

Heurística para Obtenção de uma Base Ótima para o Problema de Multifluxo

V. PODESTÁ-GOMES¹, C. PERIN², Departamento de Matemática Aplicada, IMECC, Universidade Estadual de Campinas-UNICAMP, Cx.P. 6065, 13083-970 Campinas, SP, Brasil.

Resumo. Apresentamos uma heurística que utiliza uma solução “quase-ótima” interior obtida por um método primal-dual de pontos interiores para determinar uma solução básica ótima do problema de multifluxo. Esta heurística se baseia no comportamento das matrizes de *scaling* de cada produto e do acoplamento, para construir a partição nos arcos que forma uma base ótima, isto é, quais arcos formam a árvore geradora de cada produto, quais arcos formam ciclos com suas respectivas árvores geradoras e quais arcos correspondem às variáveis de folga do acoplamento.

1. Descrição do Problema

O Problema de Multifluxo (“Multicommodity Flow Problem”) surge da necessidade de planejar o transporte de p diferentes tipos de produtos a partir de determinados centros produtores (fontes) para determinados locais de consumo (destinos), ao longo de uma rede com m nós e n arcos. Além das restrições usuais de conservação de fluxo para cada produto em cada nó da rede, existem as restrições de acoplamento nos arcos, já que os diferentes produtos compartilham o mesmo conjunto de arcos. Em geral, procura-se uma solução de menor custo total de transporte. Uma das alternativas mais comuns de abordar o problema é trabalhar com o modelo de programação linear

$$\begin{array}{ll} \min & \sum_{k=1}^p (c^k)' x^k \\ \text{s.a.} & \left\{ \begin{array}{l} Ax^k = b^k, \quad k = 1, \dots, p \\ \sum_{k=1}^p x^k \leq d \\ 0 \leq x^k \leq u^k, \quad k = 1, \dots, p \end{array} \right. \end{array}$$

onde A é uma $m \times n$ matriz de incidência da rede, c^k, b^k, u^k (de dimensões compatíveis) são vetores de custo de transporte, de oferta/demanda do produto e de

¹E-mail: valeria@ime.unicamp.br

²E-mail: clovis@ime.unicamp.br

capacidade de fluxo, respectivamente, para o produto k e os vetores x^k são formados pelas variáveis de fluxo dos produtos. A função objetivo estabelece que o custo total de transporte deve ser minimizado; o primeiro grupo de restrições compõe as restrições de oferta/demanda para cada um dos produtos; as restrições que se seguem são as de acoplamento; as últimas restrições estabelecem a canalização de cada variável de fluxo.

A literatura mostra que os métodos de pontos interiores são competitivos para este problema. Dentre eles, a versão primal-dual tem mostrado grande viabilidade prática, conforme, por exemplo, Kojima, Mizuno e Yoshise [4], McShane, Monma e Shanno [5].

Resumidamente, a versão do método primal-dual de pontos interiores utilizada [3] consiste em, dado um ponto inicial interior não necessariamente factível para o problema primal e para o seu dual, e um certo parâmetro dito de centragem, encontrar uma direção de deslocamento para obtermos o próximo ponto. A direção tomada é a de Newton e, de fato, esta é a etapa do método que demanda o maior esforço computacional, onde necessitamos resolver um sistema linear com $mp + n$ equações. Obtida esta direção e usando alguma regra para obter o novo parâmetro de centragem, repetimos o processo a partir do novo ponto, até que um critério de parada seja satisfeito. No decorrer das iterações, a factibilidade para os problemas primal e dual vão sendo alcançadas, bem como a otimalidade.

Este trabalho tem por objetivo apresentar uma heurística que, a partir de uma solução “quase-ótima” fornecida pelo método primal-dual de pontos interiores, encontra uma base ótima para o problema. Antes de descrevermos a heurística, faremos uma breve descrição de uma base de multifluxo.

2. Base de multifluxo

É conhecido que, se a rede é conectada, então $\text{posto}(A) = m - 1$, e com a eliminação de qualquer linha de A obtemos uma matriz de posto completo. Além disso, adicionamos as variáveis de folga x^{p+1} às restrições de acoplamento, de maneira que a matriz dos coeficientes relativa ao problema original na sua forma padrão será:

$$\begin{bmatrix} A & 0 & \cdots & 0 & 0 \\ 0 & A & \cdots & 0 & 0 \\ & & \ddots & & \\ 0 & 0 & \cdots & A & 0 \\ I & I & \cdots & I & I \end{bmatrix},$$

onde A , com um abuso de linguagem, agora é a matriz de incidência nó-arco com uma linha retirada e I é a matriz identidade de ordem n .

