Comparação entre Variantes do Método de Pontos Interiores para Multifluxo

V. PODESTÁ-GOMES, C. PERIN, Departamento de Matemática Aplicada, Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas, UNICAMP, C.P. 6065, 13083-970, Campinas, SP, Brasil.

Resumo

Apresentamos uma comparação entre duas versões do método primal-dual de pontos interiores especializado para multifluxo: uma versão que chamamos de *Comum* e outra versão conhecida por *Preditor-Corretor*. Comparamos estas duas versões quanto ao tempo de cpu, número de iterações e número de iterações do gradiente conjugado precondicionado, na presença de diferentes pontos iniciais e de inicializações distintas no gradiente conjugado.

Abstract

In this paper we compare two specialized versions of the primal-dual interior point method for multiflow problems: one version called *Usual* and a second one known as *Predictor-Corrector*. We studied the cpu time, the number of iterations of the primal-dual method, as well as the total number of iterations of the preconditioned conjugate gradient in the presence of distinct starting points and different initializations in the preconditioned conjugate gradient.

1 Formulação do problema

O Problema de Multifluxo consiste em encontrar uma solução de mínimo custo total de transporte de p produtos através de uma rede com m nós e n arcos, onde os arcos possuem uma capacidade individual para cada produto e também uma capacidade mútua, compartilhada por todos os produtos. Além disso, existem as restrições usuais de conservação de fluxo para cada produto em cada nó da rede. A formulação mais conhecida para este problema (formulação nó-arco) é:

$$
\begin{aligned}
min \quad & \sum_{k=1}^{p} (c^k)' x^k \\
s.a. \quad & \begin{cases} Ax^k = b^k, \ k = 1, \cdots, p \\ \sum_{k=1}^{p} x^k \leq d \\ 0 \leq x^k \leq u^k, \ k = 1, \cdots, p \end{cases}
\end{aligned}
$$

Os vetores $x^k \in I\!R^n$, $c^k \in I\!R^n$, $u^k \in I\!R^n$ são, respectivamente, os vetores das variáveis de fluxo, de custos unitários de transporte e de capacidade individual

para cada produto k ; $b^k \in I\!R^m$ é o vetor de ofertas/demandas do produto k, em cada nó da rede. $A \in I\!R^{m \times n}$ é a matriz de incidência: cada uma de suas colunas corresponde a um arco da rede. É bastante conhecido que, se a rede é conectada, então é necessário remover uma linha qualquer para que a matriz A passe a ter posto completo. O primeiro grupo de restrições formam as restrições de oferta/demanda para cada produto. O vetor $d \in I\!R^n$ é o vetor de capacidade mútua dos arcos, e o segundo grupo de restrições são as restrições de acoplamento. As últimas restrições são as de canalização sobre as variáveis de fluxo.

Este é um problema de programação linear que, em sua forma padrão possui $\widehat{m} = p(m-1) + n$ restrições e $\widehat{n} = (p+1)n$ variáveis e é considerado um problema de grande porte. No entanto, sua matriz de coeficientes possui uma estrutura bem particular: ela é bloco angular com uma faixa horizontal; cada bloco é formado pela matriz de incidência da rede e a faixa é composta por matrizes identidade. Vários métodos já foram propostos para esta estrutura. Mais recentemente, os métodos de pontos interiores têm se mostrado bastante competitivos e, dentre eles, a versão primal-dual tem mostrado um bom desempenho ([1], [2],[4]), bem com sua versão preditor-corretor ([3], [5]). Neste trabalho, comparamos as duas versões do método primal-dual de pontos interiores especializado para multifluxo: a versão que chamamos de *Comum* e a versão *Preditor-Corretor*.

2 O Método primal-dual de pontos interiores especializado para multifluxo

Considere o par primal/dual de problemas lineares, resultante do problema original:

$$(P) \quad \begin{array}{ll} \min & \hat{c}'\hat{x} \\ s.a. & \left\{ \begin{array}{l} \hat{A}\hat{x} = \hat{b} \\ \hat{x} + \hat{s} = \hat{u} \\ \hat{x},\ \hat{s} \geq 0 \end{array} \right. \end{array} \qquad (D) \quad \begin{array}{ll} \max & \hat{b}'\hat{y} - \hat{u}'\hat{w} \\ s.a. & \left\{ \begin{array}{l} \hat{A}'\hat{y} - \hat{w} + \hat{z} = \hat{c} \\ \hat{w},\ \hat{z} \geq 0 \end{array} \right. \end{array}$$

onde

$$\widehat{A} = \begin{pmatrix} A & & & & \\ & A & & & \\ & & \ddots & & \\ & & & A & \\ I & I & \cdots & I & I \end{pmatrix}, \quad \widehat{c} = \begin{pmatrix} c^1 \\ c^2 \\ \vdots \\ c^p \\ 0 \end{pmatrix}, \quad \widehat{u} = \begin{pmatrix} u^1 \\ u^2 \\ \vdots \\ u^p \\ \infty \end{pmatrix}, \quad \widehat{b} = \begin{pmatrix} b^1 \\ b^2 \\ \vdots \\ b^p \\ d \end{pmatrix}$$

Nesta notação, $\widehat{A} \in R^{\widehat{m} \times \widehat{n}}$; $\widehat{c}$, $\widehat{x}$, $\widehat{s}$, $\widehat{u}$, $\widehat{z}$ e $\widehat{w} \in I\!R^{\widehat{n}}$, $\widehat{b}$ e $\widehat{y} \in I\!R^{\widehat{m}}$; $\widehat{y}$ e $\widehat{w}$ são as variáveis duais correspondentes às restrições de conservação de fluxo e de canalização, respectivamente, e $\widehat{z}$ á folga dual. Todos estes vetores são particionados por produtos (ou blocos): uma componente para cada produto k, $k = 1, \cdots, p$ e mais uma componente correspondente às restrições de acoplamento.

