UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA DE SISTEMAS

OTIMIZAÇÃO DO PLANEJAMENTO DA PRODUÇÃO EM SISTEMAS MULTIESTÁGIOS

Este exemplar corresponde à redação final da tese defendida por Regina Esther Berretta e aprovada pela Comissão Julgadora em 08/02/1993.

Orientador

REGINA ESTHER BERRETTA

Orientadores:

Prof. Dr. Paulo Morelato França
Prof. Dr. Vinicius Amaral Armentano

Tese apresentada à Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas - UNICAMP, como parte dos requisitos exigidos para a obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA.

- FEVEREIRO 1993 -

AGRADECIMENTOS

A todos que colaboraram na realização deste trabalho e em especial

- a Paulo M. França e Vinicius A. Armentano, pela orientação durante o desenvolvimento desta tese e estímulo à pesquisa;
- aos professores e funcionários do Densis, pelo apoio;
- aos meus pais Carmem e Ledyr pelo constante interesse e carinho.
- ao Tito pelo amor e apoio.
- a Cintia Rigão Scrich, Franklina Maria B. de Toledo e Zake Sabbag Neto pelas valiosas discussões sobre este trabalho.
- a Débora P. Ronconi e Marcelo Nishi pela participação em etapas deste trabalho.
- ao Walcir pelo apoio computacional.
- ao Conselho Nacional de Desenvolvimento Científico e Tecnológico pelo apoio financeiro.

RESUMO

Problemas multiestágios de dimensionamento de lotes consistem na determinação das quantidades a serem produzidas em diferentes períodos de tal modo que a demanda seja atendida. Consideram-se restrições nos recursos disponíveis e estrutura de produtos, ou seja, os produtos podem depender da compra ou produção de certos componentes. O modelo apresentado utiliza o conceito de estoque de escalão e inclui custo e tempo de preparação para que a produção de certos componentes possa ocorrer. Para a resolução deste problema, um método heurístico é desenvolvido e testado computacionalmente.. Técnicas de Busca Tabu são incorporadas ao procedimento de solução.

ABSTRACT

Lot-sizing multistage problems consist in the determination of the quantities to be produced in different periods of time to meet forecast demand. The model considers capacity restrictions on resources and product structure, i.e., the products may depend on the purchase or the production of certain components. The model presented uses the concept of echelon stock and includes setup cost and time to the ocurrence of production of certain components. A heuristic method is developed for solving the problem and computationally tested. The solution procedure includes Tabu Search techniques.

ÍNDICE

INTRODUÇÃO

Uma empresa, basicamente, é dividida em áreas de marketing, finanças e produção, como se observa na figura 1. O **marketing** tem como objetivo detectar preferências do consumidor, intervir na definição do projeto e no desenvolvimento de um produto, estabelecer previsão de demanda e elaborar o processo de distribuição. Nas **finanças** concentram-se as responsabilidades na obtenção de capital externo, administrando este capital no sentido da operacionalidade da empresa (aquisição de bens, como equipamentos, meios de transporte, armazéns, etc.), e determinar políticas de estoque. A **produção** responde pela transformação da matéria prima em produtos, provendo um conjunto específico de serviços.

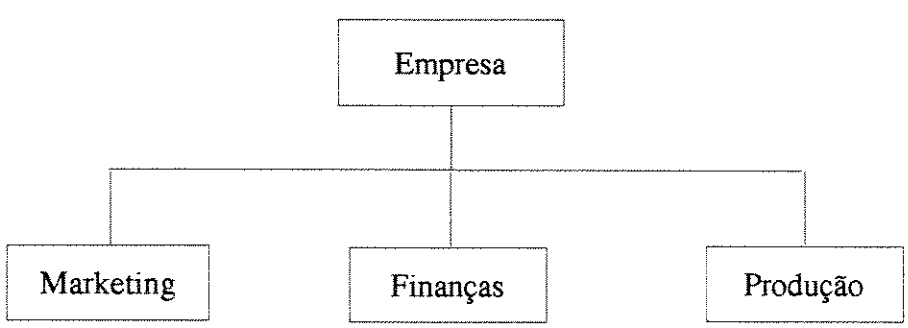

Figura 1. Principais áreas de uma empresa.

As duas primeiras áreas (marketing e finanças) ocupam-se das necessidades ou dos resultados da área de produção. Assim tem-se a necessidade fundamental de um bom gerenciamento deste setor. Trata desse gerenciamento o sistema de planejamento e controle da produção (PCP), que coordena todas as atividades desde a aquisição das matérias-primas até a entrega dos produtos acabados. A estrutura organizacional do sistema PCP reúne três níveis de planejamento distintos: estratégico, tático e operacional, como se observa na figura 2.

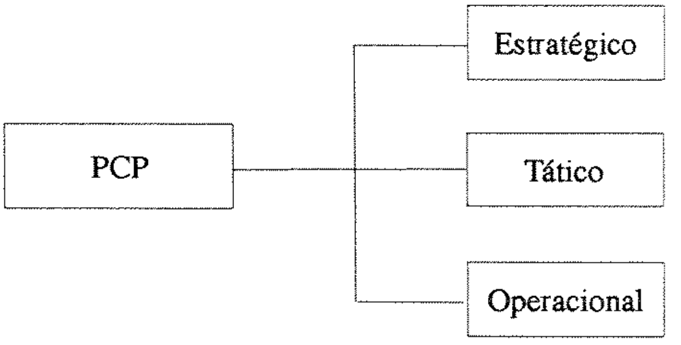

Figura 2. Níveis de planejamento de um PCP.

Ao **planejamento estratégico** cabe decidir as políticas adequadas na realização dos objetivos de longo prazo da empresa. Deve especificar o plano mestre de produção (PMP), os recursos necessários para atingir aqueles objetivos, as estratégias para aquisição de equipamentos e localização de fábricas e o planejamento da capacidade de médio prazo. Tais decisões são tomadas conjuntamente pelas áreas de marketing, finanças e produção.

O **planejamento tático** está relacionado com a implementação das estratégias definidas em nível superior (estratégico), planejando os fluxos de materiais e a capacidade para dar suporte ao plano global. Tem como objetivos o gerenciamento da demanda, o planejamento da produção e da utilização dos recursos (homens-hora, horas-máquina) e decisões sobre níveis de estoque. O horizonte de planejamento desse nível é de médio prazo.

O **planejamento operacional** trata das decisões do dia a dia, tendo como função executar os planos definidos anteriormente, ou seja, a programação detalhada da produção e compras, que consiste em seqüenciar os pedidos no centro de trabalho, a administração dos estoques, o controle da qualidade, a alocação dos pedidos de clientes em máquinas, a compra de componentes e matéria-prima e o programa de distribuição.

O trabalho desenvolvido nesta tese se concentra nas tomadas de decisões tendo em vista o planejamento tático de um PCP. O sistema central nesse nível está no planejamento da produção. Dado um horizonte de tempo, este pode ser dividido em períodos de planejamento de tal maneira que, num dado período um produto ou não é produzido ou é produzido em um lote cujo tamanho pode variar. O problema de dimensionamento de lotes consiste em determinar a quantidade e o período para produzir cada item, de tal modo que a demanda seja atendida. Outros parâmetros a serem considerados neste problema são: a estrutura dos produtos, que podem depender da compra ou produção de certos componentes; os níveis de estoque; a capacidade de produção do sistema e o tempo de produção (*lead time*) de cada produto ou componente deste, onde entende-se por tempo de produção uma média entre o tempo de produção e preparação, tempo para mover um item para o próximo estágio e tempo de espera [BILLINGTON et. al., 1983].

Uma ferramenta utilizada neste planejamento é o MRP (*Material Requirements Planning* - Planejamento de Necessidade de Materiais). Seu objetivo é converter o plano mestre de produção (PMP) num plano de fabricação de produtos finais e na produção (ou compra) de seus possíveis componentes. Baseado em previsões de demanda de cada produto final ao longo de um horizonte de planejamento de T períodos, na lista de materiais que define a estrutura do produto e no tempo de produção de cada item (*lead time*), o MRP fornece um planejamento sincronizado da produção do produto final e dos componentes que o compõem, informando a quantidade correta no período correto a ser produzida (ou comprada), de forma a poder atender a demanda prevista em cada período.

Em sua forma básica, o MRP assume que não há restrição de capacidade, isto é, qualquer quantidade de produção é possível. Se a capacidade disponível é excedida, é feita a transferência de parte da produção ou a adição de capacidade (*overtime*), na tentativa de ajustar o plano. Mas a transferência pode causar planos impraticáveis. Além disso, o plano gerado pela aplicação do MRP não fornece um plano de produção no sentido do seu custo ser o menor possível, ou seja, não são considerados os custos envolvidos na produção, no estoque e na preparação.

Decidir o tamanho dos lotes num sistema de produção, onde a capacidade de produção é limitada, objetivando um custo mínimo, é uma tarefa complexa. A inclusão de tempos e custos de preparação faz com que, matematicamente, o problema seja modelado como um problema inteiro-misto, de difícil resolução. Além disso, considerando que os produtos envolvidos no sistema possuem estrutura de produção (há produtos que dependem da produção, ou compra, de outros), o planejamento de cada produto se torna dependente do planejamento de componentes distintos. Assim, a maioria dos estudos feitos nessa área, envolve simplificações do problema, e ainda, a formulação de métodos heurísticos na resolução de tais problemas.

O objetivo deste trabalho é o desenvolvimento de um método heurístico, utilizando técnicas de Busca Tabu, com o propósito de fornecer um plano de produção, isto é, determinar quanto, quando e o que produzir, de modo que um menor custo esteja envolvido em um problema de dimensionamento de lotes com capacidade limitada de produção. Considera-se que os produtos tenham estrutura de produção e que o tempo de produção de cada produto e componente seja zero.

A tese é dividida em quatro capítulos. No capítulo 1 são apresentados modelos matemáticos para o problema de dimensionamento de lotes com capacidade finita de produção, com e sem estrutura de produto, além de uma revisão na literatura dessa área. No capítulo 2 são fornecidos os fundamentos básicos de Busca Tabu, uma técnica utilizada no desenvolvimento do método proposto. O desenvolvimento da heurística para a resolução do problema é apresentado no capítulo 3. No capítulo 4 são apresentados os resultados computacionais realizados, as conclusões e algumas sugestões para pesquisas futuras.

CAPÍTULO 1

FORMULAÇÃO DO PROBLEMA

1.1. Introdução

Este capítulo apresenta modelos matemáticos para o problema de dimensionamento de lotes com restrição de capacidade.

Seja um horizonte de tempo com T períodos, onde se deseja planejar a produção de N itens. A demanda de cada item i, no período t, é conhecida. O problema de dimensionamento de lotes tem por objetivo determinar em que períodos produzir e a quantidade a ser produzida de cada item (tamanho do lote), de modo a atender a demanda com um custo total mínimo. Os custos considerados são: produção, estoque e preparação, sendo este último um custo fixo incorrido quando se decide produzir o item i no período t. Considera-se também que a quantidade dos recursos utilizados na preparação e produção, em cada período, é limitada.

Primeiramente, é dada uma formulação para um único estágio. Em seguida, apresentam-se formulações para o problema com vários estágios.

1.2. Modelo para um único estágio

Diz-se haver um único estágio quando os itens a serem produzidos são independentes, isto é, nenhum segue ou antecede um outro.

A produção de um item no período t implica em um custo de produção, por unidade, e um custo fixo de preparação (*setup cost*), com utilização de parte dos recursos para produzir e parte para preparar. Há também um custo para cada unidade estocada no final de um período, não sendo permitido estoque negativo.

Os custos e a demanda de cada item são variantes no tempo, assim como a quantidade de recursos a ser utilizada.

Este problema pode ser formulado como um modelo de Programação Inteira-mista da seguinte forma [TRIGUEIRO et al., 1989]:

$$\min \sum_{i=1}^{N} \sum_{t=1}^{T} \left(h_{it} I_{it} + c_{it} x_{it} + s_{it} y_{it} \right) \qquad [1]$$

$$I_{i,t-1} + x_{it} - I_{it} = d_{it} \qquad i = 1,\ldots,N \; ; t = 1,\ldots,T \qquad [2]$$

$$\sum_{i=1}^{N} \left(v_{ikt} x_{it} + f_{ikt} y_{it} \right) \leq CAP_{kt} \qquad k = 1,\ldots,K \; ; t = 1,\ldots,T \qquad [3]$$

$$x_{it} \leq M_{it} y_{it} \qquad i = 1,\ldots,N ; \; t = 1,\ldots,T \qquad [4]$$

$$y_{it} = 0\,,1 \qquad i = 1,\ldots,N ; \; t = 1,\ldots,T \qquad [5]$$

$$x_{it} \; e \; I_{it} \geq 0 \qquad i = 1,\ldots,N ; \; t = 1,\ldots,T \qquad [6]$$

onde

N - número de itens.
T - número de períodos.
K - número de recursos.
M_{it} - limite superior para x_{it}.

x_{it} - quantidade de produção do item i no período t.

I_{it} - nível de estoque do item i ao fim do período t.

y_{it} - 1 se houver produção do item i no período t.
0 caso contrário.

c_{it} - custo de produção unitário do item i no período t.

s_{it} - custo de preparação do item i no período t.

h_{it} - custo unitário de estocagem do item i no período t.

f_{ikt} - quantidade do recurso k utilizado na preparação do item i no período t.

v_{ikt} - quantidade unitária do recurso k utilizado na produção do item i no período t.

CAP_{kt} - disponibilidade do recurso k no período t.

d_{it} - demanda do item i no período t.

A função objetivo **[1]** minimiza a soma dos custos de produção, estoque e preparação dos lotes de N itens ao longo de T períodos do horizonte de planejamento. Em **[2]** tem-se a equação de balanço entre as variáveis de estoque e produção, onde é determinado que a demanda de um item i em um período t (d_{it}) será suprida pela combinação da produção em t (x_{it}) e pelo estoque em t-1 ($I_{i,t-1}$) e que o excedente ficará estocado (I_{it}). A equação **[3]** representa a utilização do recurso k na preparação ($f_{ikt}\ y_{it}$) e produção ($v_{ikt}\ x_{it}$) de todos os itens em um período t, sendo limitada pela quantidade disponível do recurso no período (CAP_{kt}). A equação **[4]** fornece um limite superior para x_{it}, caso $y_{it} = 1$ e assegura $x_{it} = 0$, caso $y_{it} = 0$.

1.2.1 Métodos de resolução

Quando a capacidade de produção é considerada ilimitada, este problema pode ser resolvido, para cada item, pelo algoritmo ótimo de Programação Dinâmica, desenvolvido por Wagner e Whitin [WAGNER e WHITIN, 1958]. Se tempos e custos de preparação são ignorados, o problema pode ser resolvido por Programação Linear, ainda que capacitado.

[FLORIAN, LENSTRA e RINNOY KAN, 1980] mostraram que o problema com um único item e capacidade finita de produção é ***NP-hard*** [GAREY e JOHNSON, 1978] (mesmo com demandas iguais, custo zero de estocagem e tempos de preparação iguais a zero), portanto, o problema multi-item também é considerado ***NP-hard.***

Dizer que um problema é da classe NP significa que **não** existe um algoritmo, onde o número de operações elementares necessário para a obtenção de uma solução ótima é limitado por uma função polinomial. Portanto, é muito pouco provável que se encontre um algoritmo ótimo, com tempo computacional razoável, para sua resolução. Em [SOUZA, 1989] o problema foi abordado pelo algoritmo ótimo de decomposição cruzada [VAN ROY, 1983], mas testado apenas para problemas pequenos.

Assim, a pesquisa voltou-se para o estudo de algoritmos heurísticos, que tem a vantagem de fornecer soluções para problemas de maior porte, sem demandar muito esforço computacional. No entanto, em tais métodos, não há garantia da otimalidade global e, em alguns casos, nem mesmo de obtenção de uma solução factível.

Dentre as heurísticas desenvolvidas para o problema de dimensionamento de lotes [1-6] pode-se destacar as apresentadas por [TRIGUEIRO at al., 1989] e [SCRICH, 1992]. Em [TRIGUEIRO at al., 1989] é construída a Relaxação Lagrangeana em relação às restrições de capacidade, tornando o problema decomponível e sendo possível a utilização do algoritmo de Wagner-Whitin. Em seguida, se a solução obtida é infactível, é aplicada uma heurística, baseada na transferência de produção entre os períodos, na tentativa de factibilizar a solução. [SCRICH, 1992] desenvolveu recentemente um algoritmo heurístico, baseado em [TRIGUEIRO at al., 1989], utilizando técnicas de Busca Tabu.

1.3. Modelos para o problema multiestágio

Denomina-se multiestágio o sistema de produção onde um item final é composto por outros componentes ou itens predecessores, isto é, a produção de um item final depende da fabricação de seus componentes, que também devem ser produzidos em lotes econômicos e considerados no problema total.

Admite-se que os componentes podem ter suas próprias demandas, chamadas de demandas independentes.

A conexão entre os itens é representada por um grafo orientado, onde um item i, que é predecessor de j, implica em i > j. O item 1 é considerado sempre como item final.

Considere a seguinte notação:

S(i) = conjunto dos itens que são sucessores imediatos do item i.
P(i) = conjunto dos itens que são predecessores imediatos do item i.
R(i) = conjunto de todos os itens que são sucessores do item i.
Q(i) = conjunto de todos os itens que são predecessores do item i.

Como exemplo, suponha a seguinte estrutura

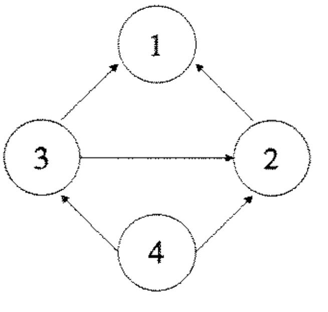

Figura 1.1. Exemplo de uma estrutura de produtos.

Temos,

S(1) = { }	P(1) = { 2, 3 }	R(1) = { }	Q(1) = { 2, 3, 4 }
S(2) = { 1 }	P(2) = { 3, 4 }	R(2) = { 1 }	Q(2) = { 3, 4 }
S(3) = { 1, 2 }	P(3) = { 4 }	R(3) = { 1, 2 }	Q(3) = { 4 }
S(4) = { 2, 3 }	P(4) = { }	R(4) = { 1, 2, 3 }	Q(4) = { }

As estruturas estudadas neste trabalho são:

Serial : onde cada item, com exceção do primeiro e do último, tem exatamente um sucessor e um predecessor (figura 1.2), isto é,

$S(i) = \{ i\text{-}1 \}$ para $i \neq 1$, $S(1) = \{ \}$.
$P(i) = \{ i+1 \}$ para $i \neq N$, $P(N) = \{ \}$.

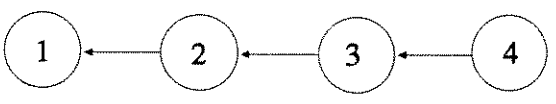

Figura 1.2. Exemplo de uma estrutura serial de produtos.

Plana : onde o item 1 tem como predecessores os itens 2,3,...,N, e estes (2,3,...,N) tem como sucessor o item 1 e nenhum item como predecessor (figura 1.3), isto é,

$S(i) = \{ 1 \}$ para $i \neq 1$, $S(1) = \{ \}$.
$P(1) = \{ 2,...,N \}$, $P(i) = \{ \}$, para $i \neq 1$.

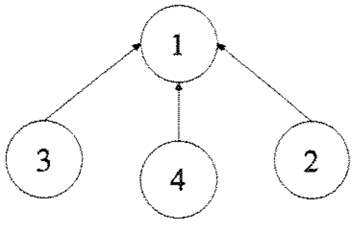

Figura 1.3. Exemplo de uma estrutura plana de produtos.

Montagem: representa uma estrutura mais complexa onde cada item pode ter vários predecessores, mas com a restrição de um único sucessor (figura 1.4).

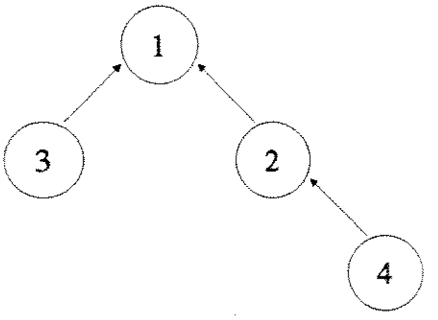

Figura 1.4. Exemplo de uma estrutura de montagem de produtos

Geral : representa a estrutura mais abrangente, não há restrição quanto ao número de sucessores e predecessores de um item, exceto os itens finais que não possuem sucessores. A figura 1.5 é um exemplo de uma estrutura geral de produtos onde possui apenas um item final, o item 1.

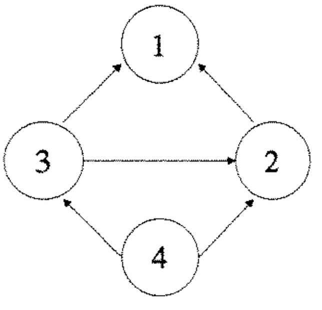

Figura 1.5. Exemplo de uma estrutura geral de produtos.

Os modelos multiestágios são geralmente formulados para o problema com estrutura geral (os modelos para as estruturas serial, plana e de montagem podem se vistos como casos particulares desta formulação). A apresentação é feita em duas fases: inicialmente considera-se o modelo utilizando estoque convencional; a seguir utiliza-se o conceito de *estoque de escalão*. Para facilitar a exposição emprega-se a palavra *item* para designar *item final* ou *componente*.

