ESTRUTURAS ESPECIAIS EM PROGRAMAÇÃO INTEIRA

Andréa Carla Gonçalves Vianna

Orientador: Dr. Marcos Nereu Arenales

Dissertação apresentada ao Instituto de Ciências Matemáticas de São Carlos, da Universidade de São Paulo, para obtenção do Título de Mestre em Ciências de Computação e Matemática Computacional.

São Carlos - SP

Junho - 1994

Aos meus pais, Valdemir e Cleide,
e ao meu marido, Carlos,
com carinho.

AGRADECIMENTOS

Ao Professor Marcos Nereu Arenales pela orientação e amizade.

Aos amigos Marco Antônio Rahal Sacoman, Leila Haddad Sacoman e Edméa Cássia Baptista pela ajuda e amizade.

À minha família pelo apoio e incentivo.

RESUMO

ESTRUTURAS ESPECIAIS EM PROGRAMAÇÃO INTEIRA

Este trabalho apresenta métodos para a resolução de Problemas de Programação Linear Inteira Mista. Estes métodos estão baseados na decomposição primal ou dual do problema. Apresenta também, um método que utiliza ambas as decomposições em um único procedimento, chamado de Método de Decomposição Híbrida.

Além disso, este trabalho apresenta um estudo de caso sobre o Problema de Planejamento e Controle de Produção numa Fundição Automatizada. Apresenta a resolução deste problema através de um algoritmo de busca heurística e seus resultados computacionais.

ABSTRACT

SPECIAL STRUCTURES IN INTEGER PROGRAMMING

This work presents methods for solving Mixed Integer Linear Programming Problems. These methods are based on primal or dual decomposition of the problem. It also presents a method which simultaneously uses both decomposition, named the Cross Decomposition Method.

Furthermore, this work shows an application on a particular Planning and Control Production Problem and its resolution by a heuristic search algorithm with computational results on a number of examples.

SUMÁRIO

	Página
NOTAÇÃO	01
1. INTRODUÇÃO	04
2. MÉTODOS DE DECOMPOSIÇÃO	09/
2.1. Método de Decomposição de Benders	09/
2.1.1. Algoritmo	19/
2.2. Relaxação Lagrangeana	21/
2.2.1. Método do Subgradiente	24/
2.2.1.1. Algoritmo	30
3. DECOMPOSIÇÃO HÍBRIDA	31
3.1. O Princípio da Decomposição Híbrida	31
3.2. Equivalência entre os Subproblemas	33
3.3. Convergência do Método	43
3.4. Algoritmo	45
4. APLICAÇÃO: PROBLEMA DE PLANEJAMENTO E CONTROLE DE PRODUÇÃO NUMA FUNDIÇÃO AUTOMATIZADA	48
4.1. Definição do Modelo	48
4.2. Resolução do Problema	51
4.3. Algoritmo	58
4.4. Resultados Computacionais	59
4.4.1. Exemplo 1	62
4.4.2. Exemplo 2	63
4.4.3. Exemplo 3	66
5. CONCLUSÕES E PERSPECTIVAS	70
6. BIBLIOGRAFIA	72

NOTAÇÃO

A notação utilizada nos próximos capítulos deste trabalho é mostrada a seguir.

- (P) : problema de otimização.
- $(\bar{P})$: problema (P) com as restrições de integralidade relaxadas.
- $v(P)$: valor ótimo do problema (P) .
- $S(P)$: conjunto das soluções viáveis de (P) .

Os problemas definidos ao longo deste trabalho são dados a seguir.

- Subproblema Primal na variável x_1

$$\text{minimizar } c_1^T \cdot x_1 + c_2^T \cdot x_2$$

$$\text{sujeito a: } A_1 \cdot x_1 \geq b - A_2 \cdot x_2 \quad (\text{SP}x_2)$$

$$x_1 \geq 0$$

- Problema Dual do Subproblema Primal ($\text{SP}x_2$)

$$\text{maximizar } (b - A_2 \cdot x_2)^T \cdot u + c_2^T \cdot x_2$$

$$\text{sujeito a: } A_1^T \cdot u \leq c_1 \quad (\text{DSP}x_2)$$

$$u \geq 0$$

- Problema Mestre Primal

$$\text{minimizar } x_0$$

$$\text{sujeito a: } x_0 \geq c_2^T \cdot x_2 + (b - A_2 \cdot x_2)^T \cdot u^i, \quad i \in I_{VP} \quad (\text{MP})$$

$$(b - A_2 \cdot x_2)^T \cdot r^j \leq 0, \quad j \in I_{RP}$$

$$x_2 \in \mathbb{Z}^{n_2}$$

- Problema Mestre Primal Relaxado

$$\text{minimizar } x_0$$

$$\text{sujeito a: } x_0 \geq c_2^T \cdot x_2 + (b - A_2 \cdot x_2)^T \cdot u^i, \quad i \in I'_{VP} \subseteq I_{VP} \quad (\text{MPR})$$

$$(b - A_2 \cdot x_2)^T \cdot r^j \leq 0, \quad j \in I'_{RP} \subseteq I_{RP}$$

$$x_2 \in \mathbb{Z}^{n_2}$$

◦ Subproblema Dual

$$\text{minimizar } \mathbf{c}^T \cdot \mathbf{x} + (\mathbf{b}_2 - \mathbf{A}^2 \cdot \mathbf{x})^T \cdot \mathbf{u}_2$$

$$\text{sujeito a: } \mathbf{A}^1 \cdot \mathbf{x} \geq \mathbf{b}_1$$

(SDu₂)

$$\mathbf{x} \in S$$

◦ Problema Mestre Dual

$$\text{maximizar } u_0$$

$$\text{sujeito a: } \mathbf{c}^T \cdot \mathbf{x}^i + (\mathbf{b}_2 - \mathbf{A}^2 \cdot \mathbf{x}^i)^T \cdot \mathbf{u}_2 \geq u_0, \quad i \in I_{VD}$$

(MD)

$$\mathbf{u}_2 \geq 0$$

◦ Problema Mestre Dual Relaxado

$$\text{maximizar } u_0$$

$$\text{sujeito a: } \mathbf{c}^T \cdot \mathbf{x}^i + (\mathbf{b}_2 - \mathbf{A}^2 \cdot \mathbf{x}^i)^T \cdot \mathbf{u}_2 \geq u_0, \quad i \in I'_{VD} \subseteq I_{VD}$$

(MDR)

$$\mathbf{u}_2 \geq 0$$

1. INTRODUÇÃO

Em problemas de otimização busca-se minimizar ou maximizar um objetivo, que depende de um número finito de variáveis. Estas variáveis podem ser independentes umas das outras, ou podem ser relacionadas por meio de uma ou mais restrições.

Um Problema de Programação Linear Inteira Mista é um problema de otimização no qual o objetivo e as restrições são expressas como funções lineares e, apresentam restrições de integralidade para algumas variáveis.

Assim, um Problema de Programação Linear Inteira Mista é escrito na forma:

minimizar $c^T \cdot x$

sujeito a: $A \cdot x \geq b$ (P)

$x \in S$

onde $c, x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ e

$$S = \left\{ x = (x_1, x_2)^T \mid x_1 \geq 0 \text{ e } x_2 \in \mathbb{Z}^{n_2} \right\}, n_2 < n.$$

Um Problema de Programação Linear Inteira Mista onde as variáveis inteiras podem assumir somente o valor zero ou um é classificado como Problema de Programação Linear Inteira Mista Zero-Um.

Uma grande parte dos problemas práticos são formulados como sendo problemas de Programação Linear Inteira Mista. Daí a importância de algoritmos e métodos de resolução computacionalmente eficientes.

O objetivo deste trabalho é apresentar métodos eficientes para resolução de problemas de Programação Linear Inteira Mista e, fazer um estudo de caso sobre o Problema de Planejamento e Controle de Produção numa Fundição Automatizada.

Portanto, são apresentados métodos de decomposição, primal e dual, que transformam o problema original em um problema equivalente (problema mestre). Esta transformação é feita de acordo com a estrutura original do problema e de

uma estratégia de resolução, fornecendo um processo iterativo para a solução do problema mestre. É apresentado também, um método híbrido proposto por ROY (1983), que utiliza ambas as decomposições em um único procedimento.

No Capítulo 2 são desenvolvidos dois métodos de otimização, baseados na decomposição primal e dual, para resolução de problemas de Programação Linear Inteira Mista: o Método de Decomposição de Benders e o Método de Relaxação Lagrangeana.

No Método de Decomposição de Benders, o Problema de Programação Linear Inteira Mista é apresentado com a seguinte estrutura particular:

$$\begin{aligned} &\text{minimizar } c_1^T \cdot x_1 + c_2^T \cdot x_2 \\ &\text{sujeito a: } A_1 \cdot x_1 + A_2 \cdot x_2 \geq b \\ &\quad x = (x_1, x_2)^T \in S \end{aligned} \tag{P}$$

onde $c_1, x_1 \in \mathbb{R}^{n_1}$, $c_2 \in \mathbb{R}^{n_2}$, $A_1 \in \mathbb{R}^{m \times n_1}$, $A_2 \in \mathbb{R}^{m \times n_2}$, $b \in \mathbb{R}^m$, e $S = \left\{ x = (x_1, x_2)^T \mid x_1 \geq 0 \text{ e } x_2 \in \mathbb{Z}^{n_2} \right\}$.

No Método de Relaxação Lagrangeana, o Problema de Programação Linear Inteira Mista é apresentado com a seguinte estrutura particular:

$$\text{minimizar } c^T \cdot x$$

$$\text{sujeito a: } A^1 \cdot x \geq b_1 \quad (P)$$

$$A^2 \cdot x \geq b_2$$

$$x \in S$$

onde $c, x \in \mathbb{R}^n$, $A^1 \in \mathbb{R}^{m_1 \times n}$, $A^2 \in \mathbb{R}^{m_2 \times n}$, $b_1 \in \mathbb{R}^{m_1}$, e $b_2 \in \mathbb{R}^{m_2}$
e $S = \left\{ x = (x_1, x_2)^T \mid x_1 \geq 0 \text{ e } x_2 \in \mathbb{Z}^{n_2} \right\}$.

O Método de Decomposição Híbrida, que explora simultaneamente a estrutura primal e dual de um problema, é apresentado no Capítulo 3.

Neste caso, o Problema de Programação Linear Inteira Mista é apresentado em sua forma mais geral:

$$\text{minimizar } c_1^T \cdot x_1 + c_2^T \cdot x_2$$

$$\text{sujeito a: } A_1^1 \cdot x_1 + A_2^1 \cdot x_2 \geq b_1 \quad (P)$$

$$A_1^2 \cdot x_1 + A_2^2 \cdot x_2 \geq b_2$$

$$x = (x_1, x_2)^T \in S$$

onde $c_1, x_1 \in \mathbb{R}^{n_1}$, $c_2 \in \mathbb{R}^{n_2}$, $A_1^1 \in \mathbb{R}^{m_1 \times n_1}$, $A_2^1 \in \mathbb{R}^{m_1 \times n_2}$,
 $A_1^2 \in \mathbb{R}^{m_2 \times n_1}$, $A_2^2 \in \mathbb{R}^{m_2 \times n_2}$, $b_1 \in \mathbb{R}^{m_1}$, $b_2 \in \mathbb{R}^{m_2}$ e
 $S = \left\{ x = (x_1, x_2)^T \mid x_1 \geq 0 \text{ e } x_2 \in \mathbb{Z}^{n_2} \right\}$.

No Capítulo 4 é feito um estudo de caso sobre o Problema de Planejamento e Controle de Produção numa Fundi-

ção Automatizada. Um modelo matemático é apresentado para o problema e, é desenvolvido uma técnica de busca heurística para a resolução do mesmo. É apresentado também, um algoritmo para esta técnica e alguns resultados computacionais.