Resumidamente, o método primal-dual de pontos interiores especializado para multifluxo consiste em, dado um ponto inicial interior $(\widehat{x}, \widehat{s})$ para o problema primal (P), não necessariamente factível, um ponto inicial interior $(\widehat{y}, \widehat{w}, \widehat{z})$ para o problema dual (D), não necessariamente factível 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 é necessário resolver um sistema linear com $\widehat{m}$ 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 vai sendo alcançada, bem como a otimalidade.

A diferença entre as duas versões apresentadas está na obtenção da direção de deslocamento. Na versão *Comum* resolvemos um sistema linear; na versão *Preditor-Corretor* são resolvidos dois sistemas lineares a cada iteração do método primal-dual.

Nos dois casos, estamos utilizando o método do *gradiente conjugado precondicionado* para resolver os sistemas lineares, onde a estrutura da matriz $\widehat{A}$ é aproveitada. Nesta resolução, usamos uma combinação de dois precondicionadores: o *Diagonal*, nas primeiras iterações do método primal-dual e o *Floresta Geradora Máxima*, nas iterações posteriores (ver [6]).

A direção de deslocamento é particionada de acordo com as variáveis primais e duais: $\Delta\widehat{x},\ \Delta\widehat{s},\ \Delta\widehat{y},\ \Delta\widehat{w},\ \Delta\widehat{z}$. Desta maneira, o método primal-dual pode ser resumido no seguinte algoritmo:

Algoritmo primal-dual

- Fornecer um ponto inicial interior $(\widehat{x},\ \widehat{s})$ interior para o primal e $(\ \widehat{y},\ \widehat{w},\ \widehat{z})$ para o dual.
- Repetir :

  obter o parâmetro de centragem μ ;
  obter a direção de deslocamento $\Delta\widehat{y}$, resolvendo um sistema linear;
  calcular $\Delta\widehat{x},\ \Delta\widehat{s},\ \Delta\widehat{w},\ \Delta\widehat{z}$ a partir de $\Delta\widehat{y}$;
  teste da razão: calcular λ_P e λ_D;
  $(\widehat{x},\ \widehat{s}) \leftarrow (\widehat{x}, \widehat{s}) + \lambda_P\ (\Delta\widehat{x},\ \Delta\widehat{s})$;
  $(\widehat{y},\ \widehat{w},\ \widehat{z}) \leftarrow (\widehat{y},\ \widehat{w},\ \widehat{z}) + \lambda_D\ (\Delta\widehat{y},\ \Delta\widehat{w},\ \Delta\widehat{z})$;

até que um critério de parada seja satisfeito.

O passo da obtenção da direção de deslocamento é o que diferencia as versões primal-dual *Comum* e *Preditor-Corretor*, além de ser o passo computacionalmente mais caro do algoritmo. Este passo será detalhado a seguir.

2.1 Primal-Dual Comum

Na versão primal-dual *Comum*, o sistema linear que encontra a direção de deslocamento é:

$$\underbrace{\begin{bmatrix} \widehat{A} & 0 & 0 & 0 & 0 \\ I & I & 0 & 0 & 0 \\ 0 & 0 & \widehat{A}' & -I & I \\ \widehat{Z} & 0 & 0 & 0 & \widehat{X} \\ 0 & \widehat{W} & 0 & \widehat{S} & 0 \end{bmatrix}}_{\widehat{Q}} \begin{bmatrix} \Delta\widehat{x} \\ \Delta\widehat{s} \\ \Delta\widehat{y} \\ \Delta\widehat{w} \\ \Delta\widehat{z} \end{bmatrix} = \begin{bmatrix} \widehat{b} - \hat{A}\widehat{x} \\ \widehat{u} - \widehat{x} - \widehat{s} \\ \widehat{c} - \widehat{A}'\widehat{y} + \widehat{w} - \widehat{z} \\ \mu\mathbf{1} - \widehat{X}\widehat{Z}\mathbf{1} \\ \mu\mathbf{1} - \widehat{S}\widehat{W}\mathbf{1} \end{bmatrix} \tag{1}$$

onde $\widehat{X} = diagonal(X^1, \cdots, X^p, X^{p+1})$, com $X^k = diagonal(x_j^k),\ \forall k,\ j = 1, \cdots, n$; $\widehat{S},\ \widehat{W},\ \widehat{Z}$ têm definições similares à matriz $\widehat{X}$, isto é, cada uma delas é uma matriz diagonal por blocos com $\widehat{n}$ elementos positivos (matrizes formadas pelas respectivas soluções correntes) e $\mathbf{1}$ é o vetor coluna com $\widehat{n}$ componentes iguais a 1.

