UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA ELÉTRICA

DEPARTAMENTO DE ENGENHARIA DE SISTEMAS

DIMENSIONAMENTO DE LOTES DE MÚLTIPLOS ITENS COM RESTRIÇÃO DE CAPACIDADE

CINTIA RIGÃO SCRICH

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

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.

- SETEMBRO 1992 -

AGRADECIMENTOS

A todos que colaboraram na realização deste trabalho e em especial

- a Vinicius A. Armentano e Paulo M. França, pela orientação durante o desenvolvimento desta tese e estimulo à pesquisa;
- aos professores e funcionários do DENSIS, pelo apoio;
- a minha mãe, pelo constante interesse, preocupação e amor;
- aos amigos do DENSIS e DCA pelo companheirismo e pelas horas de alegria;
- a Regina Esther Berretta, Franklina Maria B. de Toledo e Zake Sabbag Neto pelas valiosas discussões sobre este trabalho;
- a Débora P. Ronconi, Ricardo Lüders e Marcelo Nishi pela participação em etapas deste trabalho;
- a Walcir Fontanini e Takaaki Ohishi pelo apoio computacional;
- ao Conselho Nacional de Desenvolvimento Científico e Tecnológico pelo apoio financeiro.

RESUMO

O problema de dimensionamento de lotes de múltiplos itens com restrição de capacidade consiste na determinação das quantidades a serem produzidas em diferentes períodos de tempo na presença de restrição no recurso disponível. O modelo apresentado neste trabalho considera que a produção de um item em um dado período incorre em um custo fixo e um tempo de preparação. Para a resolução deste problema um método heurístico é desenvolvido. Além disso, é feita uma adaptação das técnicas de Busca Tabu a este método. Experimentos computacionais são apresentados e analisados.

ABSTRACT

The multi-item single-level capacitated lot-sizing problem consists in the determination of the amounts being produced in different periods of time in the presence of restriction on the available resources. The model presented in this work considers that the item production implies a setup cost and a setup time. To solve this problem a heuristic method is developed. Besides, an adaptation of Tabu Search techniques for this method is done. Computational experiments are presented and analysed.

ÍNDICE

INTRODUÇÃO

Desde a Revolução Industrial e com a explosão da importância do setor secundário na economia internacional, o setor da manufatura vem alterando o seu foco de atenção e concentrando esforços nos pontos vitais das relações comerciais em prol da sobrevivência no mercado.

Estas transformações podem ser analisadas sob dois aspectos: o do processo e o do produto, e a partir daí, resumir a evolução das metas do setor, destacando-se três fases para cada aspecto considerado:

Processo:	Produto:
- Produção	- Preço
- Tecnologia	- Diversidade
- Metodologia Administrativa/Gerencial	- Qualidade/Produtividade

No início, com a ausência de concorrência, os fabricantes só se preocupavam em produzir. Com o aumento da concorrência, mostrou-se a necessidade da diferenciação de preços com diminuição de custos. Com o desenvolvimento de novas tecnologias pôde-se oferecer variedade ao consumidor. E por fim, voltou-se a atenção ao aprimoramento da administração empresarial visando a garantia da qualidade a um preço competitivo. É exatamente neste ponto que se pode perceber a importância capital de um bom planejamento na empresa e onde se insere o problema estudado neste trabalho, o qual, pode em poucas palavras, ser apresentado como a decisão de o que, quanto e quando produzir ao menor custo possível, isto é, o planejamento tático da produção deve apresentar soluções ao clássico problema de se decidir em quais períodos produzir quais itens (produtos), e quais serão os tamanhos dos lotes nos períodos de produção, tais que atendam à demanda prevista para um certo horizonte, minimizando-se os custos de produção, estoque e preparação da máquina.

Esta classe de problema de planejamento é comum na literatura e em especial na prática. Com o objetivo de se aproximar ainda mais dos casos reais encontrados no dia a dia das indústrias, será considerada a existência de limitação nos recursos, isto é, os meios de produção têm uma disponibilidade limitada.

Apesar da crescente modernização dos sistemas de manufatura e da sofisticação das novas técnicas gerenciais e administrativas, o trabalho também introduz como parâmetro, um tempo de preparação ("setup time"). Isto é justificado porque mesmo o mais moderno conceito de sistema produtivo - o Sistema Flexível de Manufatura - não consegue eliminar por completo o tempo gasto em preparação. Também os sistemas de classificação e codificação de peças, que formam famílias de itens similares e que fundamentam as técnicas de Tecnologia de Grupo, apesar de implicarem em drásticas reduções, ainda apresentam um tempo gasto na troca entre peças de mesma família. E por fim, mesmo a filosofia de gestão da fabricação sem estoques - o Just-in-Time - acusa tempos de preparação razoáveis, pois tem um calcanhar de Aquiles na imposição pelo mercado de maior diversidade no mix de produtos.

Além disso, menores tempos de preparação implicariam em ciclos menores de manufatura, o que acarretaria um aumento no número de preparações, fazendo com que o tempo total gasto em preparações pouco se altere.

Em relação à formulação matemática, a presença de tempos e custos de preparação de máquinas sugere um problema de programação inteira-mista com variáveis binárias representando cada combinação item-período, isto é, a decisão de produzir-não produzir determinado produto em determinado período.
Como problemas típicos requerem um planejamento de vários itens em longos horizontes de tempo, o problema assume grandes proporções, fazendo com que na maioria dos casos seja resolvido através de procedimentos heurísticos que levam a soluções sub-ótimas.

O trabalho a seguir apresenta os passos da elaboração de um método heurístico para a resolução do problema acima descrito. O método proposto parte de uma solução inicial obtida pelo algoritmo de Wagner e Whitin. Sua estrutura, basicamente, se compõe de uma fase de factibilização seguida de um processo de melhoria da solução e outra fase de busca de novas soluções. Também é utilizado o conceito de Busca Tabu, a qual pode ser entendida como uma estratégia que busca direcionar o método no sentido de evitar ciclagem e exploração de soluções que já foram pesquisadas.

Assim, o Capítulo I trata da formulação matemática do problema e da apresentação de heurísticas de resolução do mesmo. No Capítulo II, é apresentado um resumo dos fundamentos da Busca Tabu. O algoritmo desenvolvido e as técnicas utilizadas na implementação da Busca Tabu ao método se encontram detalhados no Capítulo III. No Capítulo IV são feitos testes computacionais e análise comparativa dos resultados obtidos e, no Capítulo V, são apresentadas as conclusões e feitas algumas sugestões para pesquisas futuras.

CAPÍTULO I

O PROBLEMA DE DIMENSIONAMENTO DE LOTES

I.1. FORMULAÇÃO

O problema de dimensionamento de lotes de múltiplos itens consiste em planejar a produção de N itens em um horizonte de tempo finito T. O objetivo é determinar a quantidade que deve ser produzida de cada item (tamanho do lote) em cada período de modo a atender as demandas conhecidas e minimizar os custos fixos e variáveis de produção e os custos de estocagem. A produção dos itens em cada período compartilha um único recurso (hora de máquina, homens-hora) com capacidade limitada.

Este problema pode ser formulado matematicamente como o seguinte problema de programação inteira:

$$(P)\ \text{Min} \sum_{i=1}^{N} \sum_{t=1}^{T} S_{i,t}\, y_{i,t} + c_{i,t}\, x_{i,t} + h_{i,t}\, I_{i,t} \qquad (1)$$

sujeito a

$$I_{i,t} = I_{i,t-1} + x_{i,t} - d_{i,t} \qquad \forall\ i,t \qquad (2)$$

$$\sum_{i=1}^{N} b_{i,t}\, x_{i,t} + s_{i,t}\, y_{i,t} - C_t \leq 0 \qquad \forall\ t \qquad (3)$$

$$x_{i,t} \leq M\, y_{i,t} \qquad \forall\ i,t \qquad (4)$$

$$y_{i,t} \in \{0,1\} \qquad \forall\ i,t$$

$$I_{i,t} \geq 0\ ,\ x_{i,t} \geq 0 \text{ e inteiro} \qquad \forall\ i,t$$

$$I_{i,0} = 0 \qquad \forall\ i$$

onde:

N = número de itens;

T = número de períodos;

$S_{i,t}$ = custo de preparação do item i no período t;

$c_{i,t}$ = custo unitário de produção do item i no período t;

$h_{i,t}$ = custo unitário de estocagem do item i no período t;

$d_{i,t}$ = demanda do item i no período t;

$I_{i,t}$ = estoque do item i no final do período t;

$b_{i,t}$ = tempo de processamento unitário do item i no período t;

$s_{i,t}$ = tempo de preparação para a produção do item i no período t;

C_t = capacidade do recurso no período t em unidades de tempo;

M = um número grande;

$x_{i,t}$ = produção do item i no período t;

$$y_{i,t} = \begin{cases} 1 & \text{se o item i é produzido no período t} \\ 0 & \text{caso contrário} \end{cases}$$

No modelo (P), a função objetivo (1) expressa a minimização dos custos fixos e variáveis de produção e dos custos de estocagem. A restrição (2) representa a equação de balanço de estoque para cada item em cada período e (3) representa a restrição de capacidade em cada período. Note que a produção de um item i num período t implica em um tempo de preparação e um tempo proporcional à quantidade produzida do item. A restrição (4) garante a incidência de um custo fixo e de um tempo de preparação de um item i no período t quando sua produção $x_{i,t}$ é positiva.

I.2. MÉTODOS DE RESOLUÇÃO

A complexidade computacional de um problema está relacionada com o esforço computacional requerido pelos algoritmos conhecidos para sua resolução.

Problemas resolvidos por algoritmos polinomiais, isto é, em que o número de operações elementares necessário para a obtenção da solução ótima é limitado, no pior caso, por uma função polinomial do tamanho do problema, pertencem à classe-P. Já, os problemas para os quais não são conhecidos algoritmos de resolução polinomiais são classificados como NP-completos ou NP-hard [1]. Estes são considerados complexos e de difícil tratamento

Em [1] foi provado que quando N = 1 e $s_{i,t} = 0$, o problema (P) é NP-hard. Quando os tempos de preparação são desprezíveis ($s_{i,t} = 0$) é fácil verificar que o problema (P) é factível se e somente se

$$\sum_{k=1}^{t} \sum_{i=1}^{N} b_{i,k}\, d_{i,k} \le \sum_{k=1}^{t} C_k \qquad \text{para todo } t$$

No entanto, quando $s_{i,t} > 0$, foi mostrado em [2] que o problema de factibilidade de (P) é NP-completo. Estes resultados mostram que é improvável resolver problemas de médio a grande porte através de algoritmos ótimos em tempos computacionais razoáveis, o que motivou o desenvolvimento de métodos heurísticos.

Em [17] é feita uma pesquisa bibliográfica na área de dimensionamento de lotes e são citados diversos métodos desenvolvidos para os problemas das quatro categorias abaixo:

1. único estágio e capacidade ilimitada
2. único estágio e capacidade limitada
3. múltiplos estágios (item final possui vários componentes) e capacidade ilimitada
4. múltiplos estágios e capacidade limitada

A categoria 2 é um caso particular do problema (P) que está sendo considerado aqui, pois os tempos de preparação são desprezados ($s_{i,t} = 0$) e a categoria 4 é um caso mais geral, pois são consideradas estruturas de produção serial, de montagem e geral [19].

Para o caso em que $s_{i,t} = 0$, foram propostos na literatura dois algoritmos exatos [3] e [4] com aplicação limitada a problemas pequenos, e diversos métodos heurísticos. Em [5] é feita uma classificação e comparação de diversas heurísticas. Estas se classificam basicamente em três categorias:

1. Heurísticas período a período
2. Heurísticas de melhoria
3. Heurísticas baseadas em programação matemática

I.2.1. Heurísticas período a período

As heurísticas que fazem parte deste grupo são encontradas em [6], [7] e [8]. Elas têm início no período um e, basicamente, determinam o tamanho dos lotes para o período em questão. Quando todos os lotes estão estabelecidos passam para o próximo período. O processo é então repetido até o final do horizonte de planejamento T.

A determinação do tamanho dos lotes visa satisfazer as demandas do período e também é baseada nos chamados índices de prioridade que são calculados para todos os itens e para todos os períodos futuros. Estes índices podem ser diferentes de uma heurística para outra, mas têm como principal objetivo o de incluir demandas futuras nos lotes atuais, visando obter ganhos em relação aos custos de preparação e estocagem, porém sem ultrapassar o limite da restrição de capacidade.

Além das diferenças em relação aos índices de prioridade, as heurísticas se diferem também na maneira pela qual garantem factibilidade. Em [6], quando um período infactível é encontrado, usa-se um mecanismo de realimentação ("feedback") onde são feitas transferências de demandas deste período para períodos anteriores, a um mínimo custo adicional, até que a infactibilidade seja removida. Em [7] e [8] é feito um cálculo a priori da demanda líquida requerida $(\max \{0, d_{i,t} - I_{i,t-1}\})$ no período t, onde t é o período onde os lotes serão estabelecidos. Para os períodos onde a quantidade demandada excede a capacidade disponível, a produção total ou parcial de alguns itens daquele período deve ser alocada a períodos anteriores a um mínimo custo adicional. Isto é feito para todo t.

Uma vantagem destas heurísticas período a período é que as decisões de tamanho de lotes tomadas nos primeiros períodos do horizonte não são muito influenciadas por incertezas nas demandas em períodos mais distantes, e isto se torna importante quando trabalha-se em ambientes de horizonte rolante ("rolling-schedule"), onde somente as decisões dos primeiros períodos são efetivadas, e quando erros nas previsões são parte da realidade.

I.2.2. Heurísticas de melhoria

As heurísticas aqui consideradas são encontradas em [9] e [10]. A principal característica delas é que têm como ponto de partida uma solução que leva em consideração o horizonte todo e que, em geral, é infactível pois é obtida ignorando-se as restrições de capacidade. Partindo da solução inicial, as heurísticas baseiam-se em transferências de lotes, tentando factibilizar os períodos infactíveis e/ou reduzir custos. Estas transferências podem ser feitas para frente (de t para τ, onde $\tau > t$) ou para trás (de t para τ, onde $\tau < t$) e a quantidade a ser transferida é determinada de acordo com os objetivos em questão.

Existem vários modos de implementar estas rotinas de transferências e a sequência em que estas são aplicadas pode ter uma grande influência tanto no tempo computacional como na qualidade da solução.

Em geral, estas heurísticas de melhoria produzem soluções melhores que as anteriores, mas isto só foi testado para problemas pequenos e não há garantia que para problemas grandes isso também ocorrerá de modo que haja uma justificativa em termos de *qualidade de solução x tempo*. Vale ressaltar aqui que uma característica das heurísticas período a período é o baixo tempo computacional mesmo para problemas grandes.

I.2.3. Heurísticas baseadas em programação matemática

Algumas heurísticas que fazem parte deste grupo estão resumidas em [5], e são baseadas em técnicas de programação matemática tais como relaxação Lagrangeana e "branch and bound". Em [11] novos métodos heurísticos foram propostos e baseiam-se em técnicas de partição de conjuntos ("set partitioning").

Quando $s_{i,t} > 0$, a literatura de métodos de resolução de (P) é bastante escassa. Em [12] um problema semelhante a (P) foi abordado pelo algoritmo ótimo de decomposição cruzada [13], tendo-se também observado que o método é aplicável apenas a problemas pequenos. Em [14] e [16] foram apresentados dois métodos heurísticos, nos quais, contrói-se a relaxação Lagrangeana em relação às restrições (3) e a cada passo de maximização da função dual tenta-se obter uma solução factível através de uma heurística. As diferenças dos dois métodos situam-se nos algoritmos usados para maximizar a função dual e nas heurísticas utilizadas para tentar obter soluções factíveis. As heurísticas, apesar de diferentes, baseiam-se em transferências de lotes tentando factibilizar os períodos infactíveis e/ou reduzir custos.