Vale mencionar que neste trabalho o tempo de produção (*lead time*) de cada item é considerado nulo para todos os itens, não havendo portanto o problema de sincronização da produção dos diversos itens.

1.3.1 Formulação em estoque convencional

Deve-se observar que a produção e o estoque de um item devem ser suficientes para suprir sua própria demanda, mais eventualmente uma quantidade para compor o lote dos itens sucessores. Define-se $\mathbf{r_{i,j}}$ como a quantidade do item i necessária para compor 1 unidade do item j. Considera-se também que o tempo de produção (*lead time*) é zero.

Como um exemplo, considere a estrutura da figura 1.5, onde

$r_{2,1} = 2$; $r_{3,1} = 3$; $r_{4,2} = 2$ e $r_{4,3} = 1$.

e que os itens 2,3 e 4 não tenham demanda independente. Suponha que foi decidido produzir 10 unidades do item 1, e que não haja estoque de qualquer item. Então, a produção do item 2 deve ser de pelo menos 20 unidades (10*2), do item 3 de 30 unidades (10*3) e do item 4 de pelo menos 70 unidades (20*2 + 30).

Logo, a equação de balanço entre as variáveis de estoque e produção deve determinar que x_{it} e I_{it} devem suprir d_{it} e $\sum_{j \in S(i)} r_{ij} x_{jt}$.

Assim, o modelo pode ser formulado como [CLARK, 1990]

MODELO [MI]

$$\min \sum_{i=1}^{N} \sum_{t=1}^{T} (h_{it} I_{it} + c_{it} x_{it} + s_{it} y_{it}) \qquad [7]$$

$$I_{i,t-1} + x_{it} - I_{it} = d_{it} + \sum_{j \in S(i)} r_{ij} x_{jt} \qquad i = 1,...,N ;\ t = 1,...,T \qquad [8]$$

$$\sum_{i=1}^{N} (v_{ikt} x_{it} + f_{ikt} y_{it}) \leq CAP_{kt} \qquad k = 1,...,K ;\ t = 1,...,T \qquad [9]$$

$$x_{it} \leq M_{it} y_{it} \qquad i = 1,...,N ;\ t = 1,...,T \qquad [10]$$

$$y_{it} = 0 , 1 \qquad i = 1,...,N ;\ t = 1,...,T \qquad [11]$$

$$x_{it} \text{ e } I_{it} \geq 0 \qquad i = 1,...,N ;\ t = 1,...,T \qquad [12]$$

O modelo [MI] é idêntico ao modelo [1-6], a menos das restrições [8] que representam as equações de balanço entre as variáveis de estoque e produção, onde é assegurado que a demanda de um item i em um período t (d_{it}) e a quantidade para os lotes de itens sucessores ($\sum_{j \in S(i)} r_{ij} x_{jt}$) serão supridas pela combinação da produção em t (x_{it}) e pelo estoque em t-1 ($I_{i,t-1}$) e que o excedente ficará estocado (I_{it}).

1.3.2 Formulação em estoque de escalão

Uma formulação que apresenta vantagens pode ser obtida adotando-se o conceito de ***estoque de escalão*** introduzido por [CLARK e SCARF, 1960] e implementado por [AFENTAKIS et al., 1984].

- **Estoque de escalão**

Estoque de escalão de um item é a quantidade total do item presente no sistema, incluindo a quantidade do item em estoque mais a quantidade do item contida no estoque de seus sucessores.

Como exemplo, considere a estrutura

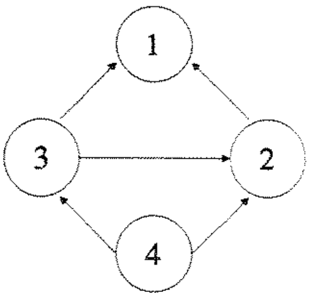

Figura 1.6. Exemplo de uma estrutura de produtos.

A quantidade do item 1 existente no sistema é apenas o seu estoque, já que este não possui sucessor. Logo, seu estoque de escalão é seu próprio estoque convencional,

$$E_{1t} = I_{1t}$$

O item 2, além de ter seu próprio estoque, está presente no item 1 (S(2) = { 1 }), portanto seu estoque de escalão é

$$E_{2t} = I_{2t} + r_{21}\, I_{1t}$$

No caso do item 3, tem-se

$E_{3t} = I_{3t} +$	o próprio estoque
$r_{31}\, I_{1t} + r_{32} \,.\, r_{21}\, I_{1t}$	quantidade do item 3 no estoque do item 1.
$r_{32}\, I_{2t}$	quantidade do item 3 no estoque do item 2.

Analogamente para o item 4,

$$\begin{aligned} E_{4t} = I_{4t} &+ \\ & r_{43}\, I_{3t} + \\ & r_{42}\, I_{2t} + r_{43}\, r_{32}\, I_{2t} + \\ & r_{43}\, r_{32}\, r_{21}\, I_{1t} + r_{43}\, r_{31}\, I_{1t} + r_{42}\, r_{21}\, I_{1t} \end{aligned}$$

O estoque de escalão do item i no período t é definido como

$$\boxed{E_{it} = I_{it} + \sum_{j \in S(i)} r_{ij}\, E_{jt}}$$

Utilizando este conceito, é preciso definir o ***custo de estoque de escalão*** em termos de custo de estoque convencional.

■ Custo de estoque de escalão

O custo de estoque de escalão é definido a partir do custo de estoque convencional, de modo que haja um abatimento de custos de estoque convencional de itens predecessores.

Logo, ao E_{it}, é atribuído um novo custo, que é definido como

$$\boxed{e_{it} = h_{it} - \sum_{j \in P(i)} r_{ji} h_{jt}}$$

A equivalência $\sum_{i=1}^{N} \sum_{t=1}^{T} e_{it} E_{it} = \sum_{i=1}^{N} \sum_{t=1}^{T} h_{it} I_{it}$ é mostrada no apêndice I.

- **Reformulando o problema**

O modelo [MI] pode, então, ser reformulado em termos de estoque de escalão. Primeiramente é examinada a equação de balanço [8], entre as variáveis de estoque e produção.

Como ilustração, considere a equação para o item 2, para o exemplo anterior. Utilizando estoque convencional, tem-se a equação de balanço dada por

$$I_{2,t-1} + x_{2t} - I_{2t} = d_{2t} + r_{21} x_{1t} \qquad [13]$$

sendo x1t determinado a partir de

$$I_{1,t-1} + x_{1t} - I_{1t} = d_{1t} \qquad [14]$$

Substituindo [14] em [13] temos

$$I_{2,t-1} + r_{21} I_{1,t-1} + x_{2t} - I_{2t} - r_{21} I_{1t} = d_{2t} + r_{21} d_{1t}$$

Isto é,

$$E_{2,t-1} + x_{2t} - E_{2t} = d_{2t} + r_{21} d_{1t}$$

Logo, a equação de balanço utilizando estoque de escalão não depende do tamanho dos lotes dos itens sucessores, mas sim, de suas demandas.

Assim a equação de balanço entre as variáveis de estoque de escalão e produção para o item 2 fica definida como

$$E_{2,t-1} + x_{2t} - E_{2t} = d_{2t} + r_{21}\, d_{1t}$$

Definindo ***demanda de escalão*** como a contabilização das demandas independentes e dependentes (a demanda dos itens sucessores), e dada, matematicamente, por

$$D_{it} = d_{it} + \sum_{j \in S(i)} r_{ij}\, D_{jt}$$

a equação pode ser reescrita como

$$E_{i,t-1} + x_{it} - E_{it} = D_{it}$$

Falta impor que o estoque de um item seja maior que zero ($I_{it} \geq 0$). Pela definição de E_{it} temos,

$$E_{it} \geq \sum_{j \in S(i)} r_{ij}\, E_{jt}$$

Temos assim, a seguinte formulação em estoque de escalão:

MODELO [ME]

$$\min \sum_{i=1}^{N} \sum_{t=1}^{T} (e_{it} E_{it} + c_{it} x_{it} + s_{it} y_{it}) \qquad [15]$$

$$E_{i,t-1} + x_{it} - E_{it} = D_{it} \qquad i = 1,...,N ;\ t = 1,...,T \qquad [16]$$

$$\sum_{j \in S(i)} r_{ij} E_{jt} - E_{it} \leq 0 \qquad i = 1,...,N ;\ t = 1,...,T \qquad [17]$$

$$\sum_{i=1}^{N} (v_{ikt} x_{it} + f_{ikt} y_{it}) \leq CAP_{kt} \qquad k = 1,...,K ;\ t = 1,...,T \qquad [18]$$

$$x_{it} \leq M_{it} y_{it} \qquad i = 1,...,N ;\ t = 1,...,T \qquad [19]$$

$$y_{it} = 0 , 1 \qquad i = 1,...,N ;\ t = 1,...,T \qquad [20]$$

$$x_{it} \ e \ E_{it} \geq 0 \qquad i = 1,...,N ;\ t = 1,...,T \qquad [21]$$

A equivalência entre as formulações **[MI]** e **[ME]** é mostrada no apêndice I.

Utilizando o conceito de estoque de escalão, retira-se a dependência entre os itens que aparece na equação [8] do modelo [MI]. A dependência encontra-se na equação [17] no problema [ME]. Porém, a aplicação da Relaxação Lagrangeana nas restrições [17] e [18] tornam o problema resultante completamente decomponível em N problemas monoestágios, independentes e com capacidade infinita de produção. Isso torna possível a utilização do algoritmo de Wagner-Within para a resolução do problema lagrangeano.

1.3.3 Métodos de resolução

No problema abordado neste trabalho, determinar o tamanho dos lotes é uma tarefa complexa. Em [McCLAIN e MAES, 1991] provou-se que determinar a existência de solução factível, para problemas com tempos de preparação (*setup times*) , é **NP-Completo** . Esta é uma das razões pela qual poucas pesquisas existem incluindo *setup-time*. Encontrar a solução ótima é um problema ***NP-hard***, pois, como dito anteriormente, o problema monoestágio já é ***NP-hard*** [FLORIAN, LENSTRA e RINNOY KAN, 1980]. Assim, os métodos que garantem solução ótima apresentam soluções apenas para problemas pequenos, dentro de um tempo computacional razoável. Por esta razão, há um maior desenvolvimento de métodos heurísticos.

A maior parte das pesquisas trata de estrutura serial e de montagem. Uma revisão nesta área de trabalho encontra-se em [BAHL, RITZMAN e GUPTA, 1987] e em [BILLINGTON at al., 1983].

Quando se considera **capacidade ilimitada** de produção, há **algoritmos ótimos** baseados em Programação Dinâmica que utilizam formulação em redes, para o caso de estrutura serial em [ZANGWILL, 1969], [LOVE, 1972] e [KONNO, 1988], entre outros. Em [AFENTAKIS et al., 1984] considera-se estrutura de montagem e é desenvolvido um algoritmo de *Branch and Bound* utilizando Relaxação Lagrangeana, mas viável apenas para problemas pequenos. [AFENTAKIS e GAVISH, 1986] transformaram estruturas gerais em estruturas equivalentes de montagem, para aplicar o mesmo método de *Branch and Bound*, mas, na transformação, são utilizados itens artificiais fazendo com que o número total de itens do problema, seja maior que o original. Há também, para estruturas de montagem, algoritmos de Programação Dinâmica e *Branch and Bound* em [CROWSTON, WAGNER e WILLIAMS, 1973] e [CROWSTON e WAGNER, 1973].

Alguns exemplos de **métodos heurísticos** existentes são: [BLACKBURN e MILLEN, 1982], que utilizam uma idéia de modificação dos custos e aplicam regras do problema multi-item não capacitado. Porém, eles não tratam estrutura geral. Em [GRAVES, 1981], foi considerado um caso particular de estrutura geral, mas não há testes para problemas grandes. [WILLIAMS, 1981] discute 7 heurísticas que podem ser usadas em estruturas de montagem, apresentando testes para 11000 problemas. No entanto, trabalha com uma formulação que considera demanda contínua no tempo. [CROWSTON, WAGNER, HENSHAW, 1972] testaram 4 heurísticas que foram comparadas com o algoritmo de Programação Dinâmica desenvolvido por [CROWSTON, WAGNER e WILLIAMS, 1973].

Para o caso de **capacidade finita de produção** temos pouca pesquisa realizada, devido à complexidade do problema. O trabalho de [BILLINGTON et al., 1983] analisa uma formulação matemática incluindo restrições de capacidade e considerando tempo de produção (*lead time*) diferente de zero, apresentando uma técnica de decomposição para a redução das estruturas dos produtos, mas sem fornecer um método de resolução.

Algoritmos ótimos desenvolvidos para este problema são poucos e para casos especiais. Em [STEINBERG e NAPIER, 1980] foi desenvolvido um procedimento ótimo, modelando o sistema como uma rede generalizada capacitada, com arcos de custos fixos. Consideram estrutura geral e capacidade apenas como limite para a variável x, mas é computacionalmente viável apenas para problemas de pequeno porte (testado para 3 itens, com 4 níveis e 20 períodos). [GABBAY, 1979] formulou um modelo multi-item para estruturas seriais, com recurso limitado para cada nível e consumo zero para a preparação deste recurso. Desenvolveu um algoritmo de Programação Dinâmica, mas sem testes computacionais. No trabalho de [CHIU e LIN, 1988] tem-se um estudo da topologia das estruturas serial e de montagem, para o caso de 1 item final. Utilizando o conceito de estoque de escalão e, sob algumas suposições, as condições para uma solução factível e as características da solução ótima são discutidas. É apresentado um algoritmo de Programação Dinâmica, podendo ser usado no problema sem ou com restrição de capacidade, sendo esta considerada para o último item ou para cada estágio.

Em [CLARK, 1990] é abordado o problema capacitado para estrutura de montagem e geral, com tempo de produção diferente de zero. São desenvolvidas duas abordagens de solução: uma de cortes fortes e uma utilizando Relaxação Lagrangeana. Além disso, é desenvolvido uma heurística baseada em transferências de produção, entre os períodos, para uso em problemas de maior porte.

[ZAHORIK, THOMAS e TRIGUEIRO, 1984] fonecem um algoritmo ótimo para o problema com 3 períodos e estrutura paralela (várias estruturas em série independentes), com utilização de uma formulação em rede e restrição de capacidade em um nível. Supõem custos lineares e não consideram custos de preparação (*setup cost*). No caso de T períodos, descrevem uma heurística baseada na formulação anterior.

Pela pequena dimensão dos problemas resolvidos pelos algoritmos ótimos, há um maior número de algoritmos heurísticos em desenvolvimento. [LAMBRECHT e VANDEREECKEN, 1978] estudam a estrutura serial, considerando custos côncavos e restrições de capacidade apenas no último item (final). O sistema é representado por rede, com exclusão da restrição de capacidade, que se analisa usando a teoria de rede de custos côncavos (baseado em [ZANGWILL, 1969]). Dois procedimentos são discutidos. O primeiro, um algoritmo de decomposição, que resolve o problema não capacitado, sem considerar o item final, e uma vez concluído, adiciona o item final segundo uma estratégia. O segundo, trata-se de um algoritmo de um estágio, que tenta simplificar o primeiro procedimento proposto.

Em [BLACKBURN e MILLEN, 1984] é considerado estrutura de montagem e quantidade máxima de produção de componente em cada período. Aplicam uma técnica de modificação de custos na função objetivo, para incluir os efeitos de restrição de capacidade em diferentes estágios. Em seguida, aplicam um método para cada estágio, a partir do item final. Os testes são realizados com 5 itens e 24 períodos.

[BAHL e RITZMAN, 1984] propuseram um modelo não linear, e uma heurística iterativa decide o tamanho dos lotes dos itens finais, supondo, nos níveis dos outros componentes, uma política lote-por-lote . [McCLAIN e MAES, 1991] resolvem por programação linear e testam três heurísticas diferentes para arredondar as variáveis binárias. Consideram capacidade por nível, mas é testado apenas para problemas pequenos (3 itens em série e 10 períodos).

A proposta deste trabalho é desenvolver uma heurística para o caso de capacidade limitada de produção e estrutura geral de produto, baseada na heurística sugerida por [CLARK, 1990] e com a aplicação de técnicas de Busca Tabu.

CAPÍTULO 2

FUNDAMENTOS DE BUSCA TABU

2.1. Introdução

Busca Tabu [GLOVER, 1988, 1989a, 1989b, 1990a, 1990b] é uma estratégia heurística que tem por objetivo direcionar as operações de um algoritmo, na tentativa de encontrar novas soluções, explorando diferentes regiões, tentando superar a ciclagem e a otimalidade local, que geralmente ocorrem em algoritmos heurísticos.

Inicialmente proposta por [GLOVER, 1977], Busca Tabu tem sido aplicada a diversos problemas [GLOVER, 1992, MULLER, 1990, PUREZA, 1990, SCRICH, 1992] e, na maior parte dos casos, tem melhorado o desempenho das heurísticas, produzindo soluções de melhor qualidade.

Neste capítulo são apresentados os fundamentos básicos de Busca Tabu utilizados neste trabalho.

2.2. O Princípio de Busca Tabu

Algoritmos de busca local geram uma seqüência de soluções através de operações realizadas sobre uma vizinhança de cada solução. Estas operações são chamadas de ***movimentos***, que acarretam a transição de uma solução à outra. Nos algoritmos heurísticos de [CLARK, 1990] e [SCRICH, 1992], um exemplo de movimento para o problema de dimensionamento de lotes é a transferência de uma parte da produção de um período para outro. A cada transferência, uma nova solução é gerada.

Depois de uma seqüência de movimentos, pode-se chegar a uma solução que não pode ser melhorada, segundo os passos determinados pelo algoritmo, e assim a heurística termina em uma solução, não havendo garantias que esta seja a melhor solução do problema.

Além disso, certos procedimentos podem ciclar em determinada região, sendo incapazes de sair pela própria característica dos passos que estes utilizam. A heurística já citada anteriormente, que transfere partes da produção para períodos posteriores e depois para períodos anteriores, por exemplo, pode ciclar quando as quantidades transferidas para frente são as mesmas que as transferidas para trás.

Uma maneira de tentar superar a solução local e/ou a ciclagem é proibir certos movimentos de serem refeitos ou de serem desfeitos, com o objetivo de forçar novas soluções não pesquisadas.

O princípio da Busca Tabu é o de restringir a busca, pela classificação de certos movimentos como proibidos, chamados de ***movimentos tabu***, na tentativa de prevenir a ciclagem e permitir a exploração de novas regiões. Depois de um certo número de iterações, estes movimentos tabu são liberados, podendo ser realizados, caso sejam escolhidos.

A Busca Tabu reúne estes dois aspectos, proibir e liberar movimentos, em uma estrutura denominada ***lista tabu***.

Depois que um movimento é realizado, este é armazenado na lista, determinando que o próprio movimento está proibido de ser refeito, ou o reverso deste, dependendo de como será feita a consulta à lista.

Para contrabalancear as restrições tabu, tornando a busca mais flexível, é utilizado um mecanismo capaz de liberar um movimento de sua condição tabu se este é um movimento suficientemente interessante para a exploração de novas regiões. Esta liberação é feita pelo ***critério de aspiração***.

A seguir, são analisados maiores detalhes dos conceitos apresentados acima.

2.3. Lista Tabu

Lista tabu é a estrutura utilizada para fornecer os movimentos que estão sendo considerados proibidos.

Primeiramente, é preciso definir quais elementos serão armazenados na lista, isto é, que irão caracterizar um movimento. Estes elementos são chamados de ***atributos***. Assim, a cada movimento realizado, os atributos associados a este movimento são armazenados na lista. E, a cada nova iteração, a lista é consultada de modo que nenhum movimento que possua tais atributos seja efetuado.

Como um exemplo, considere o movimento já citado, onde há mudança de parte da produção de um item de um período a outro. Este movimento pode ser caracterizado através de diferentes atributos.

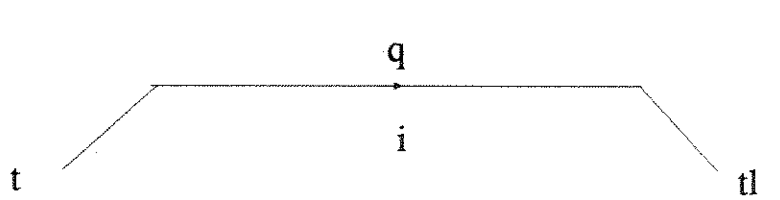

Figura 2.1. Exemplo de um movimento.

O movimento, ilustrado na figura 2.1, trata da transferência de uma quantidade q do item i do período t para o período tl. Os atributos que podem ser utilizados para caracterizar este movimento são:

(q,i,t,tl) - proíbe a quantidade q do item i de sair de t e ir a tl ou mesmo a quantidade q do item i de sair de tl e voltar a t.

(i,t,tl) - proíbe o item i de sair de t e ir a tl ou o item i de sair de tl e voltar a t.

(i,t) - proíbe o item i de sair de t ou de voltar a t.

(i,tl) - proíbe o item i de ir a tl ou de voltar a tl.