Com algumas manipulações nas equações deste sistema, ele fica reduzido a um sistema linear de dimensão bem menor, $\widehat{m}$, que é

$$(\widehat{A}\widehat{\Theta}\widehat{A}')\ \Delta\widehat{y} = \widehat{A}\widehat{\Theta}\ \{[\ \mu((\widehat{S})^{-1} - (\widehat{X})^{-1}) - \widehat{W} + \widehat{Z}\ ]\mathbf{1} + (\ \widehat{c} - \widehat{A}'\widehat{y} + \widehat{w} - \widehat{z}\ ) - (\widehat{S})^{-1}\widehat{W}(\ \widehat{u} - \widehat{x} - \widehat{s}\ )\} + (\ \widehat{b} - \widehat{A}\widehat{x}\ )$$

ou, de uma maneira mais conveniente,

$$B\ \Delta\widehat{y} = r \tag{2}$$

onde a matriz $\widehat{\Theta} = [(\widehat{X})^{-1}\widehat{Z} + (\widehat{S})^{-1}\widehat{W}]^{-1}$ é uma matriz diagonal, com elementos positivos, denominada *matriz de scaling* do método primal-dual.

A matriz B do sistema linear (2) é simétrica e definida positiva, sendo então adequado o uso do gradiente conjugado precondicionado na resolução deste sistema.

As outras componentes da direção ($\Delta\widehat{x},\ \Delta\widehat{s},\ \Delta\widehat{w},\ \Delta\widehat{z}$) podem ser obtidas a partir de $\Delta\widehat{y}$ (ver [6]).

2.2 Preditor-Corretor

Esta variante do método primal-dual foi apresentada por Mehrotra [5] e Lustig, Marsten, Shanno [3]. A diferença fundamental entre o método primal-dual *Comum* e esta variante está na maneira de calcular a direção de deslocamento: no *Preditor-Corretor*, a direção é obtida resolvendo dois sistemas lineares a cada iteração, onde a matriz dos coeficientes é a mesma, mas os lados direitos são distintos.

Primeiro resolvemos

$$\begin{bmatrix} \widehat{A} & 0 & 0 & 0 & 0 \\ I & I & 0 & 0 & 0 \\ 0 & 0 & \widehat{A}' & -I & I \\ \widehat{Z} & 0 & 0 & 0 & \widehat{X} \\ 0 & \widehat{W} & 0 & \widehat{S} & 0 \end{bmatrix} \begin{bmatrix} \bar{\Delta}\widehat{x} \\ \bar{\Delta}\widehat{s} \\ \bar{\Delta}\widehat{y} \\ \bar{\Delta}\widehat{w} \\ \bar{\Delta}\widehat{z} \end{bmatrix} = \begin{bmatrix} \widehat{b} - \widehat{A}\widehat{x} \\ \widehat{u} - \widehat{x} - \widehat{s} \\ \widehat{c} - \widehat{A}'\widehat{y} + \widehat{w} - \widehat{z} \\ -\widehat{X}\widehat{Z}\mathbf{1} \\ -\widehat{S}\widehat{W}\mathbf{1} \end{bmatrix}$$

que é o mesmo sistema linear (1), mas com novas variáveis $\bar{\Delta} = (\ \bar{\Delta}\widehat{x},\ \bar{\Delta}\widehat{s},\ \bar{\Delta}\widehat{y},\ \bar{\Delta}\widehat{w},\ \bar{\Delta}\widehat{z}\ )$, chamadas de *direções afins*.
O segundo sistema linear a ser resolvido é

$$\begin{bmatrix} \widehat{A} & 0 & 0 & 0 & 0 \\ I & I & 0 & 0 & 0 \\ 0 & 0 & \widehat{A}' & -I & I \\ \widehat{Z} & 0 & 0 & 0 & \widehat{X} \\ 0 & \widehat{W} & 0 & \widehat{S} & 0 \end{bmatrix} \begin{bmatrix} \Delta\widehat{x} \\ \Delta\widehat{s} \\ \Delta\widehat{y} \\ \Delta\widehat{w} \\ \Delta\widehat{z} \end{bmatrix} = \begin{bmatrix} \widehat{b} - \widehat{A}\widehat{x} \\ \widehat{u} - \widehat{x} - \widehat{s} \\ \widehat{c} - \widehat{A}'\widehat{y} + \widehat{w} - \widehat{z} \\ \mu\mathbf{1} - \widehat{X}\widehat{Z}\mathbf{1} - \bar{\Delta}\widehat{x}\bar{\Delta}\widehat{z}\mathbf{1} \\ \mu\mathbf{1} - \widehat{S}\widehat{W}\mathbf{1} - \bar{\Delta}\widehat{s}\bar{\Delta}\widehat{w}\mathbf{1} \end{bmatrix}$$

Note que temos a mesma matriz $\widehat{Q}$ nos dois sistemas lineares. A diferença entre eles está apenas no lado direito: nas duas últimas componentes do segundo sistema temos a presença dos termos não lineares $\bar{\Delta}\widehat{x}\bar{\Delta}\widehat{z}$ e $\bar{\Delta}\widehat{s}\bar{\Delta}\widehat{w}$.

Esta variante reduz o número de iterações do algoritmo anterior, mas exige que resolvamos agora dois sistemas lineares em cada iteração do método primal-dual. No entanto, como a matriz destes sistemas é a mesma matriz $\widehat{Q}$, os cálculos utilizados para construí-la (ou parte dela) são efetuados uma única vez.