O objetivo desta pesquisa é propor um novo método heurístico para o caso $s_{i,t} > 0$ e o estudo da aplicação de Busca Tabu para este problema.

CAPÍTULO II

FUNDAMENTOS DA BUSCA TABU

II.1. INTRODUÇÃO

A Busca Tabu é uma estratégia heurística utilizada na resolução de problemas de otimização combinatória. Foi desenvolvida para direcionar métodos que fazem uso de um conjunto de movimentos para transformar uma solução em outra [22,23,24,25].

Esta estratégia é altamente flexível e possui uma grande capacidade de adaptação a estruturas de diferentes problemas e objetivos. Foi introduzida por Glover para problemas não lineares de recobrimento ("covering") [21] e depois aplicada a uma coleção diversa de problemas desde programação de atividades ("scheduling") até telecomunicações, tendo se mostrado eficiente na obtenção de soluções de alta qualidade [26,27].

II.2. ELEMENTOS BÁSICOS

A principal característica da Busca Tabu é a de permitir que se saia de um ótimo local, atingido por algum procedimento heurístico, e se continue o processo de busca de novas e melhores soluções.

Em geral, os procedimentos heurísticos realizam movimentos que representam a transição de uma solução para outra. Para o caso específico de dimensionamento de lotes, uma das heurísticas citadas anteriormente faz uso de transferência de quantidades para obter novas soluções e esta transferência pode ser caracterizada como um movimento.

Considere então, que uma heurística tenha atingido um ótimo local, realizando os melhores movimentos a cada passo. Se a partir daí, se quiser dar continuidade à busca de novas soluções, é necessário realizar um movimento de piora para que se saía deste ótimo local. Entretanto, se depois disso reavaliarmos os movimentos disponíveis, existirá a possibilidade de efetivarmos um movimento que seja exatamente o reverso daquele realizado anteriormente, pois estamos considerando que agora o movimento a ser escolhido corresponde ao melhor dentre os possíveis. Isto caracteriza o fenômeno da ciclagem, que é capaz de comprometer o processo de busca de novas soluções. Aqui se mostra necessário então, um elemento restritivo na busca, que é um dos elementos chave da Busca Tabu: a *restrição tabu*. Esta restrição classifica certos movimentos como *tabu* (ou proibidos) tendo como objetivo prevenir a ciclagem e permitir a exploração de novas regiões.

Estas restrições tabu são, entretanto, contrabalanceadas pela aplicação de *critérios de aspiração*. Estes critérios de aspiração têm como objetivo permitir a anulação do *status tabu* de um movimento se este for suficientemente atrativo, tornando assim a busca mais flexível. Este "esquecimento" estratégico da restrição tabu é também considerado um elemento chave da Busca Tabu.

A restrição tabu e o critério de aspiração da Busca Tabu agem através de *atributos*, que são os elementos descritivos dos movimentos.

Como exemplo, considere a transferência das quantidades já citada para o problema de dimensionamento de lotes. Suponha, então, que se efetivou uma transferência do item i na quantidade q do período t para o período tl. Um atributo que poderia ser adotado para o movimento de transferência seria (i,t,tl), onde i é o item, t é o período origem e tl o período destino da transferência executada. Este atributo preveniria o movimento reverso no seguinte sentido: o item i estaria proibido de ser transferido do período tl para o período t.

Um outro atributo que poderia ser adotado para o movimento descrito acima seria (i,t), onde i é o item e t o período de onde o item foi transferido. Neste caso, o movimento reverso proibido seria o de transferir o item i para o período t. Observe que este atributo é mais restritivo que o anterior pois engloba uma maior quantidade de movimentos.

Quando um movimento é realizado, os atributos correspondentes são armazenados em uma *lista tabu* e, sendo esta lista o agente de restrição da busca, é evidente que sua composição e forma de atualização têm um papel fundamental na geração de soluções.

II.3. LISTAS TABU

A lista tabu define um conjunto de restrições para os movimentos que apresentam propriedades particulares (dependendo dos atributos escolhidos previamente). Ela armazena os últimos m atributos, onde m é um parâmetro. Cada vez que um novo elemento é adicionado ao final da lista, o mais antigo na lista é retirado do começo (estratégia FIFO).

A principal função de uma lista tabu é impedir um movimento de ser revertido.

Em uma heurística, durante o processo de escolha dos melhores movimentos a cada passo, se um movimento candidato contém um (ou mais) atributo(s) da lista, ele é considerado tabu (ou proibido). Uma vez que um atributo é removido da lista os movimentos que o contêm perdem seu status tabu e se tornam livres para serem escolhidos.

Outra forma de remover o status tabu de um movimento é utilizando critérios de aspiração. Um critério de aspiração pode ser aplicado a um atributo, durante o período em que ele permanece na lista tabu. Sua função é proporcionar uma flexibilidade adicional na escolha de bons movimentos, permitindo que o status tabu seja eliminado caso o critério de aspiração seja satisfeito. A definição de um critério de aspiração deve ter como principal objetivo o de garantir a não ocorrência de ciclagem.

Para as restrições tabu que são baseadas em um único atributo de movimento, é preferível que se selecione um atributo cujo status tabu restrinja menos rigidamente a escolha de movimentos disponíveis, deixando assim a busca mais flexível.

Vale ressaltar que uma lista tabu devidamente caracterizada e dimensionada deve ser capaz de impedir ciclagens e isso não significa que o tamanho m da lista deva ser grande já que, desta forma, o espectro de escolhas poderia ser reduzido.

Um aspecto importante em relação ao tratamento da lista tabu é quando todos os movimentos disponíveis são tabus. Neste caso é comum a escolha do movimento de menor piora para ser efetivado. Quando são feitas referências ao melhor ou pior movimento, estas avaliações podem ser baseadas tanto na mudança do valor da função objetivo produzida pelo movimento como em medidas de atração locais. Isto significa que melhores ou piores movimentos não estão necessariamente relacionados com acréscimo ou decréscimo no valor da função objetivo.

II.4. ESTRATÉGIA DE OSCILAÇÃO

Em alguns problemas onde as restrições confinam as soluções a regiões bastante restritas, sugere-se o uso de uma estratégia de oscilação que tem como objetivo induzir a exploração de novas regiões, cruzando certos limites que normalmente não são acessíveis pelo método. Tais limites podem ser representados entre factibilidade e infactibilidade ou podem ser valores de funções ao redor dos quais a busca tende a pairar.

Em algumas aplicações de programação inteira, uma boa solução pode não ser acessível exceto pelo cruzamento de uma região de infactibilidade.

Suponha então, por exemplo, uma função de avaliação $E(x)$, onde x é uma solução candidata, expressa na forma $E(x) = aF(x) + bO(x)$, onde $F(x)$ denota a medida de factibilidade e $O(x)$ a de otimalidade. O objetivo de separar $F(x)$ e $O(x)$ nesta representação é permitir que uma *diferenciação qualitativa* de factibilidade e otimalidade seja implementada pelo ajuste de a e b como um aspecto dinâmico do processo de busca.

Utilizando então a manipulação desses parâmetros a e b e uma lista tabu para impedir ciclagens, pode-se governar a profundidade de penetração para dentro da região factível ou para fora desta, antes de determinar a trajetória de volta à periferia.

II.5. FUNÇÕES DE MEMÓRIA DE MÉDIO E LONGO PRAZO

As restrições tabu e o critério de aspiração são a base do que chamamos de função de memória de curto prazo da Busca Tabu. Esta função tem por objetivo promover uma exploração agressiva na busca de soluções, pois sempre tenta fazer o melhor movimento possível sujeito a escolhas disponíveis.

Além desta, a Busca Tabu emprega as funções de memória de médio e longo prazo, cujos objetivos são promover intensificação local e diversificação global da busca, respectivamente.

A função de memória de médio prazo age armazenando e comparando características de um certo número de melhores soluções encontradas durante um certo período da busca. As características que são comuns para todas ou para a grande maioria destas soluções são tomadas como atributos locais da busca. O método, então, irá procurar novas soluções que exibam estas características, restringindo ou penalizando os movimentos disponíveis durante o período de intensificação da busca. Para o problema de dimensionamento de lotes, um atributo local da busca poderia ser o número de variáveis $y_{i,t} = 1$ de cada item i nas melhores soluções obtidas e uma forma de intensificação seria a penalização de movimentos que fizessem com que esse número diminuísse ou aumentasse muito.

A função de memória de longo prazo, cujo objetivo é diversificar a busca, emprega princípios que são o reverso daqueles da memória de médio prazo. Esta função age estimulando a busca em regiões que contrastam com aquelas examinadas até então. Esta abordagem, entretanto, contrasta também com a de métodos que tentam diversificar a busca gerando aleatoriamente uma série de soluções iniciais, pois estes não possuem mecanismos de aprendizado do passado.

A mémoria de longo prazo é usada então para determinar um critério de avaliação que possa ser usado por uma heurística para produzir novos pontos de partida. Este critério procura penalizar as características que a memória de longo prazo encontra como predominantes em execuções anteriores do processo de busca.

CAPÍTULO III

HEURÍSTICAS PARA A RESOLUÇÃO DO PROBLEMA

III.1. INTRODUÇÃO

Além do desenvolvimento de um método heurístico para o problema de dimensionamento de lotes de múltiplos itens com restrição de capacidade, este trabalho tem também como objetivo estudar os eventuais ganhos que a aplicação da Busca Tabu consegue introduzir em uma heurística para este problema que, por ser combinatório, mostra-se adequado à aplicação desta técnica.

Neste capítulo é apresentado o método heurístico desenvolvido e as estratégias utilizadas na incorporação da técnica de Busca Tabu ao método.

III.2. DESCRIÇÃO DA HEURÍSTICA

O método heurístico desenvolvido é constituído de três fases:

1. Factibilização
2. Melhoria da solução factível
3. Busca de novas soluções factíveis

O ponto de partida para a fase 1 é a solução obtida via algoritmo de Wagner e Whitin [15] aplicado aos N itens e que, em geral, é uma solução infactível, pois para aplicar Wagner-Whitin as restrições de capacidade (3) são relaxadas, isto é, ignoradas.

Caso a solução obtida de Wagner-Whitin seja factível, então ela é a solução *ótima* do problema.

Fase 1 - Factibilização

Esta fase consiste na aplicação de passos regressivo ("backward") e progressivo ("forward") no tempo, com o objetivo de obter factibilidade a um mínimo custo adicional. Estes passos são baseados em transferências de quantidade de itens dos períodos infactíveis para períodos anteriores (regressivo) ou posteriores (progressivo). A seguir são descritos os principais pontos sobre os quais as transferências se baseiam.

Considere, então, um período t qualquer que esteja infactível, isto é, um período onde a restrição (3) esteja violada, e seja Excesso[t] a medida desta violação.

$$\text{Excesso}[t] = \sum_{i=1}^{N} b_{i,t}\, x_{i,t} + \sum_{i=1}^{N} s_{i,t}\, y_{i,t} - C_t$$

- quantidade a ser transferida: a escolha desta quantidade é feita para todos os itens com produção no período t.

No passo regressivo, a quantidade $q_{i,t}$ a ser transferida é escolhida entre $x_{i,t}$, que é toda a produção do item i naquele período, e $\text{Excesso}[t]/b_{i,t}$ que significa a quantidade do item i necessária para eliminar a infactibilidade. Para o caso onde $\text{Excesso}[t]/b_{i,t}$ é maior que $x_{i,t}$, somente $x_{i,t}$ é considerada.

No passo progressivo, para garantir que o estoque seja não negativo, a quantidade de um item i a ser transferida do período t para um período τ, onde $\tau > t$, é limitada por:

$$\min I_{i,k} \qquad t \le k \le \tau - 1$$

e então, a escolha neste passo é feita da seguinte maneira:

$$q_{i,t} = \min \{ x_{i,t} ,\ \text{Excesso}[t]/b_{i,t} ,\ \min I_{i,k} \}$$

- período alvo da transferência(tl): é o período destino da transferência.

No passo regressivo, para cada item estudado, os períodos considerados alvos vão de t-1 até o primeiro período de produção do item antes de t. Considerando que a transferência para trás causa, na maioria das vezes, um acréscimo de custo devido ao custo de estoque, se a quantidade escolhida for mudada para o período da produção anterior do item, então economiza-se o custo de preparação. Entretanto, apesar da alternativa acima parecer a melhor, não se deve descartar a possibilidade de transferências para períodos intermediários, pois o acréscimo ou decréscimo de custo depende muito da relação entre custo de estoque e custo de preparação para cada item i.

No passo progressivo os períodos alvos da transferência vão de t+1 até T, pois neste caso de transferências para frente sempre há diminuição no custo de estoque. Vale ressaltar aqui que, embora pareça uma escolha muito abrangente, em geral as transferências são feitas até no máximo o primeiro período da produção do item depois de t, pois a limitação de estoque já citada, restringe a maioria das possibilidades.

- escolha da melhor transferência

Para os passos regressivo e progressivo, a escolha da transferência que será efetivada é feita segundo o critério da Razão. Esta Razão fornece o custo por unidade de infactibilidade retirada do período t e é calculada da seguinte forma:

Razão = Custo_adicional/Redução

Este cálculo é feito para todas as transferências candidatas e é escolhida a de menor Razão.

O cálculo do Custo_adicional é baseado na variação causada nos custos de produção, estoque e preparação, e também em uma penalização sobre os Excessos em t e no período alvo causados, caso a transferência que está sendo analisada seja efetivada (penalização somente nos casos de Excesso positivo).

Sejam Excesso[t] e Excesso[tl] os excessos resultantes nos períodos origem (t) e alvo (tl) caso a transferência candidata seja executada. Teremos então, para o passo regressivo

$$\begin{aligned} \text{Custo\_adicional} = {} & q_{i,t}\,(c_{i,tl} - c_{i,t}) + q_{i,t}\,(h_{i,t-1} + h_{i,t-2} + \ldots + h_{i,tl}) + \\ & + S_{i,tl}\ (\text{se } x_{i,tl} = 0) - S_{i,t}\ (\text{se } q_{i,t} = x_{i,t}) + \\ & + \text{penalização de } \max\{\,0, \text{Excesso}[t]\,\} \text{ e } \max\{\,0, \text{Excesso}[tl]\,\} \end{aligned}$$

e, para o passo progressivo

$$\begin{aligned} \text{Custo\_adicional} = {} & q_{i,t}\,(c_{i,tl} - c_{i,t}) - q_{i,t}\,(h_{i,t} + h_{i,t+1} + \ldots + h_{i,tl-1}) + \\ & + S_{i,tl}\ (\text{se } x_{i,tl} = 0) - S_{i,t}\ (\text{se } q_{i,t} = x_{i,t}) + \\ & + \text{penalização de } \max\{\,0, \text{Excesso}[t]\,\} \text{ e } \max\{\,0, \text{Excesso}[tl]\,\} \end{aligned}$$

A penalização citada nas fórmulas está detalhada mais à frente.

O cálculo da Redução fornece a redução no valor do Excesso em t se $x_{i,t}$ for reduzido por $q_{i,t}$, sendo que a Redução é no máximo Excesso[t], ou seja:

se (Redução > Excesso[t]) então Redução = Excesso[t]
senão Redução = Redução

Por exemplo: seja t = 3, Excesso[3] = 20 e suponha que o item 1 está sendo considerado para a transferência com $q_{1,3}$ = 10 ($q_{1,3} = x_{1,3}$). Então a redução em t é $b_{1,3}\ q_{1,3} + s_{1,3}$. Se $b_{1,3}$= 1, $s_{1,3}$ = 15 então Redução = 25 e neste caso assume-se Redução = 20. Se $b_{1,3}$ = 1 e $s_{1,3}$ = 5 então Redução = 15 e este valor é mantido.