2. MÉTODOS DE DECOMPOSIÇÃO

Neste capítulo são apresentados dois dos principais métodos de otimização para Problemas de Programação Linear Inteira Mista. Inicialmente, é apresentado o Método de Decomposição de Benders (BENDERS, 1962; LASDON, 1970), que particiona o problema original, trabalhando com subproblemas mais fáceis de serem resolvidos. A seguir, é apresentado o Método de Relaxação Lagrangeana (GEOFFRION, 1974), que trabalha com a estrutura dual do problema, relaxando as restrições que tornam o problema difícil de ser resolvido.

2.1. Método de Decomposição de Benders

O Método de Decomposição de Benders explora a estrutura primal do Problema de Programação Linear Inteira Mista. O método fixa inicialmente valores para as variáveis

inteiras obtendo um Problema de Programação Linear. Um problema mestre coordena as mudanças nas variáveis inteiras.

Considera-se o seguinte Problema de Programação Linear Inteira Mista:

$$\begin{aligned} &\text{minimizar } c_1^T \cdot x_1 + c_2^T \cdot x_2 \\ &\text{sujeito a: } A_1 \cdot x_1 + A_2 \cdot x_2 \geq b \\ &\quad x = (x_1, x_2)^T \in S \end{aligned} \tag{P}$$

onde $c_1, x_1 \in \mathbb{R}^{n_1}$, $c_2 \in \mathbb{R}^{n_2}$, $A_1 \in \mathbb{R}^{m \times n_1}$, $A_2 \in \mathbb{R}^{m \times n_2}$, $b \in \mathbb{R}^m$,
e $S = \left\{ x = (x_1, x_2)^T \mid x_1 \geq 0 \text{ e } x_2 \in \mathbb{Z}^{n_2} \right\}$.

Ao fixar valores para x_2 , obtém-se um subproblema linear (SP x_2) em x_1 que pode ser facilmente resolvido.

$$\begin{aligned} &\text{minimizar } c_1^T \cdot x_1 + \underbrace{c_2^T \cdot x_2}_{\text{constante}} \\ &\text{sujeito a: } A_1 \cdot x_1 \geq \underbrace{b - A_2 \cdot x_2}_{\text{constante}} \\ &\quad x_1 \geq 0 \end{aligned} \tag{SP}_{x_2}$$

Porém, os valores a serem atribuídos a x_2 não podem ser arbitrários, é necessário que eles satisfaçam as restrições do problema. Portanto, x_2 deve pertencer ao conjunto

$$P = \left\{ x_2 \in \mathbb{Z}^{n_2} \mid \exists x_1 \in \mathbb{R}^{n_1} \text{ com } A_1 \cdot x_1 \geq b - A_2 \cdot x_2 \text{ e } x_1 \geq 0 \right\}.$$

Assim, se o conjunto P não for vazio, o problema (P) tem solução e, é equivalente a:

$$\underset{x_2 \in P}{\text{minimizar}} \left\{ c_2^T \cdot x_2 + \underset{x_1 \geq 0}{\text{mínimo}} \left\{ c_1^T \cdot x_1 \mid A_1 \cdot x_1 \geq b - A_2 \cdot x_2 \right\} \right\}. \quad (2)$$

Isto corresponde à idéia inicial de fixar a variável x_2 , resolver o problema de programação linear em relação a variável x_1 e, escolher um valor melhor para a variável x_2 .

Considera-se u o vetor das variáveis duais associadas às restrições do subproblema (SPx_2) . O seu problema dual é da forma:

$$\begin{aligned} &\text{maximizar} \quad (b - A_2 \cdot x_2)^T \cdot u + c_2^T \cdot x_2 \\ &\text{sujeito a:} \quad A_1^T \cdot u \leq c_1 \\ &\quad \quad \quad u \geq 0 \end{aligned} \quad (DSPx_2)$$

Como as restrições deste subproblema não dependem da variável x_2 , este será denotado simplesmente por (DSP).

Pelo Teorema da Dualidade de Programação Linear, se o conjunto de soluções duais viáveis de (DSP) for vazio, ou seja, se $S(DSP) = \left\{ u \mid A_1^T \cdot u \leq c_1, u \geq 0 \right\} = \phi$, o subproblema (SPx_2) não tem solução viável ou não tem solução

ótima (valor da função objetivo tende a $-\infty$). Como $x_2 \in P$ então (SPx_2) tem solução viável e, se o conjunto de soluções viáveis do dual for vazio, (SPx_2) não tem solução ótima. Consequentemente o problema (P) não tem solução ótima, isto é, a função objetivo decresce indefinidamente em um raio da região de viabilidade.

Caso o problema dual seja inviável, o máximo é dado por $-\infty$, e pode-se escrever o problema (P), utilizando o Teorema da Dualidade:

$$\underset{x_2 \in P}{\text{minimizar}} \left\{ c_2^T \cdot x_2 + \overset{\substack{\text{dual de} \\ (2)}}{\text{máximo}} \left\{ (b - A_2 \cdot x_2)^T \cdot u \mid A_1^T \cdot u \leq c_1, u \geq 0 \right\} \right\}.$$

Se o conjunto de soluções duais viáveis de (DSP) não for vazio, então o dual tem solução ótima em um ponto extremo do conjunto ou a função objetivo tende a $+\infty$. Esta última possibilidade implica no subproblema primal (SPx_2) inviável, contradizendo a hipótese inicial de $x_2 \in P$. Portanto, como existe um número P finito de pontos extremos do conjunto de soluções duais, u^i , $i = 1, \dots, n_p$, escreve-se o problema (P) como:

$$\underset{x_2 \in P}{\text{minimizar}} \left\{ c_2^T \cdot x_2 + \text{máximo} \left\{ (b - A_2 \cdot x_2)^T \cdot u^i \mid i \in I_{VP} \right\} \right\}$$

onde $I_{VP} = \{1, 2, \dots, n_p\}$.

Desta maneira, o problema (P) é equivalente a:

minimizar x_0

sujeito a: $x_0 \geq c_2^T \cdot x_2 + (b - A_2 \cdot x_2)^T \cdot u^1, i \in I_{VP}$

$x_2 \in P$

O conjunto P (descrito em termos da existência de x_1) pode ser equivalentemente representado em termos de inequações pelo Lema de Farkas-Minkowski: *Existe $x \geq 0$ tal que $B \cdot x = a \Leftrightarrow a^T \cdot r \geq 0$ para todo r tal que $B^T \cdot r \geq 0$.*

Aplicando o Lema de Farkas-Minkowski ao sistema linear

$$\begin{cases} A_1 \cdot x_1 \geq b - A_2 \cdot x_2 \\ x_1 \geq 0 \end{cases}$$

x_2 é viável, se e somente se,

$$(b - A_2 \cdot x_2)^T \cdot r \leq 0 \quad (2.1)$$

para todo r tal que

$$A_1^T \cdot r \leq 0, r \geq 0.$$

Considera-se o cone $C =$

$$\left\{ r \mid A_1^T \cdot r \leq 0, r \geq 0 \right\} \text{ que é um poliedro e é gerado por um}$$

número R finito de raios extremos. Logo, existem vetores r^j , $j = 1, \dots, n_R$, de modo que qualquer vetor $r \in \mathbb{C}$ possa ser escrito como uma combinação linear não-negativa dos demais raios extremos, ou seja,

$$r = \sum_{j=1}^{n_R} \lambda_j \cdot r^j, \quad (2.2)$$

$$\lambda_j \geq 0, \quad j = 1, \dots, n_R. \quad (2.3)$$

Substituindo (2.2) e (2.3) em (2.1) obtém-se

$$\sum_{j=1}^{n_R} \lambda_j \cdot (b - A_2 \cdot x_2)^T \cdot r^j \leq 0$$

que é válido para todo $\lambda_j \geq 0$, se e somente se,

$$(b - A_2 \cdot x_2)^T \cdot r^j \leq 0, \quad j \in I_{RP} \quad (2.4)$$

onde $I_{RP} = \{1, 2, \dots, n_R\}$.

Portanto, $x_2 \in \mathbb{Z}^{n_2}$ é viável se e somente se satisfizer o sistema dado por (2.4). Então, o conjunto $\mathbb{P}$ pode ser escrito como

$$\mathbb{P} = \left\{ x_2 \in \mathbb{Z}^{n_2} \mid (b - A_2 \cdot x_2)^T \cdot r^j \leq 0, \quad j \in I_{RP} \right\}.$$

Substituindo o conjunto P pelas restrições que o definem, obtém-se o Problema Mestre Primal (MP):

minimizar x_0

sujeito a: $x_0 \geq c_2^T \cdot x_2 + (b - A_2 \cdot x_2)^T \cdot u^i, i \in I_{yp}$

$(b - A_2 \cdot x_2)^T \cdot r^j \leq 0, j \in I_{RP}$

$x_2 \in \mathbb{Z}^{n_2}$

nao nosstou quem era o conjunto (MP) Ito e Ito.

cujas restrições são chamadas de cortes de Benders ou cortes primais.

Portanto, a solução $\{x_1, x_2\}$ do problema (P) será determinada pela resolução do problema mestre (MP), $\{x_0, x_2\}$, e então pela solução do problema linear (SP x_2), ou do seu problema dual (DSP), para determinar x_1 .

Porém, o número de restrições do problema mestre (MP) é igual à soma do número de pontos extremos e raios extremos, o que em geral é muito grande, mesmo se o problema for de dimensão moderada. Entretanto, na solução ótima somente uma pequena parte destas restrições serão ativas, sugerindo assim, o uso da estratégia de relaxação.

O termo relaxação geralmente é usado com a seguinte idéia: um problema de minimização (R) é uma relaxação de um problema de minimização (Q) se o conjunto de soluções viáveis de (R) contiver o conjunto de soluções viáveis de (Q)

e, a função objetivo do problema (R) for menor ou igual à função objetivo de (Q) sobre o conjunto viável de (Q). Deste modo, observe que (SPx_2) é uma relaxação de (P) para todo $x_2 \in \mathbb{Z}^{n_2}$.

Relaxa-se o Problema Mestre (MP), obtendo desta maneira um Problema Mestre Relaxado, (MPR), formado por um subconjunto de restrições de (MP):

$$\begin{aligned} &\text{minimizar } x_0 \\ &\text{sujeito a: } x_0 \geq c_2^T \cdot x_2 + (b - A_2 \cdot x_2)^T \cdot u^i, \quad i \in I'_{VP} \subseteq I_{VP} \\ &\quad (b - A_2 \cdot x_2)^T \cdot r^j \leq 0, \quad j \in I'_{RP} \subseteq I_{RP} \\ &\quad x_2 \in \mathbb{Z}^{n_2} \end{aligned} \quad (\text{MPR})$$

A inviabilidade do problema (P) é detectada pela inviabilidade de (MPR).

Se o problema (MPR) tiver solução ótima, esta solução será ótima para o problema mestre (MP) se e somente se, esta solução satisfizer todas suas restrições, ou seja, é necessário e suficiente que a solução $\{\bar{x}_0, \bar{x}_2\}$ de (MPR) satisfaça as seguintes restrições

$$\begin{cases} \bar{x}_0 \geq c_2^T \cdot \bar{x}_2 + (b - A_2 \cdot \bar{x}_2)^T \cdot u^i, \quad i \in I_{VP} \\ (b - A_2 \cdot \bar{x}_2)^T \cdot r^j \leq 0, \quad j \in I_{RP} \end{cases}$$

Porém, estas restrições não estão disponíveis

explicitamente, ou seja, não se tem disponível todos os pontos extremos e raios extremos do problema (P). Mas, para verificar a otimalidade do problema mestre (MP), resolve-se o problema dual (DSP) que testará a otimalidade ou gerará a restrição violada de (MP).