Transformações similares às que foram feitas para transformar o sistema linear (1) no sistema reduzido (2) também podem ser efetuadas nesta variante, de modo que os dois sistemas lineares resolvidos em cada iteração do primal-dual possuem a mesma dimensão do sistema (2), isto é, $\widehat{m}$ equações e incógnitas, sendo também um sistema simétrico e definido positivo, onde podemos utilizar o gradiente conjugado precondicionado.

3 Pontos iniciais

3.1 Ponto inicial I

Este ponto inicial está descrito em [6]. Começamos com $(\widehat{x}^0,\ \widehat{s}^0,\ \widehat{y}^0,\ \widehat{w}^0,\ \widehat{z}^0)$ que satizfaz

$\widehat{x}^0,\ \widehat{s}^0 > 0$ (interior),
$\widehat{A}'\widehat{y}^0 - \widehat{w}^0 + \widehat{z}^0 = \widehat{c}$ (dual factível),
$\widehat{x}^0 + \widehat{s}^0 = \widehat{u}$ (canalizações respeitadas).

Solução dual: consideramos $\widehat{y}^0$ um vetor onde $(y^k)^0$ seja qualquer, $k = 1, \cdots, p$, mas $(y^{p+1})^0 < 0$. Para cada bloco k, e para cada arco $a_t = (i, j),\ t = 1, \cdots, n$ da

rede, calculamos:

$$\tilde{c}_{ij}^k = c_{ij}^k - (y_i^k)^0 + (y_j^k)^0 - (y_t^{p+1})^0 \ .$$

Seja $M = \max_{k,(ij)}\{\ |\tilde{c}_{ij}^k|\} + \max_t\{(|y_t^{p+1})^0|\},\ \forall k,\ \forall (i,j)$.

Para garantirmos $\widehat{w}^0 > 0$ e $\widehat{z}^0 > 0$, tomamos, para $\forall k$ e $\forall (i,j)$:
se $\tilde{c}_{ij}^k > 0$, então $(w_t^k)^0 = M$ e $(z_t^k)^0 = \tilde{c}_{ij}^k + M$,
se $\tilde{c}_{ij}^k \leq 0$, então $(z_t^k)^0 = M$ e $(w_t^k)^0 = -\tilde{c}_{ij}^k + M$.

Desta maneira, todas as restrições duais ficam satisfeitas. Em nossa implementação, iniciamos com $(y^k)^0 = 0,\ k = 1, \cdots, p$ e $(y^{p+1})^0 = -(1,\ 1, \cdots,\ 1)'$, de modo que $\tilde{c}_{ij}^k = c_{ij}^k + 1,\ \ \forall k,\ \ \forall (i,j)$.

Solução primal: para cada arco $a_t = (i,j)$ calculamos $soma = \sum_{k=1}^p u_t^k$.
Se $soma < 2d_t$, então $(x_t^k)^0 = u_t^k/2$; senão, $(x_t^k)^0 = (u_t^k\ d_t)/2soma,\ k = 1, \cdots, p$.
$(x_t^{p+1})^0 = d_t - \sum_{k=1}^p (x_t^k)^0,\ \ t = 1, \cdots, n$ e $\widehat{s}^0 = \widehat{u} - \widehat{x}^0$.
Assim, as únicas equações que poderão não estar satisfeitas são as de conservação de fluxo nos nós dos produtos ($Ax^k = b^k$).

3.2 Ponto inicial II

Este ponto inicial é uma extensão do ponto sugerido por Mehrotra [5] para problemas gerais de Programação Linear. Não sendo específico para redes, ele deixa de tirar proveito da estrutura de nosso problema. De qualquer maneira, este ponto requer um trabalho maior para ser construído.

Num primeiro passo, encontramos um ponto $(\widehat{x},\ \widehat{s},\ \widehat{y},\ \widehat{w},\ \widehat{z})$ que satisfaz todas as restrições do par primal/dual, mas não necessariamente as restrições de não negatividade:

$$\widehat{y} = (\widehat{A}\widehat{A'})^{-1}\widehat{A}\widehat{c}$$

$$\widehat{z} = 0.5\ (\widehat{c} - \widehat{A'}\widehat{y})$$

$$\widehat{w} = -\widehat{z}$$

$$\widehat{x} = 0.5\widehat{u} - \widehat{A'}(\widehat{A}\widehat{A'})^{-1}(0.5\widehat{A}\widehat{u} - \widehat{b})$$

$$\widehat{s} = \widehat{u} - \widehat{x}$$

Note que, tanto a construção de $\widehat{x}$ quanto a construção de $\widehat{y}$ requerem a resolução de um sistema linear quadrado de dimensão $\widehat{m}$.

