MÉTODO DE PONTOS INTERIORES APLICADO AO PROBLEMA DE FLUXO MULTIPRODUTO

Juliana Verga, Elma Pereira Santos
Faculdade de Engenharia Elétrica e de Computação - Universidade Estadual de Campinas
Av. Albert Einstein, 400 - Caixa Postal 6101 - CEP:13083-852 - Campinas - SP - Brasil
juverga@dt.fee.unicamp.br, elmaps@gmail.com

Aurelio Ribeiro Leite de Oliveira
Instituto de Matemática, Estatística e Computação Cientifica - Depto de Matemática Aplicada
Universidade Estadual de Campinas
Cidade Universitária, CEP:13083-859 - Campinas - SP - Brasil
aurelio@ime.unicamp.br

Wesley Vagner Inês Shirabayashi
Depto de Matemática - Universidade Estadual de Maringá
Av. Colombo, 5790, cep 87020-900, Jd. Universitário - Maringá - PR - Brasil
wvishirabayashi@uem.br

RESUMO

Este trabalho apresenta uma proposta de solução para o problema de fluxo multiproduto através do método de pontos interiores primal-dual seguidor de caminho. O problema foi modelado através de um grafo, cujos nós representam pontos de oferta e demanda de produtos, os quais trafegam pelos arcos da rede. O método proposto visa encontrar a solução ótima do problema de forma eficiente através da solução dos sistemas lineares com método dos gradientes conjugados precondicionado.

PALAVRAS-CHAVE. Fluxo Multiproduto. Métodos de Pontos Interiores. Programação Linear. Teoria de Grafos.

ABSTRACT

This work presents an approach for solving the multicommodity flow problem through of primal dual path following interior-point method. The problem was modeled through a graph whose nodes represent points of supply and demand of commodities, which pass through arcs of the network. The proposed method aims to find the optimal solution of the problem of efficiently form through the solution linear systems with method of preconditioned conjugate gradient.

KEYWORDS. Multicommodity Flow. Interior-Points Methods. Linear Programming. Graph Theory.

1 Introdução

A teoria de grafos é comumente utilizada para resolver problemas que podem ser representados na forma de redes. Ela fornece uma modelagem intuitiva de um problema, facilitando a implementação de algoritmos que auxiliam na obtenção da sua solução (Ahuja et al, 1993). Atualmente, problemas bem conhecidos como caminho mínimo, fluxo de custo mínimo, alocação, entre outros são muito estudados utilizando esta teoria (Ford & Fulkerson, 1962).

Métodos de pontos interiores possuem uma abordagem na qual a procura por soluções é realizada no interior do ortante positivo até atingir o ótimo. Desde o surgimento dos métodos de pontos interiores para otimização linear, códigos computacionais baseados nessas idéias vêm se apresentando como alternativas eficientes para solução de problemas de grande porte (Adler et al, 1989, Kojima et al, 1993, Wright, 1996).

O problema de fluxo multiproduto surge quando vários produtos compartilham os arcos em uma rede e competem pela capacidade dos mesmos. O objetivo deste problema é determinar o fluxo dos produtos em cada arco, a um custo mínimo, de modo a atender as restrições de capacidade dos arcos e as restrições de conservação de fluxo (Chagas, 2005).

As restrições de capacidade dos arcos limitam o fluxo dos produtos, de modo que em nenhum arco trafegue uma quantidade de produtos superior à capacidade suportada por ele. Já as restrições de conservação de fluxo gerenciam o fluxo dos produtos pelos arcos da rede, que saem de um ponto de oferta e chegam em um ponto de demanda. Cada produto pode ser transportado de um ou vários nós origem para um ou vários nós destino da rede (grafo) (Chagas, 2005).

Geralmente é associado um custo referente ao trânsito em cada arco, por unidade de fluxo. Estes custos podem ser os mesmos para todos os produtos ou podem ser diferentes para cada produto.