Resolvendo o problema dual (DSP), pode-se encontrar os seguintes resultados:

1. O problema (DSP) inviável, o que implica no problema (P) sem solução ótima (função objetivo tende a $-\infty$) e término de procedimento.
2. O problema (DSP) sem solução ótima (função objetivo tende a $+\infty$), isto é, foi identificado um raio de subida $\bar{r} \geq 0$ tal que $A_1^T \cdot \bar{r} \leq 0$ e $(b - A_2 \cdot \bar{x}_2)^T \cdot \bar{r} > 0$.

Assim, a restrição $(b - A_2 \cdot \bar{x}_2)^T \cdot \bar{r} \leq 0$ não é satisfeita, e logo $\{\bar{x}_0, \bar{x}_2\}$ não é solução do problema mestre (MP). Portanto, a restrição $(b - A_2 \cdot x_2)^T \cdot \bar{r} \leq 0$ é adicionada à (MPR), produzindo um novo problema, que exclui $\{\bar{x}_0, \bar{x}_2\}$.

Se o problema (DSP) for resolvido pelo método simplex, então $\bar{r}$ é identificado, juntamente com um ponto extremo $\bar{u}$ (isto é, a função cresce indefinidamente na semi-reta $\{\bar{u} + \lambda \cdot \bar{r}, \lambda \geq 0\}$).

É conveniente também verificar se $\bar{x}_0 \geq c_2^T \bar{x}_2 + (b - A_2 \bar{x}_2)^T \bar{u}$ para o ponto extremo $\bar{u}$. Caso isto não ocorra, adiciona-se também a seguinte restrição ao novo problema relaxado: $x_0 \geq c_2^T x_2 + (b - A_2 x_2)^T \bar{u}$.

3. O problema (DSP) apresenta solução ótima, ou seja, obtém-se a solução em um vértice $\bar{u}$. Porém, se

$$\bar{x}_0 < c_2^T \bar{x}_2 + (b - A_2 \bar{x}_2)^T \bar{u}$$

então a restrição $x_0 \geq c_2^T x_2 + (b - A_2 x_2)^T \bar{u}$ é acrescentada em (MPR), excluindo-se $\{\bar{x}_0, \bar{x}_2\}$.

4. Finalmente, a solução ótima de (DSP) fornece um vértice $\bar{u}$ tal que

$$\bar{x}_0 = c_2^T \bar{x}_2 + (b - A_2 \bar{x}_2)^T \bar{u}.$$

Já que $\bar{u}$ é um vértice ótimo de (DSP) que maximiza $(b - A_2 x_2)^T u$, então

$$\bar{x}_0 \geq c_2^T \bar{x}_2 + (b - A_2 \bar{x}_2)^T u^i, \quad i \in I_{vp}.$$

E, além disso, $(b - A_2 \bar{x}_2)^T r^j \leq 0$, para todo $j \in I_{rp}$, pois, caso contrário, (DSP) não teria solução ótima. Portanto, a solução $\{\bar{x}_0, \bar{x}_2\}$ é ótima para o problema mestre e o procedimento termina.

Enquanto o teste de otimalidade não for verificado (caso 4) são adicionadas novas restrições ao problema mestre relaxado, e novos problemas relaxados são resolvidos.

Se o problema mestre relaxado for inviável, o problema (P) também será inviável e o procedimento é interrompido. Se o objetivo do problema relaxado tender a $-\infty$, nada se pode concluir em relação ao problema mestre ou ao problema (P), pois este caso pode ter ocorrido pela relaxação de restrições.

2.1.1. Algoritmo

Com os resultados obtidos é possível escrever o seguinte algoritmo para o Método de Benders:

- Passo 0 - Iniciação das variáveis

Fazer $\bar{x}_0 = -\infty$.

Selecionar $\bar{x}_2 \in \mathbb{Z}^{n_2}$.

- Passo 1 - Resolução do Subproblema

Resolver o subproblema dual (DSP).

Se (DSP) inviável então

(P) tende a $-\infty$. FIM.

Se (DSP) tende a $+\infty$ então

Determinar um raio extremo $\bar{r}$ e um ponto extremo $\bar{u}$ tal que o objetivo de (DSP) cresça indefinidamente na semi-reta:

$$\{\bar{u} + \lambda \cdot \bar{r}, \lambda \geq 0\}.$$

Adicionar a restrição $(b - A_2 \cdot x_2)^T \cdot \bar{r} \leq 0$ à (MPR).

Verificar se $\bar{x}_0 \geq c_2^T \cdot \bar{x}_2 + (b - A_2 \cdot \bar{x}_2)^T \cdot \bar{u}$, caso contrário adicionar $x_0 \geq c_2^T \cdot x_2 + (b - A_2 \cdot x_2)^T \cdot \bar{u}$ à (MPR).

Senão

Se $\bar{x}_0 = c_2^T \cdot \bar{x}_2 + (b - A_2 \cdot \bar{x}_2)^T \cdot \bar{u}$ então

A solução é ótima. FIM.

Senão

Adicionar a restrição $x_0 \geq c_2^T \cdot x_2 + (b - A_2 \cdot x_2)^T \cdot \bar{u}$ à (MPR).

◦ Passo 2 - Resolução do Problema Mestre

Resolver o problema mestre relaxado (MPR).

Se (MPR) inviável então

(P) é inviável. FIM.

Senão

Se (MPR) tende a $-\infty$ então

Fazer $\bar{x}_0 = -\infty$.

Selecionar $\bar{x}_2 \in \mathbb{Z}^{n_2}$.

Senão

$\{\bar{x}_0, \bar{x}_2\}$ é solução ótima de (MPR).

Ir para o Passo 1.

A convergência do algoritmo em um número finito de iterações é garantida, seja identificando a inviabilidade de P, ou inexistência de solução ótima (objetivo tendendo a $-\infty$), ou ainda com uma solução ótima.

2.2. Relaxação Lagrangeana

Relaxação Lagrangeana é aplicada a problemas que apresentam um grupo de restrições *complicadas* e outro de restrições *simples* e, consiste em relaxar as restrições *complicadas*, ou seja, retirar as restrições que sem as quais o problema torna-se mais fácil de resolver. Estas restrições são colocadas como uma *penalização* na função objetivo de forma que o processo de otimização busque satisfazê-las.

Considera-se o seguinte problema:

$$\begin{aligned} &\text{minimizar } c^T \cdot x \\ &\text{sujeito a: } A^1 \cdot x \geq b_1 \\ &\quad \quad \quad A^2 \cdot x \geq b_2 \\ &\quad \quad \quad x \in S \end{aligned} \tag{P}$$

onde $c, x \in \mathbb{R}^n$, $A^1 \in \mathbb{R}^{m_1 \times n}$, $A^2 \in \mathbb{R}^{m_2 \times n}$, $b_1 \in \mathbb{R}^{m_1}$, e $b_2 \in \mathbb{R}^{m_2}$ e $S = \left\{ x = (x_1, x_2)^T \mid x_1 \geq 0 \text{ e } x_2 \in \mathbb{Z}^{n_2} \right\}$.

A Relaxação Lagrangeana do problema (P) é definida associando às restrições *complicadas* do tipo $A^2 \cdot x \geq b_2$ uma variável dual $u_2 \in \mathbb{R}^{m_2}$, também denominada de multiplicador de Lagrange, com $u_2 \geq 0$.

Considera-se o Subproblema Dual (SDu_2), também chamado de Subproblema Lagrangeano, com as restrições

$A^2.x \geq b_2$ relaxadas:

minimizar $L(x, u_2)$

sujeito a: $A^1.x \geq b_1$

(SD u_2)

$x \in S$

onde $L(x, u_2) = c^T.x + (b_2 - A^2.x)^T.u_2$, com $u_2 \geq 0$, é denominada função Lagrangeana. Por simplicidade, denota-se (SD u_2) por (SD), já que $S(SDu_2)$ não depende da variável u_2 .

Note que para toda solução x viável do problema (P) tem-se $A^2.x \geq b_2$. Então, com $u_2 \geq 0$,

$L(x, u_2) \leq c^T.x$, para todo x viável em (P) e $u_2 \geq 0$.

A importância de qualquer relaxação do problema (P) é determinada pela proximidade do valor da função objetivo do problema relaxado e o valor da função objetivo de (P). Define-se então a função dual $w(u_2)$ como sendo

$w(u_2) = \text{mínimo } L(x, u_2)$

$x \in S(SD)$

onde $S(SD) = \left\{ x \mid A^1.x \geq b_1 \text{ e } x \in S \right\}$.

E, o melhor valor de u_2 é dado pela solução ótima do problema:

$$\text{maximizar } w(u_2) \quad (D)$$

$$\text{sujeito a: } u_2 \geq 0$$

Este problema é denominado Problema Dual Lagrangeano. Sua solução é de fundamental importância para resolução do problema (P) e pode ser obtida pelo Método do Subgradiente, descrito na próxima seção.

O problema dual Lagrangeano (D) pode também ser escrito da forma:

$$\text{maximizar } \left\{ \begin{array}{l} \text{minimizar } c^T \cdot x + (b_2 - A^2 \cdot x)^T \cdot u_2 \\ \text{sujeito a: } A^1 \cdot x \geq b_1 \\ x \in S \end{array} \right\} \quad (D)$$

$$u_2 \geq 0$$

ou

$$\text{maximizar } \text{mínimo } \left\{ c^T \cdot x^i + (b_2 - A^2 \cdot x^i)^T \cdot u_2 \mid i \in I_{VD} \right\}$$

$$u_2 \geq 0$$

onde I_{VD} é o conjunto dos índices de todos os pontos extremos do menor conjunto convexo formado pelas combinações lineares convexas de todos os pontos extremos de $S(SD)$, $C\{S(SD)\}$.

Desta forma, o problema (P) é equivalente ao Problema Mestre Dual (MD):

$$\begin{aligned}
& \text{maximizar } u_0 \\
& \text{sujeito a: } c^T \cdot x^i + (b_2 - A^2 \cdot x^i)^T \cdot u_2 \geq u_0, \quad i \in I_{VD} \quad (\text{MD}) \\
& u_2 \geq 0
\end{aligned}$$

cuja restrição são chamadas de cortes duais.

Evidentemente, resolver (MD) é equivalente a resolver (P), porém requer todos os pontos extremos de $C\{S(SD)\}$ explicitamente. Então, resolve-se o Problema Mestre Dual Relaxado (MDR), abaixo, onde apenas alguns cortes são considerados.

$$\begin{aligned}
& \text{maximizar } u_0 \\
& \text{sujeito a: } c^T \cdot x^i + (b_2 - A^2 \cdot x^i)^T \cdot u_2 \geq u_0, \quad i \in I'_{VD} \subseteq I_{VD} \quad (\text{MDR}) \\
& u_2 \geq 0
\end{aligned}$$

2.2.1. Método do Subgradiente

O Método do Subgradiente para resolução do Problema Dual Lagrangeano (D) foi inicialmente proposto por SHOR (1968). Posteriormente foi estudado por POLJAK (1969), mas a idéia central está baseada nos trabalhos de AGMON (1954) e MOTZKIN & SCHOENBERG (1954), em métodos iterativos de relaxação para resolver sistemas de desigualdades.

Estes métodos consistem basicamente na perturbação, a cada iteração, de uma solução u_2^k na direção de um

subgradiente da função, ou seja,

$$u_2^{k+1} = u_2^k + \theta_k \cdot \frac{\gamma^k}{\|\gamma^k\|}$$

onde γ^k é um subgradiente da função dual ($\gamma^k \in \partial w(u_2^k)$), e $\partial w(u_2^k) = \left\{ \gamma \mid w(u_2) \leq w(u_2^k) + \gamma^T \cdot (u_2 - u_2^k) \right\}$ é o conjunto dos subgradientes de w em u_2^k .