Observe que a opção de proibir o movimento direto ou o reverso está relacionada com a forma de consulta à Lista Tabu, pois o atributo apenas determina quais elementos serão armazenados na lista. Como um exemplo, considere o atributo (i,t) e que um movimento caracterizado pelo atributo (1,2) tenha sido realizado, isto é, houve transferência de parte da produção do item 1 do período 2 para um outro período qualquer. Após a transferência, o atributo (1,2) é armazenado na lista. A seguir, outros movimentos serão avaliados e a cada candidato, primeiramente, verifica-se se seu atributo pertence à Lista Tabu. Seja o movimento candidato dado pela figura 2.1. A consulta à Lista pode ser feita verificando se o par (i, t) pertence à Lista Tabu, significando que estamos proibindo o movimento direto, ou o par (i, tl) pode ser verificado e, neste caso, estamos proibindo o movimento reverso.

A cada movimento realizado, seu atributo é armazenado na lista tabu em sua última posição. Quando a lista estiver preenchida, a cada atributo armazenado subseqüentemente, o mais antigo é liberado de sua condição tabu (estratégia FIFO).

A escolha dos atributos se faz importante à medida que, dependendo do atributo escolhido, um ou mais movimentos estão sendo proibidos simultaneamente a cada atributo armazenado na lista. Assim, a busca torna-se mais, ou menos, restritiva a partir do atributo escolhido. No exemplo anteriormente citado, o atributo (i,t) torna a busca mais restritiva que o atributo (q,i,t,tl), pois a cada par (i,t) está associado um número maior de movimentos que à quádrupla (q,i,t,tl).

Outro aspecto a ser considerado é o tamanho da lista. Este parâmetro depende do algoritmo ao qual será incorporada a Lista Tabu e do problema que está sendo resolvido. Pode-se também utilizar um tamanho de lista aleatório, onde cada movimento é tabu por um número diferente de iterações. Para a implementação da lista de tamanho aleatório é preciso utilizar uma estrutura diferente de lista que é apresentada a seguir.

2.3.1 Implementação da lista

A implementação da Lista Tabu pode ser feita utilizando-se uma estrutura de lista com a estratégia FIFO para sua atualização, no entanto, a cada movimento candidato, a consulta a esta lista, para saber se o movimento em questão é ou não tabu, pode ser lenta.

Uma maneira mais simples de armazenar o atributo de um movimento executado, pode ser uma matriz, onde cada posição desta matriz simboliza um atributo, sendo associado um valor que representa a iteração de liberação dos movimentos, caracterizados por este atributo. O valor que indica o número de iterações que um movimento ficará proibido é chamado de ***TABU TAG***. Assim, em uma iteração k, a posição que indica o atributo deste movimento receberá o valor k + TABU TAG, representando a iteração que os movimentos, caracterizados por esse atributo, poderão ser realizados. Por exemplo, na utilização do atributo (i,t), pode-se utilizar uma matriz bidimensional, onde cada posição da matriz (i,t), inicializado com 0, receberá o valor TABU TAG + k (onde k é o número da iteração corrente) após a realização de um movimento que possui o atributo (i,t). Esta estrutura torna o gerenciamento da lista mais eficiente e permite o uso de tamanho de lista aleatório, ou seja, a cada movimento realizado é sorteado um número dentro de um intervalo [a,b], representando o valor de TABU TAG para o atributo de tal movimento. Entretanto, dependendo da quantidade de atributos utilizado, a dimensão desta estrutura torna inviável o seu uso, como, por exemplo, o atributo (q,i,t,tl).

O uso de tamanho de lista aleatório tem se mostrado mais adequado pela periodicidade que determinadas ciclagens ocorrem, ou seja, ciclagens que se repetem de períodos em períodos [TAILLARD, 1990].

2.4. Critério de Aspiração

Durante o processo de busca, o uso de certo conjunto de atributos pode estar proibindo movimentos que podem ser interessantes de serem feitos. Assim, torna-se necessário um mecanismo capaz de reconhecer situações em que é possível a superação da condição tabu de um movimento. Tal mecanismo é denominado Critério de Aspiração.

A implementação do Critério de Aspiração sugere a utilização de uma função que avalie um movimento proibido, segundo um determinado critério, informando se este pode ou não ser um movimento atrativo, eliminando sua condição tabu. É desejável que a aplicação do Critério de Aspiração não comprometa a busca no sentido de criar novas ciclagens, embora na prática seja quase impossível.

A incorporação de Critério de Aspiração em conjunto com a Lista Tabu proporciona uma flexibilidade adicional na escolha de movimentos e, conseqüentemente, no processo de busca.

2.5. Oscilação Estratégica

Um aspecto de Busca Tabu está em induzir o comportamento da busca, fazendo com que sejam criadas sucessivas soluções, onde o valor da função objetivo varia. Esta indução à busca pode ser aperfeiçoada com a utilização de Oscilação Estratégica.

A Oscilação Estratégica faz com que a busca se torne mais flexível, pois diversifica a natureza das regiões exploradas. Um exemplo de tal oscilação é a exploração de regiões infactíveis e factíveis, isto é, permite-se que a busca avance para regiões infactíveis e depois retorne à região factível, na tentativa que soluções factíveis melhores possam ser alcançadas.

Para tal aplicação, considere a função E(x) expressa da forma

$$E(x) = \alpha \,.\, O(x) + \beta \,.\, F(x)$$

onde F(x) indica o grau de factibilidade da solução x e O(x) indica uma medida de otimalidade dessa mesma solução.

As funções F(x) e O(x) têm por objetivo fornecer informações quantitativas e qualitativas do problema. Deste modo, pela variação dos parâmetros α e β, podemos explorar regiões distintas, dando ênfase à factibilidade ou à otimalidade das soluções exploradas.

Como ilustração de tal processo, considere a heurística já citada anteriormente, no qual são feitas transferências de partes da produção entre os períodos. A escolha de qual a melhor transferência a ser efetuada pode ser feita levando-se em conta o valor da função objetivo, isto é, escolhendo a transferência que fornece o maior decréscimo na função objetivo, mas não permitindo que um período produza mais que o permitido (veja o movimento (s1, s2) na figura 2.2.). Neste caso estamos trabalhando, pela função E(x), com os parâmetros $\alpha = 1$ e $\beta = 0$, isto é, explorando apenas regiões factíveis do problema. Com o propósito de explorar novas regiões, pode-se escolher uma transferência permitindo que certos períodos fiquem infactíveis, isto é, produzam mais que o permitido (veja o movimento (s2, s3) na figura 2.2.). Dessa forma estamos trabalhando com os parâmetros $\alpha > 0$ e $\beta > 0$. Agora, com uma nova solução gerada, que pode ser infactível, podemos fazer a escolha da próxima transferência com o objetivo apenas de factibilizar o problema, não importando o valor da função objetivo (veja movimento (s3, s4) na figura 2.2.). Assim estamos utilizando $\alpha = 0$ e $\beta = 1$.

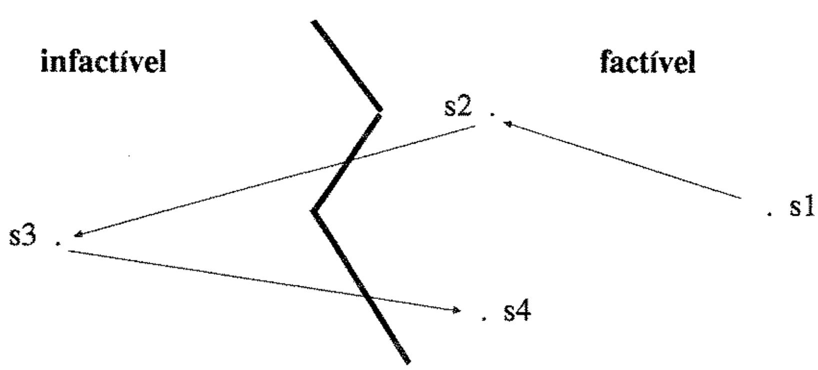

Figura 2.2.

Observe que o espaço de soluções na vizinhança da solução s4 poderia nunca ter sido visitado, sem que regiões infactíveis tivessem primeiramente sido alcançadas.

Com a utilização de Lista Tabu e manipulando os parâmetros alfa e beta da função E(x) obtemos um método mais flexível e agressivo para a exploração de novas regiões e conseqüentemente na obtenção de novas soluções.

CAPÍTULO 3

UMA HEURÍSTICA PARA A RESOLUÇÃO DO PROBLEMA

3.1. Introdução

Este capítulo apresenta uma heurística para a resolução do problema de dimensionamento de lotes multiestágio com restrição de capacidade (MODELO [ME]).

A heurística é dividida em quatro procedimentos:

P1 - Obtenção da solução inicial.
P2 - Factibilização.
P3 - Melhoria.
P4 - Busca de novas soluções.

Em P1, a solução inicial é gerada por uma aplicação seqüencial do algoritmo de Wagner-Within ao problema MI, ignorando as restrições de capacidade [9]. Assim, é provável que a solução seja infactível para o problema MI original. Nesse caso, o procedimento P2 é aplicado na tentativa de obter-se a factibilidade.

Caso uma solução factível tenha sido encontrada (em P1 ou em P2), o procedimento P3 tenta encontrar melhores soluções a partir da obtida anteriormente. Em seguida, o procedimento P4 tenta reconfigurar a solução, mesmo infactibilizando-a, de forma a ter um novo ponto de partida.

Desse modo, os procedimentos P2, P3 e P4 são executados continuamente até que um critério de parada seja atingido.

Com o propósito de se estudar os eventuais ganhos atingidos com técnicas de Busca Tabu, estas são aplicadas no desenvolvimento da heurística.

3.2. Obtenção da Solução Inicial - P1

Considere o modelo MI, apresentado no capítulo 1, ignorando as restrições de capacidade.

MODELO MI'

$$\min \sum_{i=1}^{N} \sum_{t=1}^{T} (h_{it} I_{it} + c_{it} x_{it} + s_{it} y_{it})$$

$$I_{i,t-1} + x_{it} - I_{it} = d_{it} + \sum_{j \in S(i)} r_{ij} x_{jt} \qquad i = 1,...,N \,;\; t = 1,...,T$$

$$x_{it} \leq M_{it} y_{it} \qquad i = 1,...,N \,;\; t = 1,...,T$$

$$y_{it} = 0 , 1 \qquad i = 1,...,N \,;\; t = 1,...,T$$

$$x_{it} \text{ e } I_{it} \geq 0 \qquad i = 1,...,N \,;\; t = 1,...,T$$

Como dito no capítulo 1, o algoritmo de Wagner-Within fornece uma solução ótima para o problema de dimensionamento de lotes monoestágio sem restrição de capacidade.

A idéia do procedimento P1 é aplicar este algoritmo para cada item do modelo MI' separadamente. Inicialmente, aplica-se o algoritmo para o item final, pois sua produção deve atender apenas sua demanda. A seguir, para cada item i $\in$ {2,...,N}, o algoritmo de Wagner-Within é aplicado após o cálculo de uma nova demanda (dit'), que depende da solução dos sucessores imediatos de i ($\sum_{j \in S(i)} r_{ij} x_{jt}$) e da própria demanda de i (dit), isto é,

$$d_{it}' = d_{it} + \sum_{j \in S(i)} r_{ij} x_{jt}$$

Depois de N aplicações do algoritmo de Wagner-Whitin, ou seja, uma para cada item, obtem-se uma seqüência de soluções monoestágios que, quando reunidas, formam uma solução factível para o problema MI'.

Como as restrições de capacidade foram ignoradas, provavelmente a solução é infactível para o problema MI e conseqüentemente para o problema ME. Dessa forma é utilizada o procedimento P2 na tentativa de adquirir a factibilidade.

Caso a solução encontrada seja factível, é feita a tentativa de melhora da solução através do procedimento P3.

3.3. Factibilização - P2

O procedimento proposto a seguir tem o propósito de encontrar uma solução factível para o problema [ME], a partir da solução encontrada no procedimento anterior. É baseado em [CLARK, 1990], onde são feitas transferências de partes da produção de períodos que estão infactíveis para outros períodos.

Um período t é considerado infactível se

$$\sum_{i=1}^{N} \left(f_{ikt}\, y_{it} + v_{ikt}\, x_{it} \right) > CAP_{kt} \qquad \text{para pelo menos um } k = 1,\ldots,K.$$

Dado um período t infactível, é feita a tentativa de transferir uma parte da produção de um item i para um período tl, com tl $\neq$ t.

Para cada item i que possui produção no período infactível, é analisada a possibilidade de transferir duas quantidades diferentes:

- $M_{i,tl}$ = quantidade máxima permitida, do item i, a ser transferida para o período tl, respeitando os níveis de estoque de escalão, isto é, obedecendo à restrição

$$\sum_{j \in S(i)} r_{ij} E_{jt} \leq E_{it} \qquad i = 1,\ldots,N \text{ e } t = 1,\ldots,T.$$

- Q_{itk} = quantidade suficiente do item i para eliminar exatamente o uso do excesso do recurso k no período t. A quantidade Q_{itk} é considerada apenas se $Q_{itk} < M_{i,tl}$.

Para a escolha da quantidade, item e período destino (q, i, tl) para a realização de uma transferência de um período t, é feito um cálculo que analisa a variação no custo e no excesso da utilização dos recursos que esta transferência causa, no caso de ser efetuada. Este cálculo é chamado de Teste da Razão e a tripla (q, i, tl) escolhida é a que obtiver mínima Razão.

O procedimento de factibilização (P2) é dividido em dois passos:

- PASSO REGRESSIVO (*backward*).

São analisados os períodos t = T, T-1,..., 2, nesta ordem, e a cada período t infactível são feitas transferências para períodos *anteriores* a t. Caso a solução, ao final deste passo, seja infactível, o passo progressivo é aplicado. Caso contrário, o procedimento de factibilização (P2) é finalizado.

- PASSO PROGRESSIVO (*forward*).

São analisados os períodos t = 1, 2, ..., T-1, nesta ordem, e a cada período t infactível, são feitas tentativas de transferências de parte da produção para períodos *posteriores* a t. Caso a solução encontrada seja infactível, o passo regressivo é aplicado novamente. Caso contrário, conclui-se o procedimento de factibilização (P2).

Os dois passos (regressivo e progressivo) são sucessivamente repetidos até que uma solução factível seja encontrada, ou um contador de passos atinja um limite máximo pré-especificado. Neste caso, o método falha.

A seguir, são mostrados os pseudo-códigos dos passos regressivo, progressivo e explicados os principais pontos sobre os quais as tranferências se baseiam.

■ PASSO REGRESSIVO DA FACTIBILIZAÇÃO ALGORITMO [A1]

```
para t = T até 2 faça
  E = EXCESSO ( t )
  enquanto ( E > 0 ) faça
      Rmin = Infinito
      para i = 1 até N faça
          se ( xit > 0 ) então
              τ =  período anterior à t com produção do item i, ou
                   1, caso não exista período anterior à t com produção do item i
              para tl = t - 1 até τ faça
                  M =  CALCULA_M ( i, t, tl )  { quantidade máxima permitida }
                  se (M = 0) então  tl = τ   { muda para o próximo item }
                  senão
                      para k = 0 até K faça
                          se ( k = 0 ) então     q = M
                          senão Q = CALCULA_Q (i, t, k)  {quantidade para eliminar
                                se ( Q < M )  q = Q        o excesso do recurso k}
                          fim { senão }
                          R = RAZÃO ( q, i, t, tl )
                          se ( R < Rmin )
                                qo    = q
                                io    = i
                                tlo   = tl
                                Rmin  = R
                          fim { se R < Rmin }
                      fim { para k }
                  fim { senão }
              fim { para tl }
          fim { se x > 0 }
      fim { para i }
      ATUALIZA ( qo, io, t, tlo )   { Atualiza solução, efetuando a transferência }
      E = EXCESSO ( t )            { Recalcula o excesso }
  fim { enquanto E > 0 }
fim { para t }
```

```
para t = 1 até T-1 faça
    E = EXCESSO ( t )
    enquanto ( E > 0  e  t < T ) faça
        Rmin = Infinito
        para i = 1 até N faça
            se ( xit > 0 ) então
                τ =  período posterior à t com produção do item i, ou
                     T, caso não exista período posterior à t com produção de i
                para tl = t + 1 até τ faça
                    M =  CALCULA_M ( i, t, tl )  { quantidade máxima }
                    se (M = 0) então  tl = τ    { muda para o próximo item }
                    senão
                        para k = 0 até K faça
                            se ( k = 0 ) então    q = M
                            senão Q = CALCULA_Q (i, t, k)   {quantidade para eliminar
                                  se ( Q < M )   q = Q        o excesso do recurso k}
                            fim { senão }
                            R = RAZÃO ( q, i, t, tl )
                            se ( R < Rmin )
                                  qo    = q
                                  io    = i
                                  tlo   = tl
                                  Rmin  = R
                            fim { se R < Rmin }
                        fim { para k }
                    fim { senão }
                fim { para tl }
            fim { se x > 0 }
        fim { para i }
        se existe ( qo, io, t, tlo ) então ATUALIZA ( qo, io, t, tlo ) { Atualiza solução efetuando a
                                                                          transferência }
        senão                    t = t + 1                            { muda para o próximo período }
        E = EXCESSO ( t ) { Recalcula o excesso }
    fim { enquanto E > 0 }
fim { para t }
```

- **Excesso no período t (Procedimento EXCESSO (t))**

Calcula o excedente de recursos utilizados em relação à capacidade disponível no período t , sendo dado por

$$\boxed{EXCESSO(t) = \sum_{k=1}^{K} \left[\frac{\sum_{i=1}^{N} (f_{ikt}\, y_{it} + v_{ikt}\, x_{it}) - CAP_{kt}}{CAP_{kt}} \right]^{+}} \qquad [22]$$

onde $[a]^{+} = \begin{cases} 0 & se \ a \leq 0 \\ a & se \ a > 0 \end{cases}$

- **Quantidade a ser transferida**

  - **$M_{i,tl}$ (Procedimento CALCULA_M (i, t, tl))**

Inicialmente, determina-se o quanto é possível mover do período t a um outro, de modo que os níveis de estoque de escalão permaneçam válidos, isto é, respeitando a restrição

$$\sum_{j \in S(i)} r_{ij} E_{jt} \leq E_{it} \qquad i = 1,...,N \ ; \ t = 1,...,T \qquad [23]$$

Para determinar $M_{i,tl}$, considere a estrutura a seguir

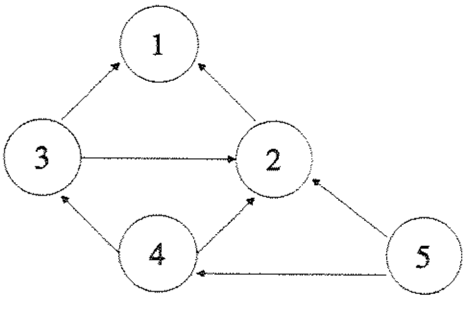

Figura 3.1. Exemplo de uma estrutura de produtos.

Suponha que exista produção do item 2 no período t e que parte de sua produção será transferida para o período tl. O período tl pode ser anterior (passo regressivo) ou posterior (passo progressivo) ao t. Os dois casos são tratados a seguir.

Transferência para um período anterior.

É analisada a transferência de uma quantidade q da produção do item 2 do período t para o período tl, de modo que tl < t.

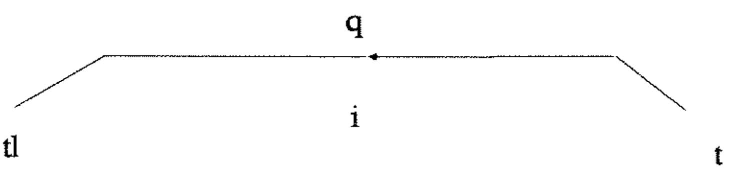

Figura 3.2. Transferência para um período anterior.

Observe, pela equação [16] do modelo [ME], dada por

$$E_{i,t-1} + x_{it} - E_{it} = D_{it}$$

que, caso seja efetuada a transferência, o estoque de escalão do item 2 do período tl ao t-1 será acrescido de q. Sendo assim, o lado esquerdo da equação [23] precisa ser analisado, isto é, a quantidade q deve ser tal que obedeça à condição

$$r_{i2}\, q + \sum_{j \in S(i)} r_{ij} E_{jt'} \leq E_{it'} \qquad t' = tl,...,t\text{-}1$$

Note, na desigualdade acima, que $2 \in S(i)$, ou ainda, $i \in P(2)$.

Isolando q, tem-se

$$q \leq \frac{E_{it'} - \sum_{j \in S(i)} r_{ij} E_{jt'}}{r_{i2}} \qquad t' = tl,\ldots,t\text{-}1 \text{ e } i \in P(2)$$

Esta condição pode ser interpretada da seguinte maneira: como haverá um acréscimo no estoque de escalão do item 2 entre os períodos tl e t-1 ($E_{2t'}$, t' = tl,...,t-1), os estoques dos itens predecessores imediatos do item 2 (itens 3, 4 e 5) precisam ser suficientes para continuarem a constituir o estoque de escalão do item 2, nos períodos tl a t-1, após a transferência.

Generalizando, para uma estrutura geral qualquer, a transferência de uma quantidade **q** do item **i** do período **t** para o período **tl** (tl < t) deve ser tal que a equação

$$q \leq \frac{E_{jt'} - \sum_{m \in S(j)} r_{jm} E_{mt'}}{r_{ji}} \qquad t' = tl,\ldots,t\text{-}1 \text{ e } j \in P(i)$$

seja satisfeita.