A seguir são mostrados os pseudo-códigos dos passos regressivo e progressivo.

```
┌ para t = T até 2 faça
│
│    • Excesso[t] = Σ_{i=1}^{N} b_{i,t} x_{i,t} + Σ_{i=1}^{N} s_{i,t} y_{i,t} - C_t
│
│    ┌ enquanto (Excesso[t] > 0) faça
│    │
│    │    ┌ para i = 1 até N faça
│    │    │
│    │    │    ┌ se (x_{i,t} > 0) então
│    │    │    │
│    │    │    │      • Q = Excesso[t]/b_{i,t}
│    │    │    │
│    │    │    │      • τ = primeiro período da produção do item i antes
│    │    │    │            de t ou período 1  caso o  item i  não tenha
│    │    │    │            produção em nenhum período antes de t.
│    │    │    │
│    │    │    │    ┌ para t1 = t-1 até τ faça
│    │    │    │    │
│    │    │    │    │    ┌ para k = 0 até 1 faça
│    │    │    │    │    │   - se (k = 0)
│    │    │    │    │    │           então q_{i,t} = x_{i,t}
│    │    │    │    │    │           senão se (Q < x_{i,t}) então q_{i,t} = Q
│    │    │    │    │    │
│    │    │    │    │    │   • calcular Custo_adicional(i,t,t1,q_{i,t})
│    │    │    │    │    │
│    │    │    │    │    │   • calcular Redução(i,t,t1,q_{i,t})
│    │    │    │    │    │
│    │    │    │    │    │   • calcular RAZÃO(i,t,t1,q_{i,t})
│    │    │    │    │    └ fim do loop k
│    │    │    │    └ fim do loop t1
│    │    │    └ fim de (x_{i,t} > 0)
│    │    └ fim do loop i
│    │
│    │    • (i*,t,t1*,q_{i*,t}) = arg min { RAZÃO(i,t,t1,q_{i,t}) }
│    │
│    │    • transferir a produção q_{i*,t} do período t para o período t1* e
│    │      fazer atualização dos dados.
│    │
│    └ fim de enquanto ( Excesso[t] > 0)
└ fim do loop t
```

$$\text{Excesso}[t] = \sum_{i=1}^{N} b_{i,t}\, x_{i,t} + \sum_{i=1}^{N} s_{i,t}\, y_{i,t} - C_t$$

PASSO PROGRESSIVO

```
┌ para t = 1 até T-1 faça
│
│    • Excesso[t] = Σ_{i=1}^{N} b_{i,t} x_{i,t} + Σ_{i=1}^{N} s_{i,t} y_{i,t} - C_t
│
│  ┌ enquanto (Excesso[t] > 0) faça
│  │
│  │  ┌ para i = 1 até N faça
│  │  │
│  │  │  ┌ se (x_{i,t} > 0) então
│  │  │  │
│  │  │  │    • Q = Excesso[t]/b_{i,t}
│  │  │  │
│  │  │  │  ┌ para tl = t+1 até T faça
│  │  │  │  │
│  │  │  │  │    • menor = min { I_{i,k} / k = t,...,tl-1 }
│  │  │  │  │
│  │  │  │  │    • q_{i,t} = min { x_{i,t} , menor, Q }
│  │  │  │  │
│  │  │  │  │    - se q_{i,t} = 0 então pare o loop tl e vá pa-
│  │  │  │  │                     ra o próximo item do
│  │  │  │  │                     loop i.
│  │  │  │  │
│  │  │  │  │    • calcular Custo_adicional(i,t,tl,q_{i,t})
│  │  │  │  │
│  │  │  │  │    • calcular Redução(i,t,tl,q_{i,t})
│  │  │  │  │
│  │  │  │  │    • calcular RAZÃO(i,t,tl,q_{i,t})
│  │  │  │  │
│  │  │  │  └ fim do loop tl
│  │  │  └ fim de (x_{i,t} > 0)
│  │  └ fim do loop i
│  │
│  │  • se existir (i*,t,tl*,q_{i*,t}) = arg min { RAZÃO(i,t,tl,q_{i,t}) }
│  │   -então transferir a produção q_{i*,t} do período t para o período
│  │    tl* e atualizar os dados.
│  │
│  │   -senão, significa que não é possível factibilizar o período t
│  │    atual e então vá para o próximo período do loop t.
│  │
│  └ fim de enquanto ( Excesso[t] > 0)
└ fim do loop t
```

Comentários:

1. Aplicando o passo regressivo obtém-se uma solução que é factível com relação à restrição (3) para os períodos T,T-1,T-2,...,2. O período 1 nesse processo pode resultar factível ou não. Se for factível, passa-se para a fase 2. Caso contrário, aplica-se o passo progressivo.

2. No passo progressivo a limitação do estoque causa uma maior dificuldade na factibilização dos períodos infactíveis. Se ao final deste passo a solução for factível, passa-se para a fase 2. Caso contrário, volta-se ao passo regressivo.

3. Para ambos os passos, existe a possibilidade de tornar infactíveis períodos que a princípio eram factíveis.

4. A cada transferência executada atualizam-se as variáveis $x_{i^*,t}$, x_{i^*,tl^*} , $y_{i^*,t}$, y_{i^*,tl^*} , os estoques do item i^* de t até tl^*-1, os Excessos de t e tl^* e o custo.

Esta fase termina quando uma solução factível é encontrada ou quando se excede um contador de tempo pré-especificado, que mede o tempo de execução da heurística. Neste último caso, o método heurístico falha e isto significa que ou o método não é capaz de achar uma solução factível ou não existe solução factível.

Em [14], a heurística desenvolvida é semelhante a esta com relação à tentativa de factibilização, utilizando transferências de quantidade de itens para trás e para frente (no tempo). Existem, porém, diferenças quanto ao critério de escolha da melhor transferência assim como nas opções de mudanças de produção. Nela, os custos utilizados nos cálculos são os custos Lagrangeanos, que são obtidos através da relaxação Lagrangeana das restrições de capacidade. Maiores detalhes sobre esta heurística são apresentados no próximo capítulo.

Fase 2 - Melhoria da solução factível

Em geral, a solução factível obtida na fase 1 resulta em alguns períodos com capacidade folgada, e então, nesta fase aplica-se um passo que tenta transferir quantidade dos itens produzidos em períodos anteriores para os períodos folgados com o objetivo de *reduzir custos sem causar infactibilidade*. Este passo pode ser considerado como progressivo no tempo e a seguir estão descritos os seus pontos principais.

Considere um período tl qualquer que esteja com a capacidade folgada, isto é, onde Excesso[tl] < 0, e que é o período destino da transferência.

- <u>período origem da transferência</u>(t)

Dado que já se conhece o período destino da transferência (período tl), os períodos que são considerados origem vão de 1 até tl-1, pois considera-se que as transferências são feitas para frente (tl deve ser maior que a origem t).

- <u>quantidade a ser transferida</u> (q): a escolha desta quantidade é feita para todos os itens com produção em t (origem).

Caso o item que está sendo estudado tenha produção no período tl, a escolha da quantidade recai sobre as seguintes possibilidades: $x_{i,t}$ que é toda a produção do item no período origem, $(-\text{Excesso}[tl]/b_{i,tl})$ que significa a quantidade do item i que pode ser transferida para tl de modo a não causar infactibilidade e $\min\{ I_{i,k}$, onde $t \leq k \leq tl-1 \}$, que é a limitação do estoque. A quantidade escolhida é então a menor dentre as possíveis acima.

Caso o item que está sendo considerado não tenha produção no período tl, a escolha recai sobre as seguintes possibilidades: $x_{i,t}$, $\min \{ I_{i,k}$, onde $t \leq k \leq tl-1 \}$ e $(-(\text{Excesso}[tl] - s_{i,tl})/b_{i,tl})$ que significa o que se pode produzir do item i na folga existente em tl, depois de contabilizada a preparação que deve ser feita resultante da existência de produção do item i no período tl. Da mesma forma, a escolha da quantidade recai sobre a menor dentre as possibilidades. Note que se a folga em tl resultar negativa então este item é descartado, pois isto significa que a sua transferência para tl causaria infactibilidade.

- <u>cálculo do custo</u>

Este cálculo é baseado na variação causada nos custos de produção, estoque e preparação caso a transferência seja executada. Para este passo, são consideradas transferências candidatas apenas aquelas que causam redução no custo e é escolhida a de maior redução.

A seguir este passo de melhoria é mostrado em pseudo-código.

PASSO DE MELHORIA

```
┌ para tl = 2 até T faça
│
│  ┌ enquanto (Excesso[tl] < 0) faça
│  │
│  │   ┌ para i = 1 até N faça
│  │   │
│  │   │     ● Q = - (Excesso[tl]/b_{i,tl})
│  │   │
│  │   │  ┌ para t = 1 até tl-1 faça
│  │   │  │
│  │   │  │     ● q = 0
│  │   │  │
│  │   │  │   ┌ se (x_{i,t} > 0) então
│  │   │  │   │
│  │   │  │   │    - se (x_{i,tl} > 0)
│  │   │  │   │        então ● q = min { Q, x_{i,t} }
│  │   │  │   │        senão ● folga = - ((Excesso[tl] - s_{i,tl})/b_{i,tl})
│  │   │  │   │              se (folga > 0) então q = min {folga, x_{i,t}}
│  │   │  │   │
│  │   │  │   │  ┌ se (q > 0) então
│  │   │  │   │  │
│  │   │  │   │  │     ● q = min { q, min { I_{i,m} / m = t,...,tl-1} }
│  │   │  │   │  │
│  │   │  │   │  │     - se ( q ≠ 0)
│  │   │  │   │  │           então ● calcular Custo_red(i,tl,t,q)
│  │   │  │   │  │
│  │   │  │   │  └ fim de (q > 0)
│  │   │  │   │
│  │   │  │   └ fim de (x_{i,t} > 0)
│  │   │  │
│  │   │  └ fim do loop t
│  │   │
│  │   └ fim do loop i
│  │
│  │   ● se existir (i*, tl, t*, q) = arg min { Custo_red(i,tl,t,q) tal
│  │     que Custo_red(i,,tl,t,q) < 0 }
│  │
│  │    -então transferir a produção q do item i* do período t* para tl e
│  │     atualizar os dados.
│  │
│  │    -senão, vá para o próximo período do loop tl.
│  │
│  └ fim de enquanto (Excesso[tl] < 0)
│
└ fim do loop tl
```

Este passo termina ou com uma solução melhor (factível e com custo menor) que a obtida na fase 1 ou com a mesma, pois não há garantia que se consiga executar transferências de modo a manter a factibilidade e reduzir o custo.

Fase 3 - Busca de novas soluções factíveis

O objetivo desta fase é obter uma nova configuração de produção dos itens nos T períodos, ou seja, uma nova solução factível, partindo da solução obtida no final da fase 2.

Para tal objetivo, são utilizadas duas estratégias diferentes:

1. Agregação de produção
2. Trocas

Na primeira, considera-se a possibilidade de transferir a produção *toda* de um item i para o primeiro período anterior de sua produção. Estas transferências têm início no período T e então pode-se considerar este passo de agregação como regressivo no tempo. As possíveis transferências neste passo são baseadas no seguinte:

- para todos os itens que têm produção no período em questão, considera-se a possibilidade de transferir toda a sua produção para o primeiro período anterior de produção deste mesmo item (que será então o período alvo). Para cada transferência possível, calcula-se o custo adicional, da mesma maneira que foi descrita na fase 1.

Este passo de agregação é mostrado a seguir em pseudo-código.

PASSO DE AGREGAÇÃO

```
┌ para t = T até 2 faça
│
│   ┌ para i = 1 até N faça
│   │
│   │     ┌ se (x_{i,t} > 0) então
│   │     │
│   │     │   ● se existir produção do item i antes de t
│   │     │
│   │     │     -então ● ts = primeiro período de produção de i antes de t.
│   │     │            ● calcular Custo_ad(i,t,ts)
│   │     │
│   │     │     -senão, o item i não tem produção em nenhum período ante-
│   │     │      rior ao período t; vá para o próximo item do loop i.
│   │     │
│   │     └ fim de (x_{i,t} > 0)
│   │
│   └ fim do loop i
│
│   ● se existir (i*, t, ts*) = arg min { Custo_ad (i,t,ts) }
│     -então transferir toda a produção do item i* do período t para o
│      período ts* e fazer as atualizações.
│
└ fim do loop t
```

Comentários:

1. Durante este passo, é feita somente uma agregação por período que é a de menor custo adicional.

2. Esta primeira estratégia, em geral, produz uma solução infactível. A factibilização é então tentada nos mesmos moldes da fase 1, porém, considerando que o passo de agregação é regressivo no tempo, inicia-se a tentativa de factibilização pelo passo progressivo.

Na segunda estratégia, de trocas, tenta-se transferir quantidades de itens tanto para trás como para frente, simultaneamente ou não.

Considere então, períodos t e k quaisquer, onde $t > k$ e itens i e j, onde i é um item com produção no período t e j um item com produção no período k (i deve ser diferente de j).

Nas transferências simultâneas tenta-se transferir uma quantidade do item i do período t para o período k e uma quantidade do item j do período k para o período t. Estas transferências são chamadas de troca dupla. As demais transferências, não simultâneas, são chamadas de troca simples e consistem em transferir quantidade ou do item i de t para k ou do item j de k para t.

Os pontos principais para as trocas dupla ou simples são os seguintes:

- <u>quantidade a ser transferida</u>

A quantidade do item i (q_i) a ser transferida para trás (de t para k) é uma parte de $x_{i,t}$. Esta parte é calculada de acordo com um parâmetro alfa que varia discretamente de 0 até um alfa máximo. Por exemplo, se alfa máximo for igual a 4 então as possibilidades de transferência do item i são: 0, $x_{i,t}/4$, $2x_{i,t}/4$, $3x_{i,t}/4$, $4x_{i,t}/4$. Para o caso em que alfa é igual a 0 não há transferência para trás e isto caracteriza uma troca simples pois a única possibilidade de transferência é a do item j do período k para o período t.

É importante ressaltar aqui que a quantidade do item i que é transferida de t para k deve ser escolhida antes da quantidade do item j (q_j) que é transferida para t, pois os cálculos para esta segunda dependem da escolha anterior. Portanto, quando estão sendo analisadas transferências para frente (de k para t), a quantidade que é escolhida é:

$$q_j = \min \{x_{j,k} \ , \ I_{j,m} \quad k \leq m \leq t-1 \text{ e } Q = (\text{folga}[t]/b_{j,t})\}$$

onde Q significa a quantidade do item j que pode ser produzida em t sem causar infactibilidade. Esta folga[t] é computada da seguinte maneira:

- se o item j já é produzido no período t então folga[t] = - Excesso[t] . Caso contrário, folga[t] = (- (Excesso[t] + $s_{j,t}$)). Se folga[t] for negativa então isto significa que o item j não pode ser transferido para o período t, pois a sua transferência causaria infactibilidade e neste caso atribui-se zero para a quantidade de j a ser transferida.

O Excesso[t] que está sendo utilizado no cálculo da folga[t] deve estar atualizado de acordo com a redução causada pela quantidade do item i que já foi escolhida para ser transferida de t.

Os casos onde a quantidade de j a ser transferida é nula, ou porque causaria infactibilidade ou por causa da limitação do estoque, também caracterizam uma troca simples.

• <u>períodos da transferência</u>

Para esta estratégia de trocas não são considerados períodos origem e destino das transferências, pois tanto o período t quanto o período k são origem e destino ao mesmo tempo. Considera-se este passo de trocas como regressivo no tempo e isto implica que t vai de T até 2 e k de t-1 até 1.