Qualquer base do problema tem dimensão $\hat{m} = (m - 1)p + n$ e tem a forma:

$$\widehat{B} = \left[\begin{array}{cccc|cc} [B^1|R^1] & 0 & \cdots & 0 & 0 & \\ 0 & [B^2|R^2] & \cdots & 0 & 0 & \\ & & \ddots & & & \\ 0 & 0 & \cdots & [B^p|R^p] & 0 & \\ [E^1|F^1] & [E^2|F^2] & \cdots & [E^p|F^p] & E^{p+1} & \end{array} \right],$$

onde

- B^k , $k = 1, \dots, p$, são bases (árvores geradoras com raiz) de A , com dimensão $(m - 1) \times (m - 1)$. Qualquer uma das matrizes R^k , $k = 1, \dots, p$, pode ser vazia.
- E^k e F^k seguem a mesma partição de $[B^k|R^k]$. E^k , F^k e E^{p+1} são matrizes formadas por colunas da matriz identidade de ordem n . A linha onde aparece o número 1 em E^k identifica qual arco foi usado em B^k , $k = 1, \dots, p$; a linha onde aparece o número 1 em F^k identifica qual arco foi usado em R^k , $k = 1, \dots, p$. As colunas de E^{p+1} identificam as variáveis de folga das restrições de acoplamento que estão na base.

Com estas partições, reescrevemos \widehat{B} :

$$\widehat{B} = \left[\begin{array}{cccc|cccc|c} B^1 & 0 & \cdots & 0 & R^1 & 0 & \cdots & 0 & 0 \\ 0 & B^2 & \cdots & 0 & 0 & R^2 & \cdots & 0 & 0 \\ & & \ddots & & & & \ddots & & \\ 0 & 0 & \cdots & B^p & 0 & 0 & \cdots & R^p & 0 \\ \hline E^1 & E^2 & \cdots & E^p & F^1 & F^2 & \cdots & F^p & E^{p+1} \end{array} \right].$$

A conclusão que se segue pode ser encontrada em Bazaraa e outros [1] e em Kennington e Helgason [2]: *cada base para o problema de multifluxo é um conjunto de colunas linearmente independente formado por três tipos de arcos: arcos que formam uma árvore geradora (com raiz) para cada produto (B^k), arcos que formam ciclos com suas respectivas árvores geradoras (R^k) e arcos correspondentes às variáveis de folga das restrições de acoplamento (E^{p+1}).*

3. Heurística para obtenção de uma base ótima

A heurística proposta determina uma solução básica ótima a partir de uma solução “quase-ótima” interior obtida ao final do método primal-dual de pontos interiores. Vamos supor que a rede associada ao problema é conectada, isto é, a dimensão de qualquer base do problema é $\widehat{m} = (m - 1)p + n$.

Nesta heurística, vamos tentar identificar os arcos que formam as árvores geradoras de cada produto (arcos de B^k), os arcos que formam ciclos com sua respectiva árvore geradora (arcos de R^k) e as variáveis de folga das restrições de acoplamento. Os arcos não básicos estarão em N^k .

Isto será feito através das matrizes de *scaling* Θ^k , para cada produto k ($k = 1, \dots, p$) e do acoplamento ($k = p + 1$), que são matrizes diagonais e positivas,

formadas a partir da solução primal-dual corrente fornecida pelo método primal-dual de pontos interiores. À medida que a solução fornecida por este método se aproxima de uma solução ótima, os elementos Θ_j^k se aproximam ou de zero ou de infinito, como podemos observar a seguir (s^k são as variáveis de folga primais da canalização e z^k são as variáveis de folga duais para cada produto k):

$$\Theta_j^k = (x_j^k s_j^k) / (s_j^k z_j^k + x_j^k w_j^k), \quad k = 1, \dots, p, \quad j = 1, \dots, n$$

$$\Theta_j^{p+1} = x_j^{p+1} / z_j^{p+1}, \quad j = 1, \dots, n.$$

Para soluções “quase-ótimas”, temos, das expressões acima:

- se $x_j^k \approx 0$, então $s_j^k \approx u_j^k \Rightarrow \Theta_j^k \approx 0$;
- se $x_j^k \approx u_j^k$, então $s_j^k \approx 0 \Rightarrow \Theta_j^k \approx 0$;
- se $0 < x_j^k < u_j^k$, então $z_j^k \approx 0$ e $w_j^k \approx 0 \Rightarrow \Theta_j^k \approx \infty$.
- se $x_j^{p+1} > 0$, então $z_j^{p+1} \approx 0 \Rightarrow \Theta_j^{p+1} \approx \infty$;
- se $z_j^{p+1} > 0$, então $x_j^{p+1} \approx 0 \Rightarrow \Theta_j^{p+1} \approx 0$.