Observe que a quantidade q deve ser não maior que o lado direito da expressão acima para todos os períodos entre tl e t-1 e para todo j ∈ P(i). Logo q deve ser o mínimo entre todos os t' e j. Assim, reunindo todas estas condições, pode-se determinar, pela equação recursiva a seguir, a quantidade de produção máxima permitida de um item i que pode ser transferida do período t para o período tl.

$$M_{i,tl} = \begin{cases} \min\left\{ M_{i,tl+1}\,,\ \min_{j \in P(i)} \left\{ \dfrac{E_{jtl} - \sum_{m \in S(j)} r_{jm} E_{mtl}}{r_{ji}} \right\} \right\} & se\ \ tl < t \\ x_{i,tl} & se\ \ tl = t \end{cases} \qquad [24]$$

Note, pela equação anterior, que na análise de transferências para períodos anteriores, sempre existe uma quantidade de pelo menos um item que pode ser transferida, pois a limitação ocorre para itens que possuem predecessores e sempre existe pelo menos um item que não possui predecessor. Pela figura 3.1., verifica-se que sempre é possível transferir parte da produção do item 5 para um período anterior.

- Transferência para um período posterior.

É analisada a transferência de uma quantidade q da produção do item 2 do período t ao período tl, de modo que tl > t.

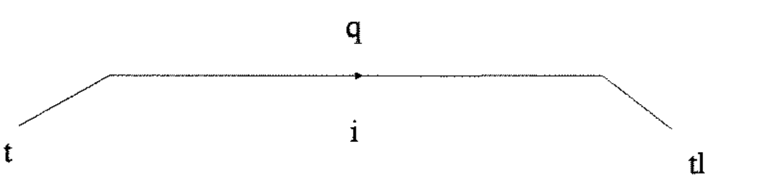

Figura 3.3. Transferência para um período posterior.

Novamente, observando a equação [16] do modelo [ME], conclui-se que, caso seja efetuada a transferência, o estoque de escalão do item 2 do período t ao tl-1 será reduzido de q. Sendo assim, o lado direito da equação [23] precisa ser analisado, isto é, a quantidade q deve ser tal que obedeça à condição

$$\sum_{j \in S(2)} r_{2j} E_{jt'} \leq E_{2t'} - q \qquad t' = t,...,tl\text{-}1$$

ou ainda,

$$q \leq E_{2t'} - \sum_{j \in S(2)} r_{2j} E_{jt'} \qquad t' = t...,tl\text{-}1$$

Esta condição pode ser interpretada da seguinte maneira: o estoque de escalão do item 2 deve continuar a ser suficiente para constituir o estoque de escalão dos itens sucessores imediatos (item 1), nos períodos t a tl-1, após a transferência, pois $E_{2t'}$ (t' = t,...,tl-1) será reduzido.

Portanto, para uma estrutura geral qualquer, a transferência de uma quantidade **q** do item **i** do período **t** para o período **tl** (tl > t) deve ser tal que a equação

$$q \leq E_{it'} - \sum_{j \in S(i)} r_{ij} E_{jt'} \qquad t' = t,\dots,tl-1$$

seja satisfeita.

Pode-se reunir todas estas condições, determinando qual a quantidade de produção máxima permitida de um item i que pode ser transferida do período t para o período tl (tl > t) pela equação recursiva a seguir

$$\boxed{M_{i,tl} = \begin{cases} \min\left\{ M_{i,tl-1} \,,\, E_{i,tl-1} - \sum\limits_{j \in S(i)} r_{ij} E_{j,tl-1} \right\} & se \;\; tl > t \\ x_{i,tl} & se \;\; tl = t \end{cases} \qquad [25]}$$

Note que, em transferências para períodos posteriores, a quantidade permitida pode ser limitada pelo estoque. Desta forma, pode haver períodos onde não existe possibilidade de transferência de produção de nenhum item, e conseqüentemente, de factibilizar o período em questão. Neste caso, a análise passa para o próximo período onde for encontrada infactibilidade.

- **Qitk** (**procedimento CALCULA_Q (i, t, tl))**

É analisada a quantidade suficiente, a ser transferida do item i, para eliminar o uso do excesso do recurso k no período t, de modo que não haja folga na capacidade do recurso k neste período.

Esta quantidade é dada por

$$Q_{itk} = \sum_{i=1}^{N} \left[(f_{ikt}\, y_{it} + v_{ikt}\, x_{it}) - CAP_{kt} \right]^{+} / \, v_{ikt} \qquad k = 1,\ldots,K. \qquad [26]$$

onde $[a]^{+} = \begin{cases} 0 \; se \; a \leq 0 \\ a \; se \; a > 0 \end{cases}$

Portanto, se Qitk < Mi,tl, considera-se a possibilidade de transferir a quantidade q = Qitk.

Assim, as quantidades de um item i candidatas para a transferência do período t para o tl, são todos os

- **Q_{itk}** para k = 1,...,K onde $Q_{itk} < M_{i,tl}$

e

- **$M_{i,tl}$**

Observe nos pseudo-códigos, que a quantidade $M_{i,tl}$ é considerada no laço de k, quando k = 0 e para k = 1,...,K são examinadas as quantidades Q_{itk}.

- **Período alvo (tl)**

Os períodos examinados como destino da transferência são todos aqueles situados entre t e o próximo período onde há produção do item analisado, isto é,

- **Passo Regressivo**

$$\mathbf{tl} = t, t\text{-}1, \ldots, \tau \qquad [27]$$

onde τ = primeiro período anterior a t, onde exista produção do item i. Caso não haja produção em nenhum período anterior a t, $\tau = 1$.

- **Passo Progressivo**

$$\mathbf{tl} = t, t+1, \ldots, \tau \qquad [28]$$

onde τ = primeiro período posterior a t, onde exista produção do item i. Caso não haja produção em nenhum período anterior a t, $\tau = T$.

Em ambos os passos, existe a possibilidade do período destino (tl) ser infactível, ou ainda, tornar-se infactível após a transferência.

- **Razão (procedimento RAZÃO (q, i, t, tl))**

Seja t um período infactível. Utiliza-se o Teste da Razão para a escolha de qual quantidade, item e período alvo, isto é, (q, i, tl), devem ser utilizados para transferência na tentativa de factibilizar o período t .

Em um período t infactível, para cada (q, i, tl) candidato, é calculada a Razão da seguinte forma:

$$\boxed{\text{Razão} = \frac{\alpha \,.\, custo\_adicional \; + \; \beta \,.\, penalidade}{excesso\_eliminado}} \qquad [29]$$

e a tripla (q, i, tl) que obtiver menor Razão é escolhida.

O <u>custo adicional</u> é o cálculo da variação no custo que uma determinada transferência (q, i, t, tl) causa e é dado, em termos percentuais, por

$$\boldsymbol{custo\_adicional} = \frac{custo\_produção \; + \; custo\_estoque \; + \; custo\_preparação}{custo\_total} \qquad [30]$$

onde

$$\text{custo\_produção} \quad = \quad q \,.\, (\, c_{i,tl} \; - \; c_{i,t} \,)$$

$$\text{custo\_estoque} \quad = \quad \begin{cases} +q \,.\, (\, e_{i,tl} \; + \; e_{i,tl+1} \; + \ldots \; + \; e_{i,t-1} \,) & para \;\; tl < t \\ -q \,.\, (\, e_{i,t} \; + \; e_{i,t+1} \; + \ldots \; + \; e_{i,tl-1} \,) & para \;\; tl > t \end{cases}$$

$$\text{custo\_preparação} \quad = \quad \begin{cases} s_{i,tl} \; - \; s_{i,t} & para \quad x_{i,t} = q \quad e \quad x_{i,tl} = 0 \\ s_{i,tl} & para \quad x_{i,t} \neq q \quad e \quad x_{i,tl} = 0 \\ -s_{i,t} & para \quad x_{i,t} = q \quad e \quad x_{i,tl} \neq 0 \\ 0 & para \quad x_{i,t} \neq q \quad e \quad x_{i,tl} \neq 0 \end{cases}$$

Note que o custo de preparação depende de todo o lote no período t ter sido transferido, e da existência ou não, de produção do item i no período tl.

O custo adicional, na maioria das vezes, é positivo em transferências para períodos anteriores, pois há um aumento no estoque e conseqüentemente no custo. A transferência de um lote inteiro pode atenuar este fato, podendo tornar o custo adicional negativo, pois há possibilidade de ocorrer uma redução nos custos da preparação. Já em transferências para períodos posteriores, o custo adicional é, geralmente, negativo, podendo ser positivo quando houver uma divisão de lotes entre períodos.

A penalidade é um valor acrescido à Razão que depende dos excessos nos períodos t e tl após a transferência. É contabilizada uma possível sobra do excesso no período t e um eventual aumento no excesso do período tl, sendo calculada por

$$\textbf{\textit{Penalidade}} = Excesso\_depois(t) + [\ Excesso\_depois(tl) - Excesso\_antes(tl)\ ] \qquad [31]$$

onde Excesso_antes(t) = EXCESSO(t) utilizando a solução anterior à transferência.

Excesso_depois(t) = EXCESSO(t) supondo que a transferência tenha sido efetuada.

Note que, pela equação [22], são contabilizados apenas os valores positivos do excesso, isto é, se num período houver folga na capacidade, o valor do excesso é zero e não um valor negativo. Desta forma, cada uma das parcelas na expressão [31] é positiva. Além disso, Excesso_depois(tl) é sempre maior ou igual a Excesso_antes(tl), pois sua produção aumenta após a transferência. Assim, a penalidade contabilizada é sempre um valor positivo. Vale ressaltar que outras formulações para o cálculo da penalidade foram testadas e a indicada em [31] foi a que obteve melhores resultados.

Após o cálculo do **custo_adicional** e da **penalidade**, é feita uma soma ponderada das parcelas de modo que seja dada uma maior importância ou para a variação do excesso ou para a alteração do custo. Note que a soma é feita com valores adimensionais, pois no cálculo do excesso e do custo_adicional os valores são representados de maneira percentuais.

Observe que, mesmo quando é dada uma maior ênfase para a alteração do custo, isto é, considerando $\alpha > \beta$, o objetivo continua sendo a eliminação do excesso, mas a escolha do item, período e quantidade se dará segundo um critério no qual o custo após a transferência tem sua importância ressaltada. Assim, quanto maior o α em relação ao β, maior a importância do custo após a transferência. Quando é feito $\beta > \alpha$, a escolha do item, período e quantidade se dará segundo um critério que determina um maior peso às variações ocorridas nos excessos após a transferência.

A idéia, dentro do algoritmo, foi considerar $\alpha = 1$ e $\beta = n$, onde n = número do ciclo, isto é,

No **primeiro** passo regressivo e progressivo $\rightarrow$ $\alpha = 1$ e $\beta = 1$.
No **segundo** passo regressivo e progressivo $\rightarrow$ $\alpha = 1$ e $\beta = 2$.
No **terceiro** passo regressivo e progressivo $\rightarrow$ $\alpha = 1$ e $\beta = 3$,

e assim por diante.

Note que, quanto maior o número de ciclos no procedimento de factibilização, isto é, o número de aplicações do passo regressivo e progressivo, maior a dificuldade em encontrar uma solução factível. Assim, a cada novo ciclo, a escolha de uma determinada transferência se faz segundo um critério que aumenta o peso da variação do excesso em relação ao peso da variação do custo.

Terminado o cálculo acima, ainda é contabilizada a variação no valor do excesso no período t em excesso eliminado. Calculado por

$$Excesso\_eliminado = Excesso\_antes(t) - Excesso\_depois(t) \qquad [32]$$

onde Excesso_antes = EXCESSO(t) utilizando a solução anterior à transferência.
Excesso_depois = EXCESSO(t) utilizando uma solução que supõe que a transferência tenha sido efetuada.

Desta forma, ao final destes cálculos, tem-se em RAZÃO, qual a alteração no custo (sendo um custo penalizado) por unidade de variação no excesso que cada tranferência (q, i, tl) causa. E dentre todos as triplas (q, i, tl) candidatas, é escolhida a que obtiver a menor RAZÃO.

3.4. Melhoria - P3

A partir de uma solução factível para o problema ME, o procedimento de melhoria P3 tenta obter uma solução com menor custo sem causar infactibilidade.

Seu mecanismo é similar ao utilizado no procedimento de factibilização, ou seja, se baseia em transferências de partes da produção entre períodos, porém, são permitidas apenas as transferências que não infactibilizam a solução e que melhoram o valor da função objetivo.

As quantidades candidatas para as transferências são as permitidas pelo estoque de escalão ($M_{i,tl}$) ou geradas aleatoriamente dentro do intervalo [0, $M_{i,tl}$], isto é, são analisadas as quantidades:

- $q = M_{i,tl}$
- $q = U\ [\ 0,\ M_{i,tl}\ ]$

onde U [a,b] representa um número gerado aleatoriamente, utilizando-se distribuição uniforme, no intervalo [a,b].

Da mesma forma que no procedimento de factibilização, as transferências são feitas para trás e, a seguir, para frente. Logo, o procedimento de melhoria também é dividido em dois passos:

- PASSO REGRESSIVO

São analisados os períodos t = T,...,2, nesta ordem, e a cada período, tenta-se transferir a quantidade $M_{i,tl}$ ou U [0, $M_{i,tl}$], onde tl é um período anterior a t. Transferências para trás, geralmente causam um aumento no custo total quando não houver transferências de todo lote, mas, sendo os custos variantes no tempo, as possibilidades são testadas, pois pode-se aumentar o custo em estoque e reduzir os custos de produção e/ou de preparação, podendo haver uma redução no custo total.

- PASSO PROGRESSIVO

São analisados os períodos $t = 1,...,T-1$, nesta ordem, e a cada período t tenta-se transferir as quantidades $M_{i,tl}$ ou $U[0, M_{i,tl}]$, onde tl é um período posterior a t. Neste passo há uma maior probabilidade de ocorrer redução nos custos, pois além dos custos de produção e/ou preparação mencionados no passo anterior, há a redução nos custos de estoque.

Em ambos os passos, a cada período t, é executada a transferência que cause a maior redução no valor da função objetivo. Para a análise das transferências, é feito o Teste da Razão, onde é calculado o **custo_adicional** como em [30] e a **penalidade** como em [31] e a Razão de cada transferência (i, q, t, tl) é definida como:

$$\textbf{\textit{Razão}} = \begin{cases} custo\_adicional & , \quad se \ custo\_adicional < 0 \ \ e \ \ penalidade = 0 \\ \infty & , \quad caso \ \ contrário \end{cases} \qquad [33]$$

Assim, a transferência (q, i, t, tl) que obtiver menor Razão é efetuada.

Existe a possibilidade de não haver transferências candidatas em um determinado período, quer seja pelo estoque de escalão ($M_{i,tl} = 0$), ou porque as transferências possíveis não atendem a restrição de reduzir o valor da função objetivo sem causar infactibilidade. Nesse caso, o próximo período, dentro da lógica do algoritmo, é analisado.

Os dois passos são executados sucessivamente até que um passo regressivo e um progressivo sejam realizados sem que nenhuma transferência tenha sido efetuada, terminando com uma solução melhor que a obtida no procedimento de factibilização, ou, no pior caso, com a mesma.

A seguir são mostrados os pseudo-códigos dos dois passos.

■ PASSO REGRESSIVO PARA MELHORIA ALGORITMO [A3]

```
para t = T até 2 faça
|
|  Rmin = Infinito
|
|  para i = 1 até N faça
|  |
|  |     se ( xit > 0 ) então
|  |     |
|  |     |     para tl = t - 1 até 1 faça
|  |     |     |
|  |     |     |     M =  CALCULA_M ( i, t, tl )  { quantidade máxima permitida }
|  |     |     |
|  |     |     |     se (M = 0) então  tl = 1    { muda para o próximo item }
|  |     |     |     senão
|  |     |     |     |
|  |     |     |     |     para k = 0 até 1 faça
|  |     |     |     |     |
|  |     |     |     |     |     se ( k = 0 ) então      q = M
|  |     |     |     |     |     senão                   q = U [ 0 , M ]
|  |     |     |     |     |
|  |     |     |     |     |     R = RAZÃO ( q, i, t, tl )
|  |     |     |     |     |
|  |     |     |     |     |     se ( R < Rmin )
|  |     |     |     |     |     |     qo    = q
|  |     |     |     |     |     |     io    = i
|  |     |     |     |     |     |     tlo   = tl
|  |     |     |     |     |     |     Rmin  = R
|  |     |     |     |     |     |
|  |     |     |     |     |     fim { se R < Rmin }
|  |     |     |     |     fim { para k }
|  |     |     |     fim { senão }
|  |     |     fim { para tl }
|  |     fim { se x > 0 }
|  fim { para i }
|
|  ATUALIZA ( qo, io, t, tlo )   { Atualiza solução }
|
fim { para t }
```

- **CALCULA_M (i, t, tl)** → Calcula $M_{i,tl}$ pela equação [24].

- **RAZÃO (q, i, t, tl)** → Calcula a Razão como em [33].

- Observe que o laço de k existe para a análise das duas quantidades para a transferência, q = M e q = U [0, M].

■ PASSO PROGRESSIVO PARA MELHORIA ALGORITMO [A4]

```
para t = 1 até T-1 faça
|
|  Rmin = Infinito
|
|  para i = 1 até N faça
|  |
|  |     se ( xit > 0 ) então
|  |     |
|  |     |     para tl = t + 1 até T faça
|  |     |     |
|  |     |     |     M =  CALCULA_M ( i, t, tl )  { quantidade máxima permitida }
|  |     |     |
|  |     |     |     se ( M = 0 ) então   tl = T   { muda para o próximo item }
|  |     |     |     senão
|  |     |     |     |
|  |     |     |     |     para k = 0 até 1 faça
|  |     |     |     |     |
|  |     |     |     |     |     se ( k = 0 ) então      q = M
|  |     |     |     |     |     senão                   q = U [ 0 , M ]
|  |     |     |     |     |
|  |     |     |     |     |     R = RAZÃO ( q, i, t, tl )
|  |     |     |     |     |
|  |     |     |     |     |     se ( R < Rmin )
|  |     |     |     |     |     |     qo    = q
|  |     |     |     |     |     |     io    = i
|  |     |     |     |     |     |     tlo   = tl
|  |     |     |     |     |     |     Rmin  = R
|  |     |     |     |     |     |
|  |     |     |     |     |     fim { se R < Rmin }
|  |     |     |     |     fim { para k }
|  |     |     |     fim { senão }
|  |     |     fim { para tl }
|  |     fim { se x > 0 }
|  fim { para i }
|
|  ATUALIZA ( qo, io, t, tlo )   { Atualiza solução }
|
fim { para t }
```

- **CALCULA_M (i, t, tl)** → Calcula $M_{i,tl}$ pela equação [25].

- **RAZÃO (q, i, t, tl)** → Calcula a Razão como em [33].

- Observe que o laço de k existe para a análise das duas quantidades para a transferência, q = M e q = U [0, M].

3.5. Busca de Novas Soluções - P4

O procedimento P4 chamado de procedimento de agregação, tem como objetivo reconfigurar os períodos onde ocorre produção, permitindo infactibilidade, de modo que ao seu final seja obtido um novo ponto de partida para a aplicação do procedimento de factibilização e/ou melhoria.

Também é baseado em transferências, mas as quantidades analisadas são sempre o lote todo, com o propósito de economizar custos e recursos de preparação em determinados períodos.

Trata-se apenas de um passo, onde são analisados para cada item a transferência do lote todo de um período t para um tl. Os períodos considerados como origem (t) e destino (tl) das transferências são determinados a partir de uma ordenação prévia dos períodos segundo critérios que envolvem custo e utilização dos recursos em cada período. A explicação da ordenação será feita posteriormente. É efetuada a primeira transferência possível pela ordem pré-especificada. Após a execução de uma transferência, o próximo item é analisado.

A cada item tenta-se transferir, no máximo, um lote e após a análise dos N itens, o procedimento termina.

A seguir é mostrado o pseudo-código e posteriormente a explicação da ordenação dos períodos.

▪ PROCEDIMENTO DE AGREGAÇÃO ALGORITMO [A5]

```
para i = 1 até N faça
|   ORDENA_PERÍODO( i, ord )
|   para t = T até 1 faça
|   |   se ( x_i,ord[ t ] > 0 ) então
|   |   |   para tl = 1 até T faça
|   |   |   |   se ( x_i,ord[ tl ] > 0 e tl <  t ) então
|   |   |   |   |   M =  CALCULA_M ( i, ord[ t ], ord[ tl ] ) {quantidade máxima permitida}
|   |   |   |   |   se ( M = x_i,ord[t] ) {verifica se a quantidade máxima é exatamente o lote todo}
|   |   |   |   |   |   ATUALIZA ( M, i, ord[ t ], ord[ tl ] ) { Atualiza solução }
|   |   |   |   |   |   tl = T; t = 1 { muda para o próximo item }
|   |   |   |   |   fim { se M = x_i,ord[t] }
|   |   |   |   fim { se  x_i,ord[tl] > 0  }
|   |   |   fim { para tl }
|   |   fim { se  x_i,ord[t] > 0  }
|   fim { para t }
fim { para i }
```

- **CALCULA_M (i, ord[t], ord[tl])** → Calcula $M_{i,ord[tl]}$ pela equação [24] se ord[tl] < ord[t].
→ Calcula $M_{i,ord[tl]}$ pela equação [25] se ord[tl] > ord[t].