Num segundo passo, é feita uma translação adequada do ponto encontrado, de modo a satisfazer as restrições de não negatividade. Para isso, são usados os parâmetros α_1 e α_2. Calculamos:

$$\lambda_P = \max\{-\alpha_1\ \min_{i,k}\{x_i^k\},\ -\alpha_1\ \min_{i,k}\{s_i^k\},\ 0.01\}$$

$$\lambda_D = \max\{-\alpha_1 \min_{i,k}\{z_i^k\},\ -\alpha_1 \min_{i,k}\{w_i^k\},\ 0.01\}$$

$$\delta_P = \lambda_P + \alpha_2 \frac{(\widehat{x} + \lambda_P \mathbf{1})'\ (\widehat{z} + \lambda_D \mathbf{1}) + (\widehat{s} + \lambda_P \mathbf{1})'\ (\widehat{w} + \lambda_D \mathbf{1})}{(\widehat{z} + \lambda_D \mathbf{1})'\ \mathbf{1} + (\widehat{w} + \lambda_D \mathbf{1})'\ \mathbf{1}}$$

$$\delta_D = \lambda_D + \alpha_2 \frac{(\widehat{x} + \lambda_P \mathbf{1})'\ (\widehat{z} + \lambda_D \mathbf{1}) + (\widehat{s} + \lambda_P \mathbf{1})'\ (\widehat{w} + \lambda_D \mathbf{1})}{(\widehat{x} + \lambda_P \mathbf{1})'\ \mathbf{1} + (\widehat{s} + \lambda_P \mathbf{1})'\ \mathbf{1}}$$

Finalmente, fazemos

$(\widehat{y})^0 = \widehat{y}$
$(\widehat{x})^0 = \widehat{x} + \delta_P \mathbf{1}$
$(\widehat{s})^0 = \widehat{s} + \delta_P \mathbf{1}$
$(\widehat{z})^0 = \widehat{z} + \delta_D \mathbf{1}$
$(\widehat{w})^0 = \widehat{w} + \delta_D \mathbf{1}$.

É possível mostrar que $(\widehat{x}^0,\ \widehat{s}^0,\ \widehat{y}^0,\ \widehat{w}^0,\ \widehat{z}^0)$ satisfaz as restrições de não negatividade. No entanto, as outras restrições primais/duais não ficam necessariamente satisfeitas. Como sugerido em [5], tomamos $\alpha_1 = 1.5$ e $\alpha_2 = 0.5$.

4 Direções iniciais para o gradiente conjugado precondicionado

A solução inicial usual para o gradiente conjugado precondicionado é a solução nula. De fato, a inicialização com esta solução forneceu bons resultados, além de ser mais simples. Tentamos algumas outras variações: por exemplo, iniciar o gradiente conjugado na iteração corrente do método primal-dual com a solução (direção) obtida pelo gradiente conjugado na iteração anterior do método primal-dual e, na versão *Preditor-Corretor*, utilizar a direção obtida ao final da execução do Preditor para se iniciar o Corretor. Estas alternativas mostraram comportamentos um pouco diferentes nas duas versões do primal-dual, dependendo ainda do ponto inicial utilizado.

5 Experimentos computacionais

Construímos um gerador de problemas de multifluxo e, através dele, vários testes computacionais foram realizados. Os problemas são gerados a partir da especificação de um *tamanho* e de uma *semente*. A rede associada ao problema é conectada e os vetores u^k e c^k, $\forall k$, são gerados com distribuição uniforme em intervalos cujos limites são fornecidos. Nestes experimentos, consideramos $c_j^k \in [0,\ 5]$, $u_j^k \in [1,\ 5]$, $\forall k$, $\forall j$. Além disso, temos também a opção de deixar as restrições de acoplamento mais "apertadas" ou mais "folgadas" (vetor d).

A implementação está feita na linguagem C^{++}, com dupla precisão. Todos os experimentos foram realizados numa Sun Ultra SPARC 10, com 300 MHz e 128 Mbytes de memória RAM.

Utilizando os pontos iniciais **I** e **II** e variando a inicialização do gradiente conjugado precondicionado, medimos o tempo de cpu, o número total de iterações do primal-dual e o número total de iterações do gradiente conjugado.

problema	m	n	p	$\widehat{m}$	$\widehat{n}$
1	50	101	10	591	1111
2	100	201	10	1191	2211
3	200	306	10	2296	3366
4	200	401	10	2391	4411
5	200	500	10	2490	5500
6	300	507	10	3497	5577
7	300	604	10	3594	6644
8	400	620	10	4610	6820
9	400	805	10	4795	8855
10	500	902	10	5901	9922
11	500	1009	10	5999	11099
12	600	1212	10	7202	13332
13	800	1515	10	9505	16665
14	1000	2017	10	12007	22187
15	1000	3006	10	12996	33066
16	2000	3100	10	23090	34100
17	100	201	20	2181	4221
18	200	408	20	4388	8568
19	300	516	20	6496	10836
20	400	814	20	8794	17094
21	1000	2028	20	22008	42588
22	2000	4044	20	44024	84924

Tabela 1: Problemas gerados

A Tabela 1 apresenta os **problemas gerados** para este trabalho. Nos problemas de 1 a 16, aumentamos o tamanho da rede (m e n) mantendo o número de produtos (p) em 10; nos problemas de 17 a 22, mantivemos o número de produtos em 20; $\widehat{m} = p(m-1)+n$ é a dimensão do sistema linear que é resolvido em cada iteração do primal-dual; $\widehat{n} = (p+1)n$ é o número de variáveis do problema de multifluxo. Os resultados a seguir se referem a estes 22 problemas.

Em todos estes resultados, utilizamos a **melhor combinação** que encontramos entre a **inicialização do gradiente conjugado precondicionado** e a utilização das duas opções de **ponto inicial**, ou seja:

• com o ponto inicial **I**, o melhor é iniciar o gradiente conjugado precondicionado com a direção nula, nas duas versões do primal-dual;

• com o ponto inicial **II**, e na versão *Comum*, como é praticamente indiferente iniciar o gradiente conjugado precondicionado com a direção nula ou com a direção obtida pelo gradiente conjugado na iteração anterior do primal-dual, optamos por utilizar a direção nula, que é mais simples. Na versão *Preditor-Corretor*, o melhor é iniciar o Corretor com a direção obtida pelo Preditor. Neste caso, ainda é um pouco melhor iniciar o Preditor com a direção nula.