Os problemas de fluxo multiproduto, de um modo geral, têm recebido muita atenção devido a aplicabilidade na resolução de problemas atuais presentes nas mais diversas áreas, como transportes, telecomunicações, dentre outras (Ahuja et al, 1993, Chagas, 2005). A dificuldade prática de se resolver esse tipo de problema aumenta rapidamente à medida que o problema cresce, referente ao número de variáveis e, principalmente, em relação ao número de produtos.

Neste trabalho, é proposto um algoritmo para abordar problemas de fluxo multiproduto baseado em um método de pontos interiores primal-dual seguidor de caminho que será abordado na Seção 4.

2 Trabalhos correlatos

Os trabalhos iniciais que tratam o problema de fluxo multiproduto foram propostos por Ford e Fulkerson (1962); e Hu (1962), no início dos anos 60.

Castro (2000) propos um método misto combinando o método dos gradientes conjugados precondicionado e a fatoração de Cholesky esparsa para resolver um sistema linear em cada iteração. O precondicionador usado foi desenvolvido explorando a estrutura do problema de fluxo multiproduto.

Resende e Veiga (1993) apresentam uma implementação do método dos gradientes conjugados precondicionado para o problema de fluxo de custo mínimo. Os autores propuseram um precondicionador que inicialmente é a diagonal da matriz de restrições do problema e depois de um determinado número de iterações passa a ser a árvore geradora que é computada através do algoritmo de Kruskal (Tarjan, 1983).

3 Formulação Matemática

Seja $G = (N, A)$ um grafo, onde N é o conjunto de nós e A é o conjunto de arcos. Cada arco é denotado por (i, j) onde $i, j \in N$. O problema de fluxo multiproduto é formulado como o seguinte problema de programação linear:

$$
\min \quad z = \sum_{k=1}^{K} \sum_{(i,j)\in A} \tilde{c}_{ij}^{k} x_{ij}^{k}
$$

$$
\text{s.a} \begin{cases} \displaystyle\sum_{j:(i,j)\in A} x_{ij}^{k} - \sum_{j:(j,i)\in A} x_{ji}^{k} = b_i^k, \forall i \in N\,, \forall k = 1,\dots,K \\ \displaystyle l_{ij} \le \sum_{k=1}^{K} x_{ij}^{k} \le u_{ij}, \ \ \forall (i,j) \in A \end{cases} \tag{1}
$$

onde:

- $x_{ij}^k \in R^n$ representa o fluxo do produto k que percorre o arco (i,j);
- $c_{ij}^k \in R^n$ representa o custo unitário do produto k associado a cada unidade de fluxo x_{ij} que percorre o arco (i,j);
- l_{ij} representa o limitante inferior da capacidade do arco (i,j) $(l_{ij} = 0)$;
- $u_{ij} \in R^n$ representa o limitante superior da capacidade do arco (i,j);
- K representa o número total de produtos;
- $b_I^k \in R^m$ representa a oferta ou demanda do produto k:
  - Se $b_I^k > 0$, i representa nó gerador do produto k;
  - Se $b_I^k < 0$, i representa nó consumidor do produto k;
  - Se $b_I^k = 0$, i representa nó de passagem do produto k;

Matricialmente temos:
Problema Primal:

$$
\min \begin{pmatrix} c_{ij}^1 & c_{ij}^2 & \dots & c_{ij}^K & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_K \\ s \end{pmatrix}
$$

$$
\begin{pmatrix} A_1 & 0 & 0 & 0 & 0 \\ 0 & A_2 & & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & A_K & 0 \\ \mathbf{I} & \mathbf{I} & \dots & \mathbf{I} & \mathbf{I} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_K \\ s \end{pmatrix} = \begin{pmatrix} b_I^1 \\ b_I^2 \\ \vdots \\ b_I^K \\ u \end{pmatrix} \tag{2}
$$

$(x, s) \ge 0$

- x_k: vetor de fluxos do produto k, $k = 1, 2, \dots, K$.
- $s \in R^n$: folga da segunda restrição;
- $A_k \in R^{m\times n}$ matriz de incidência nó-arco do produto k, onde m é o número de nós e n é o número de arcos do problema;
- $\mathbf{I} \in R^{n\times n}$.