• <u>cálculo do custo</u>

Este cálculo é baseado na variação de custo causada pelos custos de produção, estoque e preparação caso a troca (simples ou dupla) seja executada.

O pseudo código desta estratégia de trocas é mostrado a seguir.

```
┌ para t = T até 2 faça
│ ┌ para i = 1 até N faça
│ │ ┌ se (x_{i,t} > 0) então
│ │ │ ┌ para k = t-1 até 1 faça
│ │ │ │ ┌ para j = 1 até N faça
│ │ │ │ │ ┌ se (j ≠ i) e (x_{j,k} > 0) então
│ │ │ │ │ │ ┌ para α = 0 até αmax faça
│ │ │ │ │ │ │   • calcular quantidade do item i (q_i) de acordo com α
│ │ │ │ │ │ │   • atualizar Excesso[t] devido à redução causada por q_i
│ │ │ │ │ │ │   • atualizar Excesso[k] devido ao acréscimo causado
│ │ │ │ │ │ │     por q_i
│ │ │ │ │ │ │   • cálculo da quantidade de j a ser transferida
│ │ │ │ │ │ │
│ │ │ │ │ │ │       - calcular menor = min { I_{j,m} , k ≤ m ≤ t-1 }
│ │ │ │ │ │ │
│ │ │ │ │ │ │       - calcular Q = (folga[t]/b_{j,t})
│ │ │ │ │ │ │
│ │ │ │ │ │ │       - se (Q ≤ 0) ou (menor = 0)
│ │ │ │ │ │ │                 então • q_j = 0
│ │ │ │ │ │ │                 senão • q_j = min {menor, Q, x_{j,k}}
│ │ │ │ │ │ │                       • atualizar Excesso[k] devido à
│ │ │ │ │ │ │                         redução causada por q_j
│ │ │ │ │ │ │                       • atualizar Excesso [t]  devido
│ │ │ │ │ │ │                         ao acréscimo causado por q_j
│ │ │ │ │ │ │   • folga[t] = -Excesso[t] (com todas as atualizações)
│ │ │ │ │ │ │   • folga[k] = -Excesso[k] (com todas as atualizações)
│ │ │ │ │ │ │
│ │ │ │ │ │ │   - se (q_j ≠ 0) ou ( α ≠ 0) então
│ │ │ │ │ │ │         se (folga[t] ≥ 0) e (folga[k] ≥ 0) então
│ │ │ │ │ │ │                  • calcular Custo(i,t,j,k)
│ │ │ │ │ │ └ fim do loop α
│ │ │ │ │ └ fim de (j ≠ i) e (x_{j,k} > 0)
│ │ │ │ └ fim do loop j
│ │ │ └ fim do loop k
│ │ └ fim de (x_{i,t} > 0)
│ └ fim do loop i
│ • se existir (i*,t,j*,k*) = arg min {Custo(i,t,j,k) / Custo(i,t,j,k)< 0}
│   -então execute a troca relacionada a i*,t,k*,j* e atualize os dados.
│   -senão, vá para o próximo período do loop t.
└ fim do loop t
```

Comentários:

1. Para cada período t, executa-se no máximo uma troca, que é a de maior redução de custo. Caso já se tenha terminado de percorrer todos os loops e nenhuma troca tenha sido executada, então escolhe-se para ser executada somente uma troca que causar acréscimo no custo. Ela será a de menor acréscimo e será denominada de troca de menor piora.

2. Ao fim deste passo obtém-se uma solução factível, e então passa-se para a fase 2.

Esta estratégia de trocas foi inicialmente desenvolvida de modo que não se permitisse infactibilidade nos períodos. Uma segunda versão desenvolvida permite infactibilidade no período k. Essa infactibilidade permitida pode ser considerada como um Δ_{infac}, que é uma porcentagem da capacidade do período. Por exemplo, considere o período 3 como sendo o período k de uma tentativa de troca. Se sua capacidade é igual a 100 e é permitido Δ_{infac} = 15%, então o período 3 poderá ter um Excesso positivo de 15 unidades de tempo. É importante ressaltar que Δ_{infac} é o mesmo para todos os períodos naquele passo, podendo ser alterado somente quando o passo for iniciado novamente.

Pode-se então considerar que os períodos têm uma "nova" capacidade e agora as trocas devem respeitar essas "novas" capacidades. Uma conseqüência desse novo enfoque é em relação às quantidades de i e j que podem ser transferidas, pois essas mudanças de capacidade acarretam uma maior flexibilidade nas escolhas destas quantidades. Outro ponto que sofre modificações é o cálculo do custo. Neste novo cálculo são penalizados os Excessos positivos tanto do período t quanto do período k que estão sendo considerados para a troca.

A possibilidade de ter Excesso em t positivo, vem do fato do passo de trocas ser regressivo no tempo, ou seja, mesmo permitindo somente infactibilidade no período k, este período k passa a ser t em algum momento.

Exemplo: considere a transferência de um item qualquer do período 12 (t) para o período 9 (k). Sendo permitida uma infactibilidade em k, suponha que, ao se executar a transferência, Excesso[9] resulte positivo (período 9 está infactível). De acordo com o "loop" de t, nota-se que t assumirá o valor 9, e isto, consequentemente, resulta em uma infactibilidade também no período t (pois agora t = 9).

Portanto, para estes casos onde o período t é infactível, as transferências que podem ser feitas devem garantir que o novo Excesso[t] não ultrapasse este valor, isto é, não são permitidas trocas (simples ou duplas) que acarretem no período t uma infactibilidade maior do que a já existente. Isso implica que a quantidade de j a ser transferida para t no caso de Excesso[t] > 0 sofre alterações em relação ao cálculo da quantidade Q, como descrito abaixo.

Considere Excesso[t](antigo) como sendo o excesso em t antes de calcular a redução causada pela quantidade de i que será transferida de t e Excesso[t](atual) já computando essa redução. Desta forma tem-se para Q o seguinte:

- se item j tem produção em t

$$Q = (\text{Excesso}[t](\text{antigo}) - \text{Excesso}[t](\text{atual}))/b_{j,t}$$

- se item j não tem produção em t

$$Q = (\text{Excesso}[t](\text{antigo}) - \text{Excesso}[t](\text{atual}) + s_{j,t})/b_{j,t}$$

A escolha de q_j continua sendo o mínimo entre $x_{j,t}$, $I_{j,m}$ $k \leq m \leq t-1$ e Q. Se $Q < 0$, isto significa que a transferência do item j para o período t causaria uma infactibilidade maior que a antiga, e portanto atribui-se zero ao valor de q_j.

Para os casos onde Excesso[t](antigo) < 0 a escolha de q_j é feita da mesma maneira descrita no passo de trocas sem infactibilidade.

A seguir está mostrado este novo passo de troca e as diferenças em relação ao anterior estão destacadas em itálico.

PASSO DE TROCAS COM INFACTIBILIDADE

```
para t = T até 2 faça
  para i = 1 até N faça
    se (x_{i,t} > 0) então
      para k = t-1 até 1 faça
        • calcular infac = (Capacidade do período k) × Δinfac
        para j = 1 até N faça
          se (j ≠ i) e (x_{j,k} > 0) então
            para α = 0 até αmax faça
              • calcular quantidade do item i (q_i)
              • atualizar Excesso[t] devido à redução causada por q_i
              • atualizar Excesso[k] devido ao acréscimo causado
                por q_i
              • cálculo da quantidade de j a ser transferida
                - calcular menor = min { I_{j,m} , k ≤ m ≤ t-1 }
                - calcular Q ,de acordo com valor de
                                              Excesso[t](antigo)
                - se (Q ≤ 0) ou (menor = 0)
                      então • q_j = 0
                      senão • q_j = min {x_{j,t}, menor, Q}
                            • atualizar Excesso[k] devido à re-
                              dução causada por q_j
                            • atualizar Excesso[t] devido ao
                              acréscimo causado por q_j

              • folga[t] = Excesso[t] (com todas as atualizações)
              • folga[k] = Excesso[k] (com todas as atualizações)
              - se (q_j ≠ 0) ou ( α ≠ 0) então
                  se (Excesso[t](antigo) > 0) então
                     se (folga[t] ≤ Excesso[t](antigo)) e
                                (folga[k] ≤ infac) então tr = 1
                  se (Excesso[t](antigo) < 0) então
                     se (folga[t] ≤ 0) e (folga[k] ≤ infac) então
                                     tr = 1
              - se (tr = 1) então
                            • calcular Custo(i,t,j,k)
            fim do loop α
          fim de (j ≠ i) e (x_{j,k} > 0)
        fim do loop j
      fim do loop k
```

```
    └ fim de (x_{i,t} > 0)
  └ fim do loop i

  • se existir (i*,t,j*,k*) = arg min {Custo(i,t,j,k) / Custo(i,t,j,k)< 0}
    -então execute a troca relacionada a i*, t, k*, j* e atualize os dados.
    -senão, vá para o próximo período do loop t.

└ fim do loop t
```

Comentários:

1. A escolha da troca a ser executada segue os mesmos critérios do passo de troca sem infactibilidade, isto é, escolhe-se a cada período a de maior redução de custo e no caso de não existir nenhuma troca para nenhum período, então escolhe-se uma que será a de menor piora. É importante notar que para este passo, no cálculo do custo também está sendo contabilizada a penalização sobre os períodos t e k da troca. Esta penalização tem coeficiente 1 para ambos os Excessos positivos, isto é,

$$\text{Penalização} = \max \{ 0, \text{ Excesso}[t] \} + \max \{ 0, \text{Excesso}[k] \}$$

2. A aplicação deste passo em geral, resulta em uma solução infactível e a factibilização é então tentada nos mesmos moldes da fase 1, porém começando com a aplicação do passo progressivo.

Tendo sido apresentadas as três fases do método heurístico desenvolvido, será mostrado agora um esquema de seu funcionamento:

PASSO 0: Inicialização e entrada de dados.

PASSO 1: Aplicação do algoritmo de Wagner e Whitin aos N itens.

PASSO 2: Cálculo das restrições de capacidade. Se a solução resultante de Wagner-Whitin for factível então PARE. Encontrou a solução ótima do problema.
Caso contrário, vá ao passo 3.

PASSO 3: FASE 1 - Factibilização. Se o critério de parada foi satisfeito, FIM. O método falhou.
Caso contrário, se encontrou solução factível então vá ao passo 4.

PASSO 4: FASE 2 - Melhoria. Se o critério de parada foi satisfeito, FIM.
Caso contrário, vá ao passo 5.

PASSO 5: FASE 3 - Agregação ou Trocas. Se o critério de parada foi satisfeito, FIM.

Caso contrário, se a solução resultante for factível então vá ao passo 4. Senão, vá ao passo 6.

PASSO 6: FASE 1 - Factibilização, começando com o passo progressivo. Se o critério de parada foi satisfeito, FIM. Caso contrário, se a solução resultante for factível, vá ao passo 4.

O critério de parada deste algoritmo é o tempo computacional. Existe um contador de tempo em vários pontos da heurística e quando este contador atinge um nível pré especificado a heurística pára.

Devido às diferentes estratégias utilizadas na fase 3 (passo 5), optou-se por separar o método heurístico em três procedimentos que são chamados de HEUR1, HEUR2, HEUR3. HEUR1 é o procedimento que utiliza o passo de agregação na fase 3, HEUR2 utiliza o passo de trocas factíveis e HEUR3 utiliza o passo de trocas permitindo infactibilidade.

Estes três procedimentos que diferem pela fase 3, também apresentam diferenças em relação à fase 1 no que diz respeito à penalização usada no cálculo do acréscimo de custo de uma transferência candidata e à Razão utilizada.

Esta penalização, em linhas gerais, tem como objetivo evitar transferências para períodos apertados, isto é, períodos que em geral são infactíveis e para os quais a factibilização se mostra mais difícil. Para o cálculo desta penalização utiliza-se um fator de penalidade para cada período t. A inicialização deste fator é a mesma para os três procedimentos e é Fator_pen[t] = 1 para todos os períodos no início do método heurístico.

Seja então Penal a variável que computa o valor dessa penalização para cada transferência candidata e considere t e t1 como os períodos origem e alvo respectivamente. Se a transferência que está sendo considerada mantém Excesso[t] > 0 então Penal = Excesso[t] × Fator_pen[t]. Caso esta transferência também acarrete Excesso[t1] > 0 então Penal = Penal + Excesso[t1] × Fator_pen[t1]. Esta variável Penal é então adicionada ao custo da transferência que está sendo analisada.

A diferença entre os três procedimentos está na maneira como estes fatores de penalidade são atualizados.

ATUALIZAÇÃO PARA HEUR1: ao final de cada passo regressivo e progressivo , para os períodos que estiverem infactíveis, isto é, que estejam com Excesso > 0, soma-se 1 a seus fatores de penalidade. Para os restantes factíveis não há alteração no valor do fator. Portanto, para computar a penalização, tanto para os passos de factibilização como para o passo de agregação, utilizam-se os valores atualizados dos fatores. Aqui a Razão continua sendo Custo_adicional/Redução do Excesso[t].

ATUALIZAÇÃO PARA HEUR2 E HEUR3: para ambos a estratégia é a mesma e consiste em somar 1 ao fator de penalidade de *todos* os períodos a cada *composição de passos* - regressivo e progressivo, e não só àqueles onde Excesso > 0. Provavelmente, por utilizar este tipo de penalização, notou-se que a Razão com denominador igual a 1 foi melhor, então para estes procedimentos Razão = Custo_adicional.

O uso destes fatores variantes na penalização, tanto para HEUR1 quanto para HEUR2 e HEUR3, mostrou-se eficaz no sentido de encontrar soluções factíveis.

No método heurístico apresentado, esses fatores de penalidade são reinicializados ao final do passo 3, que é onde se encontra a primeira solução factível. A partir daí, esses fatores são atualizados e usados de acordo com o que já foi discutido acima.

O ajuste desses parâmetros (Razão e penalização) foi feito através de testes computacionais, onde várias possibilidades foram consideradas e as melhores foram implementadas.

III.3. IMPLEMENTAÇÃO DA BUSCA TABU NO MÉTODO

As estratégias utilizadas para a aplicação da técnica da Busca Tabu ao método heurístico são basicamente as que se relacionam com a função de memória de curto prazo e que se traduzem por restrições tabu e critério de aspiração. Para que estas estratégias possam ser implementadas é necessário caracterizar os movimentos que levam de uma solução à outra e os atributos que descrevem estes movimentos.

O movimento que é aqui considerado é a transferência de quantidade de itens, pois é sobre estas transferências que o método heurístico todo se baseia. Então transferir uma quantidade de um item de um período para outro é um movimento.

A tarefa de caracterizar um atributo de movimento é delicada, pois a escolha deste é muito importante no processo de busca e consequentemente na qualidade das soluções obtidas. Devido ao fato dos três procedimentos - HEUR1, HEUR2, HEUR3 - terem suas particularidades, as escolhas destes atributos foram diferenciadas e o detalhamento é feito separadamente.

Antes, porém, de caracterizar os atributos de cada procedimento, é mostrado como a Busca Tabu foi implantada ao método no que diz respeito às fases 1, 2 e 3.

Considere então que existe uma lista tabu que armazena os m últimos atributos dos movimentos executados. Na Inicialização (passo 0) esta lista é considerada vazia.