Supondo que o problema é *não degenerado*, vemos que se $\Theta_j^k \rightarrow 0$, então, ou $x_j^k \rightarrow 0$ ou $x_j^k \rightarrow u_j^k$; isto é, o arco j do produto k deve ser não básico. Por outro lado, se $\Theta_j^k \rightarrow \infty$, então $0 < x_j^k < u_j^k$ e o arco j do produto k deve ser básico. Resta saber se este arco está na árvore geradora do produto k ou se está fazendo um ciclo com ela. Em outras palavras, seguindo a partição que usamos anteriormente, no primeiro caso teríamos um arco fazendo parte da matriz N^k . No último caso, o arco poderia estar tanto na matriz B^k quanto na matriz R^k — e nossa maior dificuldade está exatamente em distinguir arcos de B^k de arcos de R^k .

Para o acoplamento, as conclusões são semelhantes: se $\Theta_j^{p+1} \rightarrow 0$, então $x_j^{p+1} \rightarrow 0$, isto é, a variável de folga da j -ésima restrição de acoplamento deve ser não básica. Por outro lado, se $\Theta_j^{p+1} \rightarrow \infty$, então $x_j^{p+1} > 0$, isto é, a variável de folga da j -ésima restrição de acoplamento deve ser básica.

Utilizaremos o vetor α , com n valores inteiros, para guardar a informação sobre qual produto cada um dos n arcos está representando, isto é, $\alpha_j = k$ significa que o arco j está representando o produto k (forma um ciclo com a árvore geradora do produto k) e $\alpha_j = p + 1$ significa que o arco j está representando uma variável de folga das restrições de acoplamento.

Iniciamos nossa heurística fazendo

$$\alpha_j = p + 1, \quad \text{se } \Theta_j^{p+1} > 1,$$

$$\alpha_j = 0, \quad \text{caso contrário.}$$

Vamos supor que n_a componentes do vetor α são nulas. Note que, se $n_a = 0$, temos um problema fácil de resolver, pois ele é totalmente desacoplado: todas as n variáveis de folga das restrições de acoplamento estão na base. Os outros $(m - 1)p$

arcos básicos estão formando árvores geradoras em seus produtos. Vamos então supor que $n_a > 0$.

Os arcos j tais que $\alpha_j = 0$ são identificados no vetor de n_a componentes inteiras β , isto é,

$$\beta_t = j, \text{ se } \alpha_j = 0, \quad t = 1, \dots, n_a.$$

Utilizaremos também a matriz ψ , de dimensão $p \times n_a$, formada pelos valores inteiros 0, 1 ou 2, para assinalar o estado de um arco j que está no vetor β (ou seja, ainda não sabemos qual produto este arco está representando).

Inicialmente, fazemos:

$$\psi_{k,t} = 1, \text{ se } \Theta_j^k > 1,$$

$$\psi_{k,t} = 0, \text{ caso contrário,}$$

para $k = 1, \dots, p$, $t = 1, \dots, n_a$ e $j = \beta_t$.

Um elemento que vale 1 em ψ significa que o arco correspondente pode fazer parte da árvore geradora do produto k ; se o elemento for nulo, este arco não pode fazer parte desta árvore.

Por exemplo, considere um problema com 2 produtos, 4 nós e 6 arcos, cuja rede é dada pela Figura 1. Os arcos são dados por $\{e_1, \dots, e_6\}$.