5.1 Comparação entre o número total de iterações do primal-dual

Neste item, podemos concluir claramente que o número de iteraçõs na versão *Preditor-Corretor* é sempre menor do que na versão *Comum*, independentemente do ponto inicial utilizado (colunas "p.i. I" e "p.i. II"), como pode ser observado na Tabela 2 e nos gráficos dados pelas Figuras 1 e 2. Na Figura 1, utilizamos o ponto inicial **I**, e, na Figura 2, utilizamos o ponto inicial **II**. Neste último, este fato ocorre em todos os problemas.

problema	PD Comum		Preditor-Corretor	
	p.i. I	p.i. II	p.i. I	p.i. II
1	20	23	16	17
2	23	23	16	17
3	24	24	15	19
4	30	27	22	22
5	25	27	18	19
6	26	29	18	20
7	27	32	19	20
8	27	37	23	22
9	27	31	20	22
10	31	32	21	22
11	31	31	23	26
12	31	33	24	20
13	30	32	22	23
14	33	34	23	24
15	32	35	23	22
16	35	39	25	24
17	23	28	15	18
18	26	31	19	22
19	27	36	19	19
20	31	36	32	25
21	36	38	27	29
22	40	41	30	32

Tabela 2: Número de iterações Primal-Dual

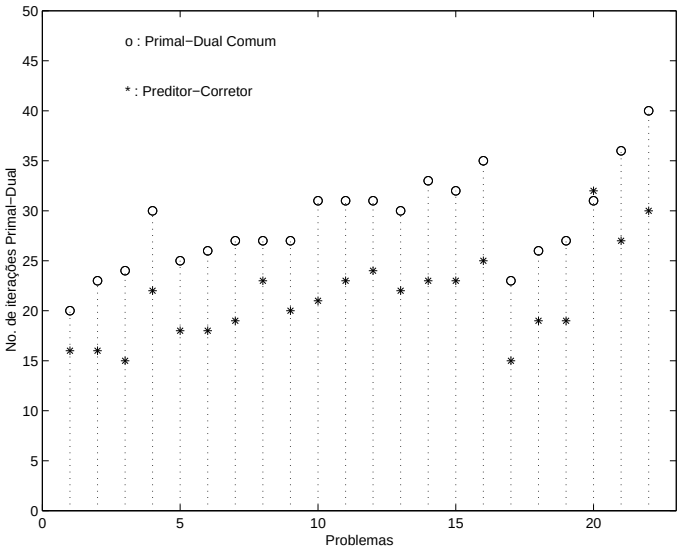

Figura 1: No. de iterações Primal-Dual (ponto inicial **I**)

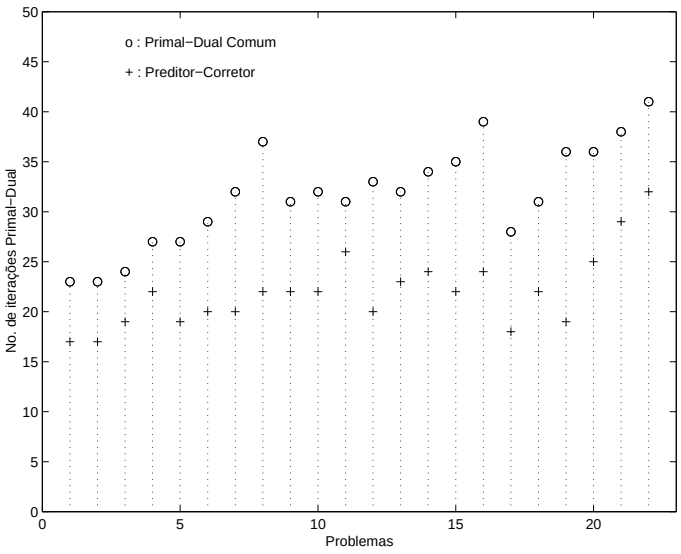

Figura 2: No. de iterações Primal-Dual (ponto inicial **II**)

5.2 Comparação entre tempos de CPU

Na Tabela 3 temos o tempo de CPU, em segundos, quando rodamos as versões primal-dual *Comum* e *Preditor-Corretor*, na presença dos dois pontos iniciais descritos (colunas "p.i. I" e "p.i. II"). Comparando as duas versões do primal-dual, vemos que na versão *Comum*, 16 problemas apresentaram tempo de CPU menor quando utilizamos o ponto inicial **I** e, na versão *Preditor-Corretor*, 14 problemas. Comparando agora os dois pontos iniciais: utilizando o ponto inicial **I**, a versão *Comum* apresentou 15 problemas com tempo de CPU menor. Utilizando o ponto inicial **II**, a versão *Comum* apresentou 12 problemas com tempo de CPU menor.

problema	PD Comum		Preditor-Corretor	
	p.i. I	p.i. II	p.i. I	p.i. II
1	8	10	14	14
2	31	33	38	34
3	82	86	119	127
4	137	136	164	168
5	94	107	108	117
6	208	241	218	244
7	296	369	286	253
8	408	732	594	703
9	438	549	405	481
10	1053	902	689	820
11	1167	1138	1118	1308
12	1705	1592	1547	1040
13	1939	2255	2187	2071
14	5286	5045	3284	5154
15	4150	4787	4665	2852
16	19485	19974	23588	13870
17	16	27	27	30
18	76	97	110	131
19	142	219	166	192
20	492	497	1009	528
21	3040	3258	2101	3022
22	17473	10347	21876	19297

Tabela 3: Tempos de CPU

Nas Figuras 3 e 4 temos a **razão entre os tempos de CPU** (CPU do *Preditor-Corretor*/CPU do *Comum*) na presença do ponto inicial **I** e do ponto inicial **II**, respectivamente. Aqui, podemos observar mais claramente que a versão *Comum* apresenta um comportamento melhor, com os dois pontos iniciais.