▪ Ordena_período

Esta função especifica a ordem no qual os períodos são analisados como origem e destino para a união de lotes, sendo que os períodos ordenados são armazenados em um vetor de nome **ord** no pseudo-código. A ordenação é feita de modo que sejam avaliados o custo de preparação do item e a quantidade de recursos <u>não</u> utilizados em cada período.

É chamado de Folga, a quantidade de recursos não utilizados no período, ou seja,

$$FOLGA(t) = \sum_{k=1}^{K} \left[\frac{CAP_{kt} - \sum_{i=1}^{N} (f_{ikt} y_{it} + v_{ikt} x_{it})}{CAP_{kt}} \right] \quad [34]$$

Para a explicação da ordenação, suponha que esteja sendo resolvido um problema como representado na figura 3.1, considerando 4 períodos como horizonte de planejamento. Suponha ainda que os procedimentos P1, P2 e P3 já tenham sido aplicados e que na execução do procedimento P4, o item 2 esteja sendo analisado para a união de lotes.

Considere que os custos de preparação do item 2 em cada período são:

período

	1	2	3	4
$s_{2,t}$	50	100	10	150

Tabela 3.1. Custo de preparação do item 2.

E que a Folga em cada período, calculada utilizando [34], seja

período

	1	2	3	4
Folga(t)	350	10	1	50

Tabela 3.2. Quantidade de recursos não utilizados em cada período.

Inicialmente, obtém-se a ordem dos períodos por ordem crescente de custo de preparação. Assim, temos:

ORDS	3	1	2	4

Tabela 3.3. Ordenação dos períodos por ordem crescente de custo de preparação

Em seguida, ordenam-se os períodos por ordem decrescente de quantidade de folga, ou seja,

ORDF	1	4	2	3

Tabela 3.4. Ordenação dos períodos por ordem decrescente de Folga.

Observe que em termos de custo de preparação (ORDS), seria interessante transferir o lote do item 2 do período 4 para o 3, ou seja, do maior para o menor custo. Mas, considerando a quantidade de Folga em cada período (ORDF), uma transferência atraente seria passar o lote do item 2, do período 3 para o 1. Desta forma, incorporou-se em uma única ordenação, custos de preparação e Folga de cada período, onde a nova ordem é determinada pela posição dos períodos em cada arranjo acima (ORDS e ORDF), ou seja,

período				posição em ORDS		posição em ORDF	
1	$\rightarrow$	peso	**3**	(2	+	1	)
2	$\rightarrow$	peso	**6**	(3	+	3	)
3	$\rightarrow$	peso	**5**	(1	+	4	)
4	$\rightarrow$	peso	**6**	(4	+	2	)

Havendo empate, optou-se pelo custo de preparação. Assim, como $s_{2,2} = 100$ e $s_{2,4} = 150$, tem-se como ordem dos períodos

ORD	**1**	**3**	**2**	**4**

Tabela 3.5. ordenação dos períodos considerando custo de preparação e Folga

Obtida a ordenação, os períodos considerados como origem são t = 4, 2, 3 e 1, nesta ordem, e os períodos considerados como destino são t = 1, 3, 2 e 4, nesta ordem, ou seja, são analisadas as transferências do lote inteiro do item 2 do período

4 ao 1, 4 ao 3, 4 ao 2, 2 ao 1, 2 ao 3 e 3 ao 1,

nesta ordem, e a primeira transferência possível, caso exista, é efetuada. A seguir, o próximo item é analisado. Ao término dos N itens, o procedimento é finalizado.

Após a execução deste procedimento, provavelmente a solução obtida é infactível. Deste modo, o procedimento de factibilização é aplicado. Caso uma solução factível seja encontrada (em P2 ou em P4), o procedimento de melhoria P3 é novamente empregado.

3.6. O Algoritmo da Heurística

A reunião dos quatro procedimentos apresentados (P1, P2, P3 e P4) formam uma heurística para resolução do problema [ME]. A seguir, é mostrado o esquema de funcionamento da heurística através de um pseudo-código.

Considere

r	= contador de iterações.
S (r)	= solução obtida na iteração r.
f (S(r))	= valor da função objetivo utilizando a solução S(r).
S*	= solução incumbente.
ITMAX	= número máximo de iterações da heurística.

■ **HEURÍSTICA** **ALGORITMO [A6]**

```
r = 0   e   seja S* infactível e f(S*) = ∞            { Inicialização }
S (r) = SOLUÇÃO DE PARTIDA                             { Procedimento P1 }
para r = 1 até ITMAX faça
  se ( S (r) é infactível ) então
        S(r) = FACTIBILIZAÇÃO                          { Procedimento P2 }
  fim
  se ( S (r) é factível ) então
        S (r) = MELHORIA                               { Procedimento P3 }
        se f ( S (r) ) < f ( S* ) então S* = S (r)
  fim {
  S (r) = AGREGAÇÃO                                    { Procedimento P4 }
fim { para r }

se ( S* é infactível ) então o método falhou.
senão S* é a melhor solução encontrada.
```

Comentários:

- ITMAX é o número de iterações máximo para a heurística. O procedimento de factibilização tem seu próprio contador interno e é finalizado quando um número máximo de iterações, também definido internamente, é atingido. O procedimento de melhoria termina quando são executados dois passos (um regressivo e um progressivo), sem que nenhuma transferência tenha sido efetuada. O procedimento agregação é finalizado quando completada a análise dos N itens.

- O procedimento de melhoria é executado apenas quando uma solução factível é encontrada. Enquanto não for encontrada, os procedimentos de factibilização e o de agregação são executados sucessivamente até que uma solução factível seja obtida ou que o contador r atinja o limite de ITMAX. O procedimento de agregação é executado independentemente da factibilidade da solução, sendo que na última iteração não é executado. O procedimento de factibilização apenas é aplicado quando uma solução infactível é obtida.

- A heurística não garante que uma solução factível seja encontrada. Tampouco é capaz de identificar se um problema é infactível, ou seja, o fato de não ser encontrada pelo menos uma solução factível até o final da heurística não implica que o problema seja infactível.

3.7. Aplicação de Busca Tabu

A Busca Tabu foi implementada em 2 etapas. Inicialmente, no procedimento de factibilização e posteriormente na heurística total. O desenvolvimento de cada uma das etapas é apresentado separadamente nos dois tópicos a seguir.

3.7.1 Busca Tabu no Procedimento de Factibilização

Como visto em 3.3, o procedimento de factibilização é composto por 2 passos. No passo regressivo são feitas transferências para períodos anteriores e a seguir, no passo progressivo, são feitas transferências para períodos posteriores. Desta forma, pela própria característica do procedimento, existe a possibilidade de haver ciclagem se as quantidades transferidas para trás forem as mesmas que as transferidas para frente. Como exemplo, considere o plano de produção a seguir.

	t_1	t_2	t_3
x_1	50	70	20
x_2	60	50	10

Tabela 3.6. Exemplo de um plano de produção

Suponha que o período t = 2 seja um período infactível, isto é, tenha Excesso positivo. Suponha ainda, que na aplicação do procedimento de factibilização as transferências efetuadas foram:

- Passo Regressivo 1 $q = 30$, $i = 1$ de $t = 2$ para $tl = 1$.

A factibilização do período 2 é atingida, mas o período 1 torna-se infactível. Inicia-se o passo progressivo.

- Passo Progressivo 1 $q = 30$ $i = 1$ de $t = 1$ para $tl = 2$.

  $q = 30$ $i = 1$ de $t = 2$ para $tl = 3$.

O período 3 está infactível. Reinicia-se o passo regressivo.

- Passo Regressivo 2 $q = 30$ $i = 1$ de $t = 3$ para $tl = 2$.

Observe que neste ponto a solução tem a mesma configuração que a inicial.

Mesmo com a utilização de α e β, alterando o critério na escolha das transferências a cada passo, pode-se verificar ciclagem no procedimento de factibilização. Desta maneira, mostra-se adequada a aplicação de Busca Tabu.

Inicialmente, é definido um movimento, na heurística proposta, como sendo a transferência de parte da produção de um item de um período a outro. Os atributos a serem considerados para caracterizar esse movimento são os conjuntos formados pelos elementos i, q, t, tl. Alguns exemplos são (q, i, t, tl), (i, t, tl), (i, t) e (q, i, t). A escolha de um determinado atributo para a aplicação de Busca Tabu é uma tarefa delicada, pois, como dito no capítulo 2, a escolha de certos atributos pode não prevenir a ciclagem, tornar a busca mais, ou menos restrita e ser ineficaz. A escolha do atributo será discutida posteriormente.

Com a definição de movimento e determinado o atributo a ser utilizado pode-se implementar Busca Tabu na estrutura do algoritmo. As técnicas utilizadas neste trabalho foram o uso da Lista Tabu e Critério de lAspiração. A Lista Tabu foi implementada utilizando a estrutura de matriz apresentada no capítulo 2. O Critério de Aspiração é discutido mais a frente.

De uma maneira geral, o procedimento de factibilização com a incorporação de Busca Tabu obteve as seguintes alterações: durante a primeira aplicação do procedimento de factibilização, no primeiro ciclo, ou seja, no primeiro passo regressivo e progressivo, a cada movimento realizado seu respectivo atributo é armazenado na Lista Tabu, de modo que esteja proibido durante **m** transferências, onde m é o tamanho da lista. Durante esta fase não há consulta à Lista. A partir do segundo ciclo, a cada movimento candidato, é verificado se o seu respectivo atributo pertence à Lista. Caso pertença, o movimento é descartado, caso contrário o movimento é avaliado. Após a escolha de um movimento, este é realizado e seu atributo é armazenado na Lista.

A partir da segunda aplicação do procedimento de factibilização, a consulta, bem como a atualização da Lista é feita em todos os ciclos, pois, realizado pelo menos um procedimento de factibilização, já existe um histórico de movimentos efetuados.

Foi utilizado um critério de aspiração, ou seja, os movimentos tabu eram avaliados por um critério e caso fossem aprovados eram liberados de sua condição tabu, mas com o uso deste critério a ciclagem foi constatada. A seguir, foi testado outro critério para liberação da condição tabu dos movimentos, um critério secundário. Neste, os movimentos proibidos eram avaliados segundo um critério apenas se não existisse movimentos disponíveis para serem realizados (todos eram tabu, ou seja, proibidos) e, a partir desse critério, o que obtivesse o melhor resultado era efetuado.

A seguir é apresentada a escolha do atributo, a manipulação da lista e o critério de liberação da condição tabu dos movimentos.

- **Escolha do atributo**

Os atributos avaliados neste trabalho foram

- **(q, i, t, tl)**

Pela aplicação do procedimento de factibilização, observa-se que a utilização deste atributo não previne ciclagem, pois, como exemplo, realizando o movimento q = 30, i = 1, t = 3 e tl = 2 (representado por m1 na figura 3.4), o atributo armazenado é (30, 1, 3, 2). Os movimentos q = 10, i = 1, t = 3 e tl = 2 e q = 20, i = 1, t = 3 e tl = 2 (m2 e m3, respectivamente na figura 3.4) não possuem seus atributos na Lista e portanto podem ser efetuados, mas a execução de m2 e m3 equivalem à execução de m1.

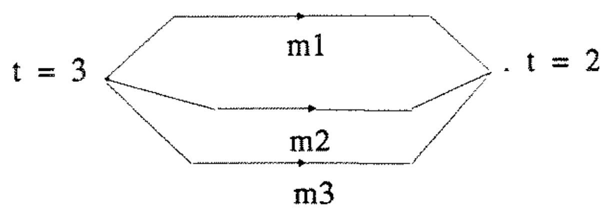

Figura 3.4. Exemplo de equivalência entre movimentos.

Assim, a utilização deste atributo faz com que movimentos que deveriam estar proibidos sejam efetuados pela composição de outros.

- **(i, t, tl)**

Com a utilização deste atributo também verificou-se o fenômeno da ciclagem. Como ilustração, suponha que tenha sido realizado o movimento q = 10, i = 1, t = 3 e tl = 2 (m1 na figura 3.5.), sendo armazenado (1, 3, 2) na Lista. Suponha também que o movimento reverso esteja proibido, ou seja, não retornar produção do item 1 do período 2 para o 3. Realizando os movimentos q = 10, i = 1, t = 2 e tl = 1 e q = 10, i = 1, t = 1 e tl = 3 (m2 e m3 na figura 3.5.), o movimento proibido foi efetuado com a composição de dois outros movimentos.

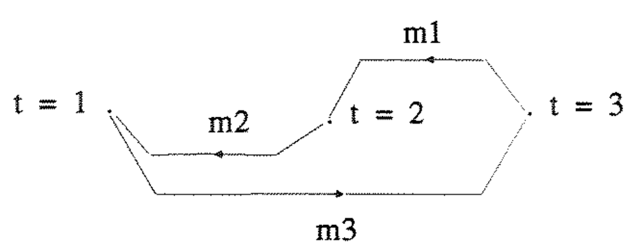

Figura 3.5. Exemplo de equivalência entre movimentos.

- **(i, t) ou (i, tl)**

A utilização de um destes atributos torna a busca mais restritiva. Como exemplo, se for realizado o movimento q = 30, i = 1, t = 3 e tl = 2 e armazenando apenas os elementos (1, 3) na LISTA, proíbe-se um conjunto maior de movimentos do que armazenar os elementos (30, 1, 3, 2). Apesar de tornar a busca mais restritiva, foram os atributos que mais preveniram a ciclagem. A utilização de (i, t) e (i, tl) mostrou-se equivalente e optou-se pelo (i, t).

■ Manipulação da Lista

Com a escolha do atributo (i, t), a estrutura para armazenar os atributos é uma matriz N x T, onde cada posição é inicializada com zero.

Seja **itfac** uma variável que armazene a quantidade de transferências realizadas no procedimento de factibilização e **LISTA** a matriz que armazenará os atributos dos movimentos proibidos.

A cada movimento (q, i, t, tl) realizado, é atribuído itfac + TAG na posição (i, t) de LISTA, onde TAG é o número de iterações em que o movimento estará proibido. No decorrer da heurística, a cada movimento candidato (q, i, t, tl), a posição (i, tl) de LISTA é verificada se possui um número maior que itfac (quantidade de transferências realizadas corrente), verificando se o movimento reverso está proibido.

Como ilustração suponha que o movimento q = 20, i = 1, t = 2 e tl = 3 (representado por m1 na figura 3.6.) foi realizado e que itfac = 10 e TAG = 5. Deseja-se proibir o movimento reverso a este, ou seja, proibir a realização de movimentos que retornem o item 1 ao período 2. Desta forma, a posição (1, 2) da matriz LISTA receberá 15 (itfac + TAG), ou seja, até a iteração 15, estão proibidos os movimentos que retornem o item 1 ao período 2.

Se um movimento q = 5, i = 1, t = 1 e tl = 2 for candidato em itfac = 13, por exemplo, tem-se que LISTA[1,2] = 15 (maior que 13) e portanto o movimento é descartado.

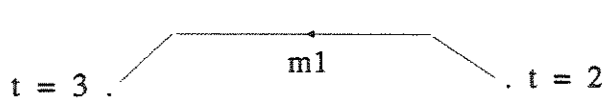

Figura 3.6. Exemplo de um movimento.

A inicialização da LISTA é feita somente no início, havendo apenas consulta e atualização até o final da heurística. A manipulação da Lista com a incorporação dos procedimentos P3 e P4 é descrito quando explicado Busca Tabu na heurística total.

- **Critério para Liberação da Condição Tabu dos Movimentos**

A princípio foi utilizado um critério de aspiração que, caso um movimento proibido passe pelo critério determinado, ele será realizado, desde que seja o melhor no Teste da Razão (o $Critério_1$ descrito a seguir). A seguir, foi testado o critério secundário que apenas é aplicado quando não há movimentos não tabu disponíveis (o $Critério_2$ descrito a seguir), já que na utilização do $Critério_1$ a ciclagem foi observada.

- **$CRITÉRIO_1$**

Se um dado movimento tabu (i, q, t, tl) produz:

$$\boxed{\text{custo\_adicional} < 0}$$

então este movimento perde sua condição tabu e é avaliado pelo Teste da Razão anteriormente definido ([29]). Ou seja, se um movimento reduz o custo, este pode ser liberado de sua condição tabu. Na implementação deste critério, observou-se ciclagem novamente.

- **$CRITÉRIO_2$**

Se não há nenhum movimento não tabu que possa ser realizado então avalie os movimentos tabu segundo o Teste da Razão abaixo e realize o que obtiver menor Razão.

$$\boxed{\text{Razão\_asp} = \frac{custo\_adicional}{excesso\_eliminado}} \qquad [35]$$

onde custo_adicional é calculado como em [30] e excesso_eliminado como em [32]

Observe que este cálculo da Razão equivale ao teste utilizado no procedimento de factibilização original (em [29]) com $\alpha = 1$ e $\beta = 0$.

Com a utilização do Critério$_2$ foram obtidos melhores resultados.

A seguir são apresentados os pseudo-códigos dos passos regressivo e progressivo utilizando o atributo (i, t), proibindo o movimento reverso e aplicando o Critério$_2$ como critério para liberação da condição tabu dos movimentos.

Sendo

TAG = número de iterações em que o atributo fica tabu (tamanho da lista).

it_fac = contador de transferências no procedimento de factibilização.

n = número de ciclos (um ciclo é um passo regressivo e um progressivo).

fase_tabu = verifica se a consulta será realizada, ou seja, se for a primeira aplicação do procedimento (r = 1) e o primeiro ciclo (n = 1), a consulta não é feita.

aspiração = se o movimento está proibido faz com que seja aplicado o critério o secundário

As alterações do algoritmo pela implementação de Busca Tabu estão destacadas em negrito e itálico.

```
se ( r = 1 e n = 1 )   então   fase_tabu = false
                        senão   fase_tabu = true
para t = T até 2 faça
  enquanto ( EXCESSO(t) > 0 ) faça
      Rmin = Rmin_asp = Infinito
      para i = 1 até N faça
          se ( xit > 0 ) então
              τ = período anterior à t com produção do item i, ou
              1, caso não exista período anterior à t com produção do item i
              para tl = t - 1 até τ faça
                  se ( fase_tabu e LISTA[ i ][ tl ] >= itfac ) então aspiração = true
                      M = CALCULA_M ( i, t, tl )
                      se (M = 0) então  tl = τ
                      senão
                          para k = 0 até K faça

                              se ( k = 0 ) então q = M
                              senão Q = CALCULA_Q (i, t, k)
                                     se ( Q < M ) q = Q

                              se ( aspiração ) então
                                  R = RAZÃO_ASP ( q, i, t, tl )
                                  se ( R < Rmin_asp )
                                      qo_asp = q; io_asp = i; tlo_asp = tl; Rmin_asp = R

                              fim { se }
                              senão
                                  R = RAZÃO ( q, i, t, tl )
                                  se ( R < Rmin )
                                      qo  = q; io = i tlo = tl; Rmin = R

                              fim { senão }
                          fim { para k }
                      fim { senão }
                  fim { se fase_tabu }
              fim { para tl }
          fim { se x > 0 }
      fim { para i }
      se existe ( qo, io, t, tlo ) então    ATUALIZA ( qo, io, t, tlo )
                                            LISTA[ io ][ t ] = TAG + itfac
      fim { se }
      senão   ATUALIZA (qo_asp, io_asp, t, tlo_asp)
              LISTA[ io_asp ][ t ] = TAG + itfac
      fim { senão }
      itfac = itfac + 1
  fim { enquanto Excesso > 0 }
fim { para t }
```

■ PASSO PROGRESSIVO DA FACTIBILIZAÇÃO COM BUSCA TABU [A8]

```
se ( r = 1 e n = 1 )  então  fase_tabu = false
                      senão  fase_tabu = true
para t = 1 até T-1 faça
  enquanto ( EXCESSO(t) > 0 e t < T ) faça
      Rmin = Rmin_asp = Infinito
      para i = 1 até N faça
         se ( xit > 0 ) então
            τ = período posterior à t com produção do item i, ou
            T, caso não exista período posterior à t com produção do item i
            para tl = t + 1 até τ faça
               se ( fase_tabu e LISTA[ i ][ tl ] >= itfac ) então aspiração = true
                  M = CALCULA_M ( i, t, tl )
                  se (M = 0) então  tl = τ
                  senão
                     para k = 0 até K faça

                        se ( k = 0 ) então q = M
                        senão Q = CALCULA_Q (i, t, k)
                              se ( Q < M ) q = Q

                        se ( aspiração ) então
                           R = RAZÃO_ASP ( q, i, t, tl )
                           se ( R < Rmin_asp )
                              qo_asp = q; io_asp = i; tlo_asp = tl; Rmin_asp = R

                        fim { se }
                        senão
                           R = RAZÃO ( q, i, t, tl )
                           se ( R < Rmin )
                              qo  = q; io = i tlo = tl; Rmin = R

                        fim { senão }
                     fim { para k }
                  fim { senão }
               fim { se fase_tabu }
            fim { para tl }
         fim { se x > 0 }
      fim { para i }
      se existe ( qo, io, t, tlo ) então    ATUALIZA ( qo, io, t, tlo )
                                            LISTA[ io ][ t ] = TAG + itfac
                                            itfac = itfac + 1
      fim { se }
      senão se existe ( qo_asp, io_asp, t, tlo_asp ) então ATUALIZA (qo_asp, io_asp, t, tlo_asp)
                                                           LISTA[ io_asp ][ t ] = TAG + itfac
                                                           it_fac = it_fac + 1
            senão t = t + 1
      fim { senão }
  fim { enquanto Excesso > 0 }
fim { para t }
```

3.7.2 Busca Tabu na Heurística

As estratégias de Busca Tabu utilizadas no procedimento de factibilização (P2) foram implementadas com o objetivo de evitar a ciclagem ocorrida internamente neste procedimento, quando transferências feitas para frente desfaziam transferências efetuadas para trás, e vice-versa.