Para a fase 1 tem-se o seguinte: durante a execução do primeiro passo regressivo e do primeiro passo progressivo (se este for necessário), constrói-se a lista tabu que contém os atributos dos movimentos que foram executados. Se ao final destes passos não for obtida uma solução factível, então a partir do segundo passo regressivo, a cada movimento candidato, a lista é consultada e só são permitidos os movimentos que não são tabus (proibidos) ou que mesmo sendo tabus satisfaçam o critério de aspiração (que é detalhado mais a frente). O movimento selecionado dentre os permitidos segue as mesmas considerações feitas quando da explicação destes passos. Portanto, esta lista tabu deve ser atualizada a cada movimento executado. Para o caso onde todos os movimentos candidatos são tabus e nenhum satisfaz o critério de aspiração, é feita a escolha do "menos-pior". Este movimento "menos-pior" será aquele que tiver a menor Razão.

O procedimento de proibições descrito visa evitar a repetição de movimentos em passos regressivo e progressivo anteriores, que verificou-se ser muito comum nos problemas.

Para os casos em que é necessária a aplicação destes passos da fase 1 para factibilizar uma solução que se tornou infactível pela aplicação da fase 3 (passos de agregação e troca infactível), então a consulta à lista é feita já a partir do primeiro passo, no caso, o passo progressivo pois após a fase 3 inicia-se por ele a tentativa de factibilização.

Na fase 2 não são feitas restrições quanto aos movimentos que possam ser executados, isto é, a lista tabu não é consultada mas apenas atualizada de acordo com os movimentos efetivados, pois dado que o passo aqui considerado é de melhoria do valor da função objetivo então é melhor deixar a busca menos restritiva afim de tentar obter o maior ganho possível.

Para a fase 3, a lista tabu também é consultada a cada movimento candidato e atualizada de acordo com os movimentos executados. Nesta fase, existem particularidades para as proibições impostas, de acordo com o passo que é utilizado.

Para o passo de agregação, os movimentos permitidos são aqueles que não são tabus ou que mesmo sendo, satisfazem o critério de aspiração. O escolhido para ser efetivado segue as considerações apresentadas quando da explicação deste passo. Para o caso onde todos são tabus, nenhum movimento é executado.

Para o passo de trocas, tanto sem infactibilidade quanto com infactibilidade, os movimentos permitidos são só aqueles que não são tabus, tanto na escolha do melhor (que é o que causa decréscimo no custo) como na escolha do de menor piora. Nestes passos de trocas não há nível de aspiração.

III.3.1. Especificação do atributo para HEUR1

Neste procedimento, durante a fase de busca de novas soluções factíveis (fase 3 - passo de agregação), as transferências consistem na mudança de *toda* a produção de um item. Com base nesta característica, estabeleceu-se como atributo de movimento a variável $y_{i,t}$. Isto significa que cada vez que executa-se um movimento, os atributos que são guardados na lista são as variáveis $y_{i,t}$ que mudaram de valor. Por exemplo: seja a produção do item 2 no período 4 igual a 20 e considere o movimento que será feito como sendo o de transferir todo este item do período 4 para o período 3, sendo que neste último não há produção deste item. Então, quando este movimento de transferência é executado, vão para a lista as variáveis $y_{2,4} = 0$ e $y_{2,3} = 1$, pois $y_{2,4}$ era 1 e passou para 0 e $y_{2,3}$ era 0 e passou para 1.

A lista tabu age então da seguinte maneira:

- se um movimento candidato causar mudança em variável(is) $y_{i,t}$ que esteja(m) com valor(es) complementar(es) na lista, então ele é considerado tabu.

Por ex: considere que as variáveis $y_{1,2} = 1$ e $y_{2,3} = 0$ estejam na lista e isto significa que é proibido deixar de produzir o item 1 no período 2 e também é proibido produzir o item 2 no período 3. Então, se um movimento candidato levar a uma solução que tenha $y_{1,2} = 0$ e/ou $y_{2,3} = 1$, ele é considerado proibido. Entretanto, essa proibição pode ser superada caso o movimento candidato satisfaça o critério de aspiração. Tal critério permite que o status tabu de um movimento seja anulado e com isso este pode ser efetivado.

O critério de aspiração escolhido neste procedimento é relativo ao Custo_adicional e é utilizado nas fases 1 e 3 (para os passos de factibilização e para o passo de agregação). Considere então a seguinte propriedade neste Custo_adicional:

$$\text{Custo\_adicional} - \text{penalização} \leq 0$$

Então caso um movimento tabu satisfaça a propriedade acima ele passa a ser permitido e se ele passar pelo critério de escolha do melhor movimento pode ser efetivado.

III.3.2. Especificação do atributo para HEUR2 e HEUR3

O atributo de movimento escolhido para HEUR2 e HEUR3 foi o mesmo dado que ambos os passos são baseados nas mesmas possibilidades de transferências.

O primeiro atributo testado foi (i,t,t1) e isto significa que, se ocorreu uma transferência de uma quantidade qualquer de um item i do período t para o período t1, então a transferência deste mesmo item do período t1 para o período t em qualquer quantidade é considerada tabu. Entretanto, utilizando este atributo verificou-se a ocorrência de ciclagem para a maioria dos problemas e isto aconteceu porque a utilização deste atributo não fornece mecanismos que permitam distinguir situações em que a reversão de um movimento ocorre através da combinação de outros. Considere por exemplo a transferência de uma quantidade do item 2 do período 3 para o período 4. O atributo que estaria na lista devido à este movimento seria (2,3,4). Suponha agora a transferência da mesma quantidade do item 2 do período 4 para o período 2. Com isso estariam na lista os atributos (2,3,4) e (2,4,2). Tais atributos entretanto não impedem que o movimento de transferir o item 2, nas mesmas quantidades anteriores, do período 2 para o período 3 seja executado. Desta forma chega-se exatamente a mesma configuração anterior e levando em consideração que a lista só é consultada para a reversão de movimentos então o primeiro movimento pode se repetir e assim por diante.

Outro atributo aqui testado foi (i,t,q) que significa que se o item i foi transferido do período t para qualquer outro período na quantidade q, então o reverso, que seria transferir o item i na quantidade q para o período t estaria proibido. Porém este atributo também não foi capaz de impedir as ciclagens pois continuaram acontecendo transferências em quantidades menores que ao final resultavam na mesma solução.

Devido a incapacidade de se impedir ciclagens, optou-se pela utilização de um atributo mais restritivo e então escolheu-se (i,t). Este atributo significa que se ocorreu a transferência do item i do período t em qualquer quantidade para qualquer período, então a transferência do item i para o período t está proibida (também em qualquer quantidade e vinda de qualquer período).

Desta forma a lista tabu armazena os atributos (i,t) e se os movimentos candidatos possuem atributos que estejam na lista, eles são tabus.

Por exemplo: suponha que o item 3 foi transferido do período 5 para o período 7. O atributo que vai para a lista é (3,5), e portanto qualquer movimento que acarretar uma transferência do item 3 para o período 5 está proibido. Para o caso de uma troca dupla, sempre vão dois atributos para a lista e basta que um deles acarrete uma reversão para que a troca seja tida como tabu. Por exemplo, suponha que estejam na lista os atributos (3,6), (5,7) e (2,4) e que a troca que está sendo estudada é transferir o item 2 do período 3 para o período 4 e transferir o item 4 do período 4 para o período 2. Note que somente a primeira transferência possui um atributo que está na lista mas isso acarreta uma proibição desta troca dupla, não impedindo no entanto a troca simples que é executar somente a segunda transferência.

Para estes procedimentos que utilizam trocas, não há critério de aspiração na fase 3. Isto significa que se todos os movimentos que acarretarem redução no custo são tabus, então escolhe-se um que cause aumento no custo, sendo que este também deve ser não tabu.

Tendo sido detalhadas as técnicas usadas para a implantação da estratégia da Busca Tabu nos três procedimentos, será dada agora a versão geral do algoritmo com Tabu.

PASSO 0: **Inicialização e entrada de dados.**

Na inicialização a lista tabu é considerada vazia.

PASSO 1: Aplicação do algoritmo de Wagner e Whitin aos N itens.

PASSO 2: Cálculo das restrições de capacidade. Se a solução resultante de Wagner-Whitin for factível então PARE. Encontrou a solução ótima do problema.

Caso contrário, vá ao passo 3.

PASSO 3: FASE 1 - Factibilização.

Para os dois primeiros passos (regressivo e progressivo) a lista tabu só é utilizada para armazenar os atributos. A partir daí, ela é consultada a cada movimento candidato e então inicia-se o processo de proibições que leva em conta os elementos da lista e o critério de aspiração. A cada movimento executado a lista é atualizada.

Se o critério de parada foi satisfeito, FIM. O método falhou.

Caso contrário, se encontrou solução factível então inicialize novamente a lista tabu como vazia, faça it = 0 e vá ao passo 4.

PASSO 4: FASE 2 - Melhoria.

Se (it = 0), então a lista tabu não é nem atualizada nem consultada.

Caso contrário, a lista só é atualizada.

Se o critério de parada foi satisfeito, FIM.

Caso contrário, vá ao passo 5.

PASSO 5: FASE 3 - Agregação ou Trocas.

Neste passo a lista tabu é consultada a cada movimento candidato e é atualizada para os movimentos efetivados. Faça it = 1.

Se o critério de parada foi satisfeito, FIM.

Caso contrário, se a solução resultante for factível então vá ao passo 4. Senão, vá ao passo 6.

PASSO 6: FASE 1 - Factibilização, começando com o passo progressivo.

Aqui a lista tabu é consultada a cada movimento candidato e atualizada de acordo com os movimentos executados.

Se o critério de parada foi satisfeito, FIM.

Caso contrário, se a solução resultante for factível, vá ao passo 4.

Observação: a variável it foi colocada para destacar que na fase 2 não é feita atualização da lista tabu depois do passo 3, que é onde se encontra a primeira solução factível e onde a lista é reinicializada como vazia.

Ao final do algoritmo tem-se a melhor solução para cada problema. Os problemas que foram testados, assim como os outros testes feitos com a finalidade de definir valores apropriados para parâmetros como tamanho de lista, penalidades, alfa, etc. estão especificados no próximo capítulo.

C A P Í T U L O IV

RESULTADOS COMPUTACIONAIS

IV.1. INTRODUÇÃO

Neste capítulo são apresentados os resultados computacionais obtidos ao longo do desenvolvimento do trabalho.

O método heurístico apresentado foi implementado em linguagem C e os testes realizados numa estação de trabalho SUN modelo SPARCstation 1+.

IV.2. CARACTERÍSTICAS DOS PROBLEMAS TRATADOS

Dos problemas que foram usados para testar os algoritmos propostos na tese, uma parte é encontrada na literatura (grupo 1) e a outra foi gerada aleatoriamente (grupos 2 a 7), somando ao todo 159 problemas.

Cada problema pode ser definido pelos seguintes dados de entrada:

1. número de itens (N)
2. número de períodos (T)
3. demandas dos itens nos períodos
4. custo unitário de estoque dos itens nos períodos
5. custo de preparação dos itens nos períodos
6. custo unitário de produção dos itens nos períodos
7. tempo de preparação dos itens nos períodos
8. tempo de processamento unitário dos itens nos períodos
9. capacidade dos períodos

Os problemas do grupo 1 foram obtidos de [14] onde consideram-se constantes a capacidade e, para cada item i, os custos de estoque e preparação e os tempos de processamento e preparação ao longo de todo o horizonte de tempo (T). Os custos de produção foram considerados nulos, pois podem ser vistos como um custo fixo a ser acrescentado à função objetivo, isto porque eles também foram considerados constantes para cada item i ao longo do tempo. A afirmação acima pode ser explicada da seguinte maneira: seja c_i o custo para cada item i (igual para todos os períodos) e $x_{i,t}$ a produção do item i em t.

$$\text{PROD} = \sum_{i=1}^{N} c_i \sum_{i=1}^{T} x_{i,t}$$

Pode-se mostrar que $I_{i,T} = 0$ [28]. Isto implica que $\sum_{t=1}^{T} x_{i,t} = \sum_{t=1}^{T} d_{i,t}$ (isto é, a somatória das produções do item i ao longo dos períodos é igual a somatória das demandas destes itens). Então:

$$\text{PROD} = \sum_{i=1}^{N} c_i \sum_{t=1}^{T} d_{i,t} \qquad \text{(i)}$$

e de (i) tem-se que PROD é uma constante, dado que já se sabem todas as demandas.

A Tabela IV.1. resume as principais características dos 39 problemas selecionados para compor o grupo 1.

Problemas	N	T	$d_{i,t}$ (*)	$h_{i,t}$ (*)	$S_{i,t}$ (*)	$s_{i,t}$ (*)	$b_{i,t}$	C_t
1-2-3	6	15	0-125	1-5	200-1000	10-50	1	728
4-5-6	6	15	0-125	1-5	200-1000	25-35	1	728
7-8-9	6	15	0-125	1-5	200-1000	30-150	1	1064
10-11-12	6	15	0-125	1-5	200-1000	78-102	1	1064
13-14-15	6	15	0-125	2,6-3,4	520-680	10-50	1	728
16-17-18	6	15	0-125	1-5	400-2000	10-50	1	728
19-20-21	6	15	0-125	2,6-3,4	1040-1360	10-50	1	728
22-23-24	6	15	0-125	1-5	200-1000	10-50	1	780
25-26-27	6	15	0-125	1-5	200-1000	10-50	1	971
28-29-30	6	15	0-125	1-5	200-1000	10-50	1	662
31-32-33	24	15	0-125	1-5	200-1000	10-50	1	2912
34-35-36	6	30	0-125	1-5	200-1000	10-50	1	754
37-38-39	24	30	0-125	1-5	200-1000	10-50	1	3016

Tabela IV.1. Características dos Problemas do Grupo 1.

(*) os dois números que aparecem nestas colunas representam os intervalos entre os quais estes dados foram gerados. A geração foi feita através de uma distribuição uniforme.

Todos os 39 problemas do grupo 1 são factíveis, ou seja, existe pelo menos uma solução que satisfaz as restrições do problema.

Os 120 problemas, que constituem os grupos 2 a 7, foram gerados aleatoriamente. Os dados que são constantes para o grupo 1, também foram considerados constantes aqui (capacidade e custos e tempos para cada item i, ao longo do horizonte de tempo). Os números de itens (N) e de períodos (T) foram previamente escolhidos, e os valores considerados foram N = 6 e T = 12, N = 24 e T = 12, N = 30 e T = 12, N = 6 e T = 24, N = 10 e T = 24.

Para cada tamanho estabelecido acima foram gerados 4 problemas, totalizando 20 problemas para cada grupo.

A geração aleatória destes problemas foi feita da seguinte maneira:

Demanda. Para cada item e a cada período, as demandas foram geradas uniformemente entre 0 e 180.

Custos de preparação. Gerados uniformemente entre 5 e 95 para cada item.

Custos de estoque por unidade. Gerados uniformemente entre 0,2 e 0,4 para cada item.

Tempos de preparação. Gerados uniformemente entre 200 e 300 para cada item.

Tempos de processamento por unidade. Gerados uniformemente entre 2 e 3 para cada item.

Capacidade. Para este parâmetro, a geração foi feita utilizando a solução lote-por-lote, isto é, para cada período t calculou-se a capacidade necessária caso as demandas dos itens, neste período, fossem totalmente produzidas, ou seja:

$$CAP[t] = \sum_{i=1}^{N} d_{i,t}\, b_{i,t} + s_{i,t} \;(\text{se } d_{i,t} > 0) \qquad \text{(ii)}$$

e então, a capacidade para os períodos foi calculada fazendo-se a média dessas capacidades obtidas em (ii):

$$C_t = \left(\sum_{t=1}^{T} CAP[t] \right) / T \quad \text{para } \forall t$$

Na verdade, nesta geração aleatória de dados, foram gerados 20 problemas, que são os do grupo 2. Os problemas dos grupos 3, 4, 5 ,6 e 7 foram gerados a partir desses 20 problemas, fazendo-se o seguinte:

Grupo 3 - neste grupo as capacidades dos problemas do grupo 2 são multiplicadas por 0,9, ou seja, estes problemas tem suas capacidades nos períodos mais apertadas em relação aos problemas do grupo 2. Os demais dados ficam inalterados.