Observação 1 *Considera-se que a matriz A_k tem posto linha completo. Caso isso não aconteça, retira-se qualquer linha dessa matriz de forma que ela fique com posto linha completo.*

Problema Dual:

$$\max \begin{pmatrix} b_I^1 & b_I^2 & \dots & b_I^K & u \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_K \\ w \end{pmatrix}$$

$$\begin{pmatrix} A_1^T & 0 & 0 & 0 & \mathbf{I} \\ 0 & A_2^T & & 0 & \mathbf{I} \\ 0 & 0 & \ddots & 0 & \vdots \\ 0 & 0 & 0 & A_K^T & \mathbf{I} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_K \\ w \end{pmatrix} \leq \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_K \\ 0 \end{pmatrix} \tag{3}$$

$(w) \leq 0$ $(-w \geq 0)$, y_k *livre*,

- $w \in R^n$
- $y_k \in R^m$, $k = 1, 2, \dots, K$.

3.1 Condições de Otimalidade

As condições de otimalidade para os problemas primal e dual são dadas por:
Primal:

$$A_1 x_1 - b_I^1 = 0$$
$$A_2 x_2 - b_I^2 = 0$$
$$\vdots$$
$$A_K x_K - b_I^K = 0$$
$$x_1 + x_2 + \dots + x_K + s - u = 0$$

Dual:

$$A_1^T y_1 - w + t_1 - c_1 = 0$$
$$A_2^T y_2 - w + t_2 - c_2 = 0$$
$$\vdots$$
$$A_K^T y_K - w + t_K - c_K = 0$$

Complementaridade:

$$X_k T_k e = 0, \;\; k = 1, 2, \dots, K$$
$$SWe = 0$$

Sendo que:

- x_k, t_k, s, $w \in R^n$ são não negativos;
- $e = (1, 1, \dots, 1)^t \in R^n$;
- $X_k = \text{diag}(x_k)$;
- $T_k = \text{diag}(t_k)$;
- $S = \text{diag}(s)$;
- $W = \text{diag}(w)$.

$$
\begin{pmatrix}
A_1x^1 - b_I^1 \\
A_2x^2 - b_I^2 \\
\vdots \\
A_Kx^K - b_I^K \\
\\
x^1 + x^2 + \ldots + x^K + s - u \\
\\
A_1^T y_1 - w + t_1 - c_1 = 0 \\
A_2^T y_2 - w + t_2 - c_2 = 0 \\
\vdots \\
A_K^T y_K - w + t_K - c_K = 0 \\
\\
X_1T_1e \\
X_2T_2e \\
\vdots \\
X_KT_Ke \\
WSe
\end{pmatrix} = 0 \quad . \tag{4}
$$

Sendo que:

- $(x_1, x_2, \ldots, x_K) \geq 0$;
- $(y_1, y_2, \ldots, y_K)$ livre;
- $(s, w, t_1, t_2, \ldots, t_K) \geq 0$;

4 Método de Pontos Interiores

A idéia básica dos métodos de pontos interiores é mover-se em direção ao ótimo através do interior da região factível do modelo de programação linear. Esta idéia já havia sido apresentada desde 1967 com o método afim-escala de Dikin (Dikin, 1967) e o método de barreira logarítmica (Fiacco, 1968). Porém estes métodos não se mostraram competitivos com o método simplex, pois exigiram significativamente mais memória que o simplex, o que era inaceitável na época em que foram propostos.

Neste trabalho é abordado o método primal-dual seguidor de caminho que dentre as muitas variantes de métodos de pontos interiores é um dos mais elegantes teoricamente e possui um bom desempenho computacional (juntamente com o preditor corretor) para problemas de programação linear.

4.1 Método Primal-Dual Seguidor de Caminho

O método primal-dual seguidor de caminho surgiu em função das baixas taxas de convergência do método Primal-Dual Afim-Escala, que permite que os produtos x_iz_i se aproximem de zero rapidamente, tornando o processo de otimização pouco eficiente.

