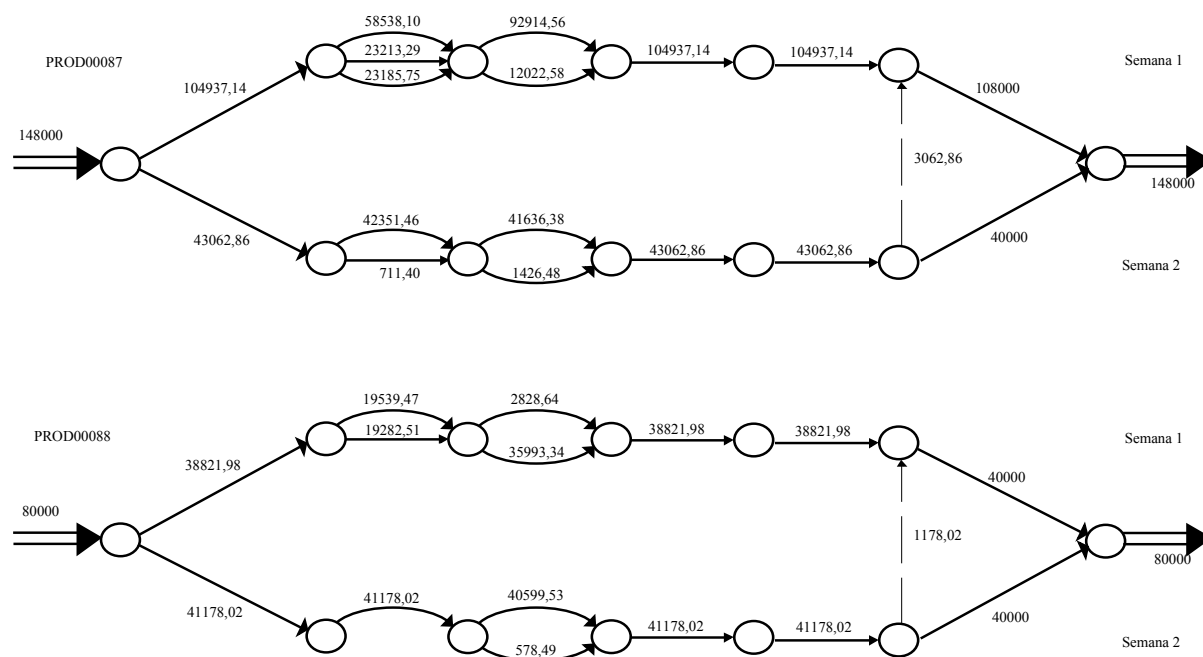


Figura 14 – Resultados do exemplo (ex02).

Tabela 3 – Resultados do exemplo ex02.

	Utilização das máquinas (%)		
	máquina	Semana 1	Semana 2
FO = 916240,88	1	54,22	58,00
m = 240058,368	2	88,53	1,48
e = 0,0020616	17	48,30	0,00
200 iterações	15	100,32	94,12
21 passos de penalidade	19	100,03	4,17
CPU = 0,16 seg.	6	32,70	17,62
	21	53,01	34,90

Tabela 4 – Resultados do exemplo completo.

Exemplo	Técnica	FO	ITER	T. 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

alguma (s) linha (s) de produção. Note que outras linhas apresentam folga que possibilita o aumento de produção.

A Figura 16 exibe o comportamento dos três algoritmos em função de variação nos fatores de demanda. O algoritmo RETRA apresenta o

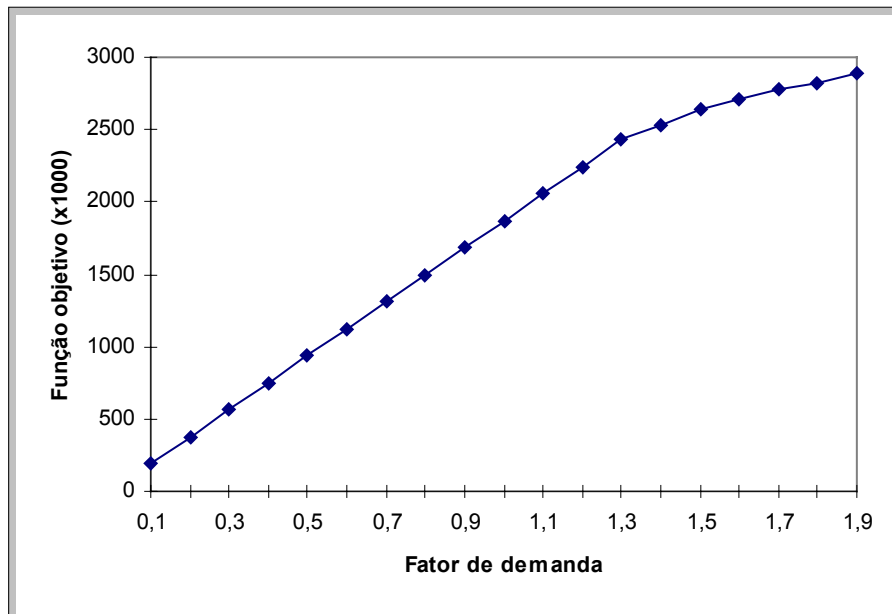


Figura 15 – Função objetivo em função da demanda.