Grupo 4 - neste grupo as capacidades são multiplicadas por 1,10, ou seja, estes problemas ficam com suas capacidades mais folgadas em relação aos problemas do grupo 2. Os demais dados ficam inalterados.

Grupo 5 - neste grupo os custos de preparação dos problemas do grupo 2 são multiplicados por 10. Os demais dados ficam inalterados.

Grupo 6 - neste grupo as capacidades dos problemas do grupo 5 são multiplicadas por 0,9. Os demais dados são iguais aos do grupo 5, ou seja, para este grupo alteram-se os custos de preparação e as capacidades.

Grupo 7 - neste grupo as capacidades dos problemas do grupo 5 são multiplicadas por 1,10. Os demais dados ficam inalterados.

Note que não há garantia que todos estes 120 problemas gerados sejam factíveis.

Um exemplo de um problema gerado de acordo com os critérios anteriores é ilustrado nas Tabelas IV.2. a IV.7.. Seja N = 6 e T = 12.

	Período											
Item	1	2	3	4	5	6	7	8	9	10	11	12
1	19	159	161	86	80	94	65	34	129	150	168	152
2	75	177	32	63	40	137	144	66	115	2	111	47
3	52	122	51	47	81	164	10	65	131	99	165	47
4	50	36	129	26	42	36	11	147	23	129	26	11
5	138	89	12	50	50	42	55	106	112	114	109	102
6	133	117	52	144	68	61	53	132	139	105	42	141

Tabela IV.2. Demandas

Item	1	2	3	4	5	6
Custo	49	40	10	82	91	93

Tabela IV.3. Custos de Preparação por Item

Item	1	2	3	4	5	6
Custo	0,27	0,23	0,30	0,30	0,27	0,27

Tabela IV.4. Custos Unitários de Estoque por Item

Item	1	2	3	4	5	6
Tempo	287	264	275	248	206	205

Tabela IV.5. Tempos de Preparação por Item

Item	1	2	3	4	5	6
Tempo	2,15	2,66	2,23	2,42	2,07	2,05

Tabela IV.6. Tempos de Processamento Unitários para cada Item

Período	1	2	3	4	5	6	7	8	9	10	11	12
Capacidade	2814	2814	2814	2814	2814	2814	2814	2814	2814	2814	2814	2814

Tabela IV.7. Capacidade por Período em Unidades de Tempo

IV.3. RESULTADOS DOS TESTES

Os detalhes da implementação do algoritmo e os testes executados são apresentados a seguir de acordo com a ordem em que foram feitos.

ETAPA 1 - Nesta etapa, foi feita a implementação do algoritmo com as seguintes fases:

FASE 1 - passos de factibilização

FASE 2 - passo de melhoria

FASE 3 - passo de agregação

Ou seja, aqui foi implementado o procedimento HEUR1.

ETAPA 2 - Foram feitos testes no procedimento HEUR1 com a finalidade de ajustar os seguintes parâmetros:

1.Razão - utilizada nos passos da FASE 1 para a escolha da melhor transferência

2.Penalidade - utilizada nos passos das FASES 1 e 3 para cálculo do Custo_adicional.

Para o parâmetro Razão, foram testadas as seguinte possibilidades:

a)Razão = Custo_adicional/Redução do Excesso[t]

b)Razão = Custo_adicional

c)Razão = Custo_adicional/$q_{i,t}$ onde $q_{i,t}$ significa a quantidade que está sendo analisada para ser transferida.

Dentre as três alternativas, a que se mostrou melhor, na média, em termos de qualidade das soluções e de número de problemas onde encontrou-se solução

factível, foi a primeira (a). Os problemas aqui considerados foram os 39 do grupo 1.

Para o parâmetro Penalidade, que é utilizado no cálculo do Custo_adicional para a penalização dos excessos positivos dos períodos origem (t) e destino (tl) resultantes da transferência candidata, foram testadas as seguintes possibilidades:

a)penalidade constante igual a 1, ou seja,

penalização = max{0,Excesso[t]} + max{0,Excesso[tl]}

b)penalidade costante igual a 2 (ou 3)

penalização = 2(max{0,Excesso[t]} + max{0,Excesso[tl]})

c)penalidade variante. Para esta alternativa, criou-se os fatores de penalidade para cada período t (Fator_pen[t]) que foram utilizados na contabilização do Custo_adicional da seguinte maneira:

$$\text{penalização} = \text{Fator\_pen}[t] \times \max\{0,\text{Excesso}[t]\} + \\ + \text{Fator\_pen}[tl] \times \max\{0, \text{Excesso}[tl]\}$$

Os testes para esta estratégia foram os seguintes:

c.1)atualização do Fator_pen somando 1 ao final de cada passo progressivo e regressivo e do passo de agregação, para os períodos que se apresentavam infactíveis.

c.2)atualização do Fator_pen somando 1 ao final de cada passo progressivo e regressivo, para os períodos que se apresentavam infactíveis.

c.3)atualização do Fator_pen somando 1 ao final de cada par de passos - regressivo e progressivo -, para *todos* os períodos.

c.4)atualização do Fator_pen multiplicando por 2 (ou 3) ao final de cada passo regressivo e progressivo, para os períodos que se apresentavam infactíveis.

A escolha da melhor estratégia também foi feita considerando-se, em média, a qualidade das soluções obtidas para os problemas e o número de problemas que resultaram factíveis de acordo com cada alternativa testada. Desta forma a que se mostrou mais atrativa foi a alternativa c.2. Aqui também foram usados os 39 problemas do grupo 1.

ETAPA 3 - Nesta etapa, foi implementada a Busca Tabu para o procedimento HEUR1 já com os parâmetros da etapa 2 ajustados. O atributo utilizado para a implementação da estratégia foi a variável $y_{i,t}$(já detalhado no capítulo anterior) e os testes executados dizem respeito ao critério de aspiração e tamanho da lista tabu.

Para critério de aspiração, utilizou-se a seguinte propriedade no Custo_adicional (utilizado nos passos de factibilização e de agregação):

$$\text{Custo\_adicional} - \text{penalização} \leq 0$$

Isto significa que, se um movimento tabu satisfaz esta propriedade, então este movimento perde seu status tabu e pode ser efetivado caso seja escolhido. Portanto, os testes que foram feitos sobre este critério de aspiração dizem respeito à sua utilização ou não, nos passos de factibilização e agregação, e tiveram como objetivo verificar se o uso deste critério acarretaria algum ganho em termos de valor de solução para os problemas e/ou perdas por possível ocorrência de ciclagem.

A conclusão a que se chegou foi que a sua utilização, tanto para os passos de factibilização como para o passo de agregação, resultou, na média, em melhores soluções e não apresentou influências em termos de ciclagem. Devido a este fato este critério foi implementado para HEUR1 nos passos das FASES 1 e 3.

Quanto ao tamanho da lista, foram testados inúmeros valores variando de 1 a 500. Considerando qualidade das soluções, em média, para os problemas testados, que novamente foram os do grupo 1, o intervalo que pareceu mais atrativo foi de 11 a 21. Dentro deste intervalo, considerando número de problemas que obtiveram solução factível e menor quantidade de problemas que apresentavam ciclagem, o tamanho da lista escolhido foi 14.

ETAPA 4 - Desenvolveu-se os passos de trocas, com e sem infactibilidade, ou seja, nesta etapa foram implementados os procedimentos HEUR2 e HEUR3. Aqui também foram feitos testes em relação à vários parâmetros:

a)Razão - utilizada na FASE 1 destes procedimentos.

b)Penalidade - utilizada também na FASE 1 para cálculo do Custo_adicional.

c)Alfa - usado nos passos de troca para cálculo da quantidade do item i a ser transferida.

d)Δinfac - utilizado para determinar a infactibilidade que é permitida no período k, para o passo de troca com infactibilidade.

No que diz respeito a Razão, as possibilidades testadas foram as mesmas da etapa 2, e de acordo com os critérios de qualidade de solução e número de problemas que obtiveram solução factível, em média, a alternativa b se mostrou melhor, ou seja, para estes procedimentos, que utlizam os passos de troca na FASE 3, a Razão que é utilizada para a escolha da melhor transferência nos passos de factibilização é Razão = Custo_adicional.

Para o parâmetro penalidade, também foram consideradas todas as possibilidades descritas anteriormente na etapa 2, e novamente de acordo com os critérios lá detalhados, para a escolha da melhor maneira de penalização dos excessos, foi escolhida a alternativa c.3.

Quanto ao parâmetro alfa, na verdade o objetivo foi escolher o valor de alfa máximo, que determina em quantas *partes* $x_{i,t}$ deve ser dividido para ser transferido. Para este teste, o intervalo de alfa máximo testado foi de 2 a 10. Em termos de qualidade de solução e número de problemas que apresentaram solução factível, tem-se os seguintes resultados:

- para o passo de trocas sem infactibilidade, o valor de alfa máximo que se apresentou melhor, em média, foi alfa máximo igual a 3 e para o passo de trocas com infactibilidade o melhor valor para alfa máximo foi 2.

Com relação à infactibilidade (Δinfac) que é permitida para o período k, no passo de trocas com infactibilidade, optou-se por se fazer uma geração aleatória de números entre alguns intervalos, isto é, a cada vez que executa-se este passo de trocas um número é gerado aleatoriamente seguindo uma distribuição uniforme. Os intervalos que foram testados para essa geração foram de 0-60 e 0-20, sendo que o segundo obteve melhores resultados. Então a cada passo de trocas um número entre 0-20 é gerado e Δinfac é igual a este número/100.

ETAPA 5 - Aqui foi implementada a Busca Tabu para os dois procedimentos desenvolvidos na etapa anterior (HEUR2 e HEUR3). Os testes feitos foram com relação ao atributo, critério de aspiração e tamanho da lista.

Os atributos que foram testados são (i,t,tl), (i,t,q) e (i,t) e o detalhamento de cada um deles já foi apresentado no capítulo anterior. Devido a ocorrência de ciclagem, na maioria dos problemas, utilizando os dois primeiros, optou-se pelo uso de (i,t).

O critério de aspiração testado foi o mesmo que está descrito na etapa 3. Os testes que foram feitos se relacionam com a sua utilização ou não, nos passos da FASE 1 (dos procedimentos HEUR2 e HEUR3) e no passo de troca com infactibilidade (para o passo de trocas factíveis, este critério não faz sentido, pois para este passo não se utiliza penalização). A conclusão que se obteve é que este critério de aspiração deve ser usado nos passos da FASE 1 - Factibilização, e não deve ser usado no passo de troca, pois para este o uso causou ciclagem em vários problemas, influenciando, portanto, a qualidade das soluções.

Quanto ao parâmetro tamanho da lista, novamente foram testados inúmeros valores. Dentre os testados, o intervalo que se mostrou mais atrativo foi de 14 a 27. Para este intervalo, a escolha foi feita de acordo com o número de problemas que resultaram factíveis e com a qualidade das soluções obtidas, o que resultou para os procedimentos HEUR2 e HEUR3, o tamanho igual a 25. Vale ressaltar que os testes para tamanho de lista foram feitos para ambos os procedimentos e o fato do melhor tamanho da lista ser o mesmo pode ser visto como coincidência ou pode ser que este tamanho tenha alguma relação com o atributo que está sendo utilizado.

Para esta etapa, as conclusões obtidas também levaram em consideração os problemas do grupo 1.

ETAPA 6 - Já tendo ajustado todos os parâmetros para os três procedimentos, HEUR1, HEUR2 e HEUR3, o objetivo agora é avaliar o desempenho destes procedimentos, com e sem a estratégia da Busca Tabu.

A avaliação do desempenho das heurísticas propostas nesta tese é feita comparando-se os resultados obtidos por elas com os resultados obtidos pela heurística TTM (proposta em [14]). Utiliza-se também o valor de um limitante inferior para a solução ótima do problema sugerido em [14]. Este limitante é o valor da função dual alcançada pela relaxação lagrangeana do problema. Nesta heurística TTM o modelo matemático usado é o mesmo usado nesta tese e utiliza-se a relaxação lagrangeana para dualizar as restrições de capacidade e incluí-las na função objetivo, isto é, definem-se

$$c'_{i,t} = c_{i,t} + \lambda_t b_{i,t}$$

$$S'_{i,t} = S_{i,t} + \lambda_t s_{i,t}$$

como os "novos" custos da função objetivo, onde λ_t são os multiplicadores de Lagrange. A seguir é apresentado o funcionamento da heurística TTM:

passo 0: Inicialização e entrada de dados.
Na inicialização os multiplicadores de Lagrange são zero e iter = 0.

passo 1: Aplicação do algoritmo de Wagner e Whitin, utilizando os custos lagrangeanos.

passo 2: Cálculo das restrições de capacidade em cada período, baseado na solução obtida do passo 1.

passo 3: Procedimento de factibilização.

passo 4: Aplicação do critério de parada. Se este for satisfeito, PARE. Senão, vá ao passo 5.

passo 5: Procedimento DUAL. Atualize os multiplicadores de Lagrange, usando subgradiente. Faça iter = iter + 1 e volte ao passo 1.

Durante cada iteração, o passo 1 produz um limitante inferior para o custo ótimo. Este limitante é usado na decisão de quando parar e na avaliação da qualidade das soluções.

No passo 3, o procedimento de factibilização utiliza transferências de quantidade de itens de períodos infactíveis para períodos anteriores e posteriores. Este procedimento executa, no máximo, quatro passos. O primeiro passo tem início no último período do horizonte e move produção de períodos infactíveis para períodos anteriores afim de eliminar as infactibilidades (é um passo regressivo no tempo). Se ele falha, então um passo que transfere quantidades para frente (passo progressivo) é tentado. Caso a infactibilidade persista, tentam-se o terceiro e quarto passos que são essencialmente repetições dos dois primeiros. O critério para transferir os itens é baseado na minimização da variação dos custos (Lagrangeanos). Permite-se a transferência de um ou mais lotes, se necessário, mas para cada período t somente um lote pode ser dividido (pode-se notar aqui as diferenças em relação ao método desenvolvido no capítulo anterior).

O método do subgradiente utilizado na solução do problema incorpora algumas das modificações propostas por [18] para reduzir o *"zig-zag"*. O tamanho dos passos, número de iterações, etc. foram ajustados de acordo com testes.

A heurística TTM executa no máximo 100 iterações. A implementação dela foi feita em linguagem FORTRAN 77. O código foi gentilmente cedido pelo autor para que pudessemos realizar os testes.

Esta heurística foi então executada para os 159 problemas, e com isso obteve-se os valores do limitante inferior para cada problema e, para aqueles em que foram encontradas soluções factíveis, o valor da melhor solução.

Os procedimentos HEUR1, HEUR2 e HEUR3 também foram executados para todos os problemas e a seguir são apresentados os resultados.

A Tabela IV.8. mostra os resultados obtidos pela execução de HEUR1, HEUR2 e HEUR3 sem a estratégia de Busca Tabu implementada e também pela execução da heurística TTM, para os problemas do grupo 1.

Uma observação importante é que para os procedimentos HEUR1, HEUR2, HEUR3 o critério de parada utilizado foi o tempo de execução, pois considerando que os passos de trocas são mais elaborados que o passo de agregação seria inviável tentarmos fazer uma comparação, por exemplo, em termos do número de iterações, pois enquanto o algoritmo com o passo de agregação demora 20 segundos para executar 1500 iterações para um problema pequeno, os algoritmos com passo de trocas demoram 5 minutos com esse mesmo número de iterações, para o mesmo problema, ou seja 15 vezes mais (em média). Em problemas grandes esta comparação se tornaria absurda.