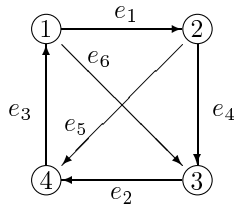


Figura 1: rede exemplo para a heurística

Vamos supor que, ao final do método primal-dual de pontos interiores, temos, para os produtos 1 e 2 e para o acoplamento, respectivamente:

$$\Theta^1 = \text{diag} (\infty, \infty, \infty, \infty, 0, 0),$$

$$\Theta^2 = \text{diag} (\infty, \infty, 0, 0, \infty, \infty),$$

$$\Theta^3 = \text{diag} (0, 0, \infty, \infty, \infty, \infty).$$

Neste caso, teremos:

vetor α inicial: $\alpha = (0, 0, 3, 3, 3, 3)$

vetor β : $\beta = (1, 2); \quad n_a = 2$

matriz ψ inicial: $\psi =$

produtos	e_1	e_2
1	1	1
2	1	1

Na próxima etapa, vamos tentar construir uma árvore geradora para cada produto k . Os arcos que podem fazer parte desta árvore são aqueles que possuem Θ_j^k e Θ_j^{p+1} grandes. Nesta primeira tentativa, ou conseguimos formar uma árvore geradora para cada produto ou conseguimos uma floresta para o produto (caso os arcos que podem fazer parte da árvore não formem um subgrafo conectado com $m - 1$ nós). Neste último caso, vamos atualizar a matriz ψ : vamos buscar no vetor β os arcos que possam completar a árvore geradora deste produto. Estes arcos são os que não formarão ciclos na floresta deste produto. Isto é feito da seguinte maneira: trocamos $\psi_{k,t} = 1$ para $\psi_{k,t} = 2$ se o produto k ainda não possui sua árvore geradora e o t -ésimo arco do vetor β (arco j) não forma um ciclo na floresta deste produto. Em outras palavras, $\psi_{k,t} = 2$ significa que o arco j tanto pode fazer parte da matriz B^k quanto da matriz R^k .

No exemplo, teríamos como primeira tentativa de árvore geradora:

produto 1:

arcos candidatos à floresta do produto 1: e_3, e_4

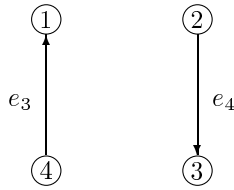


Figura 2: floresta do produto 1

produto 2:

arcos candidatos à floresta do produto 2: e_5, e_6

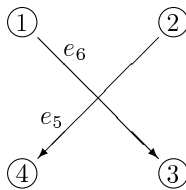


Figura 3: floresta do produto 2

Note que não temos árvore geradora para nenhum dos produtos. As correções

na matriz ψ nos levam a:

$$\psi : \begin{array}{|c|c|c|} \hline \text{produtos} & e_1 & e_2 \\ \hline 1 & 1 & 1 \\ \hline 2 & 1 & 1 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|} \hline \text{produtos} & e_1 & e_2 \\ \hline 1 & 2 & 2 \\ \hline 2 & 2 & 2 \\ \hline \end{array}$$

Continuando a heurística, pesquisamos a matriz ψ com o objetivo de atualizar os elementos nulos do vetor α pelo número do produto ao qual o arco será alocado.

Se $\psi_{k,t} = 1$, o arco j correspondente ($j = \beta_t$) será alocado ao produto k , isto é, $\alpha_j = k$. Além disso, vamos zerar $\psi_{k,t}$ para indicar que este arco já foi eliminado (alocado). De fato, esta coluna de ψ pode ser eliminada. Percorremos toda a coluna t de ψ e procuramos por elementos com valor 2: para cada linha onde ocorrer $\psi_{k,t} = 2$, o arco j será alocado na floresta do produto k correspondente (ver observação no final do algoritmo).

Em seguida, pesquisamos os elementos de ψ com valor 2 — sempre em colunas que correspondem aos arcos que ainda não foram alocados.

Vamos pesquisar as *colunas*. Se existe um único elemento igual a 2 na coluna t e este elemento ocorre na linha k de ψ , então fazemos $\psi_{k,t} = 1$ e o arco j correspondente será alocado ao produto k , isto é, $\alpha_j = k$. Novamente, esta coluna de ψ é eliminada.

Agora, vamos pesquisar as *linhas*. Se existe um único elemento igual a 2 na linha k de ψ , e este elemento está na coluna t , então o arco j correspondente é colocado na floresta do produto k (este arco é necessário para completar a árvore geradora deste produto) e fazemos $\psi_{k,t} = 0$.

(No exemplo, não temos a situação de um único elemento de valor 2 em uma linha ou coluna da matriz ψ ; os elementos de valor 2 na matriz ψ formam um ciclo.)

