

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

UM MODELO DE REDE GENERALIZADA PARA O PROBLEMA DE DIMENSIONAMENTO DE LOTES E RESOLUÇÃO ATRAVÉS DE UM ALGORITMO "BRANCH-AND-BOUND"

FRANKLINA MARIA BRAGION DE TOLEDO

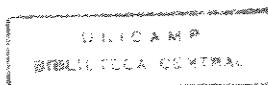
Este exemplar corresponde à redação final da tese
defendida por FRANKLINA MARIA BRAGION DE
TOLEDO
Julgada em 03.03.94
Vinicius A. Armentano
Orientador

Orientador:

Prof. Dr. Vinicius Amaral Armentano

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

- MARÇO 1994 -



À minha família

AGRADECIMENTOS

A todos que colaboraram na realização deste trabalho e em especial

- a Deus pela vida e pelos momentos em que só vi um par de pegadas na areia
- ao Professor Vinicius A. Armentano pela orientação dedicada e acessível durante o desenvolvimento desta tese e estímulo à pesquisa
- ao Professor Paulo M. França pela constante atenção e pelas importantes sugestões ao longo deste trabalho
- ao Professor Paulo de Barros Correia pela valiosa participação em etapas desta pesquisa
- aos professores e funcionários do Densis, pelo apoio
- aos funcionários da CPG pelo apoio e informações
- ao DT
- aos meus pais Waldemar e Esther pelo constante interesse e carinho
- ao meu irmão pelos conselhos e pelo carinho
- à minha família pelo constante interesse
- a Zake Neto pelas indispensáveis discussões sobre este trabalho
- à Sandra pelo apoio com o código de grafos generalizados
- à Sandra e Silvana pelas valiosas discussões sobre grafos
- à Tití pela grande mãozinha com o Trigeiro e a Regina pela ajuda na caça aos erros
- à Debora, Regina, Cintia e Lúcia pelas valiosas discussões sobre este trabalho
- ao Walcir por todas as dicas sobre computadores e ao Miguel pelo apoio no DT
- ao meu amigo coruja, Mauro, pela companhia nos horários pouco convencionais
- a todos os amigos pelos bons momentos e pela companhia
- a toda família Terra Nova pela presença mais que necessária
- a CAPES pelo apoio financeiro

Resumo

O problema de dimensionamento de lotes abordado neste trabalho consiste em planejar a produção de múltiplos itens em uma única máquina ao longo de um horizonte de tempo finito. O objetivo do problema é determinar um plano de produção de mínimo custo, que seja capaz de atender à demanda pelos itens, sem violar as restrições de capacidade de máquina. O modelo considerado inclui custos e tempos de preparação de máquina e é representado como um problema de fluxo de custo mínimo em uma rede generalizada. Um método ótimo baseado na técnica *branch-and-bound* é utilizado para sua resolução.

Abstract

The lot-sizing problem considered in this work consists in planning the production of multiple items in a single machine over a finite horizon. The objective of the problem is to determine a minimum cost production plan that meets forecast demand for the items. The model considered here includes setup time and setup cost, and is represented as a minimum cost flow in a generalized network. A branch-and-bound method is developed for solving this problem.

ÍNDICE

| | |
|------------------|---|
| INTRODUÇÃO | 1 |
|------------------|---|

CAPÍTULO 1

FORMULAÇÃO DO PROBLEMA

| | |
|----------------------|---|
| 1.1 Formulação | 4 |
|----------------------|---|

| | |
|---------------------------------|---|
| 1.2 Revisão Bibliográfica | 7 |
|---------------------------------|---|

CAPÍTULO 2

REPRESENTAÇÃO DO PROBLEMA ATRAVÉS DE UMA REDE

| | |
|----------------------|----|
| 2.1 Introdução | 12 |
|----------------------|----|

| | |
|------------------|----|
| 2.2 Modelo | 13 |
|------------------|----|

CAPÍTULO 3

UM ALGORITMO *BRANCH-AND-BOUND*

| | |
|----------------------|----|
| 3.1 Introdução | 19 |
|----------------------|----|

| | |
|--|----|
| 3.2 O Método <i>Branch-and-Bound</i> | 20 |
|--|----|

| | |
|---------------------|----|
| 3.3 Relaxação | 22 |
|---------------------|----|

| | |
|--------------------------|----|
| 3.4 Seleção de Nós | 30 |
|--------------------------|----|

| | |
|--|----|
| 3.5 Seleção da Variável de Ramificação | 31 |
|--|----|

| | |
|---------------------------|----|
| 3.6 Solução Inicial | 33 |
|---------------------------|----|

| | |
|------------------------------------|----|
| 3.7 Pré-Fixação de Variáveis | 33 |
| 3.8 O Algoritmo Proposto | 35 |

CAPÍTULO 4

TESTES COMPUTACIONAIS E ANÁLISE DE RESULTADOS

| | |
|---|----|
| 4.1 Introdução | 38 |
| 4.2 Características dos Problemas Gerados | 39 |
| 4.3 Resultados dos Testes | 42 |
| 4.3.1 Resultados e Análise Grupo1 | 42 |
| 4.3.2 Resultados e Análise Grupo2 | 50 |
| 4.3.3 Resultados e Análise Grupo3 | 56 |

CAPÍTULO 5

| | |
|-------------------------|----|
| CONCLUSÕES | 59 |
|-------------------------|----|

| | |
|---|----|
| REFERÊNCIAS BIBLIOGRÁFICAS | 61 |
|---|----|

À minha família

INTRODUÇÃO

Uma empresa de manufatura, de uma maneira geral, deve executar duas tarefas: produzir bens e comercializá-los. A tarefa de produzir depende da disponibilidade de matéria-prima, mão-de-obra, equipamentos e instalações. Já a comercialização dos bens produzidos está vinculada à estratégia de marketing da empresa, que envolve, de maneira geral, a pesquisa de mercado, a distribuição dos produtos, promoções e vendas. Entretanto, para realizar estas tarefas é necessário o investimento de capital, que deve ser obtido no mercado financeiro.

Uma empresa pode ser dividida em três setores fundamentais: **finanças**, **marketing** e **produção**. Os setores de **finanças** e **marketing** estão diretamente ligados às necessidades e resultados do setor de **produção**. Assim, o setor de **produção** merece uma atenção especial quanto ao seu gerenciamento que é responsável pelo planejamento e controle da produção.

Segundo Ackoff (1974) em [Oliveira, 1986], existem três tipos de filosofias de planejamento: filosofia da **satisfação**, cujo objetivo é produzir “suficientemente bem”; filosofia de **otimização**, que visa produzir da melhor maneira possível; e a filosofia da **adaptação** que busca o ajuste da empresa à mudanças de mercado, aos avanços tecnológicos e ao aumento na concorrência.

Ainda quanto ao planejamento, este pode, segundo [Oliveira, 1986], ser subdividido em três tipos: planejamento **estratégico**, planejamento **tático** e planejamento **operacional**. Esses três tipos de planejamento podem ser relacionados aos níveis de tomada de decisão numa “pirâmide organizacional”, como mostra a Figura 1 [Oliveira, 1986].

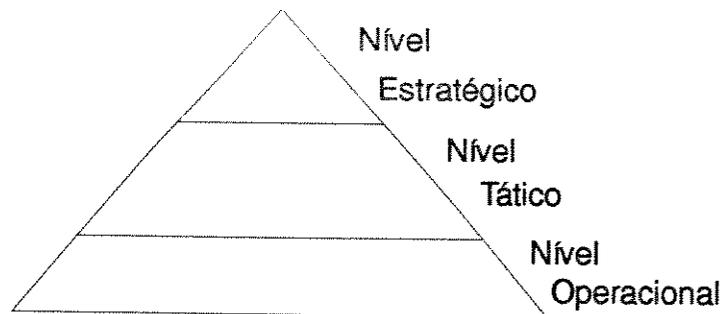


Figura 1 - Tipos de Planejamento e Níveis de Decisão.

O planejamento **estratégico** está relacionado ao mais alto nível de tomada de decisões, das quais participam os setores de produção, marketing e finanças. Neste nível são definidas as metas globais da empresa e as políticas adequadas para atingí-las. Estas decisões determinam os objetivos da empresa a longo prazo.

O planejamento **tático** é responsável pela utilização eficiente dos recursos disponíveis, a fim de cumprir os objetivos determinados no planejamento estratégico. Nesta etapa devem ser tomadas as decisões de médio prazo, que afetam áreas da produção. Por exemplo, no planejamento tático são tomadas decisões como: quanto e quando produzir, quais os níveis de estoque, a alocação de mão-de-obra e de equipamentos, e o transporte de mercadorias.

O último tipo de planejamento, o planejamento **operacional**, está relacionado ao dia-a-dia da empresa, ou seja, neste nível são tomadas as decisões de curto prazo, cujo objetivo é definir e executar programas detalhados de produção.

O trabalho desenvolvido nesta tese se concentra nas tomadas de decisões de médio prazo - portanto no planejamento **tático** da produção - e segue a filosofia de **otimização**, buscando a melhor maneira possível de produzir. Uma vez definido o horizonte de planejamento de médio prazo, este pode ser subdividido em períodos

menores, cuja demanda de cada um dos produtos e a quantidade de recursos disponíveis em cada um dos períodos é préestabelecida pelo planejamento tático. A partir destas informações o planejamento tático deve obter um plano de produção capaz de realizar tais determinações, resolvendo assim o problema de quanto e quando produzir e quais os níveis de estoques de produção, ou, em outras palavras, resolver um problema de dimensionamento de lotes.

O objetivo deste trabalho é determinar um plano de produção de mínimo custo, que seja capaz de resolver o problema de dimensionamento de lotes de múltiplos itens em uma máquina, ao longo de um horizonte de tempo composto de um número finito de períodos, considerando a disponibilidade de recursos.

O custo total de um plano de produção envolve o custo de preparação de máquina (*setup cost*), o custo variável de produção e o custo de estoque em cada período. A fim de tornar o modelo estudado mais real foi também considerado o tempo gasto na preparação da máquina (*setup time*). A existência dos custos e tempos de preparação de máquina faz com que matematicamente este problema seja formulado como um problema de programação inteira com variáveis binárias (0-1), representando as situações de produzir ou não produzir um item em um dado período.

Neste trabalho é proposta a representação do modelo como uma rede generalizada e um método ótimo para resolvê-lo foi desenvolvido. O método ótimo é um algoritmo *branch-and-bound* especializado, onde em cada nó é resolvido um problema de fluxo mínimo em uma rede generalizada. Devido à complexidade do problema não é possível aplicar o algoritmo a problemas de grande porte, que devem ser resolvidos de maneira heurística. Este trabalho pode ser classificado como uma pesquisa de base, uma vez que sua principal contribuição é permitir a avaliação de métodos heurísticos, os quais podem ser aplicados a problemas de grande porte, como a maioria dos casos reais.

Assim, o capítulo 1 trata da formulação matemática do problema e da apresentação de uma revisão bibliográfica. No capítulo 2, é exposta a representação do problema através de uma rede generalizada. O algoritmo desenvolvido e as estratégias utilizadas se encontram detalhadas no capítulo 3. No capítulo 4 são apresentados os testes computacionais realizados e uma análise dos resultados obtidos e, no capítulo 5, são apresentadas as conclusões e sugestões para pesquisas futuras.

CAPÍTULO 1

FORMULAÇÃO DO PROBLEMA

1.1 Formulação

O problema de dimensionamento de lotes estudado consiste em planejar a produção de N itens em um horizonte de T períodos. O objetivo do problema é determinar um plano de produção de mínimo custo, que seja capaz de atender à demanda pelos produtos, sem violar as restrições de capacidade da máquina.

O custo total de um plano de produção envolve o custo de preparação de máquina (*setup cost*), o custo variável de produção e o custo de estoque em cada período. O custo de preparação de máquina é contabilizado sempre que é planejada a produção de um item i em um período t . Ele independe da quantidade produzida e por isso é chamado de custo fixo de produção, enquanto os custos variáveis de produção e de estoque são, em geral, considerados lineares em relação a quantidade produzida e estocada, respectivamente.

O tempo gasto na produção de um item i no período t é composto de dois elementos: o tempo de processamento, que é proporcional à quantidade do item produzida no período, e o tempo fixo de preparação da máquina (*setup time*). O tempo consumido, na produção dos itens em um período t não deve exceder a capacidade de máquina, que representa o tempo disponível para a produção naquele período.

Os tempos e custos de preparação de máquina induzem à formulação do problema utilizando variáveis binárias (0-1), que representam as situações de **produzir** ou **não produzir** um dado item i em um período t . Tem-se, então, um problema de programação inteira mista, que pode ser formulado matematicamente como:

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

$$\text{s.a. } I_{i,t-1} + x_{it} - I_{it} = d_{it} \quad i = 1, \dots, N; \quad t = 1, \dots, T \quad (2)$$

$$\sum_{i=1}^N (b_{it} x_{it} + f_{it} y_{it}) \leq C_t \quad t = 1, \dots, T \quad (3)$$

$$b_{it} x_{it} \leq \Delta_{it} y_{it} \quad i = 1, \dots, N; \quad t = 1, \dots, T \quad (4)$$

$$\Delta_{it} = \min \left\{ C_t - f_{it}, \sum_{k=t}^T b_{ik} d_{ik} \right\} \quad i = 1, \dots, N; \quad t = 1, \dots, T \quad (4')$$

$$I_{it} \geq 0, \quad x_{it} \geq 0, \quad y_{it} \in \{0, 1\} \quad i = 1, \dots, N; \quad t = 1, \dots, T \quad (5)$$

$$I_{i0} = I_{iT} = 0 \quad i = 1, \dots, N \quad (6)$$

onde:

- N - número de itens;
- T - número de períodos;
- s_{it} - custo fixo de produção do item i no período t ;
- c_{it} - custo unitário de produção do item i no período t ;
- h_{it} - custo unitário de estoque do item i no período t ;
- d_{it} - demanda do item i no período t ;
- b_{it} - tempo de processamento unitário do item i no período t ;

- f_{it} - tempo fixo de preparação da máquina para produzir o item i no período t ;
- C_t - capacidade da máquina disponível no período t (em unidades de tempo);
- I_{it} - quantidade em estoque do item i no final do período t (variável);
- x_{it} - quantidade produzida do item i no período t (variável);
- y_{it} - 1 se o item i é produzido no período t , e 0 caso contrário (variável).

No modelo (PDL) a função objetivo (1) minimiza a soma do custo variável de produção, do custo de preparação de máquina e do custo de estoque em cada um dos períodos. A equação (2) garante o atendimento da demanda de cada um dos itens em todos os períodos do horizonte de planejamento. A restrição (3) garante que o tempo gasto na produção de itens em um período não deve exceder a capacidade da máquina. O custo e o tempo de preparação devem ser contabilizados toda vez que x_{it} for positivo; esta restrição é imposta em (4). Através de (6), os estoques iniciais e finais de todos os produtos foram considerados nulos sem perda de generalidade. Isto porque, se o estoque inicial for diferente de zero ele pode ser abatido das demandas dos primeiros períodos até que se torne nulo [Johnson e Montgomery, 1974]. Quanto ao estoque final, é fácil ver que na solução ótima ele será nulo, uma vez que o problema visa minimizar custos, ou seja, se em uma solução S qualquer algum $I_{iT} > 0$, então tem-se que

$\sum_{t=1}^T x_{it} > \sum_{t=1}^T d_{it}$, logo existe uma outra solução S' de custo inferior em que

$\sum_{t=1}^T x_{it} = \sum_{t=1}^T d_{it}$ [Zangwill, 1966]. Os níveis de produção não-negativos são assegurados

pela restrição (5), onde também é caracterizado o atendimento da demanda sem atraso, uma vez que estoques negativos não são permitidos.

Em (4') é definido um limitante superior para a produção do item i no período t (x_{it}). Como foi visto anteriormente o estoque final do item i no período t é nulo, logo

pode-se concluir que x_{it} será no máximo $\sum_{k=t}^T d_{ik}$, utilizando para sua produção

$\sum_{k=t}^T b_{ik} d_{ik}$ unidades de tempo. Além disso, x_{it} é limitado pela quantidade de recurso

disponível $C_t - f_{it}$, que é a capacidade da máquina menos o tempo necessário para a preparação da máquina para produzir o item i .