Os resultados obtidos, para os mesmos problemas da Tabela IV.8., da execução de HEUR1, HEUR2 e HEUR3 com a aplicação da Busca Tabu são apresentados na Tabela IV.9. e, para facilitar a comparação, novamente são mostrados os resultados obtidos da execução de TTM.

A Tabela IV.10. mostra os resultados obtidos para os 120 problemas (dos grupos 2 a 7) da execução de HEUR1, HEUR2 e HEUR3 sem a estratégia de Busca Tabu e de TTM. Nesta tabela optou-se por mostrar, para cada grupo, os resultados em relação às médias. Esta tabela também contém o número de problemas onde encontrou-se solução factível para cada uma das heurísticas executadas e o tempo de execução. O tempo apresentado também é o tempo médio por grupo, pois em cada grupo, o tempo especificado para cada tamanho de problema é diferente.

Para esses 120 problemas, os resultados obtidos por HEUR1, HEUR2 e HEUR3 com a Busca Tabu são mostrados na Tabela IV.11., onde novamente são mostrados os resultados obtidos por TTM.

Problema	$\frac{Z^{TTM}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z^{HEUR1}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z^{HEUR2}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z^{HEUR3}-LB^{TTM}}{LB^{TTM}}$ (%)	Tempo (1) (s)	Tempo (2) (s)
1	7,898	7,270	3,658	4,221	20	3,2
2	12,711	não encontrou solução factível	não encontrou solução factível	não encontrou solução factível	20	4,2
3	5,480	7,611	6,650	4,963	20	3,4
4	2,402	8,784	7,367	5,617	20	3,3
5	6,588	7,575	5,768	7,973	20	3,3
6	2,454	5,769	3,617	3,645	20	3,1
7	3,891	6,492	5,810	5,810	20	3,0
8	6,543	6,511	7,841	6,763	20	3,0
9	2,745	2,303	2,265	2,265	20	3,0
10	2,146	5,610	4,278	4,676	20	3,0
11	0,455	1,521	0,992	0,992	20	2,8
12	0,595	3,254	3,254	3,254	20	3,0
13	8,795	7,714	8,538	6,356	20	3,5
14	6,738	7,581	7,005	7,151	20	3,3
15	3,735	7,450	9,383	5,531	20	3,6
16	9,200	11,693	6,846	6,846	20	3,2
17	4,891	5,094	3,253	3,253	20	3,7
18	3,111	6,402	3,858	2,955	20	3,4
19	1,694	10,350	6,697	4,139	20	3,2
20	8,625	13,499	12,223	7,243	20	3,7
21	1,684	5,850	7,574	6,355	20	3,2
22	7,045	11,739	8,532	6,366	20	3,5
23	3,772	12,113	7,992	5,996	20	3,0
24	9,735	11,987	não encontrou solução factível	não encontrou solução factível	20	3,9
25	0,009	0,321	0,321	0,321	20	2,9
26	0,257	1,737	0,922	0,922	20	2,9
27	0,807	2,061	2,061	2,061	20	3,0

Tabela IV.8. Resultados do Grupo 1 sem Busca Tabu.

Problema	$\frac{Z^{TTM}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z^{HEUR1}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z^{HEUR2}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z^{HEUR3}-LB^{TTM}}{LB^{TTM}}$ (%)	Tempo (1) (s)	Tempo (2) (s)
28	7,780	não encontrou solução factível	12,473	9,056	20	3,9
29	11,130	não encontrou solução factível	9,365	8,438	20	4,1
30	10,684	não encontrou solução factível	19,042	25,263	20	4,0
31	0,124	0,947	0,600	0,780	75	12,1
32	0,495	0,916	0,416	0,504	75	12,7
33	0,434	1,547	1,688	1,688	75	12,3
34	3,787	4,087	4,128	4,512	60	16,0
35	7,055	7,820	9,727	7,936	60	16,4
36	2,931	3,134	3,028	2,991	60	16,1
37	0,218	0,545	0,676	0,681	240	63,9
38	0,244	0,899	0,672	0,668	240	64,1
39	0,609	1,399	1,272	1,315	240	65,6

Tabela IV.8. (Continuação)

onde:

LB^{TTM} - valor do limitante inferior obtido pela execução de TTM.
Z^{TTM} - valor da melhor solução obtida pela execução de TTM.
Z^{HEUR1} - valor da melhor solução obtida pela execução de HEUR1.
Z^{HEUR2} - valor da melhor solução obtida pela execução de HEUR2.
Z^{HEUR3} - valor da melhor solução obtida pela execução de HEUR3.
Tempo (1)- tempo de execução (pré-especificado) para HEUR1, HEUR2 e HEUR3.
Tempo (2)- tempo de execução da heurística TTM (Esta heurística executa, no máximo, 100 iterações)

Problema	$\frac{Z^{TTM}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z_T^{HEUR1}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z_T^{HEUR2}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z_T^{HEUR3}-LB^{TTM}}{LB^{TTM}}$ (%)	Tempo(1) (s)	Tempo(2) (s)
1	7,898	6,735	4,037	3,888	20	3,2
2	12,711	25,539	17,001	23,632	20	4,2
3	5,480	9,819	4,301	5,047	20	3,4
4	2,402	6,260	2,797	2,352	20	3,3
5	6,588	6,301	5,007	6,631	20	3,3
6	2,454	4,504	2,552	2,802	20	3,1
7	3,891	4,760	4,145	2,617	20	3,0
8	6,543	6,511	5,528	7,502	20	3,0
9	2,745	2,303	1,247	1,247	20	3,0
10	2,146	5,661	3,226	2,518	20	3,0
11	0,455	1,521	0,992	1,061	20	2,8
12	0,595	3,254	2,553	1,374	20	3,0
13	8,795	7,809	6,628	5,828	20	3,5
14	6,738	8,609	6,007	5,011	20	3,3
15	3,735	5,339	8,105	4,779	20	3,6
16	9,200	10,263	7,586	7,110	20	3,2
17	4,891	5,094	3,253	5,259	20	3,7
18	3,111	6,224	3,907	3,625	20	3,4
19	1,694	5,717	6,697	5,876	20	3,2
20	8,625	8,358	12,841	7,917	20	3,7
21	1,684	6,408	7,784	5,732	20	3,2
22	7,045	12,127	6,800	4,660	20	3,5
23	3,772	7,101	6,613	3,419	20	3,0
24	9,735	8,642	7,847	5,527	20	3,9
25	0,009	0,321	0,321	0,321	20	2,9
26	0,257	1,737	0,578	0,578	20	2,9
27	0,807	2,061	2,061	1,041	20	3,0

Tabela IV.9. Resultados do Grupo 1 com Busca Tabu.

Problema	$\frac{Z^{TTM}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z_T^{HEUR1}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z_T^{HEUR2}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z_T^{HEUR3}-LB^{TTM}}{LB^{TTM}}$ (%)	Tempo(1) (s)	Tempo(2) (s)
28	7,780	12,244	12,809	11,320	20	3,2
29	11,130	13,772	9,057	9,266	20	4,1
30	10,684	18,536	14,887	10,965	20	4,0
31	0,124	0,947	0,726	0,904	75	12,1
32	0,495	0,847	0,513	0,449	75	12,7
33	0,434	1,7001	1,688	1,686	75	12,3
34	3,786	4,087	4,021	3,964	60	16,0
35	7,055	9,458	4,378	4,301	60	16,4
36	2,931	3,134	3,028	3,025	60	16,1
37	0,218	0,545	0,676	0,571	240	63,9
38	0,244	0,914	0,675	0,580	240	64,1
39	0,609	1,196	1,272	1,090	240	65,6

Tabela IV.9. (Continuação)

onde:

LB^{TTM} - valor do limitante inferior obtido pela execução de TTM.
Z^{TTM} - valor da melhor solução obtida pela execução de TTM.
Z_T^{HEUR1} - valor da melhor solução obtida pela execução de HEUR1 com Busca Tabu.
Z_T^{HEUR2} - valor da melhor solução obtida pela execução de HEUR2 com Busca Tabu.
Z_T^{HEUR3} - valor da melhor solução obtida pela execução de HEUR3 com Busca Tabu.
Tempo (1)- tempo de execução (pré-especificado) para HEUR1, HEUR2 e HEUR3.
Tempo (2)- tempo de execução da heurística TTM (Esta heurística executa, no máximo, 100 iterações)

	$\frac{Z^{TTM}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z^{HEUR1}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z^{HEUR2}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z^{HEUR3}-LB^{TTM}}{LB^{TTM}}$ (%)	PF(TTM)	PF(HEUR1)	PF(HEUR2)	PF(HEUR3)	Tempo (1) (s)	Tempo (2) (s)
Grupo 2	0,688	1,635	1,292	1,330	9	9	9	9	58	9,3
Grupo 3	1,095	2,490	4,033	3,958	2	2	2	2	115	17
Grupo 4	0,480	1,217	0,902	0,902	18	18	18	18	50	8,2
Grupo 5	9,490	12,279	11,862	16,046	8	9	9	9	79	12,5
Grupo 6	58,953	5,998	5,190	4,605	2	1	1	1	136	20,5
Grupo 7	4,324	6,877	7,061	6,421	18	18	18	18	60	9,3

Tabela IV.10. Resultados dos Grupos de 2 a 7 sem Busca Tabu.

onde:

LB^{TTM} - valor do limitante inferior obtido pela execução de TTM.

Z^{TTM} - valor da melhor solução obtida pela execução de TTM.

Z^{HEUR1} - valor da melhor solução obtida pela execução de HEUR1 (sem Busca Tabu).

Z^{HEUR2} - valor da melhor solução obtida pela execução de HEUR2 (sem Busca Tabu).

Z^{HEUR3} - valor da melhor solução obtida pela execução de HEUR3 (sem Busca Tabu).

Tempo (1) - tempo médio de execução para HEUR1, HEUR2 e HEUR3.

Tempo (2) - tempo médio de execução de TTM.

PF(TTM) - número de problemas para os quais TTM encontrou solução factível.

PF(HEUR1) - número de problemas para os quais HEUR1 (sem Busca Tabu) encontrou solução factível.

PF(HEUR2) - número de problemas para os quais HEUR2 (sem Busca Tabu) encontrou solução factível.

PF(HEUR3) - número de problemas para os quais HEUR3 (sem Busca Tabu) encontrou solução factível.

	$\frac{Z^{TTM}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z_T^{HEUR1}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z_T^{HEUR2}-LB^{TTM}}{LB^{TTM}}$ (%)	$\frac{Z_T^{HEUR3}-LB^{TTM}}{LB^{TTM}}$ (%)	PF(TTM)	PF(HEUR1)	PF(HEUR2)	PF(HEUR3)	Tempo (1) (s)	Tempo (2) (s)
Grupo 2	0,688	1,705	1,181	1,030	9	9	9	9	58	9,3
Grupo 3	1,095	2,490	3,628	2,997	2	2	2	2	115	17
Grupo 4	0,480	1,220	0,891	0,877	18	18	18	18	50	8,2
Grupo 5	9,490	12,448	12,652	13,351	8	9	9	9	79	12,5
Grupo 6	58,953	5,998	5,190	5,609	2	1	1	1	136	20,5
Grupo 7	4,324	6,454	7,295	5,777	18	18	18	18	60	9,3

Tabela IV.11. Resultados dos Grupos 2 a 7 com Busca Tabu.

onde:

LB^{TTM} - valor do limitante inferior obtido pela execução de TTM.

Z^{TTM} - valor da melhor solução obtida pela execução de TTM.

Z_T^{HEUR1} - valor da melhor solução obtida pela execução de HEUR1 com Busca Tabu.

Z_T^{HEUR2} - valor da melhor solução obtida pela execução de HEUR2 com Busca Tabu.

Z_T^{HEUR3} - valor da melhor solução obtida pela execução de HEUR3 com Busca Tabu.

Tempo (1) - tempo médio de execução para HEUR1, HEUR2 e HEUR3.

Tempo (2) - tempo médio de execução de TTM.

PF(TTM) - número de problemas para os quais TTM encontrou solução factível.

PF(HEUR1) - número de problemas para os quais HEUR1 com Busca Tabu encontrou solução factível.

PF(HEUR2) - número de problemas para os quais HEUR2 com Busca Tabu encontrou solução factível.

PF(HEUR3) - número de problemas para os quais HEUR3 com Busca Tabu encontrou solução factível.

IV.3.1. Análise dos Resultados

Da análise dos resultados, quanto à qualidade das soluções obtidas, pode-se verificar que a inclusão da Busca Tabu para os três procedimentos promoveu, para os problemas do grupo 1, um ganho de qualidade e também um aumento no número de problemas para os quais encontrou-se solução factível.

A Tabela IV.12. mostra o ganho percentual médio obtido pela Busca Tabu para cada procedimento. Esses valores foram obtidos em relação aos problemas do grupo 1 em que, tanto o procedimento sem a Busca Tabu quanto o que a utiliza encontraram solução factível.

Procedimentos	Ganho médio com aplicação da Busca Tabu
HEUR 1	0,44 %
HEUR 2	0,42 %
HEUR 3	0,50 %

Tabela IV.12. Ganho obtido para os três procedimentos com a utilização da estratégia Tabu.

Para os problemas dos grupos 2 a 7, a inclusão da Busca Tabu apresentou para o procedimento HEUR3 um ganho médio de 0,59%.. Para os procedimentos HEUR1 e HEUR2 observou-se, com a estratégia Tabu, uma piora nas soluções obtidas - perda média de 0,038% para HEUR1 e 0,13% para HEUR2. Em relação ao número de problemas para os quais se encontrou solução factível, a inclusão da Busca Tabu não causou alteração para nenhum procedimento.

Em relação à heurística TTM, pode-se notar que em alguns casos os procedimentos HEUR1, HEUR2 e HEUR3 superaram o valor das soluções por ela encontrado, mesmo sem a Busca Tabu, porém com um tempo um pouco maior, mas que ainda pode ser considerado baixo. O fato de termos pré-especificado tempos mais altos que os obtidos por TTM na resolução dos problemas, tem como justificativas a utilização da Busca Tabu, que parece começar a fazer efeito apenas após um certo número de transferências executadas, e também a utilização dos passos de troca, que utilizam um tempo significativo. Fazendo um paralelo entre a heurística TTM e HEUR2, temos que a execução de 100 iterações da primeira para um determinado problema gasta 3 segundos, enquanto que este mesmo número de iterações para a segunda heurística (HEUR2) leva 40 segundos, para o mesmo problema (obviamente tentando contar o número de iterações de modo semelhante).

Comparando a heurística TTM com o procedimento HEUR1 com a estratégia Tabu, conclue-se que HEUR1 se mostrou superior em 10,5% dos problemas em que ambos encontraram solução factível (isto em relação a todos os grupos), e tendo se mostrado semelhante em 1,04% destes problemas.

Com relação ao procedimento HEUR2, também com Busca Tabu, este se mostrou superior a TTM em 17,89% dos problemas e foi equivalente em 3,15%. Para o procedimento HEUR3 obteve-se, em 24,02% dos problemas, solução melhor que as encontradas por TTM. Para este procedimento também obteve-se equivalência em 3,15% dos problemas.

Quanto à comparação dos três procedimentos, pode-se notar que os que utilizam trocas (HEUR2 e HEUR3) se mostraram mais eficientes. Isto pode ser justificado pelo fato dos passos de troca serem mais elaborados, considerando um maior número de possibilidades de transferências, o que no passo de agregação não ocorre, pois neste as transferências são drásticas (transfere todo o lote). Um fato que pode explicar o baixo desempenho de HEUR1 (que utiliza o passo de agregação) é que foi verificada a ocorrência de ciclagem para alguns problemas, mesmo com a inclusão da Busca Tabu. Isto provavelmente acontece devido ao atributo escolhido, que não se mostra muito efetivo por ser pouco restritivo.