A partir da incorporação dos procedimentos de melhoria e agregação, formando assim a heurística, foi observada ciclagem entre os diferentes procedimentos, quando as agregações de lotes efetuadas no procedimento de agregação (P4) eram desfeitas no procedimento de factibilização (P2). Desta forma, ampliou-se a utilização de Busca Tabu em outros procedimentos da heurística.

A implementação de Busca Tabu na heurística foi feita de duas maneiras:

- com a utilização de uma **única lista** gerenciando os movimentos de todos os procedimentos.
- com a utilização de **duas listas**, uma no interior do procedimento de factibilização, como mostrado no tópico anterior, e uma segunda lista gerenciando as transferências de todos os procedimentos.

As duas implementações são discutidas a seguir.

▪ Utilizando uma única lista

Neste caso, a implementação é equivalente à apresentada no procedimento de factibilização, mas com a Lista Tabu sendo atualizada também nos procedimentos de melhoria (P3) e de agregação (P4), e ainda, com consulta à Lista Tabu sendo feita também no procedimento de agregação.

Assim, realizado um movimento (q, i, t, tl), como mostrado na figura 3.7, a restrição tabu atua sobre os movimentos representados pela figura 3.8.. Desta forma, a cada movimento (q, i, t, tl) efetuado nos procedimentos P2, P3 e P4, a posição (i, t) da matriz LISTA é atualizada com TAG + it, onde it representa um contador de transferências efetuadas durante a heurística. A cada movimento candidato nos procedimentos de factibilização (P2) e de agregação (P4), o atributo do respectivo movimento é verificado se está tabu ou não. Caso esteja tabu, o movimento é descartado. Caso contrário, o movimento é avaliado.

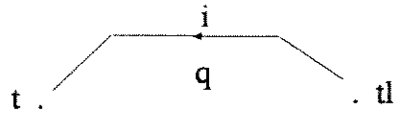

Figura 3.7.

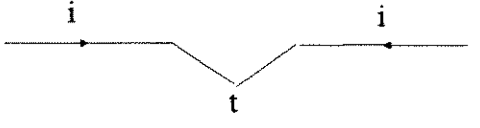

Figura 3.8.

Durante o procedimento de melhoria (P3), a LISTA é apenas atualizada, não havendo consulta, ou seja, não há restrição aos movimentos escolhidos durante esse procedimento. Assim que são realizados, ficam proibidos de serem desfeitos durante um certo número de iterações. Os movimentos efetuados durante P3, melhoram a solução, não permitindo infactibilidade e assim, torna-se interessante permití-los, tornando a busca mais flexível.

Nos procedimentos P2 e P4, caso todos os movimentos candidatos sejam tabu, o **Critério2** é aplicado.

O pseudo-código da heurística com a implementação de Busca Tabu é equivalente ao pseudo-código 6. (da heurística) com as seguintes alterações:

- Existe apenas um contador de transferências (it) para todos os procedimentos, que é inicializado com zero no início da heurística.

- Os passos Regressivo e Progressivo do procedimento de factibilização são os mesmos que os mostrados nos pseudo-códigos [A7] e [A8], com o contador itfac substituído por it.

- O procedimento de melhoria é equivalente ao mostrado nos pseudo-códigos [A3] e [A4], mas a cada movimento realizado, a LISTA é atualizada na posição (i, t), com it + TAG.

- O procedimento de agregação é equivalente ao mostrado no pseudo-código [A5], mas a cada movimento candidato, é verificado se o seu respectivo atributo está na LISTA e a cada movimento realizado, a LISTA é atualizada na posição (i, t), com it + TAG. O procedimento original (sem tabu) é alterado do mesmo modo como o procedimento de factibilização foi modificado para incorporação de Busca Tabu.

- No procedimento de agregação, ainda é implementado o **Critério2** como utilizado no procedimento de factibilização.

■ **Utilizando duas listas**

A idéia da utilização de duas listas tabu, dentro da heurística, deve-se ao fato de dois tipos de ciclagem terem sido observados, uma interna ao procedimento de factibilização e outra entre os procedimentos. Assim, uma lista tentaria prevenir a ciclagem interna ao procedimento de factibilização e a outra entre os procedimentos.

A primeira lista é utilizada como mostrado no tópico 3.7.1, onde foi mostrada a implementação de Busca Tabu apenas no procedimento de factibilização.

A segunda Lista foi utilizada para impedir que os lotes agregados durante o procedimento P4 fossem desfeitos durante a factibilização. Desta maneira, o atributo utilizado nesta Lista foi (i, tl), ou seja, dado que a produção de um lote foi para o período tl, não permite que este saia de tl durante um certo número de iterações. Assim, se um movimento (q, i, t, tl) é efetuado, a restrição proíbe o item i sair de tl.

Seja LISTA1, a matriz que representa a lista tabu no procedimento de factibilização e LISTA2, a matriz que representa a segunda lista tabu. A cada movimento (q, i, t, tl) como representado na figura 3.9, as listas 1 e 2 atuam proibindo movimentos como representado na figura 3.10 e 3.11, respectivamente.

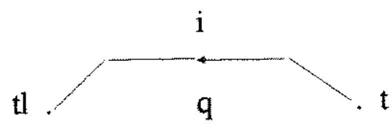

Figura 3.9.

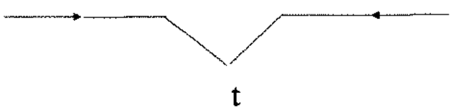

Figura 3.10. LISTA 1 - atributo (i,t).

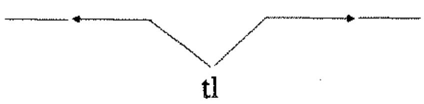

Figura 3.11. LISTA 2 - atributo (i,tl).

As atualizações e consultas às listas são feitas da maneira descrita na tabela 3.7, a seguir.

Procedimento		**LISTA1**	**LISTA2**
Factibilização (P2)	atualização	(i , t)	———
	consulta	(i , tl)	(i , t)
Melhoria (P3)	atualização	———	(i , tl)
	consulta	———	———
Agregação (P4)	atualização	———	(i , tl)
	consulta	———	(i , t)

Tabela 3.7. Manipulação da Lista Tabu quando utilizadas 2 listas.

A LISTA1 atua apenas no procedimento de factibilização. A LISTA2 se aplica na heurística inteira, sendo que no procedimento de factibilização a LISTA2 é apenas consultada para não desfazer movimentos que foram efetuados nos procedimentos de melhoria e agregação.

O critério para liberação da condição tabu foi o mesmo que o aplicado no caso de uma lista, ou seja, o **Critério2** anteriormente explicado. Assim, quando todos os movimentos forem tabu, a respectiva lista é analisada. No procedimento de factibilização as listas 1 e 2 são verificadas e todos os movimentos de ambas as listas são analisados segundo o **Critério2**. Na totalidade dos procedimentos de melhoria e agregação apenas a lista 2 é verificada, já que a lista 1 não atua como elemento restritivo nestas duas fases.

O pseudo código da heurística com a implementação de Busca Tabu é equivalente ao algoritmo 6. (da heurística) com as seguintes alterações:

- Existem dois contadores de transferências. Um para as transferências realizadas apenas no procedimento de factibilização (itfac) e outro para qualquer transferência realizada durante a heurística (it), mesmo dentro do procedimento de factibilização, sendo que os dois contadores são inicializados com zero no início da heurística.

- Os passos Regressivo e Progressivo do procedimento de factibilização são os mesmos que os mostrados nos pseudo-códigos [A7] e [A8], mas na consulta é preciso verificar as 2 listas. Assim a linha

  se (fase tabu e LISTA[i][tl] >= itfac) então aspiração = true;

  é substituída por

  se (fase tabu e LISTA1[i][tl] >= itfac e LISTA2[i][t] >= it)
  então aspiração = true;

  e a atualização é feita apenas na LISTA1. Assim a atualização

  LISTA[i][t] = TAG + itfac

  é substituída por

  LISTA1[i][t] = TAG + itfac

- O procedimento de melhoria é equivalente ao mostrado nos pseudo-códigos [A3] e [A4], mas a cada movimento realizado, a LISTA2 é atualizada na posição (i, tl) com it + TAG.

- O procedimento de agregação é equivalente ao mostrado no pseudo-código [A5], mas a cada movimento candidato, é verificado se o seu respectivo atributo está na LISTA2 e cada movimento realizado, a LISTA 2 é atualizada na posição (i, tl) com it + TAG.

Portanto, existem três versões para a heurística:

1. Heurística sem Busca Tabu.
2. Heurística com Busca Tabu, utilizando 1 lista.
3. Heurística com Busca Tabu, utilizando 2 listas.

No capítulo a seguir é discutido o desempenho de cada uma das versões, além de uma análise apenas utilizando o procedimento de factibilização.

CAPÍTULO 4

RESULTADOS COMPUTACIONAIS

4.1. Introdução

Neste capítulo são apresentados os testes computacionais realizados com o objetivo de avaliar a heurística proposta no capítulo 3.

As soluções alcançadas pela heurística são comparadas com um limitante inferior obtido através da aplicação de Relaxação Lagrangeana ao problema representado pelo modelo [ME].

Os dados utilizados nos problemas para os testes computacionais foram gerados aleatoriamente.

O método heurístico apresentado foi implementado em linguagem C e os testes foram realizados numa estação de trabalho SUN, modelo SPARCstation1+.

4.2. Obtenção de um Limitante Inferior através de Relaxação Lagrangeana.

O problema representado pelo modelo [ME], sem o conjunto de restrições [17] e [18], reúne N problemas monoestágios de dimensionamento de lotes com capacidade infinita de produção. Utilizando Relaxação Lagrangeana, dualizam-se essas duas restrições, incluindo-as na função objetivo. O problema lagrangeano resultante fornece um limitante inferior para o valor ótimo do problema representado pelo modelo [ME].

Dualizando-se as restrições [17] e [18], o problema [ME] se escreve:

$$\mathbf{Z}\ (\lambda,\mu) = \min \sum_{i=1}^{N} \sum_{t=1}^{T} (e_{it} E_{it} + c_{it} x_{it} + s_{it} y_{it})$$

$$+ \lambda_{it} \sum_{i=1}^{N} \sum_{t=1}^{T} \left[\sum_{j \in S(i)} r_{ij} E_{jt} - E_{it} \right] + \sum_{k=1}^{K} \sum_{t=1}^{T} \mu_{kt} \left[\sum_{i=1}^{N} (v_{ikt} x_{it} + f_{ikt} y_{it}) - CAP_{kt} \right]$$

$$E_{i,t-1} + x_{it} - E_{it} = D_{it} \qquad i = 1,\dots,N;\ t = 1,\dots,T$$

$$x_{it} \leq M_{it} y_{it} \qquad i = 1,\dots,N;\ t = 1,\dots,T$$

$$y_{it} = 0\,,1 \qquad i = 1,\dots,N;\ t = 1,\dots,T$$

$$x_{it}\ e\ E_{it} \geq 0 \qquad i = 1,\dots,N;\ t = 1,\dots,T$$

onde $\lambda_{it} \geq 0$ $\quad i = 1,\dots,N\ ;\ t = 1,\dots,T$

$\mu_{kt} \geq 0$ $\quad k = 1,\dots,K;\ t = 1,\dots,T$ são os multiplicadores lagrangeanos correspondentes ao conjunto de restrições [17] e [18], respectivamente.

Reordenando os termos da função objetivo:

$$\sum_{i=1}^{N}\sum_{t=1}^{T}\left[\left(s_{it}+\sum_{k=1}^{K}\mu_{kt}f_{ikt}\right)y_{it}+\left(c_{it}+\sum_{k=1}^{K}\mu_{kt}v_{ikt}\right)x_{it}\right]-\sum_{k=1}^{K}\sum_{t=1}^{T}\mu_{kt}\,CAP_{kt}$$

$$+\sum_{i=1}^{N}\sum_{t=1}^{T}e_{it}E_{it}+\sum_{i=1}^{N}\sum_{t=1}^{T}\lambda_{it}\left[\sum_{j\in S(i)}r_{ij}E_{jt}-E_{it}\right]=$$

$$\sum_{i=1}^{N}\sum_{t=1}^{T}\left[\left(s_{it}+\sum_{k=1}^{K}\mu_{kt}f_{ikt}\right)y_{it}+\left(c_{it}+\sum_{k=1}^{K}\mu_{kt}v_{ikt}\right)x_{it}\right]-\sum_{k=1}^{K}\sum_{t=1}^{T}\mu_{kt}\,CAP_{kt}$$

$$+\sum_{i=1}^{N}\sum_{t=1}^{T}\left(e_{it}-\lambda_{it}\right)E_{it}+\sum_{i=1}^{N}\sum_{t=1}^{T}\lambda_{it}\sum_{j\in S(i)}r_{ij}E_{jt}\ .$$

Substituindo $j \in S(i)$ por $j \in P(i)$ (j e i ficam invertidos), tem-se

$$\sum_{i=1}^{N}\sum_{t=1}^{T}\left[\left(s_{it}+\sum_{k=1}^{K}\mu_{kt}f_{ikt}\right)y_{it}+\left(c_{it}+\sum_{k=1}^{K}\mu_{kt}v_{ikt}\right)x_{it}\right]$$

$$+\sum_{i=1}^{N}\left[\sum_{t=1}^{T}\left(e_{it}-\lambda_{it}\right)+\sum_{j\in P(i)}r_{ji}\sum_{t=1}^{T}\lambda_{jt}\right]E_{it}-\sum_{k=1}^{K}\sum_{t=1}^{T}\mu_{kt}\,CAP_{kt}$$

Reescrevendo o problema lagrangeano:

MODELO [MR]

$$Z\ (\lambda,\mu) = \min \sum_{i=1}^{N} \left[\sum_{t=1}^{T} \left[\left(s_{it} + \sum_{k=1}^{K} \mu_{kt} f_{ikt} \right) y_{it} + \left(c_{it} + \sum_{k=1}^{K} \mu_{kt} v_{ikt} \right) x_{it} \right] \right.$$

$$\left. + \left[\sum_{t=1}^{T} \left(e_{it} - \lambda_{it} \right) + \sum_{j \in P(i)} r_{ji} \sum_{t=1}^{T} \lambda_{jt} \right] E_{it} \right] - \sum_{k=1}^{K} \sum_{t=1}^{T} \mu_{kt}\ CAP_{kt}$$

$$E_{i,t-1} + x_{it} - E_{it} = D_{it} \qquad i = 1,\ldots,N\ ;\ t = 1,\ldots,T$$

$$x_{it} \leq M_{it}\, y_{it} \qquad i = 1,\ldots,N\ ;\ t = 1,\ldots,T$$

$$y_{it} = 0\ ,1 \qquad i = 1,\ldots,N\ ;\ t = 1,\ldots,T$$

$$x_{it}\ e\ E_{it} \geq 0 \qquad i = 1,\ldots,N\ ;\ t = 1,\ldots,T$$

onde,

$\left(s_{it} + \sum_{k=1}^{K} \mu_{kt} f_{ikt} \right) =$ custo lagrangeano de preparação.

$\left(c_{it} + \sum_{k=1}^{K} \mu_{kt} v_{ikt} \right) =$ custo lagrangeano de produção.

$\left[\sum_{t=1}^{T} \left(e_{it} - \lambda_{it} \right) + \sum_{j \in P(i)} r_{ij} \sum_{t=1}^{T} \lambda_{it} \right] =$ custo lagrangeano de estoque de escalão.

$\sum_{k=1}^{K} \sum_{t=1}^{T} \mu_{kt}\ CAP_{kt} =$ parcela fixa da função objetivo.

Note que o problema [MR] decompõe-se em N problemas monoestágios de dimensionamento de lotes com capacidade infinita de produção (um para cada componente).

Portanto, para cada item i resolve-se o problema:

$$Z_i\ (\lambda,\mu) = \sum_{t=1}^{T} \left[\left(s_{it} + \sum_{k=1}^{K} \mu_{kt} f_{ikt} \right) y_{it} + \left(c_{it} + \sum_{k=1}^{K} \mu_{kt} v_{ikt} \right) x_{it} \right]$$

$$+ \left[\sum_{t=1}^{T} \left(e_{it} - \lambda_{it} \right) + \sum_{j \in P(i)} r_{ji} \sum_{t=1}^{T} \lambda_{jt} \right] E_{it}$$

$$E_{i,t-1} + x_{it} - E_{it} = D_{it} \qquad t = 1,\ldots,T$$

$$x_{it} \leq M_{it}\, y_{it} \qquad t = 1,\ldots,T$$

$$y_{it} = 0\,,1 \qquad t = 1,\ldots,T$$

$$x_{it}\ e\ E_{it} \geq 0 \qquad t = 1,\ldots,T$$

Pode-se utilizar o algoritmo de Wagner-Whitin [WAGNER e WHITIN, 1958] para a resolução de cada um dos N problemas. O limitante inferior para o problema representado pelo modelo [ME] é dado por:

$$Z\ (\lambda,\mu) = \sum_{i=1}^{N} Z_i\ (\lambda,\mu) - \sum_{k=1}^{K} \sum_{t=1}^{T} \mu_{kt}\ CAP_{kt}$$

Para maximizar a função dual (Z (λ,μ)), com o propósito de obter um melhor limitante inferior, foram realizadas 200 iterações do método de subgradiente incorporando as modificações propostas por [CAMERINI, FRATTA e MAFFIOLI, 1975].

4.3. Geração dos dados

A realização dos testes computacionais foi feita com problemas gerados aleatoriamente através de uma distribuição uniforme. Os parâmetros usados para tal geração são baseados em [CLARK, 1990].

A quantidade de itens (N), número de períodos (T) e quantidade de recursos (K) foram previamente escolhidos e divididos em três exemplos:

EX1 - $N = 10$; $T = 12$; $K = 1$.

EX2 - $N = 10$; $T = 12$; $K = 2$.

EX3 - $N = 17$; $T = 10$; $K = 1$.

Para cada exemplo, especificado acima, foram consideradas três estruturas diferentes:

SERIAL

PLANA

GERAL

onde a estrutura geral considerada foi:

para $N = 10$,

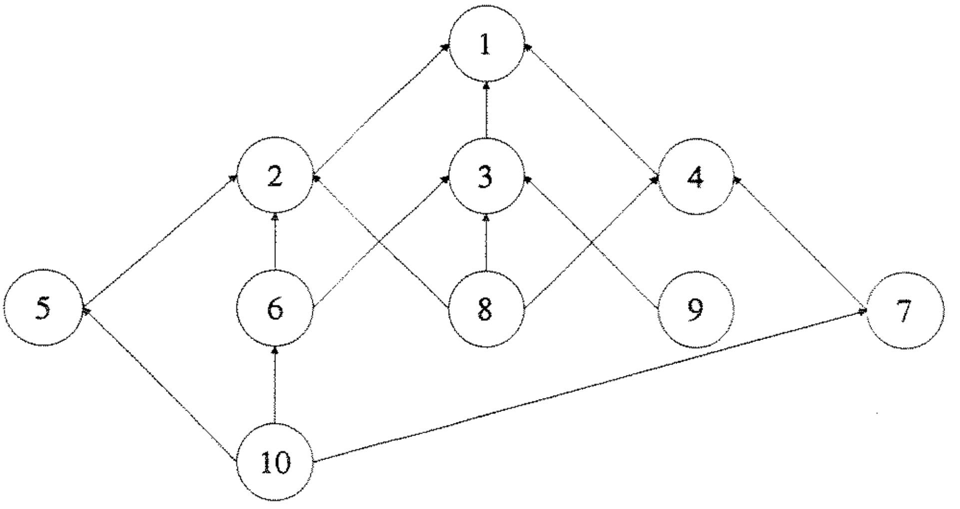

Figura 4.1. Estrutura de produtos para N = 10 encontrada em [CLARK, 1990].

e para N = 17,

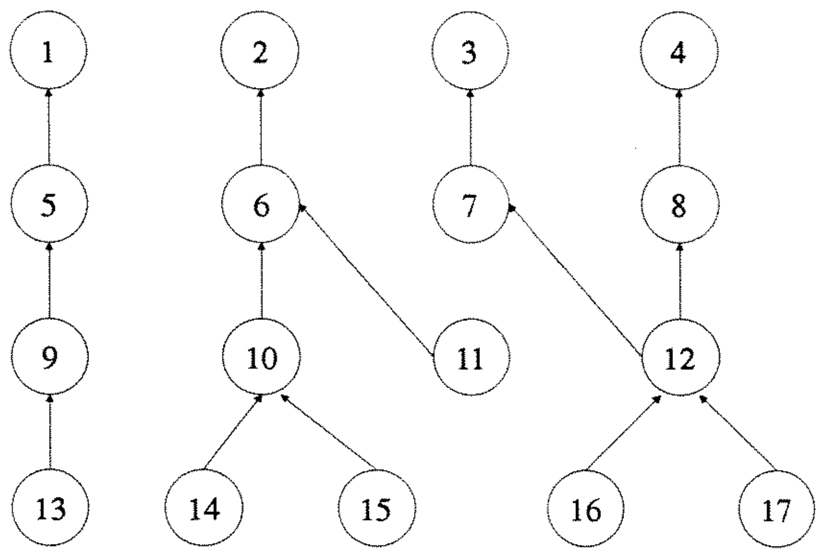

Figura 4.2. Estrutura de produtos para N = 17 encontrada em [MAES e MCLAIN, 1991].