1.2 Revisão Bibliográfica

Devido à sua importância prática, o problema de dimensionamento de lotes vem sendo objeto de pesquisa há muitos anos. Este problema já foi focado de várias maneiras e vários métodos de resolução ótimos e heurísticos foram apresentados. Nesta revisão, no entanto, serão enfatizados os trabalhos que apresentaram algoritmos ótimos de resolução para o problema estudado e para suas particularizações. Em especial será destacada a utilidade da resolução através de redes. Todos os trabalhos aqui relacionados consideram em sua formulação o custo de preparação de máquina, com exceção do trabalho de Bowman (1956). A fim de facilitar a apresentação, os trabalhos serão subdivididos nos seguintes grupos:

- um único item não capacitado;
- um único item capacitado;
- múltiplos itens com custo de preparação nulo;
- múltiplos itens com custo de preparação não nulo.

■ Um único item não capacitado

O primeiro método ótimo para o problema de planejamento da produção de um único item foi proposto em [Wagner e Within, 1958]. Os autores apresentaram um algoritmo de programação dinâmica para resolver o problema. Este algoritmo consiste em encontrar o caminho de custo mínimo em um grafo direcionado. O algoritmo proposto é bastante eficiente e tem complexidade polinomial.

No entanto, Zangwill (1969) foi o primeiro autor a ressaltar a importância do uso de redes na representação de alguns problemas de planejamento da produção. Em [Zangwill, 1969], o problema de dimensionamento de lotes foi considerado como um problema de fluxo de custo mínimo numa rede com custo côncavo nos arcos e uma única fonte. A existência de custo fixo de produção permite concluir que a função objetivo é côncava. Como as restrições do problema são lineares sabe-se que seu mínimo ocorre em um ponto extremo [Lumberger, 1984]. O autor demonstrou que os pontos extremos têm a propriedade de que para cada nó existe no máximo um fluxo incidente

positivo [Zangwill, 1966]. Esta propriedade facilitou o desenvolvimento de um algoritmo de programação dinâmica eficiente capaz de calcular um plano ótimo de produção permitindo o atraso no atendimento da demanda (*backlogging*). A Figura 2 ilustra a representação proposta, onde I_t^+ representa o estoque final no período t , I_t^- simboliza a demanda acumulada não atendida no fim do período t , x_t é a quantidade produzida no período t , e d_t é a demanda do item no período t . Quando o atraso no atendimento da demanda não é permitido devem ser desconsideradas as variáveis I_t^- e na Figura 2 devem ser retirados os arcos relativos a essas variáveis.

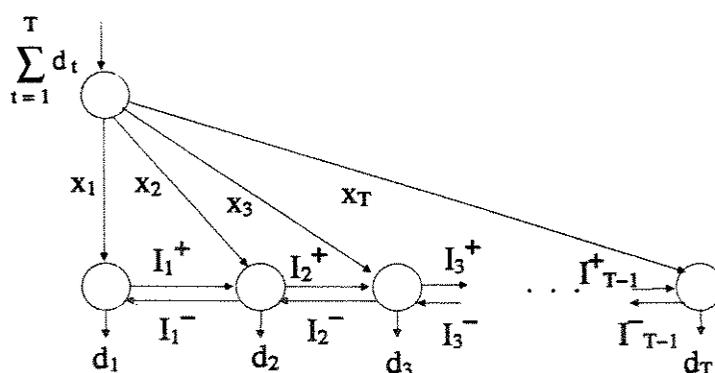


Figura 2 - Rede Proposta por Zangwill

A propriedade de ponto extremo demonstrada por Zangwill mostra que para a rede da Figura 2 apenas uma das variáveis x_t , I_t^+ e I_t^- pode ser positiva. Quando o atraso no atendimento da demanda não é permitido tem-se a propriedade devida a [Wagner e Within, 1958], $I_{t-1} x_t = 0$.

Para o problema de um único item não capacitado muitas heurísticas foram propostas e para uma revisão sugere-se [Ritchie e Tsado, 1986].

■ Um único item capacitado

Em [Bowman, 1956] foi considerada a existência de várias fontes produtoras, enquanto o custo fixo de produção foi desprezado. O objetivo é encontrar o plano de produção com menor custo total, capaz de atender à demanda respeitando os limites de capacidade das fontes. Através de manipulações algébricas foi obtida uma representação

como um problema de transportes, o qual é um caso especial de fluxo de custo mínimo em uma rede.

Em [Florian e Klein, 1974] foi abordado o problema de dimensionamento de lotes com limitação para a produção. Os autores obtiveram a caracterização de pontos extremos e a partir dela foi desenvolvido um algoritmo de programação dinâmica para resolver o problema. O algoritmo considera a capacidade constante ao longo do horizonte de planejamento. Esta caracterização pode também ser obtida através de redes [Souza, 1989].

Uma generalização do problema de Florian e Klein (1974) foi proposta por Love (1974), onde além da produção, a quantidade em estoque também é limitada. Love utilizou conceitos de fluxos em redes para caracterizar os pontos extremos do problema. Devido à maior generalidade do modelo, segundo Florian e Klein (1974), apenas uma representação parcial dos pontos extremos foi obtida.

Um estudo sobre a complexidade deste problema foi apresentado em [Florian et al., 1980], onde foi provado que encontrar a solução ótima para o problema com um único item pertence à classe NP-hard. Esta classe contém os problemas de otimização ditos intratáveis, isto é, problemas para os quais não existe um algoritmo capaz de achar uma solução ótima em tempo polinomial. Para uma definição mais detalhada e completa ver [Garey e Johnson, 1979].

■ Múltiplos itens com tempo de preparação de máquina nulo

Em [Barany et al., 1984] foi proposto um algoritmo ótimo baseado em programação matemática para resolver o problema com múltiplos itens. O algoritmo apresentado pode ser subdividido em duas fases: primeiro aplica-se um algoritmo de planos de cortes com o objetivo de obter uma boa aproximação da envoltória convexa das soluções factíveis e a seguir, um algoritmo *branch-and-bound*. Neste trabalho foram apresentados testes com no máximo 20 itens e 13 períodos. O primeiro autor a propor um algoritmo exato baseado em redes para o problema foi Evans (1985). Neste trabalho foi proposta uma nova técnica de modelagem para o problema, conceitualmente simples e única com respeito a trabalhos anteriores. O método de resolução está baseado em um algoritmo *branch-and-bound*, onde em cada nó um problema de transportes é resolvido. Este

algoritmo foi aplicado a problemas de pequeno porte, máximo de 6 períodos e 8 produtos, para a obtenção de soluções ótimas e em problemas de médio e grande porte foi utilizado de maneira heurística. Já em [Eppen e Martin, 1987] foi utilizada uma técnica de redefinição de variáveis, que foi aplicada a problemas de até 200 itens e 10 períodos. Muitos procedimentos heurísticos também foram desenvolvidos e uma revisão destes pode ser encontrada em [Maes e Wassenhove, 1988].

■ Múltiplos itens com tempo de preparação de máquina não nulo

Em muitos casos reais, a preparação da máquina para a produção de um item consome uma significativa quantidade de tempo. Durante este intervalo de tempo a máquina permanece parada, e portanto a capacidade de produção é reduzida. Este novo fator complicante redirecionou as pesquisas mais recentes, nas quais o tempo de preparação de máquina passou a ser considerado na formulação do problema de dimensionamento de lotes. Tal problema é de difícil solução pertencendo à classe NP [Garey e Johnson, 1979], isto é, não existe um algoritmo polinomial que determine a existência ou não de uma solução factível para o problema.

Em [Souza, 1989] o problema é resolvido pelo algoritmo ótimo de decomposição cruzada [Van Roy, 1983]. O método de decomposição cruzada [Van Roy, 1983] gera subproblemas que são transformados, através de manipulações algébricas, em problemas de fluxo em redes a custo mínimo. Este algoritmo, no entanto, foi testado apenas para problemas de pequeno porte, 15 produtos e 18 períodos.

Recentemente Diaby et al. (1992a) desenvolveram alguns métodos ótimos/quase ótimos para resolver este problema. O procedimento principal é um algoritmo *branch-and-bound*, no qual os limitantes (*bounds*) são gerados pela Relaxação Lagrangeana e otimização por subgradiente. Duas relaxações lagrangeanas foram utilizadas: das restrições de capacidade e das restrições de demanda. Quando é utilizada a relaxação da capacidade o problema lagrangeano é decomponível em um subproblema para cada item que pode ser resolvido pelo algoritmo de Wagner-Within. Para a relaxação da demanda o problema lagrangeano pode ser resolvido como um problema da mochila. Nos nós terminais da árvore *branch-and-bound*, retidas as decisões de onde produzir, o problema é reformulado como um problema de transportes, a partir do qual tenta-se obter uma solução factível. Este trabalho foi testado para problemas com 8 produtos e 8 períodos, para os quais

foram obtidos bons resultados. Outros testes foram realizados para problemas de maior porte, 50 itens e 12 períodos e 99 itens e 8 períodos, onde porém foi desconsiderado o custo de preparação de máquina.

Os dois trabalhos citados anteriormente consideram dois tipos de recurso: hora regular e hora extra. O custo da utilização destes recursos é incluído na função objetivo.

Devido à dificuldade deste problema, várias heurísticas foram propostas para sua resolução, das quais podem ser destacadas as apresentadas por Trigeiro et al. (1989), Lozano et al. (1991) e Scrich (1992).

Trigeiro et al. (1989) apresentaram um método heurístico, onde é construída a Relaxação Lagrangeana em relação às restrições de capacidade, o que permite decompor o problema em subproblemas - um para cada item. Os subproblemas são resolvidos pelo algoritmo de Wagner e Within e a função dual é otimizada pelo método do subgradiente. Em seguida, caso a solução obtida seja infactível, é aplicada uma heurística, baseada em transferência de produção entre os períodos, na tentativa de factibilizar a solução.

Uma heurística semelhante à apresentada por Trigeiro foi proposta em [Lozano et al., 1991], sendo que as diferenças entre os dois métodos situam-se nos algoritmos usados para maximizar a função dual e nas heurísticas utilizadas para tentar obter soluções factíveis. Scrich (1992) desenvolveu recentemente um algoritmo heurístico, baseado em Trigeiro, utilizando a técnica de Busca Tabu.

Mais recentemente foi publicado um trabalho em [Diaby, et al., 1992b] onde foi proposto um procedimento heurístico. Como em [Diaby, et al., 1992a] é feita a relaxação lagrangeana das restrições de capacidade. A cada número de iterações por subgradiente, definido pelos autores, é determinado onde cada item é produzido, isto é, os valores das variáveis binárias são determinados. A seguir é resolvido um problema de transporte, e se a solução obtida for infactível ou inconsistente com relação a algum valor das variáveis binárias definidas anteriormente, então perturba-se as quantidades produzidas a fim de obter uma solução factível e/ou consistente. Foram apresentados testes para problemas de grande porte com custo de preparação nulos - 5000 itens e 30 períodos - que são solucionados em tempos bastante baixos, obtendo resultados quase ótimos.

CAPÍTULO 2

REPRESENTAÇÃO DO PROBLEMA

ATRAVÉS DE UMA REDE

2.1 Introdução

Este capítulo propõe a representação do problema de dimensionamento de lotes através de uma rede. Primeiro, considera-se o modelo apresentado no Capítulo 1. Em seguida, é construído um modelo equivalente, que será representado como uma rede.

2.2 Modelo

O problema de dimensionamento de lotes estudado neste trabalho foi formulado no Capítulo 1 pelo modelo (PDL). Essa formulação é aqui repetida para facilitar o acompanhamento da leitura.

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

$$\text{s.a. } I_{i,t-1} + x_{it} - I_{it} = d_{it} \quad i = 1, \dots, N; \quad t = 1, \dots, T \quad (2)$$

$$\sum_{i=1}^N (b_{it} x_{it} + f_{it} y_{it}) \leq C_t \quad t = 1, \dots, T \quad (3)$$

$$b_{it} x_{it} \leq \Delta_{it} y_{it} \quad i = 1, \dots, N; \quad t = 1, \dots, T \quad (4)$$

$$\Delta_{it} = \min \left\{ C_t - f_{it}, \sum_{k=t}^T b_{ik} d_{ik} \right\} \quad i = 1, \dots, N; \quad t = 1, \dots, T$$

$$I_{it} \geq 0, \quad x_{it} \geq 0, \quad y_{it} \in \{0, 1\} \quad i = 1, \dots, N; \quad t = 1, \dots, T \quad (5)$$

$$I_{i0} = I_{iT} = 0 \quad i = 1, \dots, N \quad (6)$$

O modelo (PDL) é um problema de programação inteira mista e constitui a base para o desenvolvimento de uma representação através de uma rede com variáveis binárias (0-1).

A restrição (2) pode ser vista como uma equação de conservação de fluxo no período t em um nó de produto i (P_i), onde a produção x_{it} e o estoque disponível $I_{i,t-1}$ são arcos que incidem sobre o nó, enquanto a demanda d_{it} e a quantidade que permanecerá em estoque I_{it} deixam o nó (Figura 3). Neste trabalho os termos item e produto terão o mesmo significado.

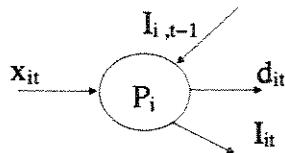


Figura 3 - Representação da Restrição (2)

A limitação do recurso é representada pela restrição (3). Um arco capacitado com custo zero ligado a um nó artificial, chamado de nó fonte (F), foi utilizado para representar esta desigualdade na rede. Este nó gera um fluxo em unidades de tempo cujo limitante superior é a soma das capacidades nos períodos. O nó F é então conectado a um nó de máquina (M) por um arco, cujo limitante superior é dado pela capacidade da máquina no período t (C_t). O fluxo que incide sobre M deve, segundo a desigualdade (3), suprir o tempo de produção e o tempo de preparação. O tempo gasto na preparação da máquina para produzir um dado item no período t é representado por uma raiz que deixa o nó de máquina, cujo fluxo é igual a f_{it} .

A equação

$$\rho_{it} = b_{it} x_{it} \quad i = 1, \dots, N; \quad t = 1, \dots, T \quad (7)$$

foi acrescentada ao modelo para permitir a união do nó de máquina ao nó de produto, convertendo o fluxo que deixa o nó de máquina (ρ_{it}), em unidades de tempo, e incide sobre o nó de produto (x_{it}), em unidades de produto.

Finalmente, pode-se representar (3) utilizando o conjunto de nós e arcos da Figura 4.

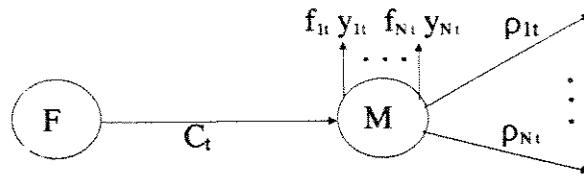


Figura 4 - Representação da Restrição (3)

A conversão do fluxo é caracterizada adicionando-se mais um parâmetro ao arco. Este parâmetro é chamado de ganho do arco (w). Na Figura 5 estão ilustradas as três situações que o ganho pode representar.

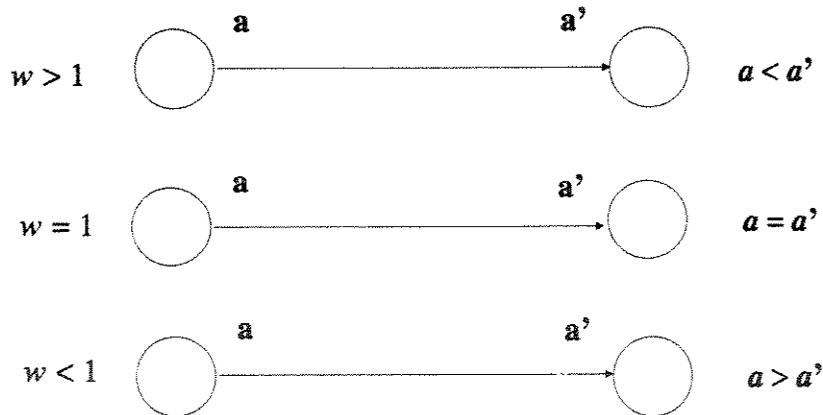


Figura 5 - Representação do Ganho

Por (7) tem-se que:

$$x_{it} = \frac{1}{b_{it}} \rho_{it} .$$

Logo o ganho no arco é definido por:

$$w_{it} = \frac{1}{b_{it}} .$$

O ganho traduz a conversão de unidade de tempo para produto, ou seja, a relação do quanto se pode produzir em uma unidade de tempo. Quando $w_{it} > 1$ pode-se produzir mais de um item i no período t em uma unidade de tempo. Se $w_{it} = 1$ pode-se produzir exatamente uma unidade do item i no período t em uma unidade de tempo, e para $w_{it} < 1$ é necessário utilizar mais de uma unidade de tempo para produzir uma unidade do item i no período t .