Considerando os procedimentos que utilizam trocas (HEUR2 e HEUR3), nota-se uma superioridade de HEUR3, que pode ser explicada pelo fato deste trabalhar com infactibilidade fazendo com que a busca fique menos restrita. Como já foi dito no capítulo II, às vezes, para que se consiga obter melhores soluções, é necessário cruzar as fronteiras entre factibilidade e infactibilidade.

As análises quanto ao desempenho dos procedimentos (HEUR1, HEUR2 e HEUR3) em relação às características dos problemas, foram feitas com base nos 120 problemas, pois estes possuem características bem definidas. Portanto, para os problemas que possuem custos de preparação alto (grupos 5, 6 e 7) nota-se que o *"gap"* da solução aumenta. Para os problemas onde as capacidades são apertadas (grupo 6), a resolução se torna mais difícil.

Quanto à variação no tamanho dos problemas, nota-se que os três procedimentos se mostraram eficientes para todos os tamanhos especificados, ou seja, pode-se dizer que tanto o tempo computacional quanto as soluções obtidas são razoáveis mesmo quando se trabalha com problemas grandes.

CAPÍTULO V

CONCLUSÃO

Neste trabalho foi proposto um método heurístico de solução para o problema de dimensionamento de lotes de múltiplos itens em um horizonte de tempo finito. O modelo considerado incorpora restrições de capacidade de produção limitada e tempos de preparação.

O método é constituído de três fases, onde para a terceira fase foram elaboradas três alternativas diferentes de passos, e devido a esta característica o método foi separado em três procedimentos heurísticos diferentes.

Também foi feita uma tentativa de melhorar o desempenho deste método através da aplicação da estratégia da Busca Tabu. Para isso foram incorporados os mecanismos de Busca Tabu simples.

Durante a implementação do método e das técnicas de Busca Tabu, foram executados vários testes computacionais, a fim de ajustar, da melhor maneira possível, os parâmetros por eles utilizados.

Depois de feitos os ajustes necessários, vários experimentos foram feitos para que se pudesse analisar o desempenho do algoritmo, ou melhor, de cada procedimento, com e sem a incorporação da Busca Tabu. De uma maneira geral, os resultados obtidos mostraram que a inclusão da Busca Tabu aos procedimentos melhorou as soluções obtidas.

Também foi feita uma comparação dos procedimentos com a heurística TTM da literatura [14]. Nesta comparação, o que se pôde perceber é que, mesmo tendo os procedimentos aqui desenvolvidos obtido resultados melhores para alguns problemas, a heurística TTM se mostrou superior, em média, para os problemas testados.

Uma conclusão importante quanto ao desempenho dos procedimentos é que sua aplicação a problemas de grande porte mostrou-se satisfatória.

Pesquisas futuras no sentido de melhorar o desempenho dos procedimentos heurísticos podem ser realizadas explorando-se algumas variações que o algoritmo permite. São elas:

- utilização de um atributo mais restritivo no procedimento que utiliza o passo de agregação na terceira fase, devido ao fato de se ter observado ciclagem em alguns problemas.

- utilização de tamanho da lista aleatório, que tem sido usado em pesquisas recentes e apresentado bons resultados. Isto porque, pelo fato dos problemas possuírem tamanho e características bem diferentes, é muito difícil encontrar um tamanho de lista (e provavelmente este não exista) que seja o melhor para todos os problemas.

- também pode-se testar a utilização de funções de memória de médio e longo prazo, que podem trazer ganhos à qualidade das soluções.

Algoritmos semelhantes ao utilizado neste trabalho estão sendo desenvolvidos para o problema em sistemas multi-estágio e para o problema com múltiplas máquinas.

APÊNDICE A

ALGORITMO WAGNER-WHITIN

Considere o seguinte modelo:

$$(P)\ \min \sum_{t=1}^{T} c_t(x_t) + h_t(I_t)$$

sujeito a

$$I_t = I_{t-1} + x_t - d_t \qquad\qquad t = 1,\dots,T$$

$$I_0 = 0,\ I_T = 0$$

$$x_t \geq 0,\ I_t \geq 0 \qquad\qquad t = 1,\dots,T$$

Supondo que as funções $c_t(x_t)$ e $h_t(I_t)$ sejam funções côncavas, segue-se que a função objetivo do problema (P) é côncava. Assumindo que o conjunto convexo das restrições de (P) é limitado, tem-se que o mínimo da função objetivo ocorre num ponto extremo deste conjunto convexo [20].

Considere $d_t > 0$, $t = 1,\dots,T$. Como se tem T restrições em (P) segue-se que somente uma das variáveis I_{t-1} e x_t deve ser positiva, isto é

$$I_{t-1}\, x_t = 0 \qquad t = 1,\dots,T \tag{1.1}$$

A dedução da propriedade (1.1) baseia-se no conceito de solução básica, que caracteriza um vértice do poliedro formado pelas restrições de (P).

Considere agora que $d_k = 0$, $k \in \{1,\dots,T\}$ e $d_t > 0$, $t \neq k$. Neste caso é factível ter $I_{k-1} = x_k = 0$, o que significa que em algum período $t \neq k$ poderia existir $I_{t-1} > 0$ e $x_t > 0$. Entretanto, é possível mostrar que esta solução não é ótima. Note que se $I_{k-1} = x_k = 0$ então a solução de custo mínimo pode ser encontrada pela decomposição do problema (P) original em dois subproblemas independentes: um envolvendo o planejamento do período 1 ao período k-1 e o outro envolvendo o planejamento dos períodos k+1 até T. O primeiro subproblema tem k-1 variáveis não nulas na solução ótima. O segundo problema tem T-k restrições e portanto T-k variáveis não nulas na solução ótima. Portanto o problema total possui T-1 variáveis não nulas na sua solução ótima, o que implica $I_{t-1}\, x_t = 0$.

De (1.1) obtém -se:

$$I_{t-1} > 0 \rightarrow x_t = 0$$

$$I_{t-1} = I_t = 0 \rightarrow x_t = d_t$$

$$I_{t-1} = 0,\ I_t > 0,\ I_{t+1} = 0 \rightarrow x_t = d_t + d_{t+1}$$

.

.

.

$$I_{t-1} = 0,\ I_t > 0,\ I_{t+1} > 0, \ldots, I_{t+k-1} > 0,\ I_{t+k} = 0 \rightarrow x_t = d_t + \ldots + d_{t+k}$$

A propriedade (1.1) permite reduzir bastante o espaço das variáveis de produção para obter uma solução através da programação dinâmica. Como mostrado acima basta considerar os seguintes valores para x_t: 0, d_t, $d_t + d_{t+1}$, ..., $d_t + d_{t+1} + \ldots + d_T$.

Vamos deduzir agora o algoritmo de Wagner-Whitin. Suponha que $I_j = 0$, $I_{j+1} > 0, \ldots, I_{k-1} > 0$, $I_k = 0$, $j = 0, 1, \ldots, T-1$. Então:

$$x_{j+1} = d_{j+1} + d_{j+2} + \ldots + d_k \quad \text{e}$$

$$I_t = x_{j+1} - \sum_{r=j+1}^{t} d_r = \sum_{r=t+1}^{k} d_r \qquad t = j+1, \ldots, k-1.$$

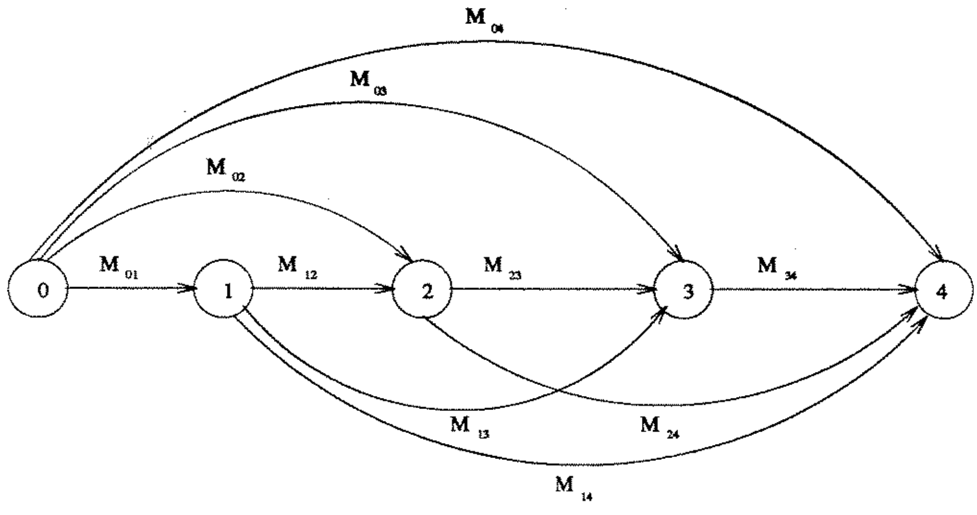
Seja M_{jk} o custo de produzir em j+1 e atender as demandas no período j+1, j+2,...,k ($j = 0,1,\ldots,T-1$; $k = j+1, j+2, \ldots, T$). Portanto,

$$M_{jk} = c_{j+1}(x_{j+1}) + \sum_{t=j+1}^{k-1} h_t(I_t)$$

$$= c_{j+1}\left(\sum_{r=j+1}^{k} d_r\right) + \sum_{t=j+1}^{k-1} h_t\left(\sum_{r=t+1}^{k} d_r\right)$$

$$= c_{j+1}\left(\sum_{r=j+1}^{k} d_r\right) + h_{j+1}\left(d_{j+2} + d_{j+3} + \ldots + d_k\right) + \ldots + h_{k-1}\left(d_k\right)$$

O problema (P) pode então ser representado pelo seguinte grafo orientado acíclico (para exemplificar T = 4):



O arco (j,k) na figura 1 representa a decisão de produzir em j+1 e atender as demandas dos períodos j+1 até k, isto é, $x_{j+1} = d_{j+1} + \ldots + d_k$. O problema (P) se reduz então a achar o caminho de custo mínimo no grafo acíclico orientado representado na figura 1 para T = 4. Este problema será resolvido através da equação da programação dinâmica com recursividade para frente.

Seja F_j o custo da solução ótima (ou o caminho de custo mínimo do nó 0 ao nó j), dado $I_j = 0$. Então

$$F_k = \min_{0 \le j \le k-1} \left[F_j + M_{jk} \right] \qquad k = 1,\ldots,T \tag{1.2}$$

$$F_0 = 0$$

A equação da programação dinâmica (1.2) constitue o algoritmo de Wagner-Whitin.

REFERÊNCIAS BIBLIOGRÁFICAS

1. Florian, M., Lenstra, J.K. and Rinnooy Kan, A.H.G., (1980), *Deterministic Production Planning: Algorithms and Complexity*, Management Science 26(7), 669-679.

2. Maes, J., McClain, J.O., Van Wassenhove, L.N., (1991), *Multilevel capacitated lotsizing complexity and LP-based heuristics*, European Journal of Operational Research 53, 131-148.

3. Barany, I., Van Roy, T.J. and Wolsey, L.A., (1984), *Strong Formulations for Multi-Item Capacitated Lot Sizing*, Management Science 30(10), 1255-1261.

4. Eppen, G.D. and Martin, R.K., (1987), *Solving Multi-Item Capacitated Lot-Sizing Problems using Variable Redefinition*, Operations Research 35(6), 832-848.

5. Maes, J. and Van Wassenhove, L., (1988), *Multi-Item Single-Level Capacitated Dynamic Lot-Sizing Heuristics: A General Review*, Journal of Operational Research Society 39(11), 991-1004.

6. Lambrecht, M.R., Vanderveken, H., (1979), *Heuristic Procedures for the Single Operation, Multi-Item Loading Problem*, AIIE Transactions 11(4), 319-326.

7. Dixon, P., Silver, E.A., (1981), *A Heuristic Solution Procedure for the Multi-Item, Single-Level, Limited Capacity, Lot-Sizing Problem*, Journal of Operations Management 2(1), 23-39.

8. Maes, J. and Van Wassenhove, L.N., (1986), *Multi Item Single Level Capacitated Dynamic Lotsizing Heuristics: A Computational Comparison (Part I: Static Case)*, IIE Transactions 18(2), 114-123.

9. Dogramaci, A., Panayiotopoulos, J.C., Adam, N.R., (1981), *The Dynamic Lot-Sizing Problem for Multiple Items under Limited Capacity*, AIIE Transactions 13(4), 294-303.

10. Karni, R., Roll, Y., (1982), *A Heuristic Algorithm for the Multi-Item Lot-Sizing Problem with Capacity Constraints*, IIE Transactions 14(4), 249-256.

11. Cattrysse, D., Maes, J., Van Wassenhove, L.N., (1990), *Set partitioning and column generation heuristics for capacitated dynamic lotsizing*, European Journal of Operational Research 46, 38-47.

12. Souza, K.X.S., Armentano, V.A., *Multi-Item Lot Sizing by the Cross Decomposition Method*, Alio-Euro Workshop on Practical Combinatorial Optimization, Rio de Janeiro, 1989.

13. Van Roy, T.J., (1983), *Cross Decomposition for Mixed Integer Programming*, Mathematical Programming 25, 46-63.

14. Trigeiro, W.W., Thomas, L.J. and McClain, J.O., (1989), *Capacitated Lot Sizing with Setup Times*, Management Science 35(3), 353-366.

15. Wagner, H.M. and Whitin, T.M., (1958), *Dynamic Version of the Economic Lot Size Model*, Management Science 5(1), 89-96.

16. Lozano, S., Larraneta, J. and Onieva, L., (1991), *Primal-dual approach to the single level capacitated lot-sizing problem*, European Journal of Operational Research 51, 354-366.

17. Bahl, H.C., Ritzman, L.P., Gupta, J.N.D., (1987), *Determining Lot Sizes and Resource Requirements: A Review*, Operations Research 35(3), 329-345.

18. Camerini, P.N., Fratta, L. and Maffioli, F., (1975), *On Improving Relaxation Methods by Modified Gradient Techniques*, Mathematical Programming Study 3, 26-34.

19. Billington, P.J., McClain, J.O. and Thomas, L.J., (1983), *Mathematical Programming Approaches to Capacity-Constrained MRP Systems: Review, Formulation and Problem Reduction*, Management Science 29(10), 1126-1141.

20. **Johnson, L.A., Montgomery, D.C., (1974), *Operations Research in Production Planning, Scheduling, and Inventory Control*, John Wiley & Sons, New York.**

21. **Glover, F., (1977), *Heuristics for Integer Programming Using Surrogate Constraints*, Decision Sciences 8, 156-166.**

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

23. Glover, F. (1989), *Tabu Search - Part I*, ORSA Journal on Computing 1(3), 190-206.

24. Glover, F., (1990), *Tabu Search - Part II*, ORSA Journal on Computing 2(1), 4-32.

25. Glover, F., (1990), *Tabu Search: A Tutorial*, Technical Report, Center for Applied Artificial Intelligence, University of Colorado, Boulder, CO.

26. Müller, F.M. (1990), *Busca Tabu na Solução de Problemas de Programação Zero-Um*, Tese de Mestrado, Faculdade de Engenharia Elétrica, UNICAMP.

27. Pureza, V.M.M. (1990), *Problemas de Roteamento de Veículos Via Metaheurística Tabu*, Tese de Mestrado, Faculdade de Engenharia Elétrica, UNICAMP.

28. Souza, K.X.S., (1989), *Planejamento da Produção de Múltiplos Itens com Restrições de Capacidade através da Decomposição Cruzada*, Tese de Mestrado, Faculdade de Engenharia Elétrica, UNICAMP.