Seja o problema de Programação Linear na forma padrão:

$$
\begin{array}{ll}
\min & f(x) = c^T x \\
s.a. & Ax = b \\
 & x \geq 0.
\end{array} \tag{5}
$$

onde $x \in R^n$ é o vetor-coluna de variáveis primais, c e b são vetores de constantes e $A \in R^{m \times n}$ é a matriz de coeficientes. A esta formulação dá-se o nome de problema primal (PP), cujo problema dual (PD) associado é dado por:

$$
\begin{array}{ll}
\max & \phi(y) = b^T y \\
s.a. & A^T y + z = c \\
 & z \geq 0, \ y \ \text{livre.}
\end{array} \tag{6}
$$

Para resolver os problemas (5) e (6), o método Primal-Dual seguidor de caminho acrescenta uma perturbação à condição de complementaridade. Assim, podemos reescrever as condições de otimalidade como:

$$\begin{aligned} Ax &= b, \qquad x \geq 0 \\ A^T y + z &= c, \qquad z \geq 0 \\ XZe &= \mu e. \end{aligned} \tag{7}$$

A idéia principal dos métodos primais-duais consiste da aplicação do método de Newton ao sistema $F(x,y,z) = 0$ (Wright, 1996):

$$F(x,y,z) = \begin{pmatrix} Ax - b \\ A^T y + z - c \\ XZe - \mu e \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \\ r_c \end{pmatrix}. \tag{8}$$

Com isso, resolve-se os problemas primal e dual, simultaneamente.

A solução de (8), dado um ponto interior (x_0, z_0), através do método de Newton, é dada por:

$$(x^{k+1}, y^{k+1}, z^{k+1}) = (x^k, y^k, z^k) - \alpha[\nabla F(x^k, y^k, z^k)]^{-1} F(x^k, y^k, z^k) \tag{9}$$

onde α é tal que $x_{k+1}, z_{k+1} > 0$. ∇F é a matriz Jacobiana do sistema, sendo expressa da forma:

$$\nabla F(x^k, y^k, z^k) = \begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z_k & 0 & X_k \end{pmatrix}. \tag{10}$$

Assim, a direção de descida d_k é dada por:

$$d^k = \begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z_k & 0 & X_k \end{pmatrix}^{-1} \begin{pmatrix} r_p^k \\ r_d^k \\ r_c^k \end{pmatrix} = \begin{pmatrix} dx^k \\ dy^k \\ dz^k \end{pmatrix}. \tag{11}$$

Resolvendo-se o sistema da Equação (11), obtém-se as direções:

$$\begin{aligned} (AD^{-1}A^T)dy &= (r_p + AD^{-1}r_d - AD^{-1}X^{-1}r_c) \\ dx &= D^{-1}(A^T dy - r_d + X^{-1}r_c) \\ dz &= X^{-1}(r_c - Zdx) \end{aligned} \tag{12}$$

onde $D = X^{-1}Z$.

O passo mais caro em todos os métodos de pontos interiores consiste em resolver o sistema linear:

$$AD^{-1}A^T dy = r \tag{13}$$

onde $r = r_p + AD^{-1}r_d - AD^{-1}X^{-1}r_c$ e D é uma matriz diagonal que varia a cada iteração do algoritmo. Enquanto para uma grande classe de sistemas lineares, (13) pode ser resolvida eficientemente através de métodos de fatoração diretos, este não é o caso para problemas de fluxo em redes. Resende e Veiga (1993), fizeram um estudo comparativo entre métodos de fatoração direta e métodos iterativos e observaram ganhos neste último para problemas de fluxo em redes.

4.2 Dedução do Algoritmo para o Problema de Fluxo Multiproduto

Primeiramente calculamos o Jacobiano das condições de otimalidade apresentadas em (4). Os resíduos primais (rp_k e rpp), resíduos duais (rd_k) e os resíduos de folga complementar (rfc e $rfcc_k$) são dados por:

$$
\begin{array}{rcll}
rp_k & = & b_k - A_k x_k, & k = 1, 2, \ldots, K \\
rpp & = & u - \sum_{i=k}^{K} x_k - s, & k = 1, 2, \ldots, K \\
rd_k & = & c - A_k^T y_k + w - t_k, & k = 1, 2, \ldots, K \\
rfc & = & \mu e - (SWe) & \\
rfcc_k & = & \mu e - (X_k T_k e), & k = 1, 2, \ldots, K.
\end{array} \tag{14}
$$