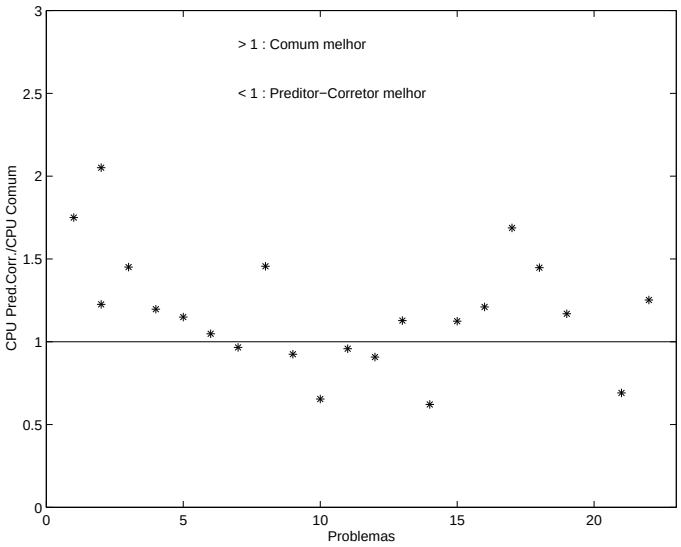

Figura 3: Razão entre tempos de CPU (ponto inicial **I**)

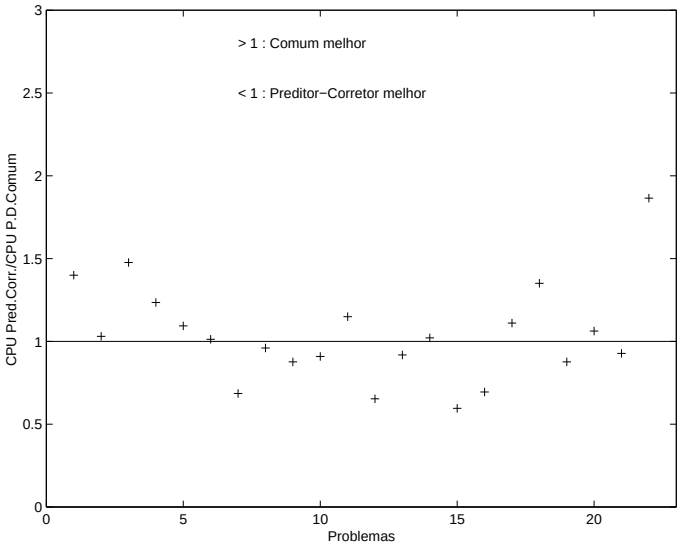

Figura 4: Razão entre tempos de CPU (ponto inicial **II**)

5.3 Total de iterações no gradiente conjugado

Na Tabela 4 temos o número total de iterações no gradiente conjugado precondicionado, com as versões *Comum* e *Preditor-Corretor*, utilizando os pontos iniciais **I** e **II**. Na versão *Preditor-Corretor*, o total de iterações do gradiente conjugado é a soma do número de iterações do Preditor com o número de iterações do Corretor. Comparando as duas versões do primal-dual, na versão *Comum* tivemos 12 problemas com um total de iterações no gradiente conjugado menor quando utilizamos o ponto inicial **I** e, na versão *Preditor-Corretor*, tivemos apenas 8 problemas. Comparando os dois pontos iniciais: com o ponto inicial **I**, a versão *Comum* apresentou 16 problemas com um total de iterações no gradiente conjugado menor; com o ponto inicial **II**, as duas versões se comportaram igualmente (11 problemas).

problema	PD Comum		Preditor-Corretor	
	p.i. I	p.i. II	p.i. I	p.i. II
1	1060	920	1952	1564
2	1886	1725	2592	2023
3	2712	2568	4680	4313
4	4230	3888	5412	5038
5	2575	2565	3150	3078
6	4550	4930	5112	5240
7	6102	7232	6156	4920
8	6858	12210	10258	11902
9	6696	7719	6400	6908
10	13330	11136	8862	10010
11	14570	13578	13777	16016
12	17329	15642	15768	10140
13	14520	16448	16654	15295
14	30987	28968	19780	29424
15	20160	22855	23046	13552
16	60830	62556	74400	43344
17	1334	1680	2175	1692
18	3094	3224	4142	3960
19	3942	5076	4275	3800
20	9548	8748	16640	7675
21	22500	31084	16011	22475
22	6440	37105	80820	71776

Tabela 4: Total de iterações no Gradiente Conjugado

As Figuras 5 e 6 mostram a **razão entre o total de iterações do gradiente conjugado** (total de iterações do *Preditor-Corretor*/ total de iterações do *Comum*) na presença dos pontos iniciais **I** e **II**, repectivamente. A versão *Comum* apresenta um comportamento melhor com o ponto inicial **I**. Com o ponto inicial **II**, as duas versões do método primal-dual se comportaram igualmente.