A existência de arcos com ganho caracteriza a rede que representa o modelo (PDL) como uma rede generalizada. A terminologia *rede generalizada*, segundo [Jensen e Barnes, 1980], é usada para indicar que existe ao menos um arco com ganho diferente de 1 fazendo parte da rede.

O modelo (PDL) é substituído pelo modelo equivalente (G), que o caracteriza como uma *rede generalizada*.

$$(G) \text{ Min } \sum_{i=1}^N \sum_{t=1}^T (s_{it} y_{it} + \frac{c_{it}}{b_{it}} \rho_{it} + h_{it} I_{it}) \quad (8)$$

$$\text{s.a. } I_{i,t-1} + \frac{\rho_{it}}{b_{it}} - I_{it} = d_{it} \quad i = 1, \dots, N; \quad t = 1, \dots, T \quad (9)$$

$$\sum_{i=1}^N (\rho_{it} + f_{it} y_{it}) \leq C_t \quad t = 1, \dots, T \quad (10)$$

$$w_{it} = \frac{1}{b_{it}} \quad i = 1, \dots, N; \quad t = 1, \dots, T \quad (11)$$

$$\rho_{it} \leq \Delta_{it} y_{it} \quad i = 1, \dots, N; \quad t = 1, \dots, T \quad (12)$$

$$\Delta_{it} = \min \left\{ C_t - f_{it}, \sum_{k=1}^T b_{ik} d_{ik} \right\} \quad i = 1, \dots, N; \quad t = 1, \dots, T$$

$$I_{it} \geq 0, \quad \rho_{it} \geq 0, \quad y_{it} \in \{0, 1\} \quad i = 1, \dots, N; \quad t = 1, \dots, T \quad (13)$$

$$I_{i0} = I_{iT} = 0 \quad i = 1, \dots, N \quad (14)$$

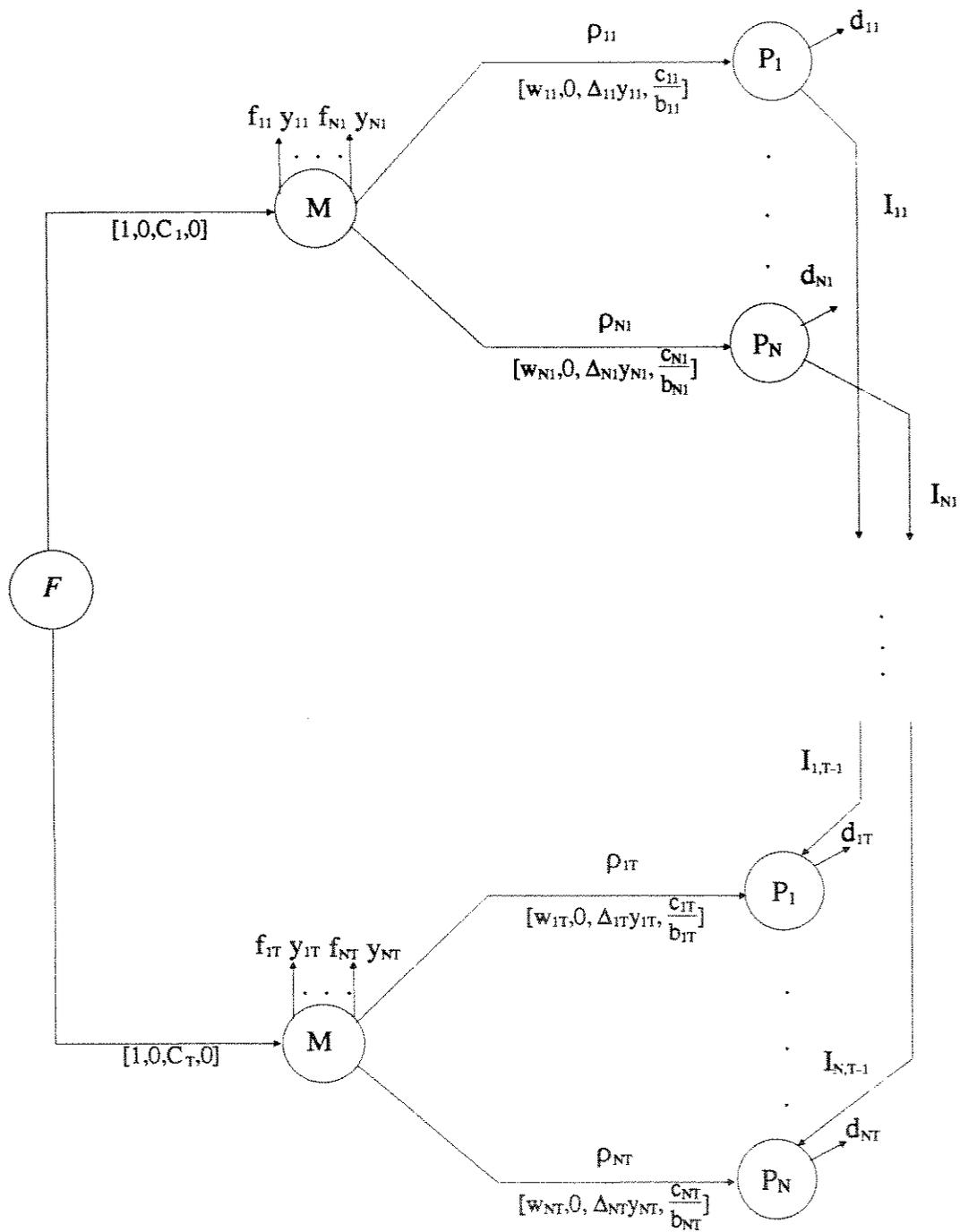


Figura 6 - Rede do Modelo (G)

Obtém-se a rede final para representação do modelo (G) unindo todos os nós e arcos citados anteriormente. Aos arcos são atribuídos quatro parâmetros: o ganho, o limite inferior, o limite superior e o custo. Esses parâmetros estão associados ao fluxo que sai do nó e são definidos a partir do modelo (G). O ganho é igual a 1 para todos os arcos exceto para aqueles que ligam os nós de máquina aos nós de produto, chamados de arcos de produção, para os quais o ganho é igual a w_{it} . O limite inferior de fluxo nos arcos é dado pelas restrições (13), e é sempre igual a 0. O limite superior de capacidade para os arcos de produção é dado pela restrição (12). Para os demais arcos assume-se que o limitante superior é infinito. O custo dos arcos é dado pela função objetivo (8). Devido à equação (14) não existem arcos de estoque incidindo sobre os nós de produto do período 1 da rede e também não existem arcos de estoque saindo dos nós de produto no último período. A Figura 6 ilustra a rede correspondente ao modelo (G).

O problema de planejamento de produção de múltiplos itens em múltiplos períodos com restrição de capacidade pode, como foi mostrado, ser representado através de uma *rede generalizada*.

CAPÍTULO 3

UM ALGORITMO *BRANCH-AND-BOUND*

3.1 Introdução

Neste capítulo é apresentado um algoritmo *branch-and-bound* para resolver o problema (PDL), onde em cada nó é resolvido um problema de fluxo a custo mínimo em uma *rede generalizada*.

Na seção 3.2 é fornecida uma visão geral do método *branch-and-bound*. A relaxação utilizada para o problema (PDL) é apresentada na seção 3.3. As regras de seleção de problema (nó) e de escolha da variável de ramificação são discutidas nas seções 3.4 e 3.5, respectivamente. Na seção 3.6 é apresentada a heurística utilizada para obter uma solução inicial para o problema, enquanto na seção 3.7 é descrita a pré-fixação de variáveis 0-1. Finalmente, na seção 3.8 é apresentado o algoritmo proposto.

3.2 O Método *Branch-and-Bound*

Este método é muito utilizado para resolver problemas de programação linear inteira e programação linear inteira mista e baseia-se em dois princípios básicos: **partição** e **relaxação**.

A **partição** consiste em dividir o conjunto de soluções factíveis S do problema original em uma família de conjuntos disjuntos $\{S^i: i = 1, \dots, k\}$. A cada um destes conjuntos é associado um problema de otimização P^i , formando-se assim, uma lista de problemas. A motivação da partição é que a otimização em conjuntos menores é mais simples e rápida.

A partição geralmente é feita de forma recursiva e pode ser representada através de uma árvore. A Figura 7 ilustra a partição completa para o caso em que duas variáveis binárias são consideradas. Como se pode notar na Figura 7, os nós descendentes de um dado nó representam a partição da região factível deste nó. Por exemplo, os nós S^{00} e S^{01} são descendentes de S^0 .

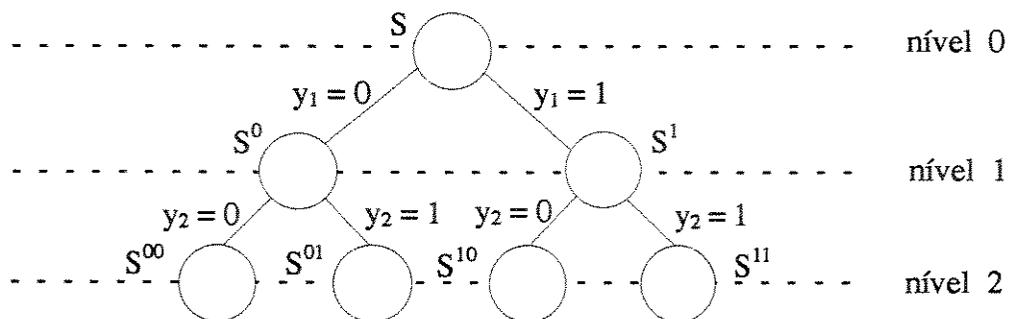


Figura 7- Árvore da Partição

A partição de um conjunto S^i é interrompida quando uma das seguintes condições é satisfeita:

1. $S^i = \emptyset$;
2. uma solução ótima de P^i é encontrada;

3. o valor de qualquer solução factível de P^i é pior que o valor da melhor solução factível atual (solução incumbente).

Quando uma dessas três condições ocorre, o nó correspondente a S^i pode ser descartado ou sondado, pois todas as suas soluções factíveis estão implicitamente enumeradas.

A verificação dessas três condições é mais fácil de ser realizada quando é utilizada a **relaxação**. Neste trabalho será utilizada a relaxação linear, que consiste em desconsiderar a integralidade das variáveis inteiras, permitindo, portanto, que elas sejam reais. Obtém-se assim um problema relaxado de programação linear P_R^i com região factível $S_R^i \supset S^i$. Suponha que o problema original é de minimização. Então as três condições anteriores podem ser verificadas através do problema relaxado da seguinte maneira:

1. $S_R^i = \emptyset$;
2. a solução ótima de P_R^i é inteira;
3. o valor ótimo do problema relaxado é maior ou igual ao valor da solução incumbente.

É claro que quanto mais elevado é o limitante inferior de um problema, mais fácil é sondar um nó, logo a árvore *branch-and-bound* terá menos nós. O ideal, portanto, é ter um problema relaxado de fácil resolução que forneça limitantes inferiores de boa qualidade. Infelizmente, em geral, estes dois atributos são antagônicos, isto é, um problema relaxado muito fácil produz, em geral, um limitante inferior pobre.

Além da qualidade do limitante, existem outros dois fatores que afetam a eficiência do método *branch-and-bound*: a escolha do próximo problema da lista e a escolha da variável de ramificação que separa o conjunto S^i em dois conjuntos, como ilustra a Figura 7. As regras utilizadas para a escolha do nó e da variável serão apresentadas nas seções 3.4 e 3.5, respectivamente.

3.3 Relaxação

Em [Correia, 1989], foi abordado um problema energético com características semelhantes ao (PDL). Foi estudado um problema de planejamento energético de curto prazo com termoeletricas de ciclo combinado. Este foi formulado como um problema de otimização em *rede generalizada* com variáveis 0-1. As variáveis 0-1 representam as duas faixas de operação da termoeletrica, onde cada faixa apresenta um rendimento diferente. A presença de variáveis 0-1 foi considerada na rede através da utilização de arcos com perda fixa, a qual não tinha sido abordado na literatura até então.

A relaxação utilizada para o problema estudado neste trabalho é similar à relaxação utilizada no modelo energético mencionado acima. Para resolver o (PDL) será adotada, então, a relaxação da integralidade de suas variáveis 0-1 (y_{it}).

Para o (PDL), a variável inteira y_{it} representa duas situações possíveis para uma máquina em relação a um item i em um período t . A primeira, onde $y_{it} = 1$, corresponde à decisão de produzir o item i no período t , acarretando o custo e o tempo de preparação de máquina. Na situação contrária, com $y_{it} = 0$, não existe produção do item i no período t , e portanto o custo e o tempo de preparação de máquina não serão contabilizados.

Através da relaxação da integralidade, os custos e tempos de produção de máquina tornam-se lineares à quantidade produzida. A Figura 8 ilustra o custo original e o custo relaxado em função da quantidade produzida. Já a Figura 9 mostra o tempo de produção original e o tempo de produção relaxado.

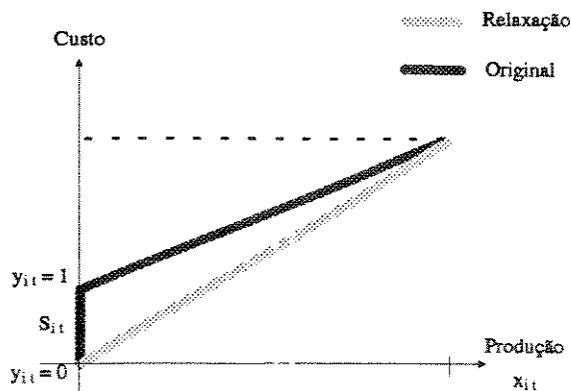


Figura 8 - Custo de Produção

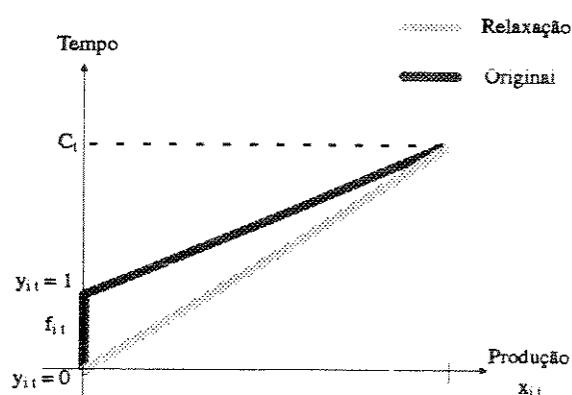


Figura 9 - Tempo de Produção

Relaxadas as variáveis y_{it} , e tendo como objetivo minimizar o custo, pode-se escrever a partir da equação (4) do modelo (PDL) que:

$$y_{it} = \frac{b_{it} x_{it}}{\Delta_{it}} \quad (15).$$

O modelo relaxado (R) é obtido substituindo (15) em (PDL). Como o modelo completo, utilizado futuramente, será construído a partir da união do problema (G') e de seu correspondente relaxado, as variáveis de (R) terão o superíndice r. O modelo (R) pode ser representado matematicamente como:

$$(R) \text{ Min } \sum_{i=1}^N \sum_{t=1}^T \left(\left(\frac{s_{it} b_{it}}{\Delta_{it}} + c_{it} \right) x_{it}^r + h_{it} I_{it} \right) \quad (16)$$

$$\text{s.a. } I_{i,t-1} + x_{it}^r - I_{it} = d_{it} \quad i = 1, \dots, N; \quad t = 1, \dots, T \quad (17)$$

$$\sum_{i=1}^N \left(\frac{f_{it}}{\Delta_{it}} + 1 \right) b_{it} x_{it}^r \leq C_t \quad t = 1, \dots, T \quad (18)$$

$$\Delta_{it} = \min \left\{ C_t - f_{it}, \sum_{k=t}^T b_{ik} d_{ik} \right\} \quad i = 1, \dots, N; \quad t = 1, \dots, T$$

$$I_{it} \geq 0, \quad x_{it}^r \geq 0 \quad i = 1, \dots, N; \quad t = 1, \dots, T \quad (19)$$

$$I_{i0} = I_{iT} = 0 \quad i = 1, \dots, N \quad (20).$$

Nota-se em (16) e (18) que o custo e o tempo de preparação de máquina foram respectivamente agregados ao custo unitário de produção e ao tempo unitário de processamento.