Note que, em todas as situações onde um arco é colocado na floresta de um produto, é verificado antes se ele não forma um ciclo nesta floresta.

Este processo é repetido enquanto pudermos encontrar os casos que descrevemos anteriormente. Como a matriz ψ tem seus valores sistematicamente alterados, ao final deste processo esta matriz apresenta a seguinte propriedade: ou ela se transformou na matriz nula, ou ela apresenta pelo menos dois coeficientes 2 em cada linha e em cada coluna. Assim, temos duas situações possíveis:

(i) a matriz ψ foi completamente zerada. Isto significa que conseguimos encontrar as árvores geradoras de todos os produtos e todos os n arcos foram alocados (ou para um produto ou para o acoplamento), isto é, o vetor α foi completamente determinado. Neste caso, finalizamos a heurística;

(ii) restam na matriz ψ vários elementos iguais a 2. Isto significa que nem todos os produtos possuem ainda suas árvores geradoras, e o vetor α ainda possui elementos nulos. Neste caso, podemos escolher qualquer produto que não possui árvore geradora, digamos, o produto k e escolhemos ainda um arco j tal que $\alpha_j = 0$ para completar a árvore deste produto ($\psi_{k,t} = 2$, para $j = \beta_t$). Repetimos o processo, até que a situação (i) seja alcançada.

No exemplo, estamos na situação (ii). Seleccionamos então o produto 1 e o arco e_1 para ser colocado em sua floresta geradora, completando a árvore geradora deste produto, conforme a Figura 4 .

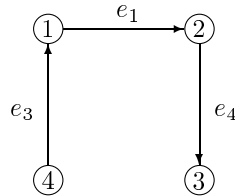


Figura 4: árvore geradora do produto 1

Isto nos leva sucessivamente à:

$$\alpha : (0, 0, 3, 3, 3, 3) \rightarrow (2, 0, 3, 3, 3, 3) \rightarrow (2, 1, 3, 3, 3, 3)$$

$$\psi : \begin{array}{|c|c|c|} \hline \text{prod.} & e_1 & e_2 \\ \hline 1 & 2 & 2 \\ \hline 2 & 2 & 2 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|} \hline \text{prod.} & e_1 & e_2 \\ \hline 1 & 0 & 1 \\ \hline 2 & 2 & 2 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|} \hline \text{prod.} & e_1 & e_2 \\ \hline 1 & 0 & . \\ \hline 2 & 1 & . \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|} \hline \text{prod.} & e_1 & e_2 \\ \hline 1 & . & . \\ \hline 2 & . & . \\ \hline \end{array}$$

E, para o produto 2, sua árvore geradora é completada com o arco e_2 , conforme a Figura 5 .

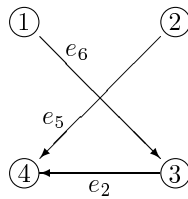


Figura 5: árvore geradora do produto 2

Agora, conseguimos as árvores geradoras para os dois produtos, o vetor α ficou completo e agora a matriz ψ tornou-se vazia. Temos então uma base para o problema.

$$\begin{aligned} B^1 &= (e_3, e_4, e_1); & R^1 &= (e_2); & N^1 &= (e_5, e_6). \\ B^2 &= (e_5, e_6, e_2); & R^2 &= (e_1); & N^2 &= (e_3, e_4). \end{aligned}$$

Em resumo, esta heurística pode ser colocada no algoritmo que se segue, onde utilizamos os seguintes vetores e matrizes:

$\alpha[1..n]$: vetor inteiro (já descrito);

$\beta[1..n_a]$: vetor inteiro (já descrito);

$peso[1..n]$: vetor real; contém os pesos dos arcos que vão compor a floresta geradora de um produto;

$\psi[1..p, 1..n_a]$: matriz inteira (já descrita);

$\Theta[1..(p+1), 1..n]$: matriz real; contém as matrizes de *scaling* de cada um dos produtos, mais a do acoplamento.

Os vetores α e β indicam os arcos que formam as matrizes R^k . Os arcos das florestas geradoras formam as matrizes B^k .