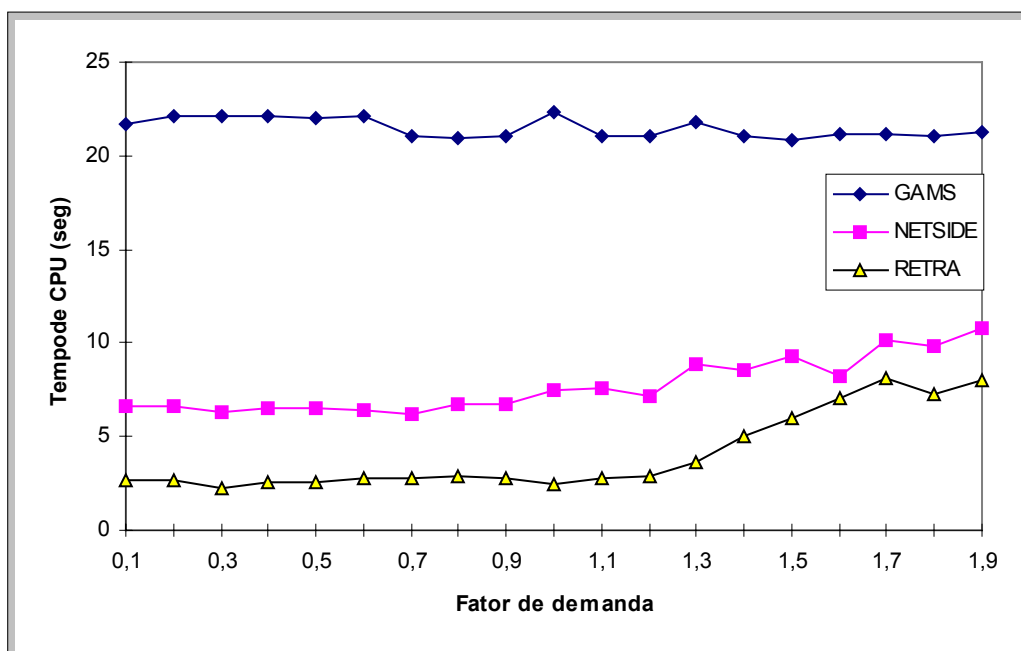


Figura 16 – Desempenho em função da demanda.

melhor desempenho. Há um aumento significativo de tempo para o sistema sobrecarregado, característica dos algoritmos de penalidade. Contudo este caso deve ser raro em aplicações

práticas. 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.

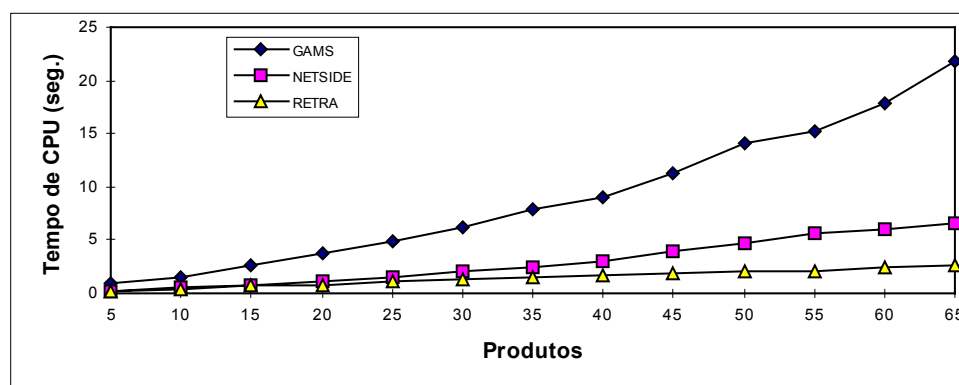


Figura 17 – Performance em função do número de produtos.

Tabela 5 – Resultados com o direcionamento da solução.

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

A Figura 17 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 por ser um pacote geral 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, ao aumento “quase linear”, se deve ao fato de a cada adição de um novo produto k resulta na adição de uma nova submatriz A_k e em uma 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 sistema de manufatura, os grafos de cada produto k têm dimensão razoavelmente iguais o que justifica também 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.

6.1 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 na qual apenas os arcos de produção formam a árvore ou base inicial, tem-se os resultados na Tabela 5. Note que o direcionamento da solução efetivamente reduziu o número de iterações e conseqüentemente o tempo de CPU.

7. Conclusões

Este artigo apresentou o planejamento da produção de sistemas de manufatura multiprodutos como um modelo de fluxo em redes com restrições adicionais. Mostrou que se as restrições adicionais fossem tratadas por técnicas de penalidade, o resultado seria um problema de fluxo em rede com função objetivo

quadrática. Apresentou a técnica de decomposição simplicial como uma forma eficiente de solução. Utilizando dois exemplos comparou os

resultados com os algoritmos de decomposição primal e o GAMS, mostrando a grande vantagem do algoritmo sugerido.