Como no Capítulo 2, o modelo será escrito de acordo com a modelagem para rede generalizada, ou seja, a variável ρ^r será acrescentada ao modelo a fim de caracterizar a conversão do fluxo de tempo em produtos. Através da equação (21) a variável ρ^r é definida por:

$$\rho_{it}^r = \left(\frac{f_{it}}{\Delta_{it}} + 1 \right) b_{it} x_{it}^r \quad (21).$$

Logo o modelo (R) pode ser reescrito como:

$$(R') \text{ Min } \sum_{i=1}^N \sum_{t=1}^T \left(\frac{s_{it} b_{it} + c_{it} \Delta_{it}}{f_{it} b_{it} + \Delta_{it} b_{it}} \rho_{it}^r + h_{it} I_{it} \right) \quad (22)$$

$$\text{s.a. } I_{i,t-1} + \frac{\Delta_{it}}{f_{it} + \Delta_{it}} \frac{1}{b_{it}} \rho_{it}^r - I_{it} = d_{it} \quad i=1, \dots, N; \quad t=1, \dots, T \quad (23)$$

$$\sum_{i=1}^N \rho_{it}^r \leq C_t \quad t=1, \dots, T \quad (24)$$

$$\Delta_{it} = \min \left\{ C_t - f_{it}, \sum_{k=1}^T b_{ik} d_{ik} \right\} \quad i=1, \dots, N; \quad t=1, \dots, T$$

$$I_{it} \geq 0, \quad \rho_{it}^r \geq 0 \quad i=1, \dots, N; \quad t=1, \dots, T \quad (25)$$

$$I_{i0} = I_{iT} = 0 \quad i=1, \dots, N \quad (26)$$

O ganho no arco relaxado é caracterizado por:

$$\omega_{it}^r = \frac{\Delta_{it}}{f_{it} + \Delta_{it}} \frac{1}{b_{it}}$$

Como em [Correia, 1989], será aplicado um algoritmo *branch-and-bound* ao (PDL) com todos os arcos originais e relaxados. Isto porque, trabalhando com a rede completa, é possível utilizar reotimização. O processo de reotimização é viabilizado através da alteração dos custos unitários \bar{c}_{it}^r e \bar{c}_{it} associados a ρ_{it}^r e ρ_{it} . Estes custos serão definidos posteriormente.

Antes de apresentar o modelo completo, deve-se destacar que o conjunto A é formado por todos as variáveis y_{it} já fixadas, ou seja, as variáveis cujos valores são 1 ou 0.

O modelo completo (MC) composto por todos os arcos originais e relaxados é formulado matematicamente como:

$$\begin{aligned}
 \text{(MC) Min } & \sum_{i=1}^N \sum_{t=1}^T [\bar{c}_{it}^r \rho_{it}^r + \bar{c}_{it} \rho_{it} + h_{it} I_{it}] + \sum_{y_{it} \in A} s_{it} y_{it} \\
 \text{s.a. } & I_{i,t-1} + \frac{\Delta_{it}}{f_{it} + \Delta_{it}} \frac{1}{b_{it}} \rho_{it}^r + \frac{1}{b_{it}} \rho_{it} - I_{it} = d_{it} \quad i = 1, \dots, N; \quad t = 1, \dots, T \\
 & \sum_{i=1}^N (\rho_{it} + \rho_{it}^r) + \sum_{y_{it} \in A} f_{it} y_{it} \leq C_t \quad t = 1, \dots, T \\
 & \rho_{it} \leq \Delta_{it} y_{it} \quad i = 1, \dots, N; \quad t = 1, \dots, T; \quad y_{it} \in A \\
 & \Delta_{it} = \min \left\{ C_t - f_{it}, \sum_{k=t}^T b_{ik} d_{ik} \right\} \quad i = 1, \dots, N; \quad t = 1, \dots, T; \\
 & A = \{ y_{it} \mid y_{it} = 1 \text{ ou } y_{it} = 0 \} \quad i = 1, \dots, N; \quad t = 1, \dots, T; \\
 & I_{it} \geq 0, \quad \rho_{it}^r \geq 0 \quad i = 1, \dots, N; \quad t = 1, \dots, T; \\
 & 0 \leq y_{it} \leq 1, \quad \rho_{it} \geq 0 \quad i = 1, \dots, N; \quad t = 1, \dots, T; \\
 & I_{i0} = I_{iT} = 0 \quad i = 1, \dots, N.
 \end{aligned}$$

O modelo (MC) é representado pela rede da Figura 10. Os parâmetros dos arcos foram suprimidos para maior clareza.

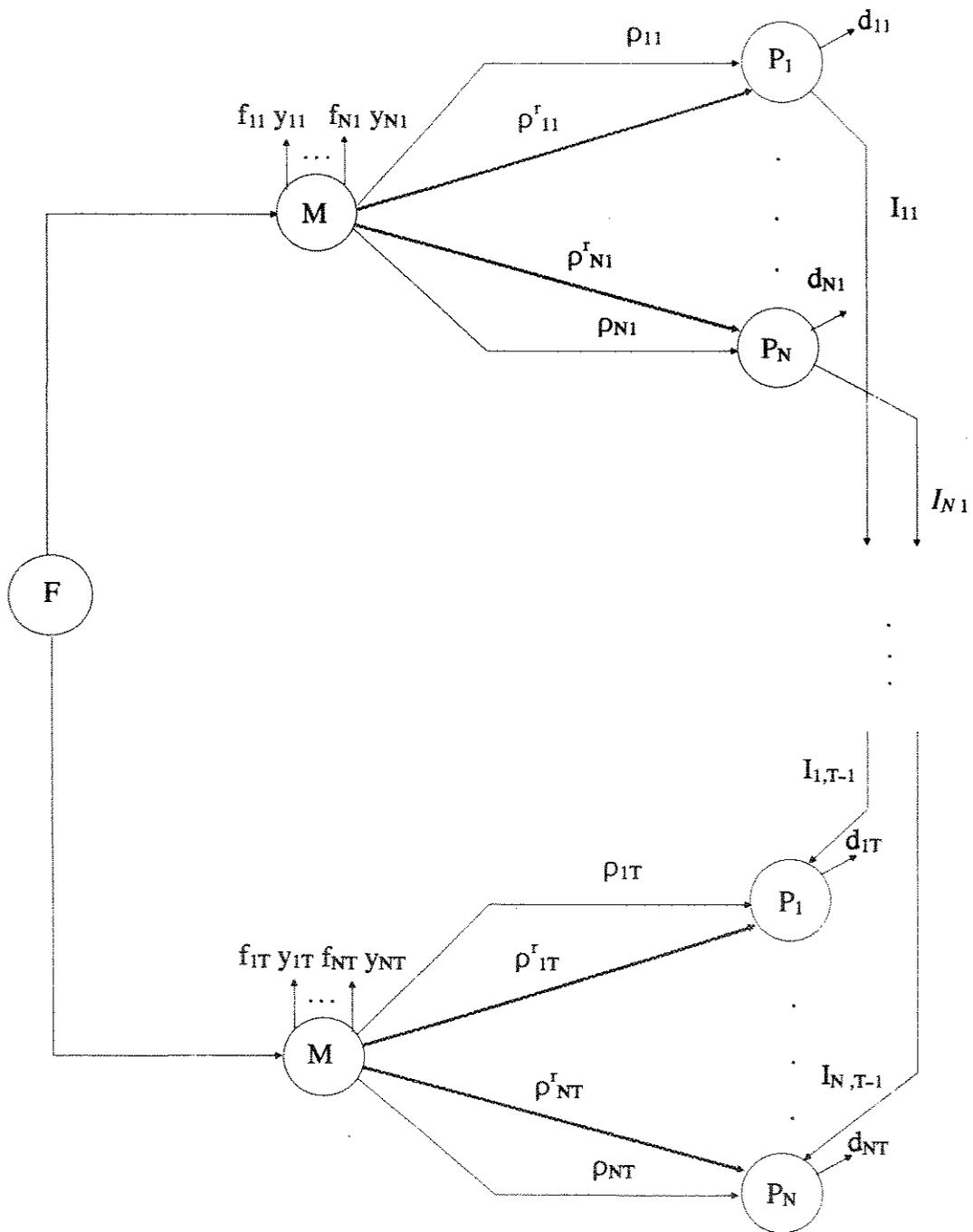


Figura 10 - Rede do Modelo (MC)

A estratégia utilizada para permitir a reotimização consiste em alterar os custos dos arcos originais e relaxados à medida que as variáveis y_{it} vão sendo fixadas. Esta estratégia, denominada **BigM**, consiste em atribuir um custo muito grande (\bar{M}) a um arco se o fluxo sobre ele não é permitido, e um custo muito pequeno ($-\bar{M}$) se o fluxo sobre ele é obrigatório. Este custo é um artifício e deve ser abatido da função objetivo. Para o (MC), são possíveis três situações de acordo com o estado da variável y_{it} : fixada em 0 ou 1, e não fixada, isto é, $0 \leq y_{it} \leq 1$. Para cada situação o fluxo pelos arcos originais e relaxados é caracterizado de maneira diferente.

Quando $y_{it} = 0$ significa que não há produção do item i no período t . Portanto não existe fluxo saindo do nó de máquina para o nó de produto, não sendo realizado nenhum gasto com a preparação da máquina. Logo, pode-se concluir que:

$$y_{it} = 0 \Rightarrow \begin{cases} \rho_{it} = 0 \\ \rho_{it}^r = 0 \\ z_{it} = 0. \end{cases}$$

A segunda situação é caracterizada por $y_{it} = 1$, ou seja, existe produção do item i no período t . A produção não escoar pelo arco relaxado porque já foram contabilizados o custo e o tempo de preparação de máquina, não sendo necessário contabilizá-los parcialmente, como é característica deste arco. Para esta situação, tem-se:

$$y_{it} = 1 \Rightarrow \begin{cases} \rho_{it} \geq 0 \\ \rho_{it}^r = 0 \\ z_{it} = f_{it}. \end{cases}$$

A última situação ocorre quando $0 \leq y_{it} \leq 1$, isto é, não é possível determinar *a priori* se haverá ou não produção do item i no período t . Caso exista produção, ela deverá escoar pelo arco relaxado para que possam ser contabilizados, mesmo que parcialmente, o custo e o tempo de preparação de máquina. Logo:

$$0 \leq y_{it} \leq 1 \Rightarrow \begin{cases} \rho_{it} = 0 \\ \rho_{it}^r \geq 0 \\ z_{it} = 0. \end{cases}$$

A Figura 11 ilustra as três situações apresentadas anteriormente. Os arcos cujos fluxos devem ser nulos foram assinalados com \times .

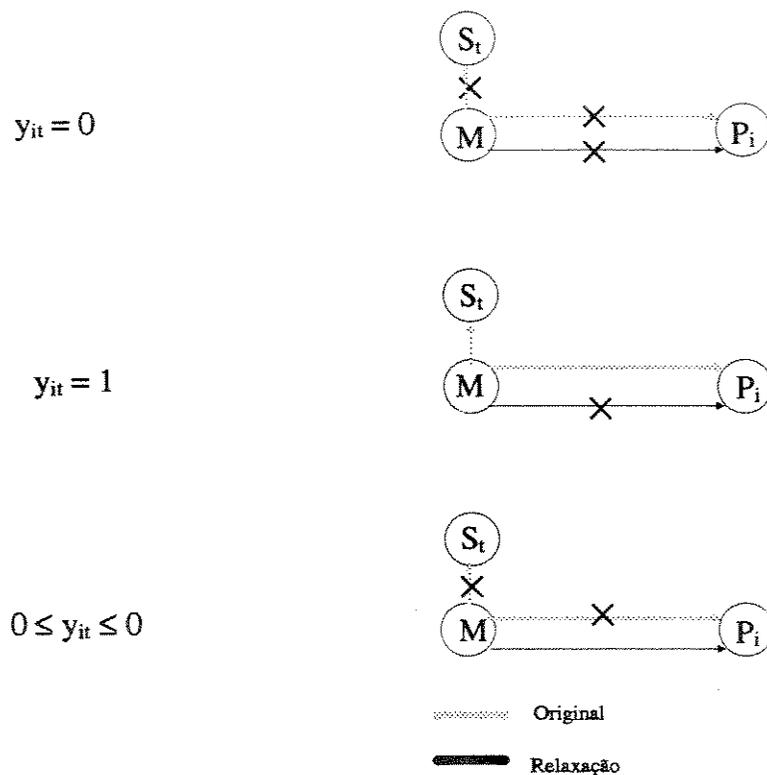


Figura 11 - Fluxo segundo o Estado das Variáveis 0-1

As características de fluxo associadas aos estados das variáveis y_{it} serão determinadas de acordo com os valores dos custos unitários \bar{c}_{it} e \bar{c}_{it}^r . Para cada uma das três situações anteriores, tem-se:

$$y_{it} = 0 \Rightarrow \begin{cases} \bar{c}_{it} = \bar{M} \\ \bar{c}_{it}^r = \bar{M} \\ s_{it} = \bar{M} \end{cases}$$

$$y_{it} = 1 \Rightarrow \begin{cases} \bar{c}_{it} = \frac{c_{it}}{b_{it}} \\ \bar{c}_{it}^r = \bar{M} \\ s_{it} = -\bar{M} \end{cases}$$

$$0 \leq y_{it} \leq 1 \Rightarrow \begin{cases} \bar{c}_{it} = \bar{M} \\ \bar{c}_{it}^r = \frac{s_{it} b_{it} + c_{it} \Delta_{it}}{f_{it} b_{it} + \Delta_{it} b_{it}} \\ s_{it} = \bar{M}. \end{cases}$$

Será utilizada reotimização sempre que o conjunto de variáveis y_{it} do problema selecionado para ser resolvido diferir em apenas um elemento do conjunto do último problema resolvido, desde que este último seja factível. Este critério foi adotado pois, devido a semelhança entre os problemas, o processo de reotimização será bastante eficiente e os erros de arredondamento causados pelo processo serão evitados.

Uma vez que os problemas diferem em apenas uma variável, sabe-se que de todos os seus arcos somente três terão custos diferentes. Estes arcos serão chamados de *arcos alterados*. O processo de reotimização identifica dentre estes arcos alterados aqueles que são básicos e não básicos. Caso exista um arco alterado básico primeiro, recalcula-se as variáveis duais do problema (potenciais dos nós). A seguir calculam-se os novos custos relativos dos arcos não básicos para obter-se a solução ótima do problema.

Logo, respeitando-se as definições anteriores, um problema pode ser reotimizado, o que melhora sensivelmente o desempenho do algoritmo.

3.4 Seleção de Nós

As regras para seleção de problemas (nós) são divididas em dois tipos: regras *a priori* e regras adaptativas.

A regra *a priori* mais conhecida é a *Last-in-First-Out* (último que chega, primeiro que sai), ou simplesmente LIFO. Esta regra determina que o problema selecionado é sempre o último que foi adicionado à lista. As principais vantagens da regra LIFO são: a facilidade de implementação e o uso reduzido de memória do computador; além disso, viabiliza o uso frequente de reotimização, agilizando sensivelmente a resolução do problema. Uma outra vantagem é que esta estratégia permite encontrar uma solução factível mais facilmente. Por outro lado, a sua principal desvantagem é que o limitante inferior da árvore *branch-and-bound* demora muito para ser alterado, dificultando a sondagem dos nós.