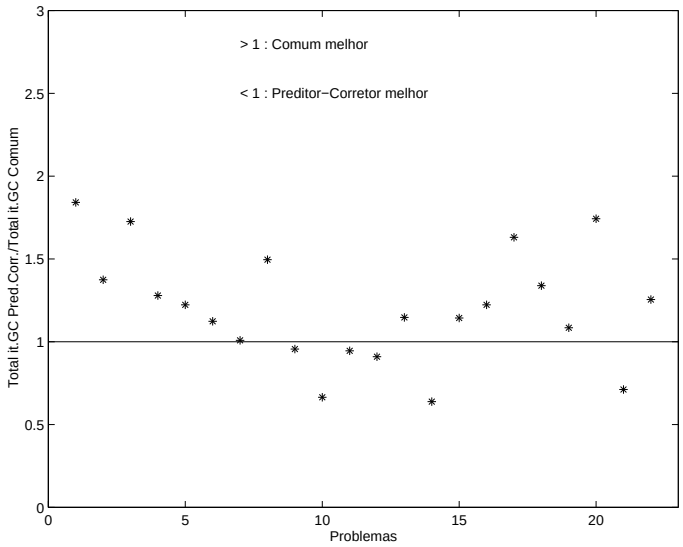

Figura 5: Razão entre total de iterações no GC (ponto inicial **I**)

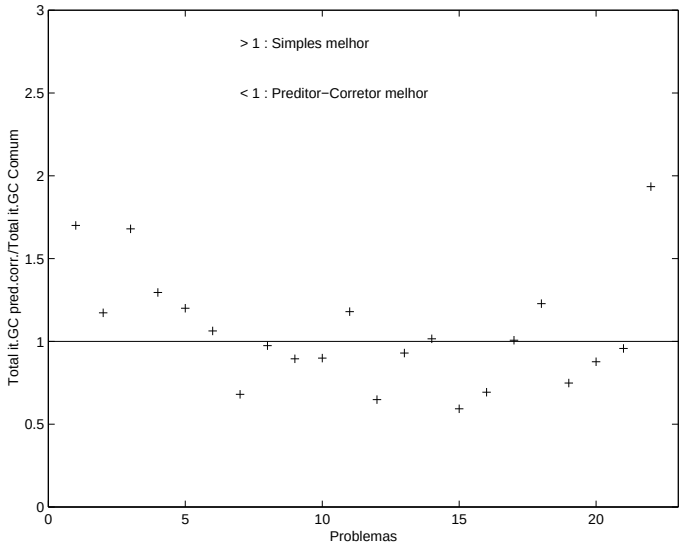

Figura 6: Razão entre total de iterações no GC (ponto inicial **II**)

6 Comentários finais

Com estes experimentos, podemos concluir que, utilizando o ponto inicial **I**, o melhor é iniciar o gradiente conjugado precondicionado com a direção nula, nas duas versões do método primal-dual. Por outro lado, utilizando o ponto inicial **II**, na versão *Comum* ainda é melhor iniciar o gradiente conjugado precondicionado com a direção nula, mas na versão *Preditor-Corretor* é um pouco melhor iniciar o Corretor com a direção obtida pelo Preditor (o Preditor é inicializado com a direção nula).

Quanto ao número de iterações primal-dual, a versão *Preditor-Corretor* apresentou um menor número de iterações do que a versão *Comum*, como já era esperado. Além disso, tanto na versão *Comum* quanto na *Preditor-Corretor*, o ponto inicial **I** apresentou um melhor comportamento.

Quanto ao tempo de CPU, a combinação que apresentou tempos menores foi a versão *Comum* e o ponto inicial **I**. Assim, um número menor de iterações no primal-dual não implica necessariamente numa redução do tempo de CPU.

Quanto ao número total de iterações do gradiente conjugado precondicionado, os experimentos não são conclusivos: parece ser um pouco melhor utilizar o ponto inicial **II** na versão *Preditor-Corretor* (como sugerido por Mehrotra) e o ponto inicial **I** na versão *Comum*.

Referências

[1] M. Kojima, N. Megiddo e S. Mizuno, A Primal-Dual Infeasible-Interior- Point Algorithm for Linear Programming, *Mathematical Programming* **61** (1993), 263-280.

[2] M. Kojima, S. Mizuno e 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.

[3] I. J. Lustig, R. E. Marsten e D. F. Shanno, On Implementing Mehrotra's Predictor-Corrector Interior Point Method for Linear Programming, *SIAM Journal on Optimization* **2** (1992), 435-449.

[4] K.A. McShane, C.L. Monma e D.F. Shanno, An Implementation of a Primal-Dual Interior Point Method for Linear Programming, *ORSA Journal on Computing* **1** (1989), 70-83.

[5] S. Mehrotra, On the Implementation of a Primal-Dual Interior Point Method, *SIAM Journal on Optimization* **2** (1992), 575-601.

[6] V. Podestá-Gomes, "Método Primal-Dual de Pontos Interiores Aplicado ao Problema de Multifluxo", Tese de Doutorado, Unicamp, 1999.