Para cada um dos problemas acima, foram gerados dados utilizando os parâmetros descritos a seguir, onde U[a,b] representa uma geração aleatória .através de distribuição uniforme entre os valores a e b.

- custo unitário de produção (c_{it}) :
  U [1,5 , 2].

- custo de preparação (s_{it}) :
  U [5 , 95] para caracterizar problemas com custo baixo de preparação.
  U [50, 950] para caracterizar problemas com custo alto de preparação.

- custo de estoque de escalão (e_{it}) :
  U [0,2 , 0,4].

- recurso unitário para produção (v_{ikt}) :
  U [150, 250] para k = 1.
  U [200, 300] para k = 2.

- recurso para preparação (f_{ikt}) :

  U [1,5 , 2,5] para k = 1.

  U [2 , 3] para k = 2.

- demanda (d_{it}) :

  U [0 , 180] para itens finais, ou seja, para todo i tal que S(i) = $\emptyset$.

  U [0 , 18] para os demais itens, ou seja, para todo i tal que S(i) $\neq$ $\emptyset$.

- $r_{ij} = 1$ para $i = 1,\ldots,N$ e $j \in S(i)$.

- capacidade (CAP_{kt}) :

  Para a determinação da capacidade em cada período t foi utilizada a solução obtida com a aplicação de uma política lote-por-lote. Nessa política, para cada período t, calcula-se a quantidade de recursos utilizada caso a produção neste período seja exatamente a demanda do período, ou seja,

$$C_{kt} = \sum_{i=1}^{N} \left(f_{ikt} + v_{ikt} D_{it} \right) \qquad se \;\; D_{it} \neq 0 \qquad \begin{array}{l} t = 1,\ldots,T \\ i = 1,\ldots,N \end{array}$$

A partir deste cálculo foi calculada uma média de modo que a quantidade de um determinado recurso k, em cada período t, seja a mesma, isto é,

$$C_{kt} \leftarrow \sum_{t=1}^{T} (C_{kt})/T \qquad \begin{array}{l} t = 1,\ldots,T \\ k = 1,\ldots,K \end{array}$$

Após calculada a capacidade disponível de cada recurso k em cada período t, os problemas foram divididos em três categorias, segundo a disponibilidade dos recursos:

Cap1 : fazendo CAP[k][t] = 0.9*Ckt $\quad \begin{array}{l} t = 1,\ldots,T \\ k = 1,\ldots,K \end{array}$ [36]

Cap2 : fazendo CAP[k][t] = Ckt $\quad \begin{array}{l} t = 1,\ldots,T \\ k = 1,\ldots,K \end{array}$ [37]

Cap3 : fazendo CAP[k][t] = 1,1*Ckt $\quad \begin{array}{l} t = 1,\ldots,T \\ k = 1,\ldots,K \end{array}$ [38]

Desta forma, os problemas que utilizarem Cap1 são os mais apertados e os com Cap3 são os mais folgados.

A cada problema gerado foi feito um cálculo da utilização dos recursos para a produção da demanda do primeiro período ($I_i0 = 0$). Caso esta utilização exceda a disponibilidade de recursos no primeiro período, este problema é descartado, pois mostra-se infactível já para o primeiro período.

Portanto, foram considerados três exemplos (EX1, EX2 e EX3), três tipos de estruturas, custos de preparação altos e baixos, três capacidades diferentes a serem utilizadas por período e ainda para cada um destes problemas foram utilizadas 20 sementes para a geração aleatória dos números, ou seja,

3 exemplos x 3 estruturas x 2 custos de preparação x 3 capacidades x 20 sementes = 1080 problemas.

4.4. Testes Realizados

A heurística foi implementada e testada em três etapas:

Etapa 1 - Implementação dos procedimentos P1 (solução de partida) e P2 (factibilização).

Etapa 2 - Implementação do procedimento P3 (melhoria).

Etapa 3 - Implementação do procedimento P4 (agregação).

Os parâmetros utilizados na heurística para obtenção dos resultados mostrados a seguir foram:

ITMAX (número de iterações máximo para a heurística) = 100.

n_max (número de ciclos máximo para o procedimento de factibilização) = 6.

4.4.1 Teste na Etapa 1

No procedimento de factibilização os testes foram realizados em duas fases. Inicialmente, foram feitos experimentos relacionados com a expressão da Razão, encontrada em [29]. A seguir, com a expressão da Razão definida, o algoritmo com a incorporação de Busca Tabu foi testado.

- **Testes realizados na expressão da Razão a ser utilizada:**

Seja a expressão da Razão dada por

$$\text{Razão} = \frac{\alpha.custo\_adicional + \beta.penalidade}{denominador} \qquad [39]$$

- No **denominador** foram testados

denominador = 1

denominador = excesso_eliminado (visto em [32]).

denominador = quantidade transferida (= q).

Utilizando o denominador igual a excesso_eliminado foram obtidos melhores resultados.

- Em relação ao parâmetro **penalidade** foram feitos os seguintes testes

PADRÃO -	sem utilizar penalidade, ou seja, $\alpha = 1$ e $\beta = 0$.
PENALIDADE1 -	considerando o mesmo peso para o custo_adicional e para a penalidade, ou seja, $\alpha = 1$ e $\beta = 1$.
PENALIDADEn -	fornecendo um peso à penalidade de acordo com o ciclo no procedimento de factibilização, ou seja, $\alpha = 1$ e $\beta = n$, onde n representa o número do ciclo no procedimento de factibilização.

Os resultados dessa fase estão contabilizados na tabela 4.1 Na tabela, **%FAC** representa a porcentagem de soluções factíveis encontradas dentre os problemas onde a heurística foi realmente testada, ou seja,

$$\%FAC = \frac{NF}{NG - ND} \cdot 100 \qquad [40]$$

onde NF = número de problemas onde foi encontrada solução factível.

NG = número de problemas gerados aleatoriamente.

ND = número de problemas descartados por serem infactíveis já no primeiro período.

E **%GAP** representa a média entre os *gaps* das soluções obtidas, onde o *gap* foi calculado como:

$$\frac{Z_O - Z_I}{Z_I} \cdot 100 \qquad [41]$$

onde Zo = valor da melhor solução obtida pela heurística.

Zl = valor do limitante inferior calculado como mostrado em 4.2.

ESTRUTURA	Custo de preparação	PADRÃO		PENALIDADE1		PENALIDADEn	
		%FAC	%GAP	%FAC	%GAP	%FAC	%GAP
Plana	baixo	69,44	1,33	83,61	1,97	83,61	1,99
	alto	72,22	17,18	78,89	11,69	82,64	13,69
	média	**70,83**	**9,41**	**81,25**	**6,69**	**83,12**	**7,81**
Serial	baixo	57,92	8,81	64,17	9,12	62,92	9,11
	alto	56,67	40,14	60,00	35,31	64,03	34,40
	média	**57,29**	**24,30**	**62,08**	**21,78**	**63,47**	**23,38**
Geral	baixo	56,06	4,20	62,93	5,19	63,16	5,24
	alto	50,87	24,86	64,69	22,16	64,69	22,37
	média	**53,47**	**14,03**	**63,81**	**13,79**	**63,93**	**13,91**
Média Total		**60,53**	**15,47**	**69,05**	**13,40**	**70,18**	**14,35**

Tabela 4.1. Testes realizados com a expressão da Razão utilizada no procedimento de factibilização.

Nesta fase, as estratégias foram avaliadas privilegiando-se a quantidade de soluções factíveis encontradas nos problemas testados, pois o propósito dos procedimentos P1 e P2 é encontrar uma solução factível inicial. Assim, a Razão que obteve os melhores resultados foi pelo teste PENALIDADEn, que definiu a expressão da Razão como sendo

$$\text{Razão} = \frac{\alpha.custo\_adicional + \beta.penalidade}{excesso\_eliminado} \qquad [42]$$

Com a expressão da razão definida, foi incorporada Busca Tabu nesta configuração do procedimento de factibilização.

- **Testes realizados com Busca Tabu**

Nesta fase, foram testados o critério de aspiração e o tamanho da lista utilizados.

- **Critério para liberação da condição tabu dos movimentos.**
  - Utilizando o critério de aspiração Critério1 foi observado o fenômeno da ciclagem.
  - O critério secundário Critério2, definido em [35], apresentou melhores resultados, e portanto, foi o adotado.

- **Tamanho da lista.**
  - Inicialmente, foi utilizando um tamanho fixo para o tamanho da lista, mas os resultados obtidos foram praticamente iguais aos apresentados pelo algoritmo sem utilizar Busca Tabu.
  - Utilizando tamanho de lista aleatório (explicado em 2.3.1, Implementação da Lista) foram obtidos melhores resultados. O uso de intervalos pequenos, bem como tamanhos de listas pequenos mostraram resultados-superiores. Os resultados a serem mostrados a seguir foram realizados com a utilização de geração aleatória de números para o TABU TAG no intervalo de [0,7].

A tabela 4.2. apresenta os resultados comparando o algoritmo sem a utilização de Busca Tabu (FST) e com a utilização de Busca Tabu (FCT). Na tabela, além de %FAC e %GAP, explicados em [40] e [41], é mostrado o tempo médio gasto para encontrar uma solução factível (**TEMPOf**), onde o tempo é obtido por uma média dos resultados FST e FCT. Vale ressaltar que os tempos em FST e FCT foram praticamente iguais. Cap1, cap2 e cap3 representam a disponibilidade de recursos por período como definido em [36], [37] e [38], respectivamente. Observe que o aumento no *gap* não significa uma piora dos resultados, já que a quantidade de soluções factíveis aumentou, e isso pode significar a inclusão de problemas com alto *gap*.

ESTRUTURA	Custo de preparação	Capacidade disponível	FST		FCT		TEMPOf (s)
			%FAC	%GAP	%FAC	%GAP	
Plana	baixo	cap1	50,83	3,48	55,00	3,60	0,79
		cap2	100,00	1,93	100,00	1,93	0,15
		cap3	100,00	1,30	100,00	1,30	0,12
	alto	cap1	52,92	26,12	55,00	25,56	1,34
		cap2	95,00	13,10	100,00	14,02	0,30
		cap3	100,00	7,67	100,00	7,67	0,17
	média		**83,13**	**7,80**	**85,00**	**8,03**	**0,35**
Serial	baixo	cap1	17,08	13,73	20,42	13,66	0,09
		cap2	71,67	9,96	78,33	10,08	0,72
		cap3	100,00	7,70	100,00	7,70	0,11
	alto	cap1	22,08	48,90	27,50	55,93	1,32
		cap2	70,00	44,21	80,00	46,58	0,39
		cap3	100,00	30,10	100,00	30,09	0,15
	média		**63,47**	**23,38**	**67,71**	**24,89**	**0,35**
Geral	baixo	cap1	15,10	9,35	15,10	9,24	0,51
		cap2	77,92	5,79	86,67	5,66	0,24
		cap3	96,48	4,16	98,15	4,20	0,33
	alto	cap1	18,43	32,47	15,10	29,05	0,15
		cap2	77,50	25,69	84,58	26,24	0,53
		cap3	98,15	17,85	98,15	17,88	0,20
	média		**63,93**	**13,91**	**66,29**	**13,71**	**0,29**
Média Total			**70,18**	**14,35**	**73,00**	**14,96**	**0,34**

Tabela 4.2. Resultados dos testes no procedimento de factibilização com e sem Busca Tabu.

4.4.2 Testes na etapa 2

Nessa etapa foi incorporado o procedimento de melhoria (P3) ao método. Os testes foram realizados com o propósito de analisar os ganhos deste procedimento ao método e ainda, a incorporação de Busca Tabu.

Inicialmente foi incluído o procedimento de melhoria ao procedimento de factibilização na configuração definida pelo teste PENALIDADEn. Este teste foi feito sem a incorporação de Busca Tabu e está mostrado na tabela 4.3 na coluna MST (melhoria sem tabu). A seguir, o procedimento P3 foi incorporado à configuração FCT (factibilização com tabu), ou seja, PENALIDADEn com Busca Tabu e está mostrado na tabela 4.3 na coluna MCT (melhoria com tabu).

Na aplicação de Busca Tabu foi utilizada apenas 1 lista. Nos testes relacionados com os parâmetros utilizados foram obtidos os mesmos resultados que os alcançados no item anterior, ou seja, a utilização de Critério2 e de lista aleatória conseguiram os melhores resultados.

A tabela 4.3. mostra %GAP e o tempo médio gasto quando ocorre a obtenção de uma solução factível (TEMPOf) nos testes que incluem o procedimento de melhoria (MST e MCT), onde o tempo foi obtido segundo uma média dos resultados MST e MCT. %FAC não consta na tabela, pois a porcentagem de soluções factíveis obtidas pelo método com e sem o procedimento de melhoria é a mesma. Logo, FST e MST têm o mesmo valor de %FAC e, FCT e MCT também.

Vale ressaltar que há diferenças entre os valores de %FAC entre FST e FCT e entre MST e MCT. Portanto, é preciso ser cauteloso na comparação entre os valores de %GAP entre os testes com e sem tabu, pois, como dito anteriormente, um aumento no número de soluções factíveis pode representar a inclusão de problemas com alto *gap*.

ESTRUTURA	Custo de preparação	Capacidade disponível	FST	MST	FCT	MCT	TEMPOf (s)
			%GAP	%GAP	%GAP	%GAP	
Plana	baixo	cap1	3,48	2,27	3,60	2,36	5,37
		cap2	1,93	1,36	1,93	1,36	5,93
		cap3	1,30	0,94	1,30	0,94	4,97
	alto	cap1	26,12	19,65	25,56	18,34	8,22
		cap2	13,10	10,97	14,02	12,01	5,72
		cap3	7,67	6,75	7,67	6,81	3,98
	média		**7,80**	**6,22**	**8,03**	**6,32**	**5,44**
Serial	baixo	cap1	13,73	13,01	13,66	12,91	1,74
		cap2	9,96	8,88	10,08	9,09	3,06
		cap3	7,70	6,68	7,70	6,68	2,28
	alto	cap1	48,90	46,63	55,93	51,85	1,50
		cap2	44,21	41,21	46,58	43,62	2,65
		cap3	30,10	27,80	30,09	27,89	2,06
	média		**23,38**	**21,58**	**24,89**	**23,01**	**2,29**
Geral	baixo	cap1	9,35	8,53	9,24	8,09	3,63
		cap2	5,79	5,02	5,66	4,94	4,69
		cap3	4,16	3,49	4,20	3,54	3,98
	alto	cap1	32,47	27,26	29,05	24,23	3,20
		cap2	25,69	23,42	26,24	24,04	5,22
		cap3	17,85	16,20	17,88	16,27	3,77
	média		**13,91**	**12,42**	**13,71**	**12,31**	**4,11**
Média Total			**14,35**	**12,73**	**14,96**	**13,31**	**4,09**

Tabela 4.3. Testes realizados com a incorporação do procedimento de melhoria ao método.

4.4.3 Testes na etapa 3

Nesta etapa, a heurística está com os quatro procedimentos incorporados (P1, P2, P3 e P4). Os testes foram realizados na heurística sem e com a utilização de Busca Tabu, utilizando 1 lista e 2 listas. Os três testes descritos acima estão contabilizados na tabela 4.4., onde HST representa a heurística sem a utilização de Busca Tabu; HCT1 indica a heurística com Busca Tabu e utilizando apenas 1 lista; e HCT2 representa a utilização de 2 listas na fase tabu.

ESTRUTURA	Custo de preparação	Capacidade disponível	HST		HCT1		HCT2		TEMPOf (s)
			%FAC	%GAP	%FAC	%GAP	%FAC	%GAP	
Plana	baixo	cap1	55,00	1,73	57,08	1,71	52,92	1,73	43,08
		cap2	100,00	1,12	100,00	1,07	100,00	1,07	34,88
		cap3	100,00	0,83	100,00	0,78	100,00	0,81	29,77
	alto	cap1	50,83	10,23	52,92	9,45	50,83	11,46	44,19
		cap2	100,00	7,85	100,00	7,28	100,00	8,73	29,03
		cap3	100,00	5,12	100,00	4,69	100,00	5,15	20,99
	média		**84,31**	**4,17**	**85,00**	**3,88**	**83,96**	**4,47**	**31,40**
Serial	baixo	cap1	22,50	10,56	20,83	10,10	25,83	10,77	8,32
		cap2	75,00	8,33	80,00	8,06	76,67	8,41	14,87
		cap3	100,00	6,10	100,00	5,99	100,00	6,06	10,95
	alto	cap1	23,75	48,12	27,50	46,91	27,50	46,30	8,06
		cap2	71,67	39,78	80,00	38,45	76,67	38,46	12,29
		cap3	100,00	26,57	100,00	25,69	100,00	26,42	9,13
	média		**65,49**	**20,67**	**68,06**	**20,55**	**67,78**	**20,64**	**10,82**
Geral	baixo	cap1	22,13	4,85	18,80	4,84	22,13	5,58	22,79
		cap2	81,25	3,67	86,67	3,40	86,25	3,73	21,90
		cap3	98,15	2,88	98,15	2,77	98,15	2,86	19,21
	alto	cap1	25,47	27,72	18,80	23,54	27,43	29,19	18,82
		cap2	79,17	19,02	84,58	17,92	88,33	21,37	18,56
		cap3	98,15	14,15	98,15	12,96	98,15	14,14	15,85
	média		**67,39**	**10,48**	**67,52**	**9,60**	**70,07**	**11,42**	**19,52**
Média Total			**72,39**	**11,11**	**73,53**	**10,77**	**73,94**	**11,60**	**21,53**

Tabela 4.4. Testes realizados na heurística com e sem Busca Tabu.

4.5. Análise dos resultados

A análise dos resultados foi feita sob três diferentes aspectos:

- Diferenças observadas em relação aos diferentes parâmetros dos problemas.
- Ganho desde o método utilizado para o teste PADRÃO até a heurística finalizada.
- Ganho com a aplicação de Busca Tabu.

- **Em relação aos diferentes parâmetros nos problemas.**

A maior dificuldade em encontrar soluções factíveis e de boa qualidade (menor *gap*) encontra-se em problemas que apresentam estrutura serial. Os problemas que possuem custos de preparação altos são os que possuem os maiores *gaps*, mas este parâmetro não influencia na busca de soluções factíveis. Os problemas considerados apertados, pela disponibilidade de recursos, são os que possuem a maior dificuldade na obtenção de soluções factíveis, o que já era esperado.

- **Diferenças observadas entre as diferentes configurações do algoritmo.**

A tabela 4.5 mostra a evolução dos resultados com a implementação dos procedimentos.

ETAPA	TESTE	%FAC	%GAP
1	PADRãO	60,5	15,5
	PENALIDADE1	69,1	13,4
	PENALIDADEn = FST	70,2	14,4
	FCT	73,0	15,0
2	MST	70,2	12,7
	MCT	73,0	13,3
3	HST	72,4	11,1
	HCT1	73,5	10,8
	HCT2	73,9	11,6

Tabela 4.5. Variação dos resultados com a incorporação dos procedimentos.

Inicialmente, deve-se lembrar, como dito anteriormente, que um aumento no *gap* em determinadas situações não significa, necessariamente, uma piora nos resultados, pois quando há um aumento no número de soluções factíveis, a inclusão de problemas com um *gap* muito alto pode ter ocorrido.

Pode-se constatar, pelos resultados, que apenas a utilização de penalidade no procedimento de factibilização (PENALIDADE1) faz com que o número de soluções factíveis encontradas pelo algoritmo cresça quase em 10 pontos percentuais. Com a incorporação do procedimento de melhoria tem-se um pequeno ganho na qualidade das soluções. Finalmente, a utilização do procedimento de agregação, completando a heurística, resulta em um aumento no número de soluções factíveis e na qualidade das soluções (menor *gap*).

Vale ressaltar, que após a incorporação da penalidade, todas as demais incorporações à heurística resultaram em melhorias pouco significativas.

- **Os ganhos obtidos com a aplicação de Busca Tabu**

A tabela 4.6 mostra os diferentes resultados sem e com a aplicação de Busca Tabu em cada etapa dos testes.

ETAPA	TESTE	%FAC	%GAP
1	FST	70,2	14,4
	FCT	73,0	15,0
2	MST	70,2	12,7
	MCT	73,0	13,3
3	HST	72,4	11,1
	HCT1	73,5	10,8
	HCT2	73,9	11,6

Tabela 4.6 Variação dos resultados com a incorporação de Busca Tabu.