Dentre as regras **adaptativas**, a mais conhecida é a regra de escolha do nó com menor limitante inferior (o limitante inferior de um nó é o valor ótimo do problema relaxado). Suas principais vantagens são: é uma regra otimista com relação a encontrar uma solução factível de boa qualidade e fornece um limitante inferior da árvore *branch-and-bound* continuamente não decrescente, facilitando assim a sondagem dos nós. No entanto, sua principal desvantagem é o grande esforço computacional exigido, pois nem sempre é possível utilizar reotimização ao longo do processo de resolução. Outra desvantagem, também computacional, é a necessidade de grande espaço disponível de memória para sua execução.

Neste trabalho foram utilizadas inicialmente as duas regras mencionadas. No entanto, testes preliminares mostraram que o desempenho da regra de seleção pelo menor limitante inferior é muito superior ao da regra LIFO. Portanto, foi adotada a regra adaptativa de menor limitante inferior.

3.5 Seleção da Variável de Ramificação

Uma vez escolhido o problema a ser particionado, deve-se então determinar a variável de ramificação que dará origem a dois problemas descendentes.

Existem dois tipos de regras para a escolha de variáveis de ramificação: por **prioridade** ou por **penalidade**. A regra de escolha por **prioridade** é mais comum e está baseada num ordenamento prévio das variáveis segundo regras apropriadas ao problema. Estas regras são empíricas e estão relacionadas ao conhecimento do problema estudado. A regra de escolha por **penalidade** determina a variável a ser fixada segundo uma estimativa do acréscimo na função objetivo ao se forçar esta variável assumir um valor inteiro; escolhe-se então a que causa menor ou maior acréscimo de acordo com a política adotada.

Neste trabalho foram utilizadas as seguintes regras de seleção de variáveis de ramificação por prioridade:

1. escolha y_{it} pelo maior s_{it} ;
2. escolha y_{it} pelo menor $\frac{d_{it}}{s_{it}}$;
3. escolha y_{it} pela menor d_{it} ;
4. escolha y_{it} pela maior d_{it} ;
5. escolha y_{it} pelo maior $\frac{s_{it}}{h_{it}}$;
6. escolha y_{it} pelo menor $\frac{s_{it}}{d_{it} h_{it}}$;
7. escolha y_{it} pela maior diferença entre o custo original e o custo relaxado ($c_{it} x_{it}^r + s_{it} y_{it} - c_{it}^r x_{it}^r$).

A regra (1) fixa no início da árvore *branch-and-bound* as variáveis com alto custo de preparação de máquina. Esta regra define dois tipos de problemas: o primeiro onde a variável foi fixada em zero ($y_{it} = 0$), que tem por objetivo reduzir o custo total

de produção; e o segundo onde a variável foi fixada em um ($y_{it} = 1$), que visa aumentar rapidamente o custo total de produção, permitindo a sondagem do nó.

A segunda regra está baseada na razão da demanda pelo custo fixo de produção, e quanto menor esta razão espera-se que seja mais vantajoso ($y_{it} = 0$) atender a demanda via estoque, pois o custo de preparação é muito alto em relação a demanda no período. A regra (3) é similar à anterior, porém ignora o custo. Assume-se que pequenas demandas possam ser atendidas pela produção em períodos anteriores sem grande prejuízo para o custo.

Pela quarta regra serão fixadas no início da árvore *branch-and-bound* as variáveis correspondentes às mais altas demandas. Esta regra tem os seguintes objetivos: quando a variável é fixada em 0, busca-se sondar o nó através da infactibilidade do problema correspondente, pois grandes demandas geralmente não podem ser atendidas pela produção em períodos anteriores devido às restrições de capacidade; quando a variável é fixada em 1, a regra tem por objetivo diminuir o custo de estoque, buscando assim boas soluções para o problema.

As regras 5 e 6 ponderam o custo fixo de produção com o custo de estoque, ou seja, visam não produzir onde o custo de estoque é baixo, tentando economizar, assim, o pagamento do custo fixo de produção.

A última regra procura fixar *a priori* a variável cuja diferença entre o custo original e o relaxado for mais significativa (ver Figura 8). Esta tem por objetivo ajustar a solução começando pelas variáveis mais distantes do custo real.

Todas as regras, exceto a quarta, são regras otimistas, ou seja, caminham por um lado em direção ao menor custo, buscando uma boa solução, e por outro, aumentando o custo, visando a sondagem dos nós.

As quatro primeiras regras foram propostas por [Evans, 1985] para o mesmo problema, porém sem considerar o tempo de preparação de máquina. As regras 5 e 6 foram desenvolvidas com base no conhecimento do problema estudado, observando o resultado de boas soluções heurísticas. A última regra de escolha de variável para ramificação é a regra clássica utilizada por [Jensen e Barnes, 1980] para o problema de custo fixo.

3.6 Solução Inicial

Um modo de aumentar a eficiência do algoritmo *branch-and-bound* é inicializá-lo com uma solução incumbente de boa qualidade. Uma heurística capaz de fornecer tal solução foi apresentada em [Trigeiro et al., 1989]. Esta heurística fornece uma boa solução inicial e também um bom limitante inferior obtida por relaxação lagrangeana das restrições de capacidade de (PDL). A partir desta heurística são extraídos os valores das variáveis y_{it} , ou seja, onde cada item deve ser produzido.

O processo de obtenção da solução inicial começa resolvendo um nó onde os valores das variáveis y_{it} são determinados pela heurística. Este procedimento pode obter uma solução inicial distinta da solução heurística, dado que somente as decisões sobre as variáveis binárias estão sendo extraídas da solução heurística. Assim, é obtida a primeira solução incumbente. Se o valor desta solução estiver suficientemente próximo do limitante inferior fornecido pela heurística, o algoritmo é finalizado tendo como solução ótima a incumbente. Caso contrário a execução do algoritmo *branch-and-bound* é iniciada a partir do nó zero.

3.7 Pré-Fixação de Variáveis

O tempo de execução do algoritmo *branch-and-bound* está relacionado com o número de variáveis 0-1 do problema, ou seja, com a dimensão do problema. Logo, quanto menor a dimensão, é mais fácil encontrar sua solução ótima.

Para o problema estudado, sabe-se que os estoques iniciais são nulos e que a demanda deve obrigatoriamente ser atendida; logo para o primeiro período é possível concluir para qualquer um dos itens ($i=1,2,\dots,N$) que:

$$y_{i1} = 1 \text{ se } d_{i1} > 0.$$

Essa conclusão pode ser estendida aos demais períodos ($t=2,3,\dots,T$) para todos os itens ($i=1,2,\dots,N$), isto é:

$$y_{it} = 1 \quad \text{se } b_{it} d_{it} > \text{Cap_Rest}_t \quad (41)$$

onde Cap_Rest_t é um limitante superior para a capacidade não utilizada até o período t .

Nesse caso a fixação da variável y_{it} decorre de que a demanda deve ser atendida, e existem apenas duas maneiras de atendê-la: pela produção do item no próprio período t ou pelo estoque proveniente de períodos anteriores a t . A capacidade restante é a capacidade ociosa nos períodos anteriores a t , sem contabilizar o tempo de preparação (mas sim o de produção) daquelas variáveis y_{it} que não satisfazem (41). Se esta não for suficiente para atender a demanda total de um dado item i no período t , então pode-se concluir que ele deverá ser produzido no período em questão, ou seja, $y_{it} = 1$. Considere o seguinte exemplo com três itens e três períodos:

Período 1

- $d_{11} = 10$ $b_{11} = 1$ $f_{11} = 5$
- $d_{21} = 20$ $b_{21} = 1$ $f_{21} = 10$
- $d_{31} = 25$ $b_{31} = 2$ $f_{31} = 15$
- $C_1 = 150$
- $y_{11} = 1$ $y_{21} = 1$ $y_{31} = 1$
- $\text{Cap_Rest}_1 = 150 - (10 \times 1 + 5) - (20 \times 1 + 10) - (25 \times 2 + 15)$
- $\text{Cap_Rest}_1 = 40$

Período 2

- $d_{12} = 40$ $b_{12} = 1$ $f_{12} = 5$
- $d_{22} = 30$ $b_{22} = 1$ $f_{22} = 10$
- $d_{32} = 25$ $b_{32} = 2$ $f_{32} = 15$
- $C_2 = 150$
- $\text{Cap_Rest}_1 = 40$
- $y_{12} = ?$ $y_{22} = ?$ $y_{32} = 1$
- $\text{Cap_Rest}_2 = 40 + 150 - 40 \times 1 - 30 \times 1 - (25 \times 2 + 15)$
- $\text{Cap_Rest}_2 = 55$

Período 3

- $d_{13} = 30$ $b_{13} = 1$ $f_{13} = 5$
- $d_{23} = 50$ $b_{23} = 1$ $f_{23} = 10$
- $d_{33} = 20$ $b_{33} = 2$ $f_{33} = 15$
- $C_3 = 150$
- $\text{Cap_Rest}_2 = 55$
- $y_{13} = ?$ $y_{23} = ?$ $y_{33} = ?$

Dependendo do problema podem ser fixadas muitas variáveis 0-1 logo no primeiro nó da árvore *branch-and-bound*, diminuindo significativamente a sua dimensão.

3.8 O Algoritmo Proposto

O algoritmo proposto é um algoritmo *branch-and-bound* adaptado através das estratégias apresentadas nas seções anteriores. Estas estratégias foram utilizadas a fim de torná-lo mais apropriado para resolver o problema de dimensionamento de lotes. Este algoritmo resolve em cada nó da árvore *branch-and-bound* um problema de rede generalizada (MC). Para resolver a rede generalizada foi utilizado o algoritmo proposto em [Jensen e Barnes, 1980].

A seguir é apresentado o pseudo-código do algoritmo proposto. Antes, porém, são definidas suas variáveis e procedimentos a fim de facilitar seu entendimento. Seja:

\bar{Z} - valor da solução incumbente do problema;

Z_L - limitante inferior para o problema;

Z_{LH} - limitante inferior lagrangeano dado pela heurística;

Z_i - valor da solução do problema relaxado P_r^i ;

Z_{L_i} - limitante inferior do problema P_i ;

Y_i - variáveis binárias da solução ótima de P_r^i ;

P^i - i-ésimo problema da lista;

P_r^i - problema P^i relaxado;

L - lista de problemas candidatos;

k - número de problemas da lista;

Solução_Inicial - procedimento utilizado para obter a solução inicial conforme explicado na seção 3.6.

Seleciona - seleciona o problema da lista L com menor limitante inferior;

Resolve - resolve o problema P^i escolhido;

Escolhe_variável_ramificação - escolhe a variável de ramificação para o problema P^i de acordo com a regra adotada;

Gerar_problemas_descendentes - gera dois problemas descendentes de P^i a partir da variável de ramificação escolhida.

Algoritmo *Branch-and-Bound*

```

 $\bar{Z} = \infty$  ,  $Z_L = Z_{LH}$  (* Inicialização de Variáveis *)
Solução_Inicial( ) (* Obtenção da Solução Inicial *)
Se  $\frac{(\bar{Z} - Z_L)}{Z_L} > 0.01$  então
  Pré_fixe_variáveis( )
  k= 1
  P1= Problema relaxado, nó zero
  L = {P1}
  ZL,1 = 0
  Enquanto k ≠ 0
    Selecciona(Pi)
    Se ZL,i > ZL então (* Atualização do Limitante Inferior*)
      ZL = ZL,i
      Se  $\frac{(\bar{Z} - Z_L)}{Z_L} < 0.01$  então FimEnquanto.
    Fim então
    L = L - {Pi} (* Atualiza Lista de Problemas *)
    k= k-1
    Resolva(Pri)
    Se (Pri factível) e (Zi <  $\bar{Z}$ ) então
      Se (Yi inteira) então
         $\bar{Z} = Z_i$  (* Atualiza Solução Incumbente *)
        Se  $\frac{(\bar{Z} - Z_L)}{Z_L} < 0.01$  então FimEnquanto.
      senão (*Sondagem, após trocar Solução *)
        (* Incumbente *)
        cont= 0
        Para i=1 até k faça
          Se Zi >  $\bar{Z}$  então
            L = L - {Pi}
            cont= cont+1
          Fim então
        Fim Para
        k= k-cont
      Fim senão
    Fim então
  senão (* Ramificação do Problema*)

```

```

    Escolhe_variável_ramificação( $P^i$ )
    Gera_Problemas_descendentes( $P^{k+1}, P^{k+2}$ )
     $L = L + \{P^{k+1}, P^{k+2}\}$ 
     $Z_{L_{k+1}} = Z_i$ 
     $Z_{L_{k+2}} = Z_i$ 
     $k = k + 2$ 
  Fim Senão
Fim então
Fim Enquanto
Fim Se  $\frac{(\bar{Z} - Z_L)}{Z_L} > 0.01$ 
Se  $\bar{Z} = \infty$  então Problema Infactível
senão Solução  $\bar{Z}$ 
FIM.

```

Note que a lista de problemas candidatos (L) contém os problemas não resolvidos com limitantes inferiores (Z_{L_i}) herdados dos respectivos nós imediatamente predecessores (nós-pai).

Este algoritmo especializado foi testado combinando a regra de escolha de nó de menor limitante inferior com as sete regras de seleção de variável de ramificação, visando obter a combinação mais eficiente.

CAPÍTULO 4

TESTES COMPUTACIONAIS E

ANÁLISE DE RESULTADOS

4.1 Introdução

Neste capítulo são apresentados os resultados computacionais obtidos ao longo do desenvolvimento deste trabalho e a análise de seus resultados.

Os problemas utilizados para testar o algoritmo foram gerados aleatoriamente e podem ser divididos em três grupos:

Grupo1- problemas gerados para definir a melhor regra de escolha da variável de ramificação;

Grupo2- problemas de maior porte, gerados para medir a influência da dimensão, custo de preparação, capacidade e demanda no desempenho do algoritmo proposto;

Grupo3- problemas gerados para comparar, mesmo que superficialmente, o desempenho do algoritmo proposto com o apresentado por Diaby et al. (1992a).

O algoritmo apresentado neste trabalho foi implementado em linguagem C e os testes foram realizados em uma estação de trabalho SUN modelo SPARCstation 10.

4.2 Características dos Problemas Gerados

Os problemas utilizados nos testes computacionais foram gerados aleatoriamente segundo uma distribuição uniforme de probabilidade. Os parâmetros utilizados nesta geração são semelhantes aos utilizados em [Trigeiro et al., 1989] e [Afentakis, 1986].

Na geração de um problema devem ser definidos os seguintes parâmetros:

1. número de itens (N);
2. número de períodos (T);
3. custo unitário de produção dos itens nos períodos;
4. custo unitário de estoque dos itens nos períodos;
5. custo de preparação da máquina nos períodos;
6. tempo de preparação da máquina nos períodos;
7. tempo unitário de processamento dos itens nos períodos;
8. demanda dos itens nos períodos.

Os custos e tempos de preparação de máquina, a capacidade disponível e os tempos de produção foram considerados constantes ao longo de todo o horizonte de planejamento. Os custos unitários de produção foram considerados nulos, pois podem ser vistos como um custo fixo a ser acrescentado ao custo total de produção, uma vez que são considerados constantes ao longo do horizonte. Esta dedução é bastante simples

e será mostrada a seguir. Seja c_i o custo para cada item i (igual para todos os períodos) e x_{it} a produção do item i no período t , então:

$$\text{Prod} = \sum_{i=1}^N c_i \sum_{t=1}^T x_{it} .$$

No Capítulo 1 foi mostrado que $\sum_{i=1}^N x_{it} = \sum_{t=1}^T d_{it}$, então,

$$\text{Prod} = \sum_{i=1}^N c_i \sum_{t=1}^T d_{it} \quad (27)$$

e de (42) tem-se que **Prod** é constante, dado que todas as demandas são conhecidas.

Os números de itens (N) e de períodos (T) foram escolhidos *a priori* e os valores utilizados foram:

Grupo1- foram escolhidos N e T de acordo com os testes apresentados por Diaby et al. (1992a), $N= 8$ e $T= 8$, a fim de permitir uma comparação entre os algoritmos;

Grupo2- $N= 10$ e $T= 6$, $N= 25$ e $T= 6$, $N= 50$ e $T= 6$, $N= 10$ e $T= 12$, e $N= 25$ e $T= 12$;

Grupo3- $N=50$ e $T= 8$, $N=50$ e $T=12$, e $N=99$ e $T=8$, também escolhidos de acordo com Diaby et al. (1992a).