As direções referentes às variáveis de folga do primal (ds) e do dual (dt_k), são dadas por:

$$ds = -W^{-1}Sdw + W^{-1}rfc$$

$$dt_k = -D_k^{-1}dx_k + X_k^{-1}rfcc_k$$

onde:
$D_k = T_k^{-1}X_k$.
Note que a matriz $W^{-1}S$ é uma matriz diagonal.
As demais direções são dadas por:

$$dx_k = D_k(-rd_k + X^{-1}rfcc_k + A_k^T dy_k - dw), \;\; k = 1, 2, \ldots, K$$

$$(A_k D_k A_k^T)dy_k = rp_k - A_k D_k(-rd_k + X^{-1}rfcc_k - dw), \;\; k = 1, 2, \ldots, K$$

$$-(D_1 + D_2 + \ldots + D_k + W^{-1}S)dw = rw - (D_1 A_1^T dy_1 + D_2 A_2^T dy_2 + \ldots + D_K A_K^T dy_K),$$

onde

$$rw = rpp - W^{-1}rfc + \sum_{k=1}^{K}(D_k rd_k - D_k X_k^{-1} rfcc_k).$$

O passo α é tal que $x_k, s, w, t_k > 0$.

Para resolver o problema de forma mais eficiente vamos usar o método dos gradientes conjugados precondicionado para resolver o sistema envolvendo as direções dy_k e dw.

4.3 Método dos Gradientes Conjugados Precondicionado

A eficiência computacional do primal dual seguidor de caminho para resolver o problema de fluxo multiproduto depende fortemente do método do gradiente conjugado precondicionado para resolver o sistema envolvendo as direções dy_k e dw em cada iteração. Tal sistema é dado por:

$$
\begin{pmatrix}
A_1 D_1 A_1^T & 0 & 0 & \ldots & -A_1 D_1 \\
0 & A_2 D_2 A_2^T & 0 & \ldots & -A_2 D_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \ldots & \ldots & A_K D_K A_K^T & -A_K D_K \\
-D_1 A_1^T & -D_2 A_2^T & \ldots & -D_K A_K^T & (W^{-1}S + \sum_{k=1}^{K} D_k)
\end{pmatrix}
\begin{pmatrix} dy_1 \\ dy_2 \\ \vdots \\ dy_K \\ dw \end{pmatrix}
=
\begin{pmatrix} ry_1 \\ ry_2 \\ \vdots \\ ry_K \\ -rw \end{pmatrix} \tag{15}
$$

onde:

$$ry_k \;\; = \;\; rp_k + A_k D_k(rd_k - X_k^{-1}rfcc_k), \;\; k = 1, 2, \ldots, K. \tag{16}$$

Para utilizar o método dos gradientes conjugados precondicionado temos que mostrar que a matriz do sistema (15) é definida positiva.

Seja

$$J = \begin{pmatrix} A_1D_1A_1^T & 0 & 0 & \dots & -A_1D_1 \\ 0 & A_2D_2A_2^T & 0 & \dots & -A_2D_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & A_KD_KA_K^T & -A_KD_K \\ -D_1A_1^T & -D_2A_2^T & \dots & -D_KA_K^T & (W^{-1}S + \sum_{k=1}^K D_k) \end{pmatrix} \quad (17)$$

e tomemos $p = [p_1\ p_2\ \dots\ , p_K, p_w]^T$ onde $p_k \in R^m,\ \ k = 1, \dots, K$(número total de produtos) e $p_w \in R^n$.