Referências Bibliográficas

- BROOKE, A.; KENDRICK, D. & MEERAUS, A.:** "GAMS – A User's Guide, Release 2.25", Boyd & Fraser Publishing Company, Massachusetts, 1992.
- CARVALHO, M.F. & SOARES, S.:** "Optimal Active Power Dispatch by Network Flow Approach", *IEEE Trans on PS*, vol.3, n.4, Nov 1988.
- CARVALHO, M.F.; FERNANDES, C.A.O.; FERREIRA, P.A.V. & TAKAHASHI, M.T.:** "Assembly Production Planning by Network Flow Approach", 9TH Symposium on Information Control in Manufacturing – INCON'98, Nancy - Metz, France, 1998.
- DEMBO, R.S.; MULVEY, J.M. & ZENIOS, S.A.:** "Large-scale nonlinear network models and their application", *Operations Research*, vol.37, n.3, pp.353-372, 1989.
- FERNANDES, C.A.O.; CARVALHO, M.F.H.; FERREIRA, P.A.V. & FILHO, O.S.S.:** "Multi-commodity Manufacturing Flow Problems by a Penalty Function Approach", *IFAC Large Scale Systems*, London, UK, 1995.
- FERNANDES, C.A.O.:** "Planejamento da Produção da Manufatura por Fluxo em Redes com Restrições Adicionais", Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 1999.
- FLORIAN, M.:** "Nonlinear Cost Network Models in Transportation Analysis", *Math. Program Study*, 26, pp.167-196, 1986.
- FRANCO, P.C.; CARVALHO, M.F. & SOARES, S.:** A Network Flow Model for Short-Term Hydro-Dominated Hydrothermal Scheduling Problems, *IEEE Trans. on Power Systems*, vol., PWRS-9, No.2, May 1994.
- JENSEN, P.A. & BARNES, W.J.:** "Network Flow Programming", Robert Krieger Publishing Company, FL, 1987.
- GLOVER, F.; JONES, G.; KARNEY, P.; KLINGMAN, D. & MOTE, J.:** "An integrated production, distribution and inventory planning system", *Interfaces*, vol.9 n.5, pp.21-36, 1979.
- HEARN, D.W.; LAWPHONGPANICH, S. & VENTURA, J.A.:** "Restricted Simplicial Decomposition: Computations and Extensions", *Mathematical Programming Study*, 31, pp.99-118, 1987.
- JAASKELAINEN, V.:** "A goal programming model for aggregate production planning", *Swedish Journal of Economics*, 2, 14-29, 1969.
- KENNINGTON, J.L. & HELGASON, R.V.:** "Algorithms for Network Programming", John Wiley & Sons, New York, 1980.
- KENNINGTON, J.L. & WHISMAN, A.:** "Netside User's Guide", Department of Computer Science and Engineering, Southern Methodist University, Dallas, 1990.
- LEWIS, HOLLY S.; SWEIGART, J.R. & MARKLAND, R.E.:** "Master Scheduling in Assemble-To-Order Environments: A Capacited Multiobjective Lot-Sizing Model", *Decision Sciences*, vol.23, pp.21-43, 1992.
- LUENBERGER, D.G.:** "Linear and Nonlinear Programming", Addison-Wesley Publishing Company, 1984.
- MULVEY, JOHN M.; ZENIOS, S.A. & AHLFELD, D.P.:** "Simplicial Decomposition for Convex Generalized Networks", *Journal of Information & Optimization Sciences*, vol.11, n.2, pp.359-387, 1990.
- MURVAY, J.M.:** "Nonlinear Network Models in Finance", *Ad. Math. Program Financial Planning*, 1, pp.253-271, 1987.
- NORBIS, M.I. & SMITH, J. MACGREGOR:** "A multiobjective, multi-level heuristic for dynamic resource constrained scheduling problems", *European Journal of Operational Research*, vol.33, no.1, p.30-41, 1988.
- ROSENTHAL R.E.:** "A nonlinear flow Algorithm for Maximization of Benefits in a Hydroelectric Power System", *Operation Research*, 29, pp.762-786, 1981.
- TRZASKALIK, T.:** "Multi-objective, multi-period planning for a manufacturing plant", *Engineering Costs and Production Economics*, vol.20, Iss.2, p.113-20, 1990.

Von HOHENBALKEN, B.: “Simplicial Decomposition in Nonlinear Programming Algorithms”, *Mathematical Programming*, 13, 49-68, 1977.

ZENIOS, S.A.; PINAR, M.Ç. & DEMBO, R.S.: “A Smooth Penalty Function Algorithm for Network-Structured Problems”, *European Journal of Operational Research*, 83, 220-236, 1995.

MANUFACTURING PRODUCTION PLANNING BY NETWORK FLOW

Abstract

This paper presents some manufacturing planning problems that have a special network structure. This characteristic encourages the utilization of dedicated tools to solve actual industry optimization problems that are of large dimension and which allow a more detailed representation of the physical system. A comparison shows the best performance of these algorithms in relation to general algorithms.

Key words: production planning, mathematical programming, network flow, optimization.