Na etapa 1 observa-se um aumento no número de soluções factíveis com o uso de Busca Tabu. Na etapa 2, onde tem-se os procedimentos P1, P2 e P3 implementados, nota-se que o uso ou não de Busca Tabu não implica em melhores soluções encontradas, mas apenas um aumento no número de soluções factíveis encontradas. Na etapa 3 tem-se um ganho que pode ser considerado pequeno no número de soluções factíveis encontradas e na qualidade das soluções.

CAPÍTULO 5

CONCLUSÕES

O objetivo deste trabalho foi o estudo do problema de dimensionamento de lotes multiestágio, e o desenvolvimento de um método heurístico para sua resolução. O modelo matemático proposto utilizou o conceito de estoque de escalão e considerou capacidade limitada de produção e tempos de preparação.

No método heurístico foi feita uma adaptação das técnicas de Busca Tabu na tentativa de melhorar o desempenho da heurística.

Os testes computacionais foram realizados em três etapas envolvendo problemas com até 17 itens e 12 períodos. Inicialmente, o teste objetivou um ajuste nos parâmetros do procedimento da Factibilização (P2) da heurística e na Busca Tabu. Na segunda etapa testou-se a inclusão do procedimento de Melhoria (P3) na heurística. Finalmente, na última etapa, a heurística completa, com duas estratégias diferentes para a Busca Tabu, foram testadas. Vale ressaltar que problemas maiores não foram testados por falta de equipamento.

Os resultados mostraram que os problemas considerando estrutura serial de produção foram os mais difíceis de serem resolvidos. Isto deve-se ao maior número de níveis que esta estrutura apresenta, em relação às estruturas gerais e planas. Além disso, observou-se que o tempo médio gasto nos problemas com estrutura serial foi menor que o tempo gasto em problemas com estruturas planas. Neste caso, a razão é um menor número de possibilidades de realização de transferências de produção que esta estrutura apresenta, pois os itens estão mais "amarrados" pela restrição de estoque de escalão. Já na estrutura plana existe um maior número de transferências de produção que podem ser avaliadas durante a execução da heurística.

A incorporação de Busca Tabu não resultou ganhos significativos nos resultados. Pode-se sugerir duas possíveis causas para este fato. A implementação de Busca Tabu usou apenas as técnicas mais simples sugeridas na literatura. Mecanismos mais sofisticados como intesificação/diversificação, análise alvo, etc. não foram utilizados. Outra possível razão envolve a própria dinâmica do algoritmo heurístico e à estrutura típica desse problema, baseada em estoque de escalão. A partir de certas configurações de soluções, as opções para transferências de produções, de acordo com o algoritmo, eram poucas. Com a utilização de Busca Tabu, há uma redução nas possibilidades de transferências ou mesmo nenhuma, o que obrigava constantemente o uso de algum critério para a retirada da condição tabu de certo movimento (aspiração ou secundário) e com isso o conjunto de movimentos realizados eram sempre os mesmos, não havendo uma diversificação de soluções atingidas.

Algumas variações no método heurístico podem ser testadas. Dentre elas pode-se citar:

- A utilização, nos procedimentos P3 e/ou P4, de uma estratégia de trocas de produção, ou seja, transferências de parte da produção de itens tanto para trás como para frente, simultaneamente ou não. Esta idéia foi utilizada em [SCRICH, 1992] para o problema de dimensionamento de lotes monoestágio, com alguma melhoria.

- A utilização de outros atributos. Como um exemplo, no procedimento de agregação, um atributo baseado na variável y_{it}.

- No procedimento P4, a utilização de diferentes mecanismos para a agregação das produções, no sentido de intensificar a busca e não diversificar, como está sendo feito.

Uma efetiva comparação dos resultados deveria ser feita utilizando soluções ótimas obtidas por um método exato, pois os *gaps* encontrados incluem um possível *gap* de dualidade, devido ao uso da Relaxação Lagrangeana.

Uma perspectiva desse trabalho é a incorporação de tempos de produção diferentes de zero. Tal consideração, torna o problema mais complexo, pois, o planejamento da produção precisa agora ser sincronizado, ou seja, apenas quando seus componentes forem comprados ou produzidos, a produção de um item pode ser programada.

APÊNDICE I

Considere as definições

$$E_{it} = I_{it} + \sum_{j \in S(i)} r_{ij} E_{jt} \tag{d1}$$

$$e_{it} = h_{it} + \sum_{j \in P(i)} r_{ji} h_{jt} \tag{d2}$$

$$D_{it} = d_{it} + \sum_{j \in S(i)} r_{ij} D_{jt} \tag{d3}$$

Proposição 1.

No caso de estrutura geral, sejam as definições (d1) e (d2), então para qualquer t ∈ {1,...,T},

$$\sum_{i=1}^{N} h_{it} I_{it} = \sum_{i=1}^{N} e_{it} E_{it}$$

Demonstração.

Para simplificar a notação, seja t fixo e considere

$I_{it} = I_i$, $h_{it} = h_i$, $E_{it} = E_i$, $e_{it} = e_i$, para i ∈ {1,...,N}

Devemos demonstrar que $\sum_{i=1}^{N} h_i I_i = \sum_{i=1}^{N} e_i E_i$.

Pela definição (d1) temos

$$\sum_{i=1}^{N} h_i I_i = \sum_{i=1}^{N} h_i \Big(E_i - \sum_{j \in S(i)} r_{ij} E_j \Big)$$

$$= \sum_{i=1}^{N} h_i E_i - \sum_{i=1}^{N} \sum_{j \in S(i)} h_i r_{ij} E_j \quad .$$

Como S(1) = $\emptyset$,

$$= \sum_{i=1}^{N} h_i E_i - \sum_{i=2}^{N} \sum_{j \in S(i)} h_i r_{ij} E_j$$

$$= \sum_{i=1}^{N} h_i E_i - \sum_{j/P(j) \neq \emptyset}^{N} \sum_{i \in P(j)} r_{ij} h_i E_j \quad .$$

Pela definição (d2) temos

$$= \sum_{i=1}^{N} h_i E_i - \sum_{j/P(j) \neq \emptyset} \Big(h_j - e_j \Big) E_j$$

$$= \sum_{i=1}^{N} h_i E_i - \sum_{j/P(j) \neq \emptyset} h_j E_j + \sum_{j/P(j) \neq \emptyset} e_j E_j$$

$$= \sum_{j/P(j) = \emptyset} h_j E_j + \sum_{j/P(j) \neq \emptyset} e_j E_j \quad .$$

Mas, para P(j) = $\emptyset$ temos ej = hj (pela definição (d2)), logo,

$$= \sum_{j/P(j) = \emptyset} e_j E_j + \sum_{j/P(j) \neq \emptyset} e_j E_j \quad ,$$

isto é, $= \sum_{j=1}^{N} e_j E_j \quad .$

Proposição 2.

No caso de estrutura geral de produção, sejam as definições (d1), (d2), (d3) e o conjunto de restrições

(R1)

$$I_{i,t-1} + x_{it} - I_{it} = d_{it} + \sum_{j \in S(i)} r_{ij} x_{jt} \qquad i = 1,\ldots,N;\ t = 1,\ldots,T \qquad \text{(r1)}$$

$$x_{it} \leq M_{it} y_{it} \qquad i = 1,\ldots,N;\ t = 1,\ldots,T \qquad \text{(r2)}$$

$$x_{it} \geq 0 \qquad i = 1,\ldots,N;\ t = 1,\ldots,T \qquad \text{(r3)}$$

$$I_{it} \geq 0 \qquad i = 1,\ldots,N;\ t = 1,\ldots,T \qquad \text{(r4)}$$

(R2)

$$E_{i,t-1} + x_{it} - E_{it} = D_{it} \qquad i = 1,\ldots,N;\ t = 1,\ldots,T \qquad \text{(r6)}$$

$$\sum_{j \in S(i)} r_{ij} E_{jt} - E_{it} \leq 0 \qquad i = 1,\ldots,N;\ t = 1,\ldots,T \qquad \text{(r7)}$$

$$x_{it} \leq M_{it} y_{it} \qquad i = 1,\ldots,N;\ t = 1,\ldots,T \qquad \text{(r8)}$$

$$x_{it} \geq 0 \qquad i = 1,\ldots,N;\ t = 1,\ldots,T \qquad \text{(r9)}$$

$$E_{it} \geq 0 \qquad i = 1,\ldots,N;\ t = 1,\ldots,T \qquad \text{(r10)}$$

Então, o conjunto de restrições (R1) é equivalente ao conjunto de restrições (R2).

Demonstração.

Primeiramente, suponha que o conjunto de restrições (R1) seja válido. Será demonstrado que (R1) implica em (R2).

Aplicando a definição (d1) na restrição (r4) tem-se que

$$E_{it} - \sum_{j \in S(i)} r_{ij} E_{jt} \geq 0 \qquad \text{para } i = 1,\dots,N\,;\; t = 1,\dots,T \;.$$

Como $S(1) = \emptyset$, $E_{1t} \geq 0$ para $t = 1,\dots,T$. Seja $i \in \{2,\dots,N\}$ e suponha que $E_{jt} \geq 0$ para $j = 1,\dots,i-1$; $t = 1,\dots,T$. Então $E_{jt} \geq 0$ para $j \in S(i)$ e $t = 1,\dots,T$ o que implica que

$$E_{it} \geq \sum_{j \in S(i)} r_{ij} E_{jt} \geq 0 \;.$$

Assim, por indução $E_{it} \geq 0$ para $i = 1,\dots,N$; $t = 1,\dots,T$.

Aplicando a definição (d1) na restrição (r1) tem-se que

$$E_{i,t-1} - \sum_{j \in S(i)} r_{ij} E_{j,t-1} + x_{it} - E_{it} + \sum_{j \in S(i)} r_{ij} E_{j,t} = d_{it} + \sum_{j \in S(i)} r_{ij} x_{j,t}$$

$$E_{i,t-1} + x_{it} - E_{it} = d_{it} + \sum_{j \in S(i)} \left[r_{ij} \left(E_{j,t-1} - E_{jt} + x_{j,t} \right) \right]$$

ou ainda,

$$E_{1,t-1} + x_{1t} - E_{1t} = d_{1t} \qquad i = 1$$

$$E_{i,t-1} + x_{it} - E_{it} = d_{it} + \sum_{j \in S(i)} \left[r_{ij} \left(E_{j,t-1} - E_{jt} + x_{j,t} \right) \right] \qquad i = 2,\dots,N$$

Para $t = 1,...,T$, seja $i \in \{2,...,N\}$ e suponha que

$$I_{j,t-1} + x_{jt} - I_{jt} = d_{jt} + \sum_{k \in S(j)} r_{jk}\, x_{kt} \qquad \textbf{para } j = 1,...,i-1\,;\ t = 1,...,T$$

implica

$$E_{j,t-1} + x_{jt} - E_{jt} = D_{jt} \qquad \text{para } j = 1,...,i-1\,;\ t = 1,...,T$$

Então, para $t \in \{1,...,T\}$, de (s1) e (s2) obtem-se

$$E_{i,t-1} + x_{it} - E_{it} = d_{it} + \sum_{j \in S(i)} [\, r_{ij}\, D_{jt} \,]$$

que pela definição (d3) e por indução tem-se

$$E_{i,t-1} + x_{it} - E_{it} = D_{it} \qquad \text{para } i = 1,...,N\,;\ t = 1,...,T$$

Assim, o conjunto de restrições (R1) (em termos de estoque convencional) implica o conjunto de restrições (R2) (em termos de estoque de escalão).

Agora, suponha que o conjunto de restrições (R2) seja válido. Vamos mostrar que (R2) implica em (R1).

Pela definição (d1), observa-se que a restrição (r7) implica em $I_{it} \geq 0$.

Aplicando a definição (d2) na restrição (r6) tem-se que

$$I_{i,t-1} + \sum_{j \in S(i)} r_{ij}\, E_{j,t-1} + x_{it} - I_{it} - \sum_{j \in S(i)} r_{ij}\, E_{jt} = D_{it}$$

$$I_{i,t-1} + x_{it} - I_{it} = D_{it} + \sum_{j \in S(i)} r_{ij} \left(E_{jt} - E_{j,t-1} \right) \quad .$$

Utilizando a restrição (r6)

$$I_{i,t-1} + x_{it} - I_{it} = D_{it} + \sum_{j \in S(i)} r_{ij} \left(x_{jt} - D_{jt} \right)$$

$$I_{i,t-1} + x_{it} - I_{it} = \sum_{j \in S(i)} r_{ij}\, x_{jt} + D_{it} - \sum_{j \in S(i)} r_{ij}\, D_{jt}$$

Com (d3) tem-se

$$I_{i,t-1} + x_{it} - I_{it} = d_{it} + \sum_{j \in S(i)} r_{ij}\, x_{jt}$$

Assim, o conjunto de restrições (R2) implica no conjunto de restrições (R1).

BIBLIOGRAFIA

AFENTAKIS P. e GAVISH B. (1986), *Optimal Lot-Sizing Algorithms for Complex Product Structures*, Operations Research, vol 34, no. 2, pp 237-249.

AFENTAKIS P., GAVISH B. e KARMAKAR U. (1984), *Computationally Efficient Optimal Solutions to the Lot-Sizing Problem in Multistage Assembly Systems*, Management Science, vol 30, no. 2, pp 222-239.

BAHL H. C., RITZMAN L. P. (1984), *An Integrated Model for Master Scheduling, Lot Sizing and Capacity Requirements Planning,* Journal of the Operational Research Society , vol. 35, no. 5, pp 389-399.

BAHL H. C., RITZMAN L. P. e GUPTA J. N. D. (1987), *Determining Lot-Sizes and Resouce Requirements: A Review*, Operations Research, vol. 35, no. 3, pp 237-249.

BILLINGTON P. J., McCLAIN J. O. e THOMAS L. J. (1983), *Mathematical Programming Approaches to Capacity MRP Systems: Review, Formulation and Problem Reduction*, Management Science, vol. 29, no. 10, pp 1126-1141.

BLACKBURN J. D. e MILLEN R. A. (1984), *Simultaneous Lot-Sizing and Capacity Planning in Multi-Stage Assembly Processes*, European Journal of Operational Research, vol. 16, pp 84-93.

BLACKBURN J. D. e MILLEN R. A. (1982), *Improved Heuristics for Mutistage Requirements Planning Systems*, Management Science, vol. 28, no. 1, pp 44-56.

CAMERINI. P. M., FRATTA L., e MAFFIOLI F. (1975), *On Improving Ralaxation Methods by Modified Gradient Techniques*, Mathematical Programming Study 3, pp 26-34.

CHIU H. N. e LIN T. M. (1988), *An Optimal Lot-Sizing Model for Multi-Stage Series/Assembly Systems*, Computers and Operations Research, vol. 15, no. 5, pp 403-415.

CLARK A. J. e SCARF H. (1960), *Optimal Policies for a Multi-Echelon Inventory Problem*, Management Science, vol. 6, pp 475-490.

CLARK, A. R. (1990), *Problemas Multiestágios de Dimensionamento de Lotes com Tempo Não-Zero de Produção e Capacidade Finita*, tese de doutorado, Faculdade de Engenharia Elétrica, UNICAMP.

CROWSTON W. B. e WAGNER M. H. (1973), *Dynamic Lot-Size Models for Multi-Stage Assembly Systems*, Management Science, vol.20, no. 1, pp 14-21.

CROWSTON W. B. e WAGNER M. H. e WILLIAMS J. F. (1973), *Economic Lot Size Determination in Multi-Stage Assembly Systems*, Management Science, vol.19, no. 5, pp 517-527.

CROWSTON W. B. e WAGNER M. H. e HENSHAW A. (1972), *A Comparison of exact and heuristic routines for Lot Size Determination in Multistage Assembly Systems*, AIIE Transactions, vol.4, no. 4, pp 313-317.

FLORIAN M., LENSTRA J. K. e RINNOY KAN, A. H. G., (1980), *Deterministic Production Planning Algorithms and Complexity*, Management Science, vol. 26, no. 7, pp. 669-679.

GABBAY H. (1979), *Multi-Stage Production Planning*, Management Science, vol. 25, no. 11, pp 1138-1148.

GAREY, M. e JOHNSON, D. (1978), *Computers, Complexity and Intractibility: A Guide to Theory of NP-Completeness*, W. H. Freeman, San Francisco.

GLOVER, F. (1977), *Heuristics for Integer Programming Using Surrogate Constraints*, Decisions Science, vol. 8, pp. 156-166.

GLOVER, F. (1986), *Future Paths for Integer Programming and Links to Artificial Intelligence*, Computers and Operations Research, vol. 13, no. 5, pp. 533-549.

GLOVER, F. (1988), *Tabu Search*, Tecnical Report, Center for Applied Artificial Intelligence, University of Colorado, Boulder, CO.

GLOVER, F. e GREENBERG, H. J. (1989), *New Approaches for Heuristic Search: A Bilateral Linkage with Artificial Intelligence*, European Journal of Operational Research, vol. 39, no. 2, pp. 119-130.

GLOVER, F. (1989a), *Candidate List Strategies and Tabu Search*, CAAI Research Report, University of Colorado, Boulder, CO.

GLOVER, F. (1989b), *Tabu Search - Part I*, ORSA Journal on Computing, vol 1, no. 3, pp. 190-206.

GLOVER, F. (1990a), *Tabu Search: A Tutorial*, Tecnical Report, Center for Applied Artificial Intelligence, University of Colorado, Boulder, CO.

GLOVER, F. (1990b), *Tabu Search - Part II*, ORSA Journal on Computing, vol 2, no. 1, pp. 4-32.

GLOVER, F. (1990c), *Issues and Methods for Applying Target Analysis to Tabu Search*, ORSA Journal on Computing, CAAI Research Report, University of Colorado, Boulder (abril).

GRAVES S. C. (1981), *Multi-Stage Lot-Sizing: An Iterative Procedure, Multilevel Production and Inventory Control Systems: Theory and Practice*, L. B. Schwarz (ed), North-Holland, Amsterdam, cap. 4, pp. 95-110.

JOHNSON L. A. e MONTGOMERY D. C. (1974), *Operations Research in Production Planning, Scheduling and Inventory Control*, John Wiley e Sons, New York.

KONNO H. (1988), *Minimum Concave Cost Production System: A Further Generalization of Multi-Echelon Model*, Mathematical Programming, vol. 2, pp 132-136.

LAMBRECHT M. e VANDER EECKEN J. (1978), *A Facilities in Series Capacity Constrained Dynamic Lot-Size Model*, European Journal of Operational Research, vol. 2, pp 42-49.

LOVE, S. F. (1972), *A Facilities in Series Inventory Model with Nested Schedules*, Management Science, vol. 18, no. 5, pp327-338.

MAES J., McCLAIN J. O. (1991), *Multilevel Capacitated LotSizing Complexity ans LP based Heuristics*, European Journal of Operational Research, vol. 53, pp 131-148.

MULLER, F. M. (1990), Busca Tabu na Solução de Problemas de Programação Zero-um, Tese de mestrado, Faculdade de Engenharia Elétrica, UNICAMP.

PUREZA, M. M. VITÓRIA (1990), Problemas de Roteamento de Veículos via Metaheurística Tabu, Tese de mestrado, Faculdade de Engenharia Elétrica, UNICAMP.

SCRICH C. R. (1992), *Dimensionamento de Lotes de Múltiplos Itens com restrição de Capacidade*, Tese de mestrado, Faculdade de Engenharia Elétrica, UNICAMP.

SOUZA, K. X. S. (1989), *Planejamento da Produção de Múltiplos Itens com Restrição de Capacidade Através da Decomposição Cruzada*, Tese de mestrado, Faculdade de Engenharia Elétrica, UNICAMP.

STEINBERG E. e NAPIER H. A. (1980), *Optimal Multi-Level Lot Sizing for Requirements Planning Systems*, Management Science, vol. 26, no. 12, pp. 1258-1271.

TAILLARD, E. (1990), *Robust Taboo Search for the Quadratic Assigment Problem*, Working paper ORWP 90/10. Département de Mathématiques, École Polytechnique Fédérale de Lausanne, Suiça.

TRIGUEIRO W. W., THOMAS L. J. e McCLAIN J. O. (1989), *Capacited Lot Sizing With Setup Times*, Management Science, vol. 35, no. 3, pp 353-366.

VAN ROY, T. J., (1983), *Cross Decomposition for Mixed Integer Programming*, Mathematical Programming, vol. 25, pp. 46-63.

WAGNER, H. M. e WHITIN, T. M. (1958), *Dynamic Version of the Economic Lot Size Model*, Management Science, vol. 5, no. 1, pp. 89-96.

WILLIAMS J. F. (1981), *Heuristic Techniques for Simultaneous Scheduling*, Management Science, vol. 27, no. 3, pp 336-352.

ZAHORIK A., THOMAS L. J. e TRIGUEIRO W. W. (1984), *Network Programming Models for Production Scheduling in Multistage Multi-Item Capacitates Systems*, Management Science, vol. 30, no. 3, pp 308-325.

ZANGWILL, W. I (1969), *A Backlogging Model and a Multi-Echelon Model of a Dynamic Economic Lot-Size Production System - A Network Approch*, Management Science., vol 15, no. 9, pp 506-527.