Não há nenhuma garantia que γ^k seja uma direção de subida, dessa forma a convergência do algoritmo depende essencialmente da escolha do tamanho do passo θ_k ($\theta_k > 0$) em cada iteração. Há muitos resultados teóricos que determinam o tamanho do passo a ser usado, mas um resultado mais geral é dado, a seguir, pelo Teorema 2.1.

A idéia do método é gerar uma seqüência de soluções $\{u_2^k\}$ através da regra:

$$u_2^{k+1} = \text{proj}_U \left(u_2^k + \theta_k \cdot \frac{\gamma^k}{\|\gamma^k\|} \right)$$

onde $U = \{ u_2 \mid u_2 \geq 0 \}$, e proj_U é o operador projeção de $\mathbb{R}^{m_2}$ em U .

- **Teorema 2.1** (POLJAK, 1967): Considera-se w uma função côncava e contínua de $\mathbb{R}^{m_2}$ em $\mathbb{R}$ e, U um conjunto convexo. Se θ_k é tal que

$$\lim_{k \rightarrow \infty} \theta_k = 0 \quad \text{e} \quad \sum_{k=1}^{\infty} \theta_k = \infty$$

então a sequência $\{u_2^k\}$ com $\gamma^k \in \partial w(u_2^k)$ é tal que

$$\lim_{k \rightarrow \infty} \sup w(u_2^k) = w(u_2^*) = \text{maximizar } w(u_2) \\ \text{sujeito a: } u_2 \geq 0$$

Prova:

É suficiente mostrar que $\forall \varepsilon > 0, \exists k$ tal que

$$u_2^k \in V(\varepsilon) = \left\{ u_2 \in U \mid w(u_2) \geq w(u_2^*) - \varepsilon \right\}.$$

Suponha por absurdo que $u_2^k \notin V(\varepsilon)$, para todo k . Ou seja, $w(u_2^k) < w(u_2^*) - \varepsilon$, para todo u_2^k .

Como a função w é contínua em $\mathbb{R}^{m_2}$ então $\exists \delta > 0$ tal que:

$$\|u_2 - u_2^*\| \leq \delta \Rightarrow w(u_2) \geq w(u_2^*) - \varepsilon$$

Tem-se $\gamma^k \neq 0$, para qualquer k (caso contrário, u_2^k seria um ponto de ótimo e o algoritmo terminaria).

Considera-se $\underline{u}_2^k$ definido por:

$$\underline{u}_2^k = \underline{u}_2^* - \delta \cdot \frac{\gamma^k}{\|\gamma^k\|} \quad (2.5)$$

Tem-se que $\|\underline{u}_2^k - \underline{u}_2^*\| = \delta$ e portanto $w(\underline{u}_2^k) \geq w(\underline{u}_2^*) - \varepsilon$.

Pelo fato de $w(\underline{u}_2^k) < w(\underline{u}_2^*) - \varepsilon \Rightarrow w(\underline{u}_2^k) - w(\underline{u}_2^k) < 0$.

Como γ^k é o subgradiente da função w no ponto $\underline{u}_2^k$, $w(\underline{u}_2^k) - w(\underline{u}_2^k) \leq (\gamma^k)^T \cdot (\underline{u}_2^k - \underline{u}_2^k)$ para todo k .

Logo,

$$(\gamma^k)^T \cdot (\underline{u}_2^k - \underline{u}_2^k) \leq w(\underline{u}_2^k) - w(\underline{u}_2^k) < 0 \quad (2.6)$$

Como o operador projeção é uma contração (MINOUX, 1986), tem-se:

$$\|\underline{u}_2^{k+1} - \underline{u}_2^*\| = \left\| \text{proj}_U \left[\underline{u}_2^k + \theta_k \cdot \frac{\gamma^k}{\|\gamma^k\|} \right] - \text{proj}_U \left[\underline{u}_2^* \right] \right\| \leq \left\| \underline{u}_2^k + \theta_k \cdot \frac{\gamma^k}{\|\gamma^k\|} - \underline{u}_2^* \right\|$$

Então,

$$\left\| \mathbf{u}_2^{k+1} - \mathbf{u}_2^* \right\|^2 \leq \left\| \mathbf{u}_2^k - \mathbf{u}_2^* \right\|^2 + \theta_k^2 + 2 \cdot \theta_k \cdot \frac{(\gamma^k)^T \cdot (\mathbf{u}_2^k - \mathbf{u}_2^*)}{\left\| \gamma^k \right\|} \quad (2.7)$$

Isolando $\mathbf{u}_2^*$ em (2.5) e substituindo na expressão $(\gamma^k)^T \cdot (\mathbf{u}_2^k - \mathbf{u}_2^*)$ obtém-se:

$$(\gamma^k)^T \cdot (\mathbf{u}_2^k - \mathbf{u}_2^*) = (\gamma^k)^T \cdot \left[\mathbf{u}_2^k - \underline{\mathbf{u}}_2^k - \delta \cdot \frac{\gamma^k}{\left\| \gamma^k \right\|} \right]$$

$$(\gamma^k)^T \cdot (\mathbf{u}_2^k - \mathbf{u}_2^*) = (\gamma^k)^T \cdot (\mathbf{u}_2^k - \underline{\mathbf{u}}_2^k) - \delta \cdot \left\| \gamma^k \right\| \quad (2.8)$$

Substituindo (2.8) em (2.7) obtém-se:

$$\left\| \mathbf{u}_2^{k+1} - \mathbf{u}_2^* \right\|^2 \leq \left\| \mathbf{u}_2^k - \mathbf{u}_2^* \right\|^2 + \theta_k^2 + 2 \cdot \theta_k \cdot \frac{(\gamma^k)^T \cdot (\mathbf{u}_2^k - \underline{\mathbf{u}}_2^k)}{\left\| \gamma^k \right\|} - 2 \cdot \delta \cdot \theta_k$$

De (2.6) tem-se $(\gamma^k)^T \cdot (\mathbf{u}_2^k - \underline{\mathbf{u}}_2^k) < 0$ e, como $\theta_k > 0$ (por hipótese), então:

$$\left\| \mathbf{u}_2^{k+1} - \mathbf{u}_2^* \right\|^2 \leq \left\| \mathbf{u}_2^k - \mathbf{u}_2^* \right\|^2 + \theta_k \cdot (\theta_k - 2 \cdot \delta)$$

Como $\{\theta_k\}$ é uma sequência que tende para zero

quando k tende para infinito, então $\exists K \in \mathbb{N}$ tal que $k \geq K \Rightarrow \theta_k \leq \delta$.

Assim, para todo $k \geq K$,

$$\|u_2^{k+1} - u_2^*\|^2 < \|u_2^k - u_2^*\|^2 - \theta_k \cdot \delta$$

Somando sobre k , $k = K, \dots, K+P$, obtém-se:

$$\delta \cdot \sum_{k=K}^{K+P} \theta_k \leq \|u_2^K - u_2^*\|^2 - \|u_2^{K+P+1} - u_2^*\|^2 \leq \|u_2^K - u_2^*\|^2$$

Escolhendo P suficientemente grande, tem-se uma contradição pelo fato que $\sum_{k=1}^{\infty} \theta_k = \infty$. ■

O resultado acima nada diz em relação a velocidade de convergência, mas AGMON (1954) mostrou convergência linear na aplicação deste procedimento. Uma outra estratégia para a escolha do passo, proposta por POLJAK (1967), onde a sequência $\{u_2^k\}$ tem convergência geométrica, tem o passo θ_k definido por:

$$\theta_k = \alpha_k \cdot \frac{\{w(u_2^*) - w(u_2^k)\}}{\|\gamma^k\|^2} \quad (2.9)$$

onde $0 < \alpha_k \leq 2$, $\forall k$ e, $w(u_2^*)$ é o valor ótimo de (D).

2.2.1.1. Algoritmo

Utilizando os resultados anteriores, pode-se escrever o seguinte algoritmo para o Método do Subgradiente:

◦ Passo 0 - Iniciação das variáveis

Escolher $\varepsilon > 0$ (precisão desejada).

Determinar $\max > 0$ (número máximo de iterações).

Fazer $k = 0$ (número da iteração).

Escolher $u_2^0 \geq 0$ (aproximação inicial).

Determinar θ_0 e $\gamma^0 \in \partial w(u_2^0)$.

◦ Passo 1

Se $\|\gamma^k\| < \varepsilon$ então u_2^k é a solução. FIM.

Se $k \geq \max$ então u_2^k nem sempre é solução. FIM.

$$\text{Calcular } u_2^{k+1} = \text{proj}_U \left[u_2^k + \theta_k \cdot \frac{\gamma^k}{\|\gamma^k\|} \right].$$

Determinar θ_{k+1} e $\gamma^{k+1} \in \partial w(u_2^{k+1})$.

Fazer $k = k + 1$.

Ir para o Passo 1.

O valor de θ_k é dado por (2.9) e, o valor utilizado de $w(u_2^*)$ no cálculo de θ_k é uma estimativa do valor ótimo da função. Para o problema (P), $\gamma^k = (b_2 - A^2 \cdot x^k)$ é um subgradiente de w em u_2^k .

3. DECOMPOSIÇÃO HÍBRIDA

Muitos métodos para resolução de Problemas Lineares Inteiros Mistos estão baseados na decomposição primal, produzindo, por exemplo, um algoritmo de Decomposição de Benders, ou na decomposição dual, produzindo, por exemplo, um algoritmo baseado na Relaxação Lagrangeana. Estes métodos exploram a estrutura primal ou a estrutura dual do problema. A idéia básica da Decomposição Híbrida (ROY, 1983) é usar a decomposição primal e dual em um único procedimento, explorando simultaneamente as duas estruturas.

3.1. O Princípio da Decomposição Híbrida

No Método de Decomposição de Benders, o subproblema primal é uma limitação do problema original, no qual algumas variáveis primais têm valores fixos. A cada ite-

ração do algoritmo, estes valores são ajustados pelo problema mestre.

Na decomposição dual, relaxa-se as restrições que tornam o problema difícil de ser resolvido e, obtém-se um subproblema dual. Assim, o algoritmo resolve sucessivamente um subproblema dual e um problema mestre ajusta as variáveis duais até que o ótimo dual seja alcançado.

O princípio da decomposição, na sua forma primal ou dual, tem produzido vários algoritmos eficientes para resolução de Problemas de Programação Linear Inteira Mista. A técnica permite tirar vantagens da estrutura especial do problema, resolvendo-se uma seqüência de subproblemas fáceis. Conseqüentemente, explora-se a estrutura primal ou dual do problema.

A idéia básica da Decomposição Híbrida é usar ambos os subproblemas em um único procedimento. O valor ótimo do subproblema primal (SPx_2) como sendo um limitante superior da solução ótima do problema original e, o valor ótimo do subproblema dual (SDu_2) como um limitante inferior.

Entretanto, um método que resolva sucessivamente um subproblema primal e um subproblema dual ($sp \rightleftharpoons sd$), onde x_2 em (SPx_2) é modificado pela solução de (SDu_2), que tem u_2 modificado pela solução dual de (SPx_2), não tem sua

convergência garantida. É necessário, às vezes, interromper o ciclo $sp \rightleftarrows sd$ e, gerar novos cortes para o problema mestre, que resolvido a partir da solução atual, gera uma nova solução, que será utilizada como ponto inicial na continuação do ciclo $sp \rightleftarrows sd$.

O procedimento do Método da Decomposição Híbrida pode ser resumido na Figura 3.1, onde TC_P (teste de convergência primal) e TC_D (teste de convergência dual) controlam a seqüência dos subproblemas primal e dual e do problema mestre.