Então, tem-se ao todo nove tipos de problemas testados. Para cada tipo apresentado foram gerados aleatoriamente 5 problemas, como se segue, onde $U[a,b]$ representa uma geração aleatória através de distribuição uniforme entre os valores a e b :

- **demanda**

Foram gerados dois tipos de demanda:

a. $U [0,200]$ (constante)

b. $U [a,b]$ onde os extremos de geração crescem de dez unidades a cada período, no início $a= 0$ e $b= 200$ (crescente).

A demanda do tipo b é semelhante à proposta em [Trigeiro et al., 1989].

- **custo fixo de preparação de máquina**
 - a. U[5, 95] - para caracterizar problemas com custo de preparação baixo;
 - b. U[50, 950] - para caracterizar problemas com custo de preparação alto;
- **custo de estoque**
U[2, 4]
- **tempo de preparação de máquina**
U[10, 50]
- **capacidade**
a capacidade foi gerada segundo a política lote-por-lote, isto é, para cada período t, calcula-se a capacidade necessária para produzir exatamente as demandas pelos itens neste período, ou seja:

$$CAP_t = \sum_{i=1}^N d_{it} b_{it} + f_{it} \quad (\text{se } d_{it} > 0) \quad (28).$$

Então, a capacidade para os períodos foi calculada fazendo-se a média destas capacidades obtidas em (43):

$$C_t = \frac{\sum_{t=1}^T CAP_t}{T}$$

Para os problemas foram testados dois níveis de capacidade diferentes:

- a. 1,1. C_t , considerada apertada;
- b. 1,2. C_t , considerada normal.

Para cada N e T escolhidos *a priori* são analisados problemas variando três fatores: capacidade (apertada e normal), custo de preparação de máquina (alto e baixo) e demanda (crescente e constante). Obtem-se assim oito classes distintas de problemas e portanto, 40 problemas (8 classes x 5 problemas) para cada par (N,T) dado.

Ao todo foram gerados 360 problemas que foram divididos entre os grupos da seguinte maneira: 40 problemas para o Grupo1, 200 problemas para o Grupo2 e 120

problemas para o Grupo3. Para o Grupo3 os custos de preparação de máquina foram considerados nulos como em [Diaby et al., 1992a].

4.3 Resultados dos Testes

O algoritmo proposto visa encontrar a solução ótima para o problema de dimensionamento de lotes estudado neste trabalho, mas como esses problemas são de difícil solução foi imposto ao algoritmo um limite máximo para seu tempo de execução. O limite escolhido foi meia hora de execução do programa, considerado razoável. Nas tabelas apresentadas a seguir será relatado o número de problemas para os quais o algoritmo foi interrompido ao atingir o tempo limite, não garantindo portanto que a solução encontrada é a ótima. Vale destacar que foram consideradas soluções ótimas aquelas cujos resultados estão a menos de 1% do limitante inferior do problema.

Como o algoritmo utiliza uma heurística de boa qualidade para obter a solução inicial é também relatado o número de problemas para os quais a heurística forneceu a solução ótima diretamente, ou seja, a solução heurística está a menos de 1% do limitante inferior lagrangeano do problema, também fornecido pela heurística.

4.3.1 Resultados e Análise Grupo1

O Grupo1 de problemas - 8 itens e 8 períodos - foi gerado para definir a melhor regra de ramificação para o algoritmo proposto. Para analisar as regras de ramificação apresentadas no Capítulo 3 foram gerados somente problemas factíveis. Os resultados obtidos foram comparados aos apresentados por Diaby et al. (1992a).

Os resultados foram analisados segundo as seguintes medidas de desempenho:

- NPH** - número de problemas para os quais a solução inicial fornecida pela heurística está a menos de 1% do limitante inferior lagrangeano do problema (fornecido pela heurística), ou seja, a solução ótima é a própria solução inicial;
- NPT** - número de problemas para os quais a execução do algoritmo é interrompida pelo tempo limite;
- NN** - número médio de nós gerados (média calculada entre os problemas para os quais a heurística não encontrou diretamente a solução ótima);
- NTL** - número de problemas em que o limitante inferior da árvore *branch-and-bound* superou o limitante inferior fornecido pela heurística;
- NPI** - número de problemas infactíveis;
- GAP1** - média percentual da diferença entre a melhor solução obtida e o limitante inferior do problema, calculada para os problemas onde foi encontrada a solução ótima. Esta diferença é expressa como:

$$\text{GAP1} = \frac{Z^* - Z_L}{Z_L} \cdot 100,$$

onde

Z^* é a solução ótima para o problema,

Z_L é o limitante inferior para o problema;

- GAP2** - indica a média percentual da diferença entre o custo da melhor solução obtida e o limitante inferior do problema. Esta é calculada para os problemas onde não foi possível garantir que a solução encontrada é ótima. A diferença é calculada de maneira idêntica ao GAP1.
- Taxa1** - diferença percentual entre a solução incumbente e a solução do problema resolvido no nó zero da árvore *branch-and-bound*; esta diferença é calculada para os problemas onde a solução inicial não está a menos de 1% do limitante inferior lagrangeano. Calculada por:

$$\text{Taxa1} = \frac{\bar{Z} - Z_{LO}}{Z_{LO}} \cdot 100$$

onde

\bar{Z} é a solução incumbente do problema

Z_{LO} é o limitante inferior para o problema no nó zero da árvore *branch-and-bound*;

CMU - capacidade média utilizada ao longo do horizonte de planejamento;

Tempo - tempo médio (em segundos) necessário para resolver um problema (a média foi calculada considerando-se somente os problemas para os quais a solução ótima foi obtida).

Na tabela a seguir são apresentados os resultados obtidos para os exemplos gerados com baixo custo de preparação de máquina.

| Cap. | Demanda | Regra | NPH | NPT | NTL | NN | GAP1 (%) | GAP2 (%) | Taxal (%) | CMU (%) | Tempo (s) |
|----------|-----------|-------|-----|-----------|-----|-------|----------|----------|-----------|---------|------------|
| Normal | Crescente | 1 | 3 | 0 | 2 | 18550 | 0.05 | --- | 127 | 83 | 632 |
| | | 2 | 3 | 1 | 1 | 26755 | 0.26 | 1.06 | 127 | 83 | 442 |
| | | 3 | 3 | 2 | 0 | 42510 | 0.08 | 1.55 | 127 | 83 | --- |
| | | 4 | 3 | 0 | 2 | 11980 | 0.17 | --- | 127 | 83 | 350 |
| | | 5 | 3 | 2 | 1 | 40220 | 0.08 | 1.04 | 127 | 83 | --- |
| | | 6 | 3 | 0 | 2 | 22360 | 0.05 | --- | 127 | 83 | 715 |
| | | 7 | 3 | 1 | 1 | 28625 | 0.06 | 1.06 | 127 | 83 | 230 |
| Normal | Constante | 1 | 4 | 0 | 1 | 40050 | 0.26 | --- | 129 | 82 | 1565 |
| | | 2 | 4 | 1 | 0 | 38510 | 0.12 | 1.79 | 129 | 82 | --- |
| | | 3 | 4 | 1 | 0 | 37670 | 0.12 | 1.79 | 129 | 82 | --- |
| | | 4 | 4 | 0 | 1 | 2060 | 0.20 | --- | 129 | 82 | 60 |
| | | 5 | 4 | 1 | 0 | 42710 | 0.12 | 1.79 | 129 | 82 | --- |
| | | 6 | 4 | 0 | 1 | 5600 | 0.14 | --- | 129 | 82 | 170 |
| | | 7 | 4 | 1 | 0 | 46220 | 0.12 | 1.79 | 129 | 82 | --- |
| Apertada | Crescente | 1 | 0 | 1 | 4 | 31420 | 0.0 | 3.82 | 129 | 90 | 850 |
| | | 2 | 0 | 3 | 2 | 35100 | 0.0 | 2.73 | 129 | 90 | 1130 |
| | | 3 | 0 | 5 | 0 | 38700 | 0.0 | 3.39 | 129 | 90 | --- |
| | | 4 | 0 | 1 | 4 | 20530 | 0.0 | 1.79 | 129 | 90 | 300 |
| | | 5 | 0 | 4 | 1 | 39465 | 0.0 | 3.28 | 129 | 90 | 505 |
| | | 6 | 0 | 2 | 3 | 37270 | 0.0 | 2.83 | 129 | 90 | 790 |
| | | 7 | 0 | 2 | 3 | 31095 | 0.0 | 2.64 | 129 | 90 | 535 |
| Apertada | Constante | 1 | 2 | 2* | 1 | 26810 | 0.0 | 3.00 | 100 | 89 | 95 |
| | | 2 | 2 | 3* | 0 | 39290 | 0.0 | 3.23 | 100 | 89 | --- |
| | | 3 | 2 | 3* | 0 | 36765 | 0.0 | 3.23 | 100 | 89 | --- |
| | | 4 | 2 | 0 | 3* | 4840 | 0.0 | --- | 100 | 89 | 180 |
| | | 5 | 2 | 3* | 0 | 36795 | 0.0 | 3.23 | 100 | 89 | --- |
| | | 6 | 2 | 0 | 3* | 10680 | 0.0 | --- | 100 | 89 | 410 |
| | | 7 | 2 | 2* | 1 | 28240 | 0.0 | 3.33 | 100 | 89 | 185 |

Tabela 1 - Problemas com Custo de Preparação Baixo

* - para um desses problemas o algoritmo obteve uma solução 0,5% melhor que a solução fornecida pela heurística.

A fim de facilitar a comparação entre as regras foi traçado o gráfico da Figura 12. A classe de problemas gerados com capacidade apertada e demanda crescente foi escolhida para traçar o gráfico por apresentar o maior número de problemas não resolvidos diretamente pela heurística. As linhas deste gráfico representam os cinco problemas testados, onde para cada regra de ramificação é apresentado o tempo necessário para resolver cada um dos problemas. Como pode-se notar as linhas se concentram na quarta regra de ramificação, onde o tempo de resolução para os cinco problemas é inferior ao tempo computado para as demais regras.

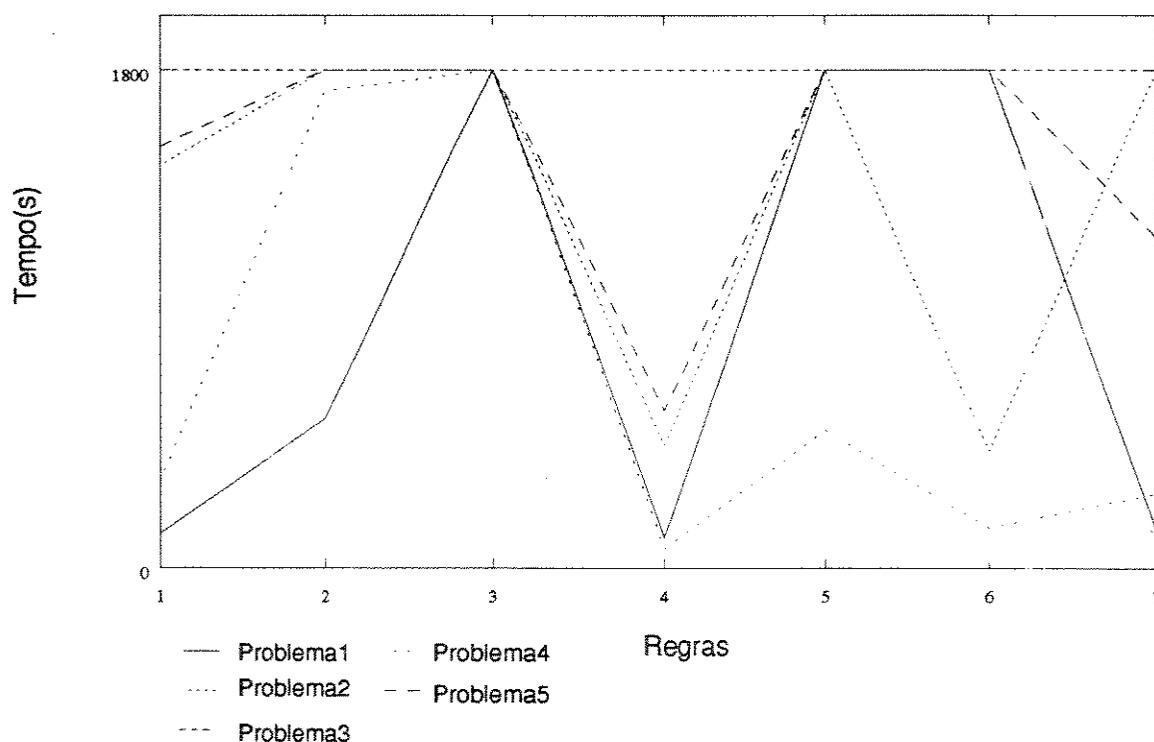


Figura 12 - Gráfico para Custo de Preparação Baixo

Em relação ao tempo para as quatro classes de problemas, as duas melhores regras foram: 4 e 7, 4 e 6, 4 e 5, e 4 e 1, como destacado em negrito na Tabela 1. Para a classe de problemas com capacidade normal e demanda constante a quarta regra é quase três vezes mais eficiente que a regra sete, a segunda melhor. Para as demais classes o tempo apresentado pela quarta regra é aparentemente superior ao tempo da regra de melhor desempenho. No entanto, esta constatação imediata é errônea uma vez que a regra quatro apresenta um número menor de problemas para os quais a otimalidade

não foi atingida (ver NPT Tabela 1) e isoladamente para cada problema oferece um desempenho sempre superior as demais (ver Figura 12).

Portanto, como pode-se notar, a regra quatro é a mais eficiente para problemas com baixo custo de preparação de máquina, apresentando ainda um comportamento bastante estável para todas as classes deste problema. Em resumo, para a regra quatro o algoritmo proposto foi capaz de encontrar a solução ótima para 90% dos problemas gerados com custo de preparação de máquina baixo, sendo necessário em média 220 segundos para garantir a otimalidade. Pode-se concluir ainda que devido a boa qualidade da solução heurística, mesmo para os problemas onde a otimalidade da solução não é garantida, foi possível verificar que a solução obtida esteve sempre a menos de 4% do limitante inferior do problema. Outro dado relevante é que em 85% dos casos a solução fornecida pela heurística foi a solução ótima para os problemas, sendo que em apenas 5% dos problemas o algoritmo obteve uma solução de melhor qualidade, embora as variáveis de decisão (y_{it}) da solução fornecida pela heurística sejam as mesmas da solução ótima obtida.

Na Tabela 2 são apresentados os resultados obtidos para problemas com alto custo de preparação de máquina. Para estes problemas um novo dado foi acrescentado à tabela, a fim de facilitar a comparação entre as regras. Seja:

Taxa2 - diferença percentual média entre a solução incumbente e o valor do limitante inferior da árvore *branch-and-bound*, calculado por:

$$\text{Taxa2} = \frac{\bar{Z} - Z_{LA}}{Z_{LA}} \cdot 100$$

onde

Z_{LA} é o limitante inferior para a árvore *branch-and-bound*.