Algoritmo (*obtenção de uma base ótima*)

• (*Inicialização*)

$n_a \leftarrow 0$;

para $k \leftarrow 1, \dots, p$ e $j \leftarrow 1, \dots, n$, fazer:

se $\Theta[k, j] > 1$ então $\alpha[j] \leftarrow p + 1$

senão

$\alpha[j] \leftarrow 0$;

$n_a \leftarrow n_a + 1$;

$\beta[n_a] \leftarrow j$;

para $k \leftarrow 1, \dots, p$ e $t \leftarrow 1, \dots, n_a$, fazer:

se $\Theta[k, \beta[t]] > 1$ então $\psi[k, t] \leftarrow 1$

senão $\psi[k, t] \leftarrow 0$;

• (*primeira tentativa de árvore geradora*)

para $k \leftarrow 1, \dots, p$, fazer:

para $j \leftarrow 1, \dots, n$, fazer:

se $\Theta[k, j] > 1$ e $\Theta[p+1, j] > 1$

então $peso[j] \leftarrow \Theta[k, j]$;

senão $peso[j] \leftarrow -1$;

montar a floresta do produto k ;

se for desconectada, então

para $t \leftarrow 1, \dots, n_a$, fazer:

se $\psi[k, t] = 1$ e o arco $j = \beta[t]$ pode ser conectado à floresta do produto k sem formar ciclo, então $\psi[k, t] \leftarrow 2$;

• (*correção da matriz ψ*)

Enquanto for possível encontrar na matriz ψ um elemento igual a 1 ou um único elemento igual a 2 em uma linha ou coluna, repetir os procedimentos seguintes:

-(*procurar 1*)

para $k \leftarrow 1, \dots, p$ e para $t \leftarrow 1, \dots, n_a$, fazer:

se $\psi[k, t] = 1$ e $\alpha[\beta[t]] = 0$

então

```

 $\alpha[\beta[t]] \leftarrow k$  ;
 $\psi[k, t] \leftarrow 0$  ;
para  $\bar{k} = 1, \dots, p$  fazer:
  se  $\psi[\bar{k}, \beta[t]] = 2$  então colocar o arco  $\beta[t]$  na floresta
    do produto  $\bar{k}$  e eliminar a coluna  $t$  de  $\psi$  ;
-(procurar único 2 na coluna)
para  $t \leftarrow 1, \dots, n_a$ , fazer:
  se existe um único 2 na coluna  $t$  ( e ocorre na linha  $k$ )
    então
       $\psi[k, t] \leftarrow 1$  ;
       $\alpha[\beta[t]] \leftarrow k$  ;
      eliminar a coluna  $t$  de  $\psi$  ;

-(procurar único 2 na linha)
para  $k \leftarrow 1, \dots, p$ , fazer:
  se existe um único 2 na linha  $k$  ( e ocorre na coluna  $t$ )
    então colocar o arco  $j = \beta[t]$  na floresta do produto  $k$  ;

• (teste de parada)
Se (o número de árvores geradoras for  $p$ ) e  $(\alpha[j] \neq 0, \forall j = 1, \dots, n)$ 
então fim do procedimento
senão
  selecionar  $j = \beta[t]$  tal que  $\alpha[j] = 0$  ;
  selecionar  $k$  tal que este produto não possua árvore geradora;
  colocar o arco  $j$  na floresta do produto  $k$ 
  voltar à correção da matriz  $\psi$  .

```

Obs.: o procedimento “colocar o arco j na floresta do produto k ” inclui atualizações na matriz ψ tais como:

- (i) $\psi[k, t] = 0$;
- (ii) se a árvore geradora deste produto foi completada com este arco, então, percorrer a linha k de ψ e procurar elementos iguais a 2: se $\psi[k, \cdot] = 2$, fazer $\psi[k, \cdot] = 1$;
- (iii) caso a árvore geradora não tenha sido completada, percorrer ainda a linha k de ψ e procurar elementos iguais a 2. Verificar, então, se o arco correspondente não forma ciclo com esta árvore geradora. Neste caso, estes elementos de ψ são trocados por 1.

4. Experimentos Computacionais

Esta heurística foi implementada, juntamente com o método primal-dual de pontos interiores aplicado ao problema de multifluxo. Vários experimentos computacionais foram efetuados para testá-la, através de um gerador de problemas de multifluxo. Embora nossa exposição tenha considerado problemas *não degenerados*, em vários testes consideramos problemas *degenerados*. Para problemas não degenerados, esta heurística se mostrou bastante robusta: de todos os testes efetuados, apenas em

dois casos tivemos falha. No entanto, para problemas degenerados, ela quase nunca obteve sucesso.