FIGURA 3.1 Algoritmo da Decomposição Híbrida.

3.2. Equivalência entre os Subproblemas

O objetivo desta seção é mostrar que o subproblema primal (SPx_2) é equivalente ao problema mestre dual,

e similarmente, que o subproblema dual (SDu_2) pode ser considerado um problema mestre na decomposição primal.

- **Definição 3.1:** Considera-se os problemas de programação linear inteira mista, (Q) e (Q') . O problema (Q) é equivalente a (Q') em relação a um subconjunto de variáveis, se a solução ótima de (Q) , nestas variáveis, é solução ótima de (Q') , e vice-versa.

Por exemplo, o problema (P) e o problema mestre primal (MP) são equivalentes em relação à variável x_2 , por construção. Analogamente, o problema dual (D) e o problema mestre dual (MD) são equivalentes em relação à variável u_2 .

Considera-se a seguinte notação:

- $S(DSPu_2)$: região viável das restrições de (DSP) com u_2 fixo
- $Iu_2 = \left\{ i \mid u^i = (u_1^i, u_2)^T \text{ é um ponto extremo de } S(DSPu_2) \right\}$
- $\bar{I}u_2 = \left\{ i \in Iu_2 \mid u^i = (u_1^i, u_2)^T \text{ é um ponto extremo de } S(DSP) \right\}$
- $(MPu_2) = (MP \mid i \in Iu_2)$.

E, considera-se, as seguintes inclusões:

$$\bar{I}u_2 \subseteq Iu_2, \bar{I}u_2 \subseteq I_{VP}$$

$$Iu_2 \subseteq I_{VP} \text{ se e somente se } \bar{I}u_2 = Iu_2.$$

Para uma melhor compreensão destas inclusões, observe o exemplo ilustrado na Figura 3.2.

FIGURA 3.2 Exemplo das inclusões que envolvem a região viável do problema (DSP).

Considera-se também, as seguintes desigualdades:

$$v(MP \mid i \in \bar{I}u_2) \leq v(MPu_2) \equiv v(MP \mid i \in Iu_2) \leq v(MP).$$

O próximo teorema demonstra que o subproblema Lagrangeano (SDu_2) é equivalente ao Problema Mestre Primal Relaxado (MPu_2) em relação à variável x_2 .

- **Teorema 3.1:** O Problema Mestre Primal $(MPu_2) = (MP \mid i \in Iu_2)$ é equivalente ao subproblema dual Lagrangeano (SDu_2) em relação a variável primal x_2 , e $v(MPu_2) = v(SDu_2)$.

Prova:

O Problema (MPu_2) é dado por:

$$\begin{aligned} &\text{minimizar } x_0 \\ &\text{sujeito a: } x_0 \geq c_2^T \cdot x_2 + (b - A_2 \cdot x_2)^T \cdot u^i, \quad i \in Iu_2 \\ &\quad (b - A_2 \cdot x_2)^T \cdot r^j \leq 0, \quad j \in I_{RP} \\ &\quad x_2 \in \mathbb{Z}^{n_2} \end{aligned} \quad (MPu_2)$$

ou

$$\begin{aligned} &\text{minimizar } \max_{x_2 \in \mathbb{P}} \left\{ c_2^T \cdot x_2 + (b - A_2 \cdot x_2)^T \cdot u^i \mid i \in Iu_2 \right\} \\ &\quad x_2 \in \mathbb{P} \end{aligned}$$

ou

$$\begin{aligned} &\text{minimizar } \left\{ \begin{array}{l} \text{maximizar } b^T \cdot u + (c_2^T - u^T \cdot A_2) \cdot x_2 \\ \text{sujeito a: } (A_1^1)^T \cdot u_1 \leq c_1 - (A_1^2)^T \cdot u_2 \\ \quad u_1 \geq 0 \end{array} \right\} \\ &\quad x_2 \in \mathbb{P} \end{aligned}$$

já que uma solução ótima do problema de maximização é uma solução básica (ponto extremo do poliedro).

O problema interno tem u_2 e x_2 fixos. Obtendo o dual deste problema interno, o problema mestre (MPu₂) pode ser escrito como:

$$\text{minimizar } \left\{ \begin{array}{l} \text{minimizar } b_2^T \cdot u_2 + (c_2^T - u_2^T \cdot A_2^2) \cdot x_2 + (c_1^T - u_2^T \cdot A_1^2) \cdot x_1 \\ \text{sujeito a: } A_1^1 \cdot x_1 \geq b_1 - A_2^1 \cdot x_2 \\ x_1 \geq 0 \end{array} \right\}$$

$x_2 \in \mathbb{P}$

ou

$$\begin{array}{ll} \text{minimizar} & c^T \cdot x + (b_2 - A^2 \cdot x)^T \cdot u_2 \\ \text{sujeito a:} & A^1 \cdot x \geq b_1 \\ & x \in S \end{array} \quad (\text{SDu}_2)$$

Como todos estes problemas foram formulados na variável x_2 , a prova está completa. ■

Um inconveniente do Problema Mestre de Benders é a inexistência da solução ótima da iteração inicial. O algoritmo da Decomposição Híbrida contorna este problema utilizando o valor da variável x_2 , obtido da solução ótima do último subproblema Lagrangeano (SDu₂).

Dois problemas equivalentes, não implica que sejam de mesma dificuldade. Uma maneira de medir a dificuldade de um problema (Q) é utilizar o gap relativo do problema relaxado ($\bar{Q}$), dado por:

$$h(Q) = \frac{v(Q) - v(\bar{Q})}{v(Q)}.$$

- **Corolário 3.1:** Os problemas (SDu_2) e (MPu_2) são de mesma dificuldade, ou seja, $h(SDu_2) = h(MPu_2)$.

Prova:

Pelo Teorema 3.1, tem-se $v(SDu_2) = v(MPu_2)$.
Reescrevendo a prova do Teorema 3.1 para $\overline{(MPu_2)}$, ou seja, substituindo $x_2 \in \mathbb{Z}^{n_2}$ por $x_2 \geq 0$, obtém-se que $v(\overline{SDu_2}) = v(\overline{MPu_2})$ e a prova está completa. ■

Entretanto resolver o subproblema Lagrangeano (SDu_2) ao invés de resolver (MPu_2) é computacionalmente mais atrativo, desde que a estrutura especial das restrições seja explorada.

Em paralelo com a equivalência do subproblema dual (SDu_2) com o problema mestre primal (MPu_2) , o próximo teorema diz respeito ao uso do subproblema primal (SPx_2) como um problema mestre dual.

Define-se:

- $S(SPx_2)$: região viável das restrições de (SPx_2) com x_2 fixo
- $Ix_2 = \left\{ i \mid x^i = (x_1^i, x_2)^T \text{ é um ponto extremo de } S(SPx_2) \right\}$

$$\circ \bar{I}x_2 = \left\{ i \in Ix_2 \mid x^i = (x_1^i, x_2^i)^T \text{ é um ponto extremo de } S(SP_{x_2}) \right\}$$

$$\circ (MD_{x_2}) = (MD \mid i \in Ix_2)$$

$$\bar{I}x_2 \subseteq Ix_2, \bar{I}x_2 \subseteq I_{VD}$$

$$Ix_2 \subseteq I_{VD} \text{ se e somente se } Ix_2 = \bar{I}x_2$$

$$v(MD \mid i \in \bar{I}x_2) \geq v(MD_{x_2}) \equiv v(MD \mid i \in Ix_2) \geq v(MDR).$$

- **Teorema 3.2:** O Problema Mestre Dual $(MD_{x_2}) = (MD \mid i \in Ix_2)$ e o Subproblema de Benders (SP_{x_2}) são equivalentes em relação a variável u_2 , e $v(MD_{x_2}) = v(SP_{x_2})$.

Prova:

O problema mestre dual (MD_{x_2}) é dado por:

maximizar u_0

sujeito a: $c^T \cdot x^i + u_2^T \cdot (b_2 - A^2 \cdot x^i) \geq u_0, i \in Ix_2$

$$u_2 \geq 0$$

ou

$$\begin{array}{ll} \text{maximizar} & \text{mínimo} \left\{ c^T \cdot x^i + u_2^T \cdot (b_2 - A^2 \cdot x^i), i \in Ix_2 \right\} \\ & u_2 \geq 0 \end{array}$$

ou

$$\text{maximizar } \left\{ \begin{array}{l} \text{minimizar } c^T \cdot x + (b_2 - A^2 \cdot x)^T \cdot u_2 \\ \text{sujeito a: } A_1^1 \cdot x_1 \geq b_1 - A_2^1 \cdot x_2 \\ x_1 \geq 0 \end{array} \right\}$$

$$u_2 \geq 0$$

Já que a solução ótima do problema de minimização interno é um ponto extremo de sua região viável, e todos estes pontos são dados pelo conjunto Ix_2 , então tomando o dual do problema interno, com u_2 e x_2 fixos, pode-se reescrever o problema em forma dual (MD x_2) como sendo:

$$\begin{aligned} &\text{maximizar } c_2^T \cdot x_2 + (b_2 - A_2^2 \cdot x_2)^T \cdot u_2 + (b_1 - A_2^1 \cdot x_2)^T \cdot u_1 \\ &\text{sujeito a: } (A_1^1)^T \cdot u_1 \leq c_1 - (A_1^2)^T \cdot u_2 \\ &u \geq 0 \end{aligned}$$

ou

$$\begin{aligned} &\text{maximizar } c_2^T \cdot x_2 + (b - A_2 \cdot x_2)^T \cdot u \\ &\text{sujeito a: } A_1^T \cdot u \leq c_1 \\ &u \geq 0 \end{aligned}$$

que é o problema dual de

$$\begin{aligned} &\text{minimizar } c_1^T \cdot x_1 + c_2^T \cdot x_2 \\ &\text{sujeito a: } A_1 \cdot x_1 \geq b - A_2 \cdot x_2 \quad (\text{SP}x_2) \\ &x_1 \geq 0 \end{aligned}$$

Como estes problemas estão formulados na va-

riável u_2 , a prova está completa. ■

Os resultados obtidos podem ser observados na Figura 3.3.

FIGURA 3.3 Equivalência entre os subproblemas.

O próximo teorema mostra como é possível verificar se uma solução é ótima no algoritmo da Decomposição Híbrida.

◦ **Teorema 3.3:** Considera-se x^* uma solução ótima de (P) e u_2^* uma solução ótima de (D). Então, as seguintes afirmações são equivalentes:

(a) A relaxação Lagrangeana em relação a restrição $A^2 \cdot x \geq b_2$ não possui *gap dual*, isto é, $v(P) = v(D)$;

(b) Existe uma solução ótima dual u^+ de (SPx_2^*) com

$$v(SDu_2^+) = v(SP x_2^*);$$

(c) Existe uma solução ótima primal x^* de (SDu_2^*) com $v(SP x_2^+) = v(SDu_2^*)$.