Esta taxa foi utilizada para facilitar a comparação entre as regras uma vez que para 30% dos problemas o algoritmo proposto foi interrompido ao atingir o limite de tempo determinado, não obtendo a solução ótima.

| Cap. | Demanda | Regra | NPH | NPT | NTL | NN | GAP1 (%) | GAP2 (%) | Taxa1 (%) | Taxa2 (%) | CMU (%) | Tempo (s) |
|----------|-----------|-------|-----|-----------|-----|-------|----------|-------------|-----------|-------------|---------|-------------|
| Normal | Crescente | 1 | 5 | 0 | 0 | --- | 0.31 | --- | --- | --- | 77.5 | --- |
| | | 2 | 5 | 0 | 0 | --- | 0.31 | --- | --- | --- | 77.5 | --- |
| | | 3 | 5 | 0 | 0 | --- | 0.31 | --- | --- | --- | 77.5 | --- |
| | | 4 | 5 | 0 | 0 | --- | 0.31 | --- | --- | --- | 77.5 | --- |
| | | 5 | 5 | 0 | 0 | --- | 0.31 | --- | --- | --- | 77.5 | --- |
| | | 6 | 5 | 0 | 0 | --- | 0.31 | --- | --- | --- | 77.5 | --- |
| | | 7 | 5 | 0 | 0 | --- | 0.31 | --- | --- | --- | 77.5 | --- |
| Normal | Constante | 1 | 4 | 1 | 0 | 39145 | 0.27 | 1.96 | 69.4 | 19.0 | 74 | --- |
| | | 2 | 4 | 1 | 0 | 40500 | 0.27 | 1.96 | 69.4 | 31.8 | 74 | --- |
| | | 3 | 4 | 1 | 0 | 39320 | 0.27 | 1.96 | 69.4 | 34.3 | 74 | --- |
| | | 4 | 4 | 1 | 0 | 38900 | 0.27 | 1.96 | 69.4 | 19.9 | 74 | --- |
| | | 5 | 4 | 1 | 0 | 35080 | 0.27 | 1.96 | 69.4 | 28.4 | 74 | --- |
| | | 6 | 4 | 1 | 0 | 38300 | 0.27 | 1.96 | 69.4 | 24.7 | 74 | --- |
| | | 7 | 4 | 1 | 0 | 40500 | 0.27 | 1.96 | 69.4 | 23.2 | 74 | --- |
| Apertada | Crescente | 1 | 2 | 3* | 0 | 34930 | 0.14 | 1.30 | 84.6 | 29.0 | 84 | --- |
| | | 2 | 2 | 3* | 0 | 35765 | 0.14 | 1.30 | 84.6 | 40.1 | 84 | --- |
| | | 3 | 2 | 3* | 0 | 37040 | 0.14 | 1.30 | 84.6 | 47.1 | 84 | --- |
| | | 4 | 2 | 3* | 0 | 36990 | 0.14 | 1.30 | 84.6 | 35.4 | 84 | --- |
| | | 5 | 2 | 3* | 0 | 33830 | 0.14 | 1.30 | 84.6 | 43.2 | 84 | --- |
| | | 6 | 2 | 3* | 0 | 35570 | 0.14 | 1.30 | 84.6 | 40.7 | 84 | --- |
| | | 7 | 2 | 3* | 0 | 36740 | 0.14 | 1.30 | 84.6 | 30.6 | 84 | --- |
| Apertada | Constante | 1 | 2 | 2 | 1 | 35695 | 0.78 | 1.14 | 47.5 | 10.1 | 82 | 1225 |
| | | 2 | 2 | 3 | 0 | 38955 | 0.78 | 1.78 | 47.5 | 18.5 | 82 | --- |
| | | 3 | 2 | 3 | 0 | 38850 | 0.78 | 1.78 | 47.5 | 21.5 | 82 | --- |
| | | 4 | 2 | 3 | 1 | 38690 | 0.78 | 1.27 | 47.5 | 12.9 | 82 | --- |
| | | 5 | 2 | 3 | 1 | 36420 | 0.78 | 1.41 | 47.5 | 16.6 | 82 | --- |
| | | 6 | 2 | 2 | 1 | 38290 | 0.70 | 1.15 | 47.5 | 14.6 | 82 | 1720 |
| | | 7 | 2 | 2 | 1 | 35870 | 0.75 | 1.15 | 47.5 | 11.7 | 82 | 1280 |

Tabela 2 - Problemas com Custo de Preparação Alto

* - para um desses problemas o algoritmo obteve uma solução melhor que a solução fornecida pela heurística.

O gráfico da Figura 13 foi traçado de modo análogo ao gráfico anterior, onde, no entanto, o tempo é substituído pela Taxa2, que permite avaliar o quão distante o limitante inferior da árvore está da solução incumbente do problema. Pode-se assim estimar a regra pela qual a solução ótima será atingida mais facilmente. Para traçar este gráfico foi escolhida a classe de problemas gerados com capacidade apertada e demanda crescente, dado que esta classe apresenta o maior número de problemas para os quais o algoritmo teve sua execução interrompida pelo tempo limite. As linhas do gráfico representam os problemas para os quais a heurística não forneceu a solução ótima diretamente. Como pode-se notar a regra 1 fornece valores menores para a Taxa2, podendo ser considerada a melhor regra para esta classe de problemas.

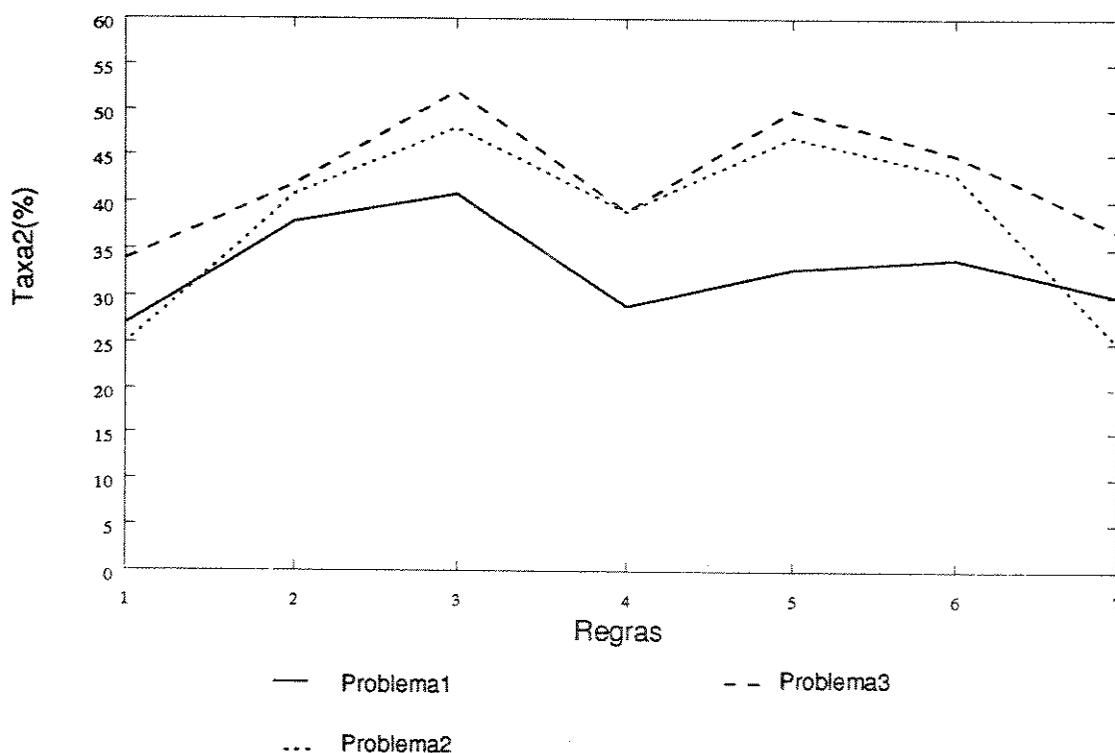


Figura 13 - Gráfico para Custo Preparação Alto

Dentre as quatro classes de problemas, a avaliação da melhor regra foi realizada com base na três últimas, uma vez que para a primeira a solução inicial fornecida pela heurística foi a solução ótima para todos os problemas. Considerando as três últimas classes, as duas melhores regras formam: 1 e 4, 1 e 7, e 1 e 7. A primeira regra se

mostrou sempre superior as demais, portanto, pode-se concluir que para problemas com alto custo de preparação de máquina a primeira regra é a que melhor se adapta.

Constatou-se ainda que apenas para 5% dos exemplos o algoritmo obteve uma solução melhor que a solução inicial fornecida pela heurística, e ainda, que mesmo para os problemas em que a otimalidade não pôde ser garantida, a solução incumbente encontrada sempre se encontra a menos de 2% do valor do limitante inferior do problema.

Diaby et al. (1992a) propuseram um algoritmo exato para o qual foram apresentados resultados de 16 problemas resolvidos em um computador CDC Cyber 730. Os resultados obtidos revelam que, para os exemplos testados, os GAPs foram sempre inferiores a 1% e o tempo computacional necessário para resolver os exemplos é em média igual a 40 segundos. O algoritmo proposto neste trabalho pode ser considerado bom comparado ao anterior, pois seus GAPs também são inferiores a 1% e o tempo computacional foi em média 220 segundos Sun modelo SPARCstation 10. Considerando que os problemas e os computadores são distintos, pode-se concluir que ambos apresentam bons resultados. Para problemas com alto custo de preparação de máquina os resultados do algoritmo proposto não puderam ser comparados aos do algoritmo apresentado por Diaby et al. (1992a), pois os autores não especificaram as faixas de geração dos parâmetros.

4.3.2 Resultados e Análise Grupo2

Uma vez definidas as melhores regras de ramificação para problemas com alto e baixo custo de preparação de máquina, passou-se ao teste dos problemas do Grupo2. Este grupo de problemas foi gerado para avaliar o desempenho do algoritmo proposto de acordo com os fatores mencionados anteriormente. Diaby et al. (1992a) não apresentam resultados para este grupo de problemas não sendo possível estabelecer uma comparação para este caso. As tabelas a seguir mostram os resultados dos testes para cada um dos diferentes pares (N,T) de problemas gerados, sendo em seguida feita uma análise dos resultados obtidos.

| Demanda | T | N | NPH | NPT | NTL | NPI | NMT | NN | GAP1 (%) | GAP2 (%) | Taxa1 (%) | Taxa2 (%) | CMU (%) | Tempo (s) |
|-----------|----|----|-----|-----|-----|-----|-----|-------|----------|----------|-----------|-----------|---------|-----------|
| Crescente | 6 | 10 | 4 | 1 | 1 | 0 | 0 | 5770 | 0.14 | --- | 78 | 1 | 83 | 150 |
| | | 25 | 5 | 0 | --- | 0 | 0 | --- | 0.06 | --- | --- | --- | 83 | --- |
| | | 50 | 5 | 0 | --- | 0 | 0 | --- | 0.08 | --- | --- | --- | 83 | --- |
| | 12 | 10 | 0 | 5 | 0 | 0 | 2 | 22040 | --- | 2.72 | 218 | 48 | 83 | --- |
| | | 25 | 4 | 1 | 0 | 0 | 0 | 5590 | 0.71 | 1.26 | 190 | 111 | 83 | --- |
| Constante | 6 | 10 | 5 | 0 | --- | 0 | 0 | --- | 0.05 | --- | --- | --- | 82 | --- |
| | | 25 | 5 | 0 | --- | 0 | 0 | --- | 0.00 | --- | --- | --- | 82 | --- |
| | | 50 | 5 | 0 | --- | 0 | 0 | --- | 0.00 | --- | --- | --- | 82 | --- |
| | 12 | 10 | 3 | 2 | 0 | 0 | 0 | 22400 | 0.32 | 1.22 | 180 | 38 | 82 | --- |
| | | 25 | 5 | 0 | --- | 0 | 0 | --- | 0.00 | --- | --- | --- | 82 | --- |

Tabela 3 - Problemas Custo Baixo e Capacidade Normal

| Demanda | T | N | NPH | NPT | NTL | NPI | NMT | NN | GAP1 (%) | GAP2 (%) | Taxa1 (%) | Taxa2 (%) | CMU (%) | Tempo (s) |
|-----------|----|----|-----|-----|-----|-----|-----|-------|----------|----------|-----------|-----------|---------|-----------|
| Crescente | 6 | 10 | 3 | 0 | 2 | 0 | 0 | 6190 | 0.47 | --- | 96 | 1 | 90 | 170 |
| | | 25 | 4 | 1 | 0 | 0 | 0 | 14615 | 0.76 | 1.32 | 83 | 27 | 90 | --- |
| | | 50 | 5 | 0 | --- | 0 | 0 | --- | 0.19 | --- | --- | --- | 90 | --- |
| | 12 | 10 | 0 | 5 | 0 | 0 | 2 | 18880 | --- | 4.73 | 178 | 65 | 89 | --- |
| | | 25 | 0 | 5 | 0 | 0 | 3 | 4380 | --- | 1.88 | 148 | 123 | 89 | --- |
| Constante | 6 | 10 | 3 | 0 | 1 | 1 | 0 | 4170 | 0.32 | --- | 70 | 1 | 89 | 110 |
| | | 25 | 4 | 0 | --- | 1 | 0 | --- | 0.13 | --- | --- | --- | 89 | --- |
| | | 50 | 5 | 0 | --- | 0 | 0 | --- | 0.03 | --- | --- | --- | 89 | --- |
| | 12 | 10 | 2 | 1 | 0 | 2 | 0 | 23000 | 0.74 | 1.27 | 194 | 48 | 90 | --- |
| | | 25 | 4 | 0 | 0 | 1 | 0 | --- | 0.15 | --- | --- | --- | 89 | --- |

Tabela 4 - Problemas Custo Baixo e Capacidade Apertada

| Demanda | T | N | NPH | NPT | NTL | NPI | NMT | NN | GAP1 (%) | GAP2 (%) | Taxa1 (%) | Taxa2 (%) | CMU (%) | Tempo (s) |
|-----------|----|----|-----|-----|-----|-----|-----|-------|----------|----------|-----------|-----------|---------|-----------|
| Crescente | 6 | 10 | 5 | 0 | --- | 0 | 0 | --- | 0.19 | --- | --- | --- | 77 | --- |
| | | 25 | 5 | 0 | --- | 0 | 0 | --- | 0.07 | --- | --- | --- | 78 | --- |
| | | 50 | 5 | 0 | --- | 0 | 0 | --- | 0.03 | --- | --- | --- | 78 | --- |
| | 12 | 10 | 5 | 0 | --- | 0 | 0 | --- | 0.31 | --- | --- | --- | 79 | --- |
| | | 25 | 5 | 0 | --- | 0 | 0 | --- | 0.20 | --- | --- | --- | 79 | --- |
| Constante | 6 | 10 | 4 | 1 | 0 | 0 | 0 | 40345 | 0.51 | 1.76 | 44 | 5 | 75 | --- |
| | | 25 | 5 | 0 | --- | 0 | 0 | --- | 0.16 | --- | --- | --- | 76 | --- |
| | | 50 | 5 | 0 | --- | --- | 0 | --- | 0.10 | --- | --- | --- | 75 | --- |
| | 12 | 10 | 3 | 2 | 0 | 1 | 0 | 20760 | 0.30 | 1.01 | 104 | 50 | 74 | --- |
| | | 25 | 4 | 1 | 0 | 0 | 0 | 4860 | 0.07 | 1.39 | 78 | 62 | 75 | --- |

Tabela 5 - Problemas Custo Alto e Capacidade Normal

| Demanda | T | N | NPH | NPT | NTL | NPI | NMT | NN | GAP1 (%) | GAP2 (%) | Taxa1 (%) | Taxa2 (%) | CMU (%) | Tempo (s) |
|-----------|----|----|-----|-----|-----|-----|-----|-------|----------|----------|-----------|-----------|---------|-----------|
| Crescente | 6 | 10 | 5 | 0 | --- | 0 | 0 | --- | 0.2 | --- | --- | --- | 84 | --- |
| | | 25 | 5 | 0 | --- | 0 | 0 | --- | 0.08 | --- | --- | --- | 85 | --- |
| | | 50 | 5 | 0 | --- | 0 | 0 | --- | 0.09 | --- | --- | --- | 85 | --- |
| | 12 | 10 | 4 | 1 | 0 | 0 | 0 | 18570 | 0.65 | 1.29 | 145 | 81 | 85 | --- |
| | | 25 | 5 | 0 | --- | 0 | 1 | --- | 0.27 | --- | --- | --- | 85 | --- |
| Constante | 6 | 10 | 2 | 1 | 1 | 1 | 1 | 23000 | 0.57 | 1.64 | 28 | 1.5 | 83 | 180 |
| | | 25 | 4 | 0 | --- | 1 | 0 | --- | 0.28 | --- | --- | --- | 83 | --- |
| | | 50 | 4 | 1 | 0 | 0 | 1 | 4070 | 0.12 | 2.79 | 26 | 21 | 83 | --- |
| | 12 | 10 | 3 | 0 | --- | 2 | 1 | --- | 0.59 | --- | --- | --- | 81 | --- |
| | | 25 | 4 | 0 | --- | 1 | 1 | --- | 0.34 | --- | --- | --- | 82 | --- |

Tabela 6 - Problemas Custo Alto e Capacidade Apertada

Tendo como base os resultados das Tabelas de 3 à 6 serão apresentadas análises em relação aos três fatores variantes do problema: capacidade, custo de preparação de máquina e demanda.