Temos que mostrar que $p^TJp > 0$, para todo $p \neq 0$.

$$\begin{aligned} p^TJp &= \sum_{i=1}^K p_i^T A_i D_i A_i^T p_i + p_w^T \left[\sum_{i=1}^K D_i + W^{-1}S\right] p_w - \sum_{i=1}^K p_w^T D_i A_i p_i - \sum_{i=1}^K p_i^T A_i D_i p_w \\ &= \sum_{i=1}^K p_i^T A_i D_i A_i^T p_i + \sum_{i=1}^K p_w^T D_i p_w - 2\sum_{i=1}^K p_i^T A_i D_i p_w + p_w^T W^{-1} S p_w \\ &= \sum_{i=1}^K \left[p_i^T A_i D_i A_i^T p_i + p_w^T D_i p_w - 2p_i^T A_i D_i p_w\right] + p_w^T W^{-1} S p_w \end{aligned}$$

Observação 2 *Uma matriz diagonal com elementos positivos na diagonal principal é definida positiva. Logo D_i e $W^{-1}S$ são definidas positivas.*

Como D_i é definida positiva podemos escrever $D_i = D_i^{1/2}D_i^{1/2}$, dai temos para $i = 1, \dots, K$:

$$p^TJp = \sum_{i=1}^K \left[(A_i^Tp_i)^T D_i^{1/2} D_i^{1/2} (A_i^T p_i) + p_w^T D_i^{1/2} D_i^{1/2} p_w - 2p_i^T A_i D_i^{1/2} D_i^{1/2} p_w\right] + p_w^T W^{-1} S p_w$$

Agora tomando $v = D_i^{1/2}(A_i^Tp_i)$ e $u = D_i^{1/2}p_w$ temos:

$$p^TJp = \sum_{i=k}^K \left[v_i^T v_i + u_i^T u_i - 2v_i^T u_i\right] + p_w^T W^{-1} S p_w,$$

ou seja,

$$p^TJp = \sum_{i=k}^K \langle v-u, v-u\rangle + p_w^T W^{-1} S p_w = \sum_{i=k}^K \parallel v-u \parallel^2 + p_w^T W^{-1} S p_w \geq 0.$$

Observe que $p^TJp = 0$ se, e somente se $p_w = 0$ e $v = u$, mas

$$v \ = \ u \ \Leftrightarrow \ A_i^T p_i \ = \ p_w \ = \ 0,\ i = 1, 2, \dots, K.$$

Lembrando que A_i tem posto linha completo para todo i, então devemos ter $p_i = 0$, $i = 1, 2, \dots, K$, e assim $p = 0$.

Portanto, $p^TJp > 0$ para todo $p \neq 0$ e J é simétrica definida positiva.

O método dos gradientes conjugados precondicionado foi feito baseando se no algoritmo 10.3.1 apresentado em Golub (1983).

Dado uma matriz simétrica definida positiva $A \in R^{n\times n}$, $b \in R^n$, um precondicionador simétrico definido positivo M e um ponto inicial x_0, o seguinte algoritmo resolve o sistema linear $Ax = b$:

$k = 0$
$r_0 = b - Ax_0$
enquanto $r_k \neq 0$
  Resolva $Mz_k = r_k$
  $k = k + 1$
  se $k = 1$
 $p_1 = z_0$

então
$\beta_k = \frac{r_{k-1}^T z_{k-1}}{r_{k-2}^T z_{k-2}}$
$p_k = z_{k-1} + \beta_k p_{k-1}$
end
$\alpha_k = \frac{r_{k-1}^T z_{k-1}}{p_k^T A p_k}$
$x_k = x_{k-1} + \alpha_k p_k$
$r_k = r_{k-1} - \alpha_k A p_k$
end

4.4 Precondicionadores

O precondicionamento é uma estratégia que deve ser aplicada aos métodos iterativos para melhorar as características de convergência de sistemas que possuam a matriz de coeficientes com autovalores dispersos. Precondicionar um sistema linear consiste em fazer com que a matriz apresente condições desejadas para que o método que está sendo aplicado resolva o sistema de forma eficiente (Silva, 2009).

Neste trabalho, construimos um precondicionador baseado no precondicionador feito por Resende e Veiga (1993). Inicialmente, o precondicionador é a diagonal da matriz de restrições do problema e posteriormente é a árvore geradora máxima. Para construir a árvore geradora máxima utilizamos o algoritmo de Kruskal (Tarjan, 1983).

A matriz diagonal constitui a forma mais simples e mais comum de precondicionador usado junto com o método do gradiente conjugado precondicionado (Golub, 1983). O precondicionador diagonal usado aqui é $M = diag(A_k D_k A_k^T)$.