Prova:

(a) $\rightarrow$ (b)

Pelo Teorema 3.2, $(SP x_2^*)$ é equivalente a (MDx_2^*) em relação à variável u_2 :

maximizar u_0

sujeito a: $c^T \cdot x^i + (b_2 - A^2 \cdot x^i)^T \cdot u_2 \geq u_0, i \in Ix_2^*$ (MDx_2^*)

$$u_2 \geq 0$$

Considera-se u^+ solução ótima dual de $(SP x_2^*)$. Então, pelo Teorema 3.2, u_2^+ é solução ótima em (MDx_2^*) . Além disso, já que x^* é solução ótima em (P), segue que:

$$u_0^* = v(MDx_2^*) = v(SP x_2^*) = v(P) = v^*$$

Suponha que:

$$v(SDu_2^+) = c^T \cdot x^k + (b_2 - A^2 \cdot x^k)^T \cdot u_2^+ < v^* = u_0^*$$

onde x^k é uma solução ótima de (SDu_2^+) e $k \in I_{vp}$. O corte dual de x^k , uma vez adicionado em (MDx_2^*) , irá perturbar a solução

ótima de (MDx_2^*) já que u_2^+ não está longe da solução viável de $(MD \mid i \in Ix_2^*+k)$. A solução ótima u_2^{++} deste novo problema mestre não pode melhorar o valor da solução ótima, já que o valor máximo $u_0^* = v^*$ já foi alcançado. Já que u_2^{++} é viável em (MDx_2^*) , ele é ótimo em (MDx_2^*) ou, equivalentemente, há algum $u^{++} = (u_1^{++}, u_2^{++})^T$ que é o ótimo dual em (SPx_2^*) . Como o conjunto I_{VD} é finito, uma das soluções dual de (SPx_2^*) deve produzir $v(SDu_2^+) = v^*$.

A implicação $(a) \rightarrow (c)$ segue a mesma linha de raciocínio de $(a) \rightarrow (b)$.

As implicações $(b) \rightarrow (a)$ e $(c) \rightarrow (a)$ são triviais.

As demais implicações decorrem da lógica: $(b) \rightarrow (c)$, pois $(b) \rightarrow (a) \rightarrow (c)$ e, $(c) \rightarrow (b)$, pois $(c) \rightarrow (a) \rightarrow (b)$. ■

3.3. Convergência do Método

Embora os subproblemas (SDu_2) e (SPx_2) sejam problemas mestres um para o outro, não significa que o problema (P) pode ser resolvido apenas pelo ciclo $sp \rightleftarrows sd$ entre estes subproblemas. Por exemplo, o algoritmo poderá nunca terminar se a Relaxação Lagrangeana não tiver gap dual nulo. São necessários, então, mais alguns resultados para definir

os testes de convergência.

◦ Lema 3.1:

(a) Considera-se u^0 solução ótima dual de (SPx_2^0) e, x^+ solução ótima de (SDu_2^0) . Então $x_2^+ \neq x_2^0$ a menos que $v(SPx_2^0) = v(P)$.

(b) Considera-se x^0 solução ótima primal de (SDu_2^0) e, u^+ solução ótima dual de (SPx_2^0) . Então $u_2^+ \neq u_2^0$ a menos que $v(SDu_2^0) = v(D)$.

Prova:

(a)

Suponha por absurdo, $v(SPx_2^0) > v(P)$ e $x_2^+ = x_2^0$.

Então, já que u^0 é solução ótima dual de (SPx_2^0) , então u_2^0 é solução ótima de (MDx_2^0) pelo Teorema 3.2 e, $c^T \cdot x^+ + (b_2 - A^2 \cdot x^+)^T \cdot u_2^0 \geq v(MDx_2^0) = v(SPx_2^0)$.

Isto contradiz x^+ ser solução ótima de (SDu_2^0) já que $c^T \cdot x^+ + (b_2 - A^2 \cdot x^+)^T \cdot u_2^0 = v(SDu_2^0) \leq v(P) < v(SPx_2^0)$.

(b) mesma linha de raciocínio de (a). ■

Suponha que o Método de Decomposição Híbrida resolva sucessivamente o subproblema primal (SPx_2^k) , obtendo

u_2^k , e o subproblema dual (SDu_2^{k+1}) obtendo x_2^{k+1} , ou seja,

$$(SPx_2^k) \longrightarrow (SDu_2^{k+1}) \longrightarrow (SPx_2^{k+2}) \longrightarrow (SDu_2^{k+3}) \longrightarrow \dots$$

Desta maneira, não se pode reaplicar o método em um número menor que 4 iterações, já que pelo Lema 3.1 tem-se $x_2^{k+2} = x_2^{k+1} \neq x_2^k$. Portanto, é possível somente que $x_2^{k+3} = x_2^k$. Um argumento similar é aplicado ao subproblema dual.

Usando este resultado pode-se incluir o número de iterações do ciclo $sp \rightleftarrows sd$ no teste de convergência. Assim, se não houver uma melhoria na solução durante a resolução dos subproblemas (SPx_2) e (SDu_2) em 4 iterações, deve-se resolver um dos problemas mestres e com uma nova solução, recomençar o ciclo $sp \rightleftarrows sd$.

3.4. Algoritmo

◦ Passo 1 - Iniciação da variáveis

Fazer $k = 0$ (número de iterações),

$\epsilon > 0$ (tolerância),

$j = 1$ (número de iterações do ciclo $sp \rightleftarrows sd$),

$\bar{v}_P = +\infty$,

$\bar{v}_D = -\infty$.

Selecionar $u_2^1 \geq 0$.

◦ Passo 2 - Subproblema Dual

Fazer $k = k + 1$.

Resolver o subproblema (SDu_2^k) . x^k é solução ótima.

Se $\bar{v}_D < v(SDu_2^k)$, $\bar{v}_D = v(SDu_2^k)$ e $j = 1$, senão $j = j+1$.

(Teste de Convergência TC_P)

Se $|\bar{v}_P - \bar{v}_D| \leq \varepsilon$ então $\bar{x}$ é solução ótima de (P). FIM.

Se $j = 4$, ir para o Passo 4(b).

Fazer $x_2^{k+1} = x_2^k$. Ir para o Passo 3.

◦ Passo 3 - SubProblema Primal

Fazer $k = k + 1$.

Resolver o subproblema (SPx_2^k) . x_1^k e u^k são soluções ótimas primal e dual, respectivamente.

Se $\bar{v}_P > v(SPx_2^k)$, $\bar{v}_P = v(SPx_2^k)$, $\bar{x} = x^k$ e $j = 1$, senão $j = j+1$.

(Teste de Convergência TC_D)

Se $|\bar{v}_P - \bar{v}_D| \leq \varepsilon$ então $\bar{x}$ é solução ótima de (P). FIM.

Se $j = 4$, ir para o Passo 4(a).

Fazer $u_2^{k+1} = u_2^k$. Ir para o Passo 2.

◦ Passo 4 - Problemas Mestres

(a) Resolver o Problema Mestre Dual. $\{\bar{u}_0, u_2^{k+1}\}$ é solução ótima.

Fazer $j = 1$.

Ir para o Passo 2.

(b) Resolver o Problema Mestre Primal. $\{\bar{x}_0, x_2^{k+1}\}$ é solução

ótima.

Fazer $j = 1$.

Ir para o Passo 3.

- **Teorema 3.5:** O algoritmo de Decomposição Híbrida é finito e resolve (P).

Prova:

Basta utilizar o Lema 3.1 e o fato que os algoritmos da Decomposição de Benders e da Relaxação Lagrangeana são finitos.■

4. APLICAÇÃO: PROBLEMA DE PLANEJAMENTO E CONTROLE DE PRODUÇÃO NUMA FUNDIÇÃO AUTOMATIZADA

4.1. Definição do Modelo

O esquema na Figura 4.1 resume as principais atividades numa fundição automatizada, a qual deve atender uma demanda de peças.

FIGURA 4.1 Esquema de produção de peças de uma fundição.

Os equipamentos da fundição são basicamente um forno e um conjunto de máquinas. O forno é alimentado por minérios de ferro, e produz uma certa quantidade de uma liga metálica. Esta liga metálica é então distribuída em máquinas, as quais são preparadas para a produção de diferentes tipos de peças em quantidades limitadas pelas tecnologias das máquinas. De cada liga é possível produzir um grupo específico de peças.

A demanda total exige a repetição destas operações (programação do forno e das máquinas) por muitos períodos. Há um custo de produção de cada peça por período (podendo representar a urgência com que uma peça deve ser produzida).

O problema consiste em programar o forno e as máquinas de modo a minimizar o custo total de produção.

São fornecidos os seguintes dados:

L: número de ligas

M: número de máquinas

T: número de períodos

P: número de tipos de peças

a_{im} : quantidade de peças do tipo i possível de ser produzida na máquina m , por unidade de tempo (tf/h)

F_t : capacidade do forno no período t (tf/h)

b_i : demanda de peças do tipo i (tf)

h_t : número de unidades de tempo no período t (h)

L_j : conjunto das peças que utilizam a liga j

$\bar{L}_j$: conjunto das peças que não utilizam a liga j

c_{it} : custo de produção de peça do tipo i no período t (\$)

Formulando o problema em função da fração de tempo utilizada para produzir uma determinada peça em cada período, tem-se as seguintes variáveis:

x_{imt} : fração de tempo utilizado para produzir peças do tipo i na máquina m no período t

$$Y_{jt} = \begin{cases} 1, & \text{se a liga do tipo } j \text{ for produzida no período } t \\ 0, & \text{caso contrário} \end{cases}$$

O modelo matemático do problema é apresentado a seguir, onde as equações (4.1) definem que apenas uma liga pode ser produzida por período, as inequações (4.2) e (4.3) garantem que se a liga j for utilizada no período t , apenas peças de L_j podem ser produzidas, as inequações (4.4) definem a capacidade de produção do forno e as inequações (4.5) definem a demanda das peças.

$$\begin{aligned} & \text{minimizar} \sum_{t=1}^T \sum_{i=1}^P \sum_{m=1}^M h_t \cdot c_{it} \cdot a_{im} \cdot x_{imt} \\ & \text{sujeito a:} \sum_{j=1}^L y_{jt} = 1, \quad t = 1, \dots, T \end{aligned} \quad (4.1)$$

$$\sum_{i \in L_j} x_{imt} \leq (1 - y_{jt}) \cdot \alpha + 1, \quad (4.2)$$

$$t = 1, \dots, T, \quad j = 1, \dots, L, \quad m = 1, \dots, M$$

$$\sum_{i \in \bar{L}_j} x_{imt} \leq (1 - y_{jt}) \cdot \alpha, \quad (4.3)$$

$$t = 1, \dots, T, \quad j = 1, \dots, L, \quad m = 1, \dots, M$$

$$\sum_{j=1}^L \sum_{m=1}^M \sum_{i \in L_j} a_{im} \cdot x_{imt} \leq F_t, \quad t = 1, \dots, T \quad (4.4)$$

$$\sum_{t=1}^T \sum_{m=1}^M a_{im} \cdot h_t \cdot x_{imt} \geq b_i, \quad i = 1, \dots, P \quad (4.5)$$

$$x_{imt} \geq 0, \quad i = 1, \dots, P, \quad m = 1, \dots, M, \quad t = 1, \dots, T$$

$$y_{jt} \in \{0, 1\}, \quad j = 1, \dots, L, \quad t = 1, \dots, T$$

com α grande

4.2. Resolução do Problema

O Problema de Planejamento e Controle de Produção numa Fundição Automatizada pode ser resolvido através do Método de Decomposição Híbrida, apresentado no capítulo anterior, uma vez que a estrutura da matriz dos coeficientes é bem conveniente. A Figura 4.2 ilustra como os elementos não nulos são distribuídos.

FIGURA 4.2 Estrutura da matriz dos coeficientes do problema.

As matrizes A , B , C são esparsas e apresentam estruturas muito particulares. Observa-se que as primeiras restrições, designadas por *restrições lógicas* são necessárias apenas para definir que tipo de liga será produzido num determinado período e, então, definir as possíveis peças a serem produzidas. As últimas restrições tem uma estrutura de *transporte generalizado*. Estas observações sugeriram um método de busca heurística, que é discutido a seguir.

A idéia é, numa fase inicial, relaxar as restrições lógicas e fazer a programação do forno. Nesta fase uma seqüência de problemas de transporte é resolvida. Numa fase final é efetuada a programação das máquinas, onde T problemas de transporte generalizado são resolvidos.