■ Capacidade

Quando a capacidade disponível para os problemas é reduzida nota-se que em geral:

1. aumenta o número de problemas infactíveis
2. aumenta o número de problemas em que o algoritmo pára ao atingir o tempo limite
3. aumentam os GAPs
4. melhora a qualidade do limitante inferior obtido no nó zero da árvore *branch-and-bound*, ou seja, diminui a Taxa1
5. aumenta a Taxa2
6. piora a qualidade da solução inicial.

Pode-se notar também que o número de problemas onde o algoritmo obteve uma solução melhor que a solução fornecida pela heurística foi maior para os problemas mais apertados.

A solução inicial de pior qualidade conduz ao aumento no número de problemas em que o algoritmo para ao atingir o tempo limite e ao aumento nos GAPs obtidos.

A qualidade do limitante inferior do nó zero da árvore *branch-and-bound* melhora devido ao acréscimo do custo unitário relaxado \bar{c}_{it}^r (pág.32) para os primeiros períodos do horizonte, onde $\Delta_{it} = C_i - f_{it}$.

Os problemas mais apertados demoram mais para serem resolvidos, logo, um número menor de nós é percorrido, refletindo na pior qualidade da Taxa2.

■ Custo de Preparação de Máquina

Diante do aumento do custo de preparação de máquina pode-se observar que:

1. o número de problemas inactíveis é o mesmo, pois os problemas gerados com custo baixo e alto são idênticos a menos do custos de preparação (multiplicados por 10)
2. diminui o número de problemas em que o algoritmo pára ao atingir o tempo limite
3. diminuem os GAPs
4. melhora a qualidade do limitante inferior obtido no nó zero da árvore *branch-and-bound*, ou seja, diminui a Taxa1
5. aumenta a Taxa2
6. melhora a qualidade da solução inicial.

A boa qualidade da solução inicial resulta em GAPs melhores e num menor número de problemas para os quais o algoritmo pára ao atingir o tempo limite.

Quando o custo de preparação é alto o crescimento do limitante inferior da árvore é bem mais lento resultando em uma Taxa2 de pior qualidade.

■ Demanda

Ao variar o tipo de demanda pode-se observar um aumento no número de problemas inactíveis para os problemas com demanda constante. Este aumento se deve ao modo como é feito o cálculo da capacidade, média da capacidade utilizada em todos os períodos (lote-por-lote), pois para os primeiros períodos onde muitos itens são produzidos, a capacidade não é suficiente, provocando a inactibilidade do problema. O mesmo não ocorre para problemas com demanda crescente, pois estes tem o maior volume de produção concentrado nos últimos períodos, que podem ser atendidos via estoque.

Observa-se que para os problemas com demanda constante a solução do problema relaxado do nó zero da árvore *branch-and-bound* é de melhor qualidade.

O comportamento da heurística é diferente para problemas com custos de preparação de máquina distintos. Para aqueles com custo alto de preparação, a heurística apresentou um comportamento melhor para os problemas com demanda crescente refletindo em GAPs menores para estes problemas. Para os problemas com baixo custo de preparação o resultado foi inverso, sendo o comportamento da heurística melhor para os problemas com demanda constante, que por consequência apresentam GAPs menores.

A partir dos resultados obtidos foi feita uma análise do comportamento dos problemas segundo sua dimensão. O aumento da dimensão foi analisado sob dois aspectos: número de itens e número de períodos.

■ Itens

À medida que o número de itens é aumentado pode-se observar que:

1. diminui o número de problemas infactíveis
2. diminui o número de problemas em que o algoritmo pára ao atingir o tempo limite
3. diminui os GAPs
4. o limitante inferior obtido no nó zero da árvore *branch-and-bound* melhora, ou seja, diminui a Taxa1
5. aumenta a Taxa2
6. melhora a qualidade da solução inicial fornecida pela heurística.

Devido à melhor qualidade da solução inicial, os GAPs e o número de problemas em que o algoritmo pára ao atingir o tempo limite diminuem. Como os problemas com mais itens são maiores, e portanto de resolução mais lenta, constata-se um menor número de nós resolvidos durante o tempo limite, o que resulta em uma Taxa2 de qualidade inferior.

■ Períodos

Pode-se observar que à medida em que aumenta o número de períodos:

1. aumenta o número de problemas inactíveis
2. aumenta o número de problemas em que o algoritmo pára ao atingir o tempo limite
3. aumentam os GAPS
4. piora a qualidade do limitante inferior obtido no nó zero da árvore *branch-and-bound* , ou seja, aumenta a $Taxa_1$
5. aumenta a $Taxa_2$
6. piora a qualidade da solução inicial.

Como para as análises anteriores a qualidade da solução inicial influencia nos GAPS dos problemas e no número de problemas parados ao atingir o tempo limite.

A pior qualidade do limitante inferior obtido no nó zero se deve a distribuição da produção ao longo do horizonte, provocando muitas preparações da máquina.

À medida que o horizonte de planejamento é ampliado as possibilidades de atender a demanda via estoque aumentam: a demanda do item i no período t pode ser atendida pela produção em um ou mais dos $t-1$ períodos anteriores a t . Logo, quanto maior for o horizonte, maior o número de possibilidades para o atendimento da demanda, o que resulta numa $Taxa_2$ maior.

Pode-se concluir então que o número de períodos do problema prejudica muito mais o desempenho do algoritmo que o número de itens.

4.3.3 Resultados e Análise Grupo3

O terceiro grupo de problemas testados foi gerado como definido anteriormente, com uma ressalva: os custos de preparação de máquina foram desprezados. Os resultados obtidos são apresentados nas Tabelas 7 e 8.

| Demanda | T | N | NPH | NPT | NTL | NPI | NMT | NN | GAP1 (%) | GAP2 (%) | Taxa1 (%) | Taxa2 (%) | CMU (%) | Tempo (s) |
|-----------|----|----|-----|-----|-----|-----|-----|-----|----------|----------|-----------|-----------|---------|-----------|
| Crescente | 8 | 50 | 0 | 1 | 4 | 0 | 0 | 560 | 0.0 | 1.9 | 24 | 10 | 83 | 250 |
| | | 99 | 2 | 2 | 1 | 0 | 0 | 430 | 0.3 | 4.9 | 55 | 53 | 83 | 10 |
| | 12 | 50 | 1 | 4 | 0 | 0 | 0 | 790 | 0.9 | 2.5 | 120 | 99 | 83 | --- |
| Constante | 8 | 50 | 5 | 0 | --- | 0 | 0 | --- | 0.0 | --- | --- | --- | 83 | --- |
| | | 99 | 5 | 0 | --- | 0 | 0 | --- | 0.0 | --- | --- | --- | 83 | --- |
| | 12 | 50 | 5 | 0 | --- | 0 | 0 | --- | 0.0 | --- | --- | --- | 83 | --- |

Tabela 7 - Problemas com Capacidade Normal

| Demanda | T | N | NPH | NPT | NTL | NPI | NMT | NN | GAP1 (%) | GAP2 (%) | Taxa1 (%) | Taxa2 (%) | CMU (%) | Tempo (s) |
|-----------|----|----|-----|-----|-----|-----|-----|------|----------|----------|-----------|-----------|---------|-----------|
| Crescente | 8 | 50 | 1 | 4 | 0 | 0 | 0 | 1460 | 0.6 | 2.1 | 92 | 68 | 90 | --- |
| | | 99 | 2 | 3 | 0 | 0 | 0 | 330 | 0.6 | 1.4 | 172 | 137 | 90 | --- |
| | 12 | 50 | 0 | 5 | 0 | 0 | 3 | 500 | --- | 2.4 | 118 | 105 | 89 | --- |
| Constante | 8 | 50 | 4 | 1 | 0 | 0 | 0 | 100 | 0.0 | 2.0 | 90 | 70 | 91 | --- |
| | | 99 | 3 | 2 | 0 | 0 | 0 | 715 | 0.0 | 3.5 | --- | --- | 91 | --- |
| | 12 | 50 | 4 | 1 | 0 | 0 | 0 | 310 | 0.0 | 6.0 | --- | --- | 91 | --- |

Tabela 8 - Problemas com Capacidade Apertada

Nas tabelas anteriores a Taxa1 e a Taxa2 não foram calculadas para os problemas onde o limitante inferior no nó zero da árvore e o limitante inferior da árvore *branch-and-bound* são nulos, respectivamente.

Pode-se concluir pelos resultados obtidos que, em relação à qualidade da solução, o algoritmo proposto teve um comportamento bastante semelhante ao algoritmo apresentado por Diaby et al., enquanto que em relação ao tempo mostrou-se um pouco inferior.

A heurística utilizada para obter a solução inicial apresentou um comportamento muito bom, obtendo para 53% dos problemas a solução ótima diretamente. Deve-se ainda destacar que o tempo necessário para executar a heurística foi sempre inferior a 10 segundos em todos os problemas testados nos grupos 1, 2 e 3.

CAPÍTULO 5

CONCLUSÕES

O objetivo deste trabalho foi propor um método exato para o problema de dimensionamento de lotes com múltiplos itens em um horizonte de planejamento finito. O modelo tratado considera tempos de preparação de máquina e limitação da capacidade de produção disponível.

O algoritmo proposto foi testado em três etapas com os seguintes objetivos:

1. Avaliar a regra de escolha da variável para ramificação com melhor desempenho para os problemas;
2. Observar o comportamento do problema com relação a dimensão, capacidade, custo de preparação e demanda do problema são alteradas.
3. Comparar o comportamento do algoritmo proposto com o algoritmo apresentado por Diaby et al. (1992a).

4. Avaliar o desempenho da heurística proposta por Trigeiro.

Os resultados observados podem ser resumidos em:

- A qualidade da solução do problema relaxado é muito pobre e dificulta a resolução dos problemas.
- O número de períodos do horizonte de planejamento e o número de itens do problema influenciam de maneira distinta tanto na qualidade da solução quanto no desempenho computacional do algoritmo.
- O algoritmo proposto se comportou tão bem quanto o algoritmo apresentado por Diaby et al. (1992a) para problemas com 8 itens e 8 períodos (Grupo1). O Grupo2 não pode ser comparado por falta de resultados para o método proposto pelos autores. Para os problemas de maior dimensão e sem tempo de preparação (Grupo3), os resultados foram semelhantes em relação à qualidade da solução, mas mais pobres em comparação ao tempo computacional.
- A heurística apresentou um bom comportamento para os problemas testados.

Novas pesquisas podem ser realizadas para melhorar o desempenho do algoritmo proposto. Dentre estas podem ser citadas:

- Desenvolvimento de uma regra de escolha de variável de ramificação adaptativa a fim de melhorar o desempenho do algoritmo.
- Pesquisar a viabilidade de reotimização através do problema dual.

Além disso, o algoritmo proposto pode ser estendido para o problema de dimensionamento de lotes capacitado com máquinas paralelas, permitindo assim a avaliação da heurística proposta no trabalho apresentado por Sabbag (1993).

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ACKOFF, R.L., *Redesigning the future; a systems approach to societal problems*, New York, John Wiley & Sons in [Oliveira,1986].
- [2] AFENTAKIS, P., GAVISH, B., (1986), *Optimal Lot-Sizing Algorithms for Complex Product Structures*, *Operations Research*, 34(2), 237-249.
- [3] BARANY, I., van ROY, T.J., WOLSEY, L.A., (1984), *Strong Formulations for Multi-Item Capacitated Lot Sizing*, *Management Science*, 30(10), 1255-1261.
- [4] BOWMAN, E.H., (1956), *Production Scheduling by the Transportation Method of Linear Programming*, *Operations Research*, 3(1), 100-103.
- [5] CORREIA, P.B., (1988), *Um Modelo Multisetorial para Otimização do Suprimento de Energia - Eletricidade, Gás Natural e Cogeração com Biomassa*, Tese de Doutorado FEE-UNICAMP.
- [6] DIABY, M., BAHL, H.C., KARWAN, M.H., ZIONTS, S., (1992a), *Capacitated Lot-Sizing and Scheduling by Lagrangean Relaxation*, *European Journal of Operational Research*, 59, 444-458.
- [7] DIABY, M., BAHL, H.C., KARWAN, M.H., ZIONTS, S., (1992b), *A Lagrangean Relaxation Approach for Very-Large-Scale Capacitated Lot-Sizing*, *Management Science*, 38(9), 1329-1340.
- [8] EPPEN, G.D., MARTIN, R.K., (1987), *Solving Multi-Item Capacitated Lot Sizing Problems using Variable Redefinition*, *Operations Research*, 35(6), 832-848.
- [9] EVANS, J.R., (1985), *Network-Based Optimization Algorithms for the Capacitated Multi-Item Lot Sizing Problem*, *Comput. & Indus. Engng.*, 9(3), 297-305.
- [10] FLORIAN, M., KLEIN, (1974), *Deterministic Production Planning with Concave Costs and Capacity Constraints*, *Management Science*, 18(1), 12-20.

- [11] FLORIAN, M., LENSTRA, J.K., RINNOY KAN, A.H.G., (1980), *Deterministic Production Planning: Algorithms and Complexity*, Management Science, 26(7), 669-679.
- [12] GAREY, M., JOHNSON, D., (1979), *Computers, Complexity and Intractability: A Guide to Theory of NP-Completeness*, W.H. Freeman, San Francisco.
- [13] JENSEN, P.A., BARNES, J.W., (1980), *Network Flow Programming*, John Wiley & Sons, New York.
- [14] JOHNSON, L.A., MONTGOMERY, D.C., (1974), *Operations Research in Production Planning, Scheduling, and Inventory Control*, John Wiley & Sons, New York.
- [15] LOVE, S.F., (1973), *Bounded Production and Inventory Models with Piecewise Concave Costs*, Management Science, 20(3), 313-318.
- [16] LOZANO, S., LARRANETA, J., ONIEVA, L., (1991), *Primal-Dual Approach to the Single Level Capacitated Lotsizing Problem*, European Journal of Operational Research 51, 354-366.
- [17] LUENBERGER, D.G., (1984), *Linear and Nonlinear Programming*, Addison-Wesley Publishing Company, New York.
- [18] MAES, J., van WASSENHOVE, L.N., (1988), *Multi-Item Single-Level Capacitated Dynamic Lot-Sizing Heuristics: a General Review*, Journal of Operational Research Society, 39(11), 991-1004.
- [19] OLIVEIRA, D.P.R., (1986), *Planejamento Estratégico - Conceitos, Metodologia e Práticas*, São Paulo, Atlas.
- [20] RITCHIW, E., TSADO, A.K., (1986), *A Review of Lot-Sizing Techniques for Deterministic Time-Varying Demand*, Production and Inventory Management, Third Quarter, 65-79.
- [21] SABBAG, Z., (1993), *Planejamento da Produção em Máquinas Paralelas sob Restrições de Capacidade*, Tese de Mestrado FEE-UNICAMP.
- [22] SCRICH, C.R., (1992), *Dimensionamento de Lotes de Múltiplos Itens com Restrições de Capacidade*, Tese de Mestrado FEE-UNICAMP.

[23] SOUZA, K.X.S., (1989), *Planejamento da Produção de Múltiplos Itens com Restrições de Capacidade Através da Decomposição Cruzada*, Tese de Mestrado FEE-UNICAMP.

[24] TRIGEIRO, W.W, THOMAS, L.J., McCLAIN, J.O., (1989), *Capacitated Lot Sizing with Setup Times*, *Management Science*, 35(3), 353-366.

[25] van ROY, T.J., (1983), *Cross Decomposition for Mixed Integer Programming*, *Mathematical Programming*, 25, 46-63.

[26] WAGNER, H.M., WHITIN, T.M., (1958), *Dynamic Version of the Economic Lot Size Model*, *Management Science*, 5(1), 89-96.

[27] ZANGWILL, W.I., (1966), *A deterministic Multi-Period Production Scheduling Model with Backlogging*, *Management Science*, 13(1), 105-119.

[28] ZANGWILL, W.I., (1966), *Single and Multi-Commodity Minimum Concave Cost Flows in Certain Networks*, Working Paper N. 154, Center for Research in Management Science, University of California, Bekeley, California, February in [Zangwill, 1969].

[29] ZANGWILL, W.I., (1969), *A Backlogging Model and a Multi-Echelon Model of a Dynamic Economic Lot Size Production System - A Network Approach*, *Management Science* 15(9), 507-527.