Na Tabela 1, apresentamos alguns resultados, retirados dos vários testes realizados. Nestes problemas, o valor de n_a (número de arcos verdadeiramente acoplados) é pequeno (o que realmente ocorre em problemas práticos). As colunas m , n , p , \hat{m} representam, respectivamente, o número de nós da rede, o número de arcos, o número de produtos e a dimensão da base. A coluna “degenerado” indica se o problema tem degeneração primal ou dual ou se o problema não é degenerado. A coluna “sucesso” indica se a heurística obteve ou não sucesso.

m	n	p	\hat{m}	n_a	degenerado	sucesso
10	21	10	111	0	não	sim
20	40	10	230	2	não	sim
50	80	10	570	6	não	sim
50	100	10	590	4	não	não
50	102	10	592	7	não	sim
80	104	10	894	6	não	sim
100	200	5	695	29	não	sim
100	203	5	698	8	primal	não
100	204	5	699	44	dual	não
150	256	5	1001	27	primal	não
150	305	5	1050	28	primal	não
180	402	3	939	14	dual	não
180	402	3	939	43	não	sim
200	408	10	2398	15	dual	sim
200	404	10	2394	25	primal	não
300	425	5	1920	29	primal	não
300	420	5	1915	24	não	sim
300	512	3	1409	32	dual	não
300	607	5	2102	38	dual	sim
300	607	5	2102	40	primal	não
300	608	5	2103	39	não	sim

Tabela 1: *heurística para solução básica ótima*

Convém observar que, encontrada a partição básica ótima pela heurística, é necessário resolver ainda os sistemas lineares que encontram as variáveis básicas primais e duais. A solução de cada um destes sistemas envolve a resolução de p sistemas lineares triangulares, cada um com a matriz B^k de um produto, onde utilizamos uma estrutura de dados adequada.

Consideramos “sucesso” para a heurística quando as quatro condições seguintes são satisfeitas:

- (i) a solução básica primal encontrada é factível;
- (ii) a solução básica dual encontrada é factível;

- (iii) o valor das funções objetivo primal e dual são iguais, a menos de uma pequena tolerância;
- (iv) este valor é igual (a menos de uma tolerância) ao valor encontrado para a função objetivo na última iteração do método primal-dual.

A heurística “fracassa” quando:

- (i) o vetor α não ficou completo (ainda existem posições nulas);
- (ii) a matriz ψ não se tornou vazia;
- (ii) qualquer uma das condições de “sucesso” anteriores não é satisfeita.

5. Conclusões

A heurística apresentou um bom desempenho para problemas não degenerados. No entanto, ela falhou em todos os problemas que apresentaram degeneração primal, e em vários casos de degeneração dual. Finalizando, podemos colocar pelo menos duas utilidades para esta heurística:

- (i) ela pode ser utilizada para antecipar a parada do método primal-dual de pontos interiores: o número de iterações e, conseqüentemente, o tempo de cpu, podem ser reduzidos se esta heurística for utilizada. Com alguns critérios sobre a factibilidade da solução corrente apresentada pelo método primal-dual e sobre a otimalidade desta solução, podemos fazer uma tentativa de encontrar “antecipadamente” uma solução básica ótima;
- (ii) ela pode ser usada como condicionador no método dos gradientes conjugados, que foi o método utilizado para resolver o sistema linear das direções no primal-dual (de fato, apresentamos dois outros condicionadores para este método). Como condicionador, a heurística poderia ser utilizada nas últimas iterações do método primal-dual, uma vez que sua construção é computacionalmente mais cara e necessita de mais informações sobre a solução primal-dual corrente.

Referências

- [1] M.S. Bazaraa; J.J. Jarvis e H.D. Sherali, “Linear Programming and Network Flows”, John Wiley & Sons, New York, 1990.
- [2] J.L. Kennington e R.V. Helgason, “Algorithms for Network Programming”, John Wiley & Sons, New York, 1980.
- [3] M. Kojima, N. Megiddo e S. Mizuno, A primal-dual infeasible-interior-point algorithm for linear programming, *Math. Prog.*, **61** (1993), 263-280.
- [4] M. Kojima, S. Mizuno and A. Yoshise, A primal-dual interior point method for linear programming, in “Progress in Mathematical Programming” (N. Megiddo, ed.), pp.29-48, Springer-Verlag, New York, 1989.
- [5] K.A. McShane, C.L. Monma and D.F. Shanno, An implementation of a primal-dual interior point method for linear programming, *ORSA Journal on Computing*, **1** (1989), 70-83.