Relaxando o conjunto de peças de cada liga, ou seja, $L_j = \{1, 2, \dots, P\}$, $j = 1, \dots, L$ (todas as ligas podem produzir todas as peças) as restrições (4.2), (4.3) tornam-se desnecessárias, bem como a variável y . Obtém-se então, o seguinte problema (as restrições (4.4) são multiplicadas por h_t):

$$\begin{aligned}
 &\text{minimizar} \quad \sum_{t=1}^T \sum_{i=1}^P \sum_{m=1}^M h_t \cdot c_{it} \cdot a_{im} \cdot x_{imt} \\
 &\text{sujeito a:} \quad \sum_{m=1}^M \sum_{i=1}^P a_{im} \cdot h_t \cdot x_{imt} \leq F_t \cdot h_t, \quad t = 1, \dots, T \\
 &\quad \sum_{t=1}^T \sum_{m=1}^M a_{im} \cdot h_t \cdot x_{imt} \geq b_i, \quad i = 1, \dots, P \\
 &\quad x_{imt} \geq 0, \quad i = 1, \dots, P, \quad m = 1, \dots, M, \quad t = 1, \dots, T
 \end{aligned}$$

Define-se uma nova variável z_{ti} por:

$$z_{ti} = \sum_{m=1}^M a_{im} \cdot h_t \cdot x_{imt}$$

que consiste na quantidade de peças do tipo i produzidas no período t .

O problema relaxado pode ser escrito como um problema de transporte da forma:

$$\begin{aligned}
& \text{minimizar} \quad \sum_{t=1}^T \sum_{i=1}^P c_{it} \cdot z_{ti} \\
& \text{sujeito a:} \quad \sum_{i=1}^P z_{ti} \leq F_t \cdot h_t, \quad t = 1, \dots, T \\
& \quad \quad \quad \sum_{t=1}^T z_{ti} \geq b_i, \quad i = 1, \dots, P \\
& \quad \quad \quad z_{ti} \geq 0, \quad t = 1, \dots, T, \quad i = 1, \dots, P
\end{aligned}$$

e esquematizado pela Figura 4.3.

FIGURA 4.3 Esquema da programação do forno quando a escolha da liga é irrelevante.

Resolvendo este problema de transporte determina-se a quantidade de cada peça produzida em cada período. Entretanto, apenas um tipo de liga pode ser utilizado por período. Faz-se necessário a escolha das ligas que serão utilizadas na fabricação das peças. A solução obtida fornece um limitante inferior para o problema e indica a liga *mais apro-*

priada para o período, como se segue.

Para a determinação da liga que será utilizada no primeiro período da produção, é necessário verificar se é possível escolher uma liga que permita a produção das peças dada pela solução ótima do problema de transporte, isto é, as peças i tais que $z_{1i} > 0$. Se não houver uma liga que possa ser escolhida, escolhe-se a liga que menos viola as peças determinadas pela solução ótima do problema.

Determinada a liga que será utilizada no primeiro período, anula-se as variáveis que representam as quantidades das peças que não utilizam a liga escolhida. Se houve violação na escolha da liga, é necessário, resolver novamente o problema de transporte.

Para a determinação das ligas nos demais períodos, deve-se repetir o mesmo processo. Por exemplo, no período t já se tem definido as ligas dos $(t-1)$ períodos anteriores, como ilustra a Figura 4.4.

Determinadas as ligas que serão utilizadas em cada período, resta determinar como será a programação das máquinas em cada período.

Então, para cada período, resolve-se um problema de transporte generalizado para determinar a fração de

tempo utilizada em cada máquina na produção de cada tipo de peça.

FIGURA 4.4 Esquema da programação do forno no período t .

Suponha que a liga j seja produzida no período t . O problema da programação das máquinas pode ser definido por:

$$\begin{aligned}
 &\text{minimizar} \quad \sum_{m=1}^M \sum_{i=1}^P x_{imt} \\
 &\text{sujeito a:} \quad \sum_{i=1}^P x_{imt} \leq 1, \quad m = 1, \dots, M \\
 &\quad \quad \quad \sum_{m=1}^M a_{im} \cdot h_t \cdot x_{imt} = z_{ti}, \quad i = 1, \dots, P \\
 &\quad \quad \quad x_{imt} = 0, \quad m = 1, \dots, M, \quad i \notin L_j \\
 &\quad \quad \quad x_{imt} \geq 0, \quad m = 1, \dots, M, \quad i \in L_j
 \end{aligned}$$

e esquematizado pela Figura 4.5.

FIGURA 4.5 Esquema da programação das máquinas num período t .

O arco (m, i) somente será considerado se $i \in L_j$.

Portanto, resolvendo o problema da programação das máquinas para cada período, obtém-se a fração de tempo de produção de cada peça em cada máquina.

FIGURA 4.6 Árvore com todas as possibilidades de programação do forno.

É necessário observar que a técnica desenvolvida explora um único caminho entre todas as ramificações possíveis. O número de combinações possíveis para a programação do forno é L^T , como mostra a Figura 4.6.

4.3. Algoritmo

Com os resultados obtidos na seção anterior, pode-se resolver o problema de Planejamento e Controle de Produção numa Fundição Automatizada por meio do seguinte algoritmo de busca heurística:

- Passo 0

Fazer $t = 1$.

- Passo 1

Resolver o Problema de Transporte (períodos x peças).

- Passo 2

Determinar a liga j que será utilizada no período t de modo que L_j tenha a menor violação possível.

- Passo 3

Fixar $z_{ti} = 0$, para $i \notin L_j$.

Fazer $t = t + 1$.

Se $t > T$ então ir para o Passo 4.

Se não houver violação então, ir para o Passo 2, caso con-

trário ir para o Passo 1.

◦ Passo 4

Resolver o Problema de Transporte (períodos x peças).

Se a demanda não for atendida então não existe solução ótima. FIM.

Caso contrário, para cada período t , $t = 1, \dots, T$, resolver o Problema de Transporte Generalizado, determinando a fração de tempo utilizada para produzir peças do tipo i na máquina m . FIM.

4.4. Resultados Computacionais

O algoritmo da seção anterior foi implementado em linguagem Turbo Pascal 7, com aproximadamente 30 rotinas, perfazendo um total aproximado de 1100 instruções.

O programa foi desenvolvido para ser utilizado em microcomputadores compatíveis com IBM-PC, utilizando sistema operacional compatível com MS-DOS. Sua implementação, seus testes e as execuções dos exemplos foram efetuadas em microcomputador PC 386 DX40 com 8MB de memória e sistema operacional DOS 6.2. Nesta implementação foram utilizadas técnicas para aproveitar-se, ao máximo, os recursos da linguagem (O'BRIEN, 1992) e do sistema operacional (NORTON, ASHLEY & FERNANDEZ, 1993).

Devido a limitação de memória do compilador utilizado no desenvolvimento do programa, é permitido que execute exemplos com no máximo 15 ligas, 15 máquinas, 15 peças e 15 períodos.

Comparou-se os resultados obtidos através da execução de exemplos do programa desenvolvido com os resultados de um pacote computacional que resolve Problemas de Programação Linear Inteira Mista, o MILP88 V.6.02. Este pacote apresenta uma limitação em relação ao número de restrições e variáveis, permitindo no máximo 255 restrições, 64 variáveis inteiras e 1255 variáveis reais. Pôde-se então verificar, nos testes realizados com problemas pequenos, que a solução obtida pelo algoritmo foi sempre a solução ótima do problema. E, o tempo computacional apresentado pelo algoritmo foi sempre melhor, ou seja, sempre menor em relação ao tempo computacional apresentado pelo pacote, uma vez que este resolve qualquer problema de Programação Linear Inteira Mista, não explorando a estrutura do problema.

A seguir são apresentados três exemplos do problema, gerados para efeito de testes, com o mesmo número de ligas, máquinas e peças, alterando somente o número de períodos, o tempo disponível por período, a capacidade do forno por período e os custos de produção das peças.

Os dados gerais dos três exemplos são apresen-

tados a seguir e, os dados particulares junto com a solução de cada exemplo, são apresentados nas seções 4.4.1, 4.4.2 e 4.4.3.

São dados dos problemas:

- Número de ligas: $L = 6$
- Número de máquinas: $M = 10$
- Número de peças: $P = 12$
- Demanda das peças (tf):

$$b_1 = 160$$

$$b_5 = 300$$

$$b_9 = 200$$

$$b_2 = 240$$

$$b_6 = 320$$

$$b_{10} = 120$$

$$b_3 = 140$$

$$b_7 = 125$$

$$b_{11} = 350$$

$$b_4 = 300$$

$$b_8 = 100$$

$$b_{12} = 160$$

- Conjunto de peças que utilizam as ligas:

$$L_1 = \{3, 5, 8\}$$

$$L_2 = \{4, 6, 8, 12\}$$

$$L_3 = \{1, 3, 4, 5, 7, 9\}$$

$$L_4 = \{2, 10, 11, 12\}$$

$$L_5 = \{3, 10\}$$

$$L_6 = \{11\}$$

- Quantidade de peças possível de ser produzida em cada máquina:

$$a_{im} = m + i - 1, \quad i = 1, \dots, 12, \quad m = 1, \dots, 10$$

4.4.1. Exemplo 1

Neste primeiro exemplo são utilizados os seguintes dados particulares:

- Número de períodos: $T = 3$
- Capacidade do forno por período (tf/h):
 $F_t = 80, t = 1, \dots, 3$
- Número de unidades de tempo por período (h):
 $h_1 = 12 \qquad h_2 = 10 \qquad h_3 = 10$
- Custo de produção de peças do tipo i no período t (\$):
 $c_{ti} = t, t = 1, \dots, 3, i = 1, \dots, 12$

Executando o programa desenvolvido obtém-se o seguinte resultado:

- Produção de peças no período 1 utilizando a liga 3

Máquina	Peça	Tempo	Quantidade
3	9	1,0000	132,0000
4	7	0,0481	5,7704
4	9	0,4722	68,0000
5	5	0,0967	10,4485
5	7	0,9033	119,2296
6	5	1,0000	120,0000
7	5	1,0000	132,0000
8	3	0,4741	56,8889
8	4	0,2652	35,0000
8	5	0,2608	37,5515
9	1	0,3704	40,0000
9	3	0,6296	83,1111
10	1	1,0000	120,0000

Custo de produção no período: \$ 960,00

- Produção de peças no período 2 utilizando a liga 2

Máquina	Peça	Tempo	Quantidade
4	12	0,7667	115,0000
5	8	0,6140	73,6777
6	6	0,7975	87,7273
6	8	0,2025	26,3223
7	6	1,0000	120,0000
8	4	0,1364	15,0000
8	6	0,8636	112,2727
9	4	1,0000	120,0000
10	4	1,0000	130,0000

Custo de produção no período: \$ 1600,00

- Produção de peças no período 3 utilizando a liga 4

Máquina	Peça	Tempo	Quantidade
5	11	0,1806	27,0833
5	12	0,2813	45,0000
6	11	1,0000	160,0000
7	10	0,0417	6,6667
7	11	0,9583	162,9167
8	2	0,3333	30,0000
8	10	0,6667	113,3333
9	2	1,0000	100,0000
10	2	1,0000	110,0000

Custo de produção no período: \$ 2265,00

Custo total de produção: \$ 4825,00

Tempo computacional: 14,00 segundos

4.4.2. Exemplo 2

Alterando apenas o número de períodos, o tempo disponível por período, a capacidade do forno por período e os custos de produção das peças do Exemplo 1, obtém-se o exemplo a seguir.