O precondicionador árvore geradora máxima é obtido através do algoritmo de Kruskal (Tarjan, 1983). Esse precondicionador foi feito baseando-se em Resende e Veiga (1983), onde é feito um precondicionador floresta máxima.

Na k-ésima iteração do método primal dual clássico, seja N_k a submatrix de A_k com colunas correspondentes aos arcos da árvore geradora máxima. O precondicionador pode ser escrito como:

$$M = N_k D_k N_k^T$$

onde $D_k = T_k^{-1} X_k$.

5 Conclusões e trabalhos futuros

Os métodos de pontos interiores primais-duais, são bastante eficientes quando não se conhece, a priori, uma solução inicial factível. Nesses casos, e naqueles em que a dimensionalidade do problema é demasiada grande, os métodos de pontos interiores constituem uma alternativa eficaz a ser considerada.

Apresentamos uma nova abordagem de solução dos sistemas lineares oriundos dos métodos de pontos interiores para o problema de fluxo multiproduto. Mostramos que a matriz obtida é definida positiva e evitamos o cálculo da decomposição Cholesky aplicando o método dos gradientes conjugados precondicionado, para tanto desenvolvemos um novo precondicionador por blocos baseado em precondicionadores já existentes.

A implementação do método proposto está em andamento.

Como trabalho futuro podemos citar:

- Obtenção de grandes instâncias, com o objetivo de verificar o desempenho do algoritmo.

Agradecimentos

Este trabalho foi financiado pelas agências CAPES e CNPq.

Referências

Ahuja, R.E. & Magnanti, R.K., *Network Flows* Prentice Hall, Philadelphia, PA, USA, 1993.
Aurich, M.R.C., *Um Modelo de Fluxo de Potência Ótimo Linear com Reprogramação Corretiva Via Método de Pontos Interiores*, Dissertação de Mestrado, FEEC-UNICAMP, Julho, 2004.
Adler, I., Resende, M.G.C., Veiga, G. & Karmarkar, N. (1989), An Implementation of Karmarkar's Algorithm for Linear Programming, *Mathematical Programming*, 44, 297-335.
Castro, J. (2000), A Specialized Interior-Point Algorithm for Multicommodity Network Flows, *SIAM Journal on Optimization*,10, 852-877.

Chagas, R.J. *Uma Aplicação de Métodos Aproximados a Problemas de fluxo Multiproduto Inteiro*, Dissertação de Mestrado, CEFET-MG, Outubro, 2005.

Cunha, F.G.M. *Generalizações dos algoritmos Afim Escala Primal e Dual e um algoritmo próximo para programação quase convexa*, Tese de Doutorado, UFRJ, Janeiro, 2007.

Dikin, I.I. (1967), Iterative Solution of Problems of Linear and Quadratic Programming, *Soviets Math. Doklady*, 8, 674-675.

Fiacco, A.V. & Mccormick, G.P. *Nolinear Programming: Sequential Unconstrained Minimization Techniques*, Jhon Wiley Sons, 1968.

Ford, L.R. & Fulkerson, D.R., *Flows in networks*, Princeton University Press, New York, 1962.

Golub, G.H & Loan, C.F.V. , *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.

Hu, T.C. (1962), Multicommodity network flows, *Operations Research*, 11, 344-360.

Kojima, M., Meggido, N. & Mizuno, S. (1993), A Primal-Dual Infeasible-Interior Point Algorithm for Linear Programming, *Math. Programming*, 61, 263-280.

Resende, M.G.C. & Veiga (1993), An Implementation of a Network Interior Point Method, *DIMACS Series in Discrete Mathematics and Theorical Computer Science*, 12, 299-348.

Silva, J. *Uma Família de Algoritmos para Programação Linear Baseada no Algoritmo de Von Neumann*, Tese de Doutorado, IMECC-UNICAMP, Março, 2009.

Tarjan. R.E. *Data Structures and Network Algorithms*, SIAM, Philadelphia, PA, 1983.

Wright. S.J. *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, PA, USA, 1996.