- Número de períodos: $T = 6$
- Capacidade do forno por período (tf/h):
 $F_t = 40, t = 1, \dots, 6$
- Número de unidades de tempo por período (h):
 $h_1 = 12 \quad h_3 = 10 \quad h_5 = 10$
 $h_2 = 10 \quad h_4 = 12 \quad h_6 = 10$
- Custo de produção de peças do tipo i no período t (\$):
 $c_{ti} = t, t = 1, \dots, 6, i = 1, \dots, 12$

Executando o programa desenvolvido obtém-se o seguinte resultado:

- Produção de peças no período 1 utilizando a liga 4

Máquina	Peça	Tempo	Quantidade
7	11	0,5882	120,0000
8	10	0,4824	98,4000
9	2	0,9000	108,0000
9	10	0,1000	21,6000
10	2	1,0000	132,0000

Custo de produção no período: \$ 480,00

- Produção de peças no período 2 utilizando a liga 3

Máquina	Peça	Tempo	Quantidade
7	4	0,5125	51,2500
8	4	1,0000	110,0000
9	3	1,0000	110,0000
10	3	0,1250	15,0000
10	4	0,8750	113,7500

Custo de produção no período: \$ 800,00

- Produção de peças no período 3 utilizando a liga 3

Máquina	Peça	Tempo	Quantidade
7	5	0,4227	46,5000
8	5	1,0000	120,0000
9	5	1,0000	130,0000
10	1	0,8500	85,0000
10	3	0,1250	15,0000
10	5	0,0250	3,5000

Custo de produção no período: \$ 1200,00

- Produção de peças no período 4 utilizando a liga 2

Máquina	Peça	Tempo	Quantidade
8	6	0,4444	69,3333
8	8	0,5556	100,0000
9	6	0,5923	99,5128
9	12	0,1458	35,0000
10	4	0,1603	25,0000
10	6	0,8397	151,1538

Custo de produção no período: \$ 1920,00

- Produção de peças no período 5 utilizando a liga 3

Máquina	Peça	Tempo	Quantidade
8	7	0,6071	85,0000
8	9	0,1875	30,0000
9	9	1,0000	170,0000
10	1	0,7500	75,0000
10	7	0,2500	40,0000

Custo de produção no período: \$ 2000,00

- Produção de peças no período 6 utilizando a liga 4

Máquina	Peça	Tempo	Quantidade
9	11	0,3750	71,2500
9	12	0,6250	125,0000
10	11	0,7938	158,7500

Custo de produção no período: \$ 2130,00

Custo total de produção: \$ 8530,00

Tempo computacional: 15,38 segundos

4.4.3. Exemplo 3

Dobrando novamente o número de períodos e alterando a capacidade do forno nos períodos do Exemplo 1, adicionando o tempo disponível por período e os custos de produção das peças, tem-se os seguintes dados para o problema.

◦ Número de períodos: $T = 12$

◦ Capacidade do forno por período (tf/h):

$$F_t = 20, t = 1, \dots, 12$$

◦ Número de unidades de tempo por período (h):

$$h_1 = 12$$

$$h_5 = 10$$

$$h_9 = 10$$

$$h_2 = 10$$

$$h_6 = 10$$

$$h_{10} = 12$$

$$h_3 = 10$$

$$h_7 = 12$$

$$h_{11} = 10$$

$$h_4 = 12$$

$$h_8 = 10$$

$$h_{12} = 10$$

◦ Custo de produção de peças do tipo i no período t (\$):

$$c_{ti} = t, t = 1, \dots, 12, i = 1, \dots, 12$$

Executando o programa desenvolvido obtém-se o seguinte resultado:

◦ Produção de peças no período 1 utilizando a liga 3

Máquina	Peça	Tempo	Quantidade
9	7	0,6500	117,0000
10	1	0,9583	115,0000
10	7	0,0417	8,0000

Custo de produção no período: \$ 240,00

- Produção de peças no período 2 utilizando a liga 4

Máquina	Peça	Tempo	Quantidade
9	2	1,0000	100,0000
10	2	0,9091	100,0000

Custo de produção no período: \$ 400,00

- Produção de peças no período 3 utilizando a liga 3

Máquina	Peça	Tempo	Quantidade
9	3	1,0000	110,0000
10	1	0,4500	45,0000
10	3	0,2500	30,0000
10	4	0,1154	15,0000

Custo de produção no período: \$ 600,00

- Produção de peças no período 4 utilizando a liga 3

Máquina	Peça	Tempo	Quantidade
9	5	0,5306	82,7692
10	4	0,8974	140,0000
10	5	0,1026	17,2308

Custo de produção no período: \$ 960,00

- Produção de peças no período 5 utilizando a liga 3

Máquina	Peça	Tempo	Quantidade
9	5	0,4615	60,0000
10	5	1,0000	140,0000

Custo de produção no período: \$ 1000,00

- Produção de peças no período 6 utilizando a liga 2

Máquina	Peça	Tempo	Quantidade
9	12	0,3750	75,0000
10	4	0,3462	45,0000
10	6	0,5333	80,0000

Custo de produção no período: \$ 1200,00

- Produção de peças no período 7 utilizando a liga 2

Máquina	Peça	Tempo	Quantidade
9	6	1,0000	168,0000
10	6	0,4000	72,0000

Custo de produção no período: \$ 1680,00

- Produção de peças no período 8 utilizando a liga 2

Máquina	Peça	Tempo	Quantidade
9	8	0,3798	60,7692
10	4	0,7692	100,0000
10	8	0,2308	39,2308

Custo de produção no período: \$ 1600,00

- Produção de peças no período 9 utilizando a liga 3

Máquina	Peça	Tempo	Quantidade
9	9	1,0000	170,0000
10	9	0,1667	30,0000

Custo de produção no período: \$ 1800,00

- Produção de peças no período 10 utilizando a liga 4

Máquina	Peça	Tempo	Quantidade
9	11	0,1551	35,3589
10	2	0,3030	40,0000
10	10	0,2193	50,0000
10	11	0,4777	114,6411

Custo de produção no período: \$ 2400,00

- Produção de peças no período 11 utilizando a liga 6

Máquina	Peça	Tempo	Quantidade
10	11	1,0000	200,0000

Custo de produção no período: \$ 2200,00

- Produção de peças no período 12 utilizando a liga 4

Máquina	Peça	Tempo	Quantidade
10	10	0,3684	70,0000
10	12	0,4048	85,0000

Custo de produção no período: \$ 1860,00

Custo total de produção: \$ 15940,00

Tempo computacional: 19,83 segundos

A tabela a seguir mostra o tempo computacional dos exemplos apresentados, com a dimensão dos problemas na sua forma original.

	Número de Restrições	Número de Variáveis Inteiras	Número de Variáveis Reais	Tempo Computacional (s)
Exemplo 1	378	18	360	14,00
Exemplo 2	744	36	720	15,38
Exemplo 3	1476	72	1440	19,83

5. CONCLUSÕES E PERSPECTIVAS

O Método de Decomposição Híbrida, visto no Capítulo 3, tem se mostrado bastante eficiente para a resolução de alguns Problemas de Programação Linear Inteira Mista. ROY (1986) mostra que o Problema de Localização de Centrais Capacitadas converge mais rápido com o algoritmo de Decomposição Híbrida, se comparado com o Método de Decomposição de Benders ou com a Relaxação Lagrangeana, que utiliza o Método do Subgradiente.

Este método híbrido, embora tenha grande potencial para aplicação, não foi implementado neste trabalho. Deu-se preferência à implementação do algoritmo que utiliza uma busca heurística, desenvolvido no Capítulo 4, para a resolução do Problema de Planejamento e Controle de Produção numa Fundação Automatizada. Este algoritmo e sua implementa-

ção foram desenvolvidos especificamente para este trabalho.

Este algoritmo é heurístico, pois explora um único caminho entre todas as ramificações possíveis. Numa pesquisa futura pode-se desenvolver um algoritmo que investigue todas as ramificações implicitamente, isto é o Método de *Branch and Bound*.

O algoritmo heurístico implementado apresentou soluções ótimas com um baixo custo computacional quando comparado com exemplos pequenos executados pelo pacote computacional MILP88. Não foi possível comparar exemplos maiores devido a limitação do pacote MILP88, como foi observado no Capítulo 4, Seção 4.4.

É interessante verificar também, a existência de outros problemas com a mesma estrutura lógica do Problema de Planejamento e Controle de Produção numa Fundição Automatizada. Assim sendo, tem-se um algoritmo eficiente para resolução de outros problemas, além do que aqui estudado.

É necessário salientar que o desenvolvimento de algoritmos eficientes para a resolução de uma classe de problemas, como o problema aqui estudado, é de grande importância, já que os pacotes computacionais não exploram a estrutura do problema, apresentando assim um custo computacional alto.

6. BIBLIOGRAFIA

AGMON, S. The Relaxation Method for Linear Inequalities.
Canadian Journal of Mathematics, 6:382-392, 1954.

AOYANAGI, K. *Decomposição em Programação Inteira*. São Carlos, 1991. 79p. (Tese - Mestrado - Instituto de Ciências Matemáticas de São Carlos).

BAZARAA, M. S. & JARVIS, J. J. *Linear Programming and Network Flows*. New York, John Wiley & Sons, 1977. 575p.

BENDERS, J. F. Partitioning Procedures for Solving Mixed-Variables Programming Problems. *Numerische Mathematik*, 4:238-252, 1962.

COLLINS, W. J. *Programação estruturada com estudos de casos*

em Pascal. McGraw-Hill, São Paulo, 1988. 529p.

FLORENTINO, H. O. *Relaxação Lagrangeana em Programação Inteira*. São Carlos, 1990. 74p. (Tese - Mestrado - Instituto de Ciências Matemáticas de São Carlos).

GEOFFRION, A. M. Lagrangean Relaxation for Integer Programming. *Mathematical Programming Study*, 2:82-114, 1974.

* JENSEN, P. A. & BARNES, J. W. *Network Flow Programming*. New York, John Wiley & Sons, 1980. 423p.

* KENNINGTON, J. L. & HELGASON, R. V. *Algorithms for Network Programming*. New York, John Wiley & Sons, 1980. 304p.

LASDON, L. S. *Optimization Theory for Large Systems*. New York, MacMillan Publishing Company, 1970. 532p.

MINOUX, M. *Mathematical Programming: Theory and Algorithms*. New York, John-Wiley and Sons, 1986.

MOTZKIN, T. S. & SCHOENBERG, I. Relaxation Method for Linear Inequalities. *Canadian Journal of Mathematics*, 6:393-404, 1954.

NORTON, P.; ASHLEY, R.; FERNANDEZ, J. *Peter Norton MS DOS 6 Avançado*. Makron Books do Brasil Editora Ltda., São Paulo,

1993. 527p.

O'BRIEN, S. K. *Turbo Pascal: completo e total*. Makron Books do Brasil Editora Ltda., São Paulo, 1992. 742p.

POLJAK, B. T. A General Method of Solving Extremum Problems. *Soviet Mathematic Daklady*, 8(3):593-597, 1967.

POLJAK, B. T. Minimization of Unsmooth Functionals. *URSS Computational Mathematics and Mathematical Physics*, 9:14-29, 1969.

ROY, T. J. V. Cross Decomposition for Mixed Integer Programming. *Mathematical Programming*, 25:46-63, 1983.

ROY, T. J. V. A Cross Decomposition Algorithm for Capacitated Facility Location. *Operations Research*, 34(1):145-163, 1986.

SALKIN, H. M. *Integer Programming*. Massachusetts, Addison-Wesley Publishing Company, 1975. 552p.

SCHIRIJVER, A. *Theory of Linear and Integer Programming*. Chichester, John Wiley & Sons, 1986. 482p.

SHOR, N. Z. On the Rate Convergence of the Generalized Gradient Method. *Kibernetika*, 4(3):98-99, 1968.