

# O PROBLEMA LINEAR DE FLUXO EM REDE - MÉTODO PREDITOR-CORRETOR

*Luis Ernesto Torres Guardia, D.Sc.<sup>1</sup>*

**Resumo:** Neste trabalho apresentamos o método de pontos interiores preditor- corretor para resolver problemas de programação linear, e em especial aplicamos para o problema de fluxo em rede. O método em menção determina, em cada iteração, a solução de um sistema linear com a mesma matriz de coeficiente, mas com diferentes lados direitos. O sistema linear é resolvido usando o método de decomposição de Cholesky usando um comando específico do ambiente MATLAB. O método preditor – corretor é implementado usando o ambiente MATLAB e aplicado a modelos de fluxo em rede de diferentes dimensões. A experiência computacional mostra que o método de pontos interiores preditor - corretor é eficiente para resolver este tipo especial de problema linear de fluxo em rede.

**Palavras-chave:** Programação Linear; Pontos Interiores; Fluxo em Rede.

**Abstract:** In this work, we present the predictor-corrector interior-point method for linear programming problem and we apply this method to the network flow problem. At each step, the solution of a linear systems with the same matrix factors and different right-hand is determined. The respective linear system is solved by using the Cholesky factorization with a specified MATLAB command. The predictor-corrector method is implemented in MATLAB and applied for different network flow problems. The numerical experiments show that the predictor-corrector interior-point method is efficient for solving this special class of linear programming problem.

**Key words:** Linear programming; Interior point method; Network flow.

---

## 1. INTRODUÇÃO

Depois que Karmarkar (1983) apresentou seu método de pontos interiores para resolver problemas de programação linear com complexidade polinomial, diferente do tradicional método simplex, que possui complexidade exponencial, muitos outros métodos de pontos interiores foram desenvolvidos com resultados bastante interessantes. Entre eles, o método primal-dual foi que atraiu maior atenção devido a seu desempenho teórico, e as experiências numéricas, principalmente aplicadas para problemas de grande porte, também mostraram seu bom desempenho computacional, superior a outros métodos de pontos interiores.

O maior esforço computacional requerido em cada iteração do método primal-dual é determinar a solução de um sistema de equações lineares de Newton, a qual pode ser obtida usando algum método de solução, já seja direto ou iterativo. Entre os métodos diretos de solução, temos o método tradicional LU ou de Cholesky, consultar o livro de Nocedal e Wright (1999), que determina a decomposição da matriz associada ao sistema linear. Um outro método de decomposição é o algoritmo AINV de Benzi et al. (2000), que determina a decomposição da inversa da matriz correspondente ao sistema linear. Geralmente essas decomposições envolvem significativamente grande esforço computacional. Podem ser usadas

---

<sup>1</sup> Universidade Federal Fluminense - Departamento de Engenharia de Produção. Rua Passo da Pátria 156, São Domingos, Cep: 24210-240, Niterói, RJ, Brasil. tepletg@vm.uff.br

diversas vezes essas decomposições, para diferentes lados direitos, se essas soluções do sistema linear ajudam a diminuir o número total de iterações do método de pontos interiores. Essa é a idéia básica realizada por Mehrotra (1992) no seu método denominado de preditor – corretor, motivado no esquema de solução para problemas de equações diferenciais. No passo preditor, usa-se o denominado método primal – dual afim de escalonamento, para determinar uma direção de busca afim. No passo corretor, trata-se de determinar uma direção que consiga trazer, a iteração realizada no passo preditor, a um ponto suficientemente perto do caminho central. Posteriormente, Gondzio (1996) modificou esse método de tal modo a realizar diversas vezes o passo corretor de centro, e assim reduzir o número de iterações para alcançar a solução ótima do problema linear.

Entre os métodos iterativos, que determinam uma solução aproximada do sistema linear de equações de Newton, temos o algoritmo do gradiente conjugado, mas depende da escolha de um bom pré-condicionador, já que quando o método primal – dual aproxima-se à solução ótima do problema linear, a correspondente matriz do sistema linear pode-se transformar numa matriz mal condicionada.

Neste trabalho, apresentamos o método preditor – corretor de Mehrotra (1992) para resolver problemas de programação linear e depois aplicamos para determinar a solução do problema linear de fluxo em rede, o qual é um problema de grande porte devido a sua estrutura de rede que formam a matriz de restrições. Para resolver problemas de programação não linear, podemos também usar o método de pontos interiores primal – dual, em uma versão mais elaborada, ver Guardia e Mello (2007).

Para uma maior compreensão na teoria de fluxo em rede e de suas aplicações, sugere-se consultar o livro de Ahuja et al. (1993), e para um maior entendimento quanto à teoria de pontos interiores, sugere-se a leitura dos livros de Wright (1997) e Vanderbei (2001).

Este trabalho é organizado como segue: a seção 2 descreve o método preditor – corretor para resolver problemas de programação linear. Na seção 3, apresenta-se a formulação matemática do problema linear de fluxo em rede e os resultados computacionais do método mencionado acima, aplicado ao caso de fluxo em rede de diferentes

dimensões, usando o ambiente de programação MATLAB. Finalmente, na última seção apresentamos as conclusões.

## 2. MÉTODO PREDITOR – CORRETOR PARA PROGRAMAÇÃO LINEAR

Apresenta-se a seguir uma breve descrição do método de pontos interiores primal – dual para resolver o seguinte problema de programação linear, na forma padrão, denominado de primal (P) e dado por:

$$(P) \quad \begin{array}{ll} \text{minimizar} & c^T x \\ \text{sujeito a:} & Ax = b, \\ & x \geq 0, \end{array}$$

onde  $c$  é um vetor,  $c \in \mathbb{R}^n$ ,  $A$  é uma matriz e de posto completo,  $A \in \mathbb{R}^{m \times n}$ ,  $m < n$  e  $b \in \mathbb{R}^m$  são elementos dados, e  $x$  é o vetor de decisão,  $x \in \mathbb{R}^n$ .

O problema dual (D) do problema linear anterior (P) pode ser escrito na forma:

$$(D) \quad \begin{array}{ll} \text{maximizar} & b^T y \\ \text{sujeito a:} & A^T y + z = c, \\ & z \geq 0, \end{array}$$

em que  $y$  é o vetor de multiplicadores de Lagrange,  $y \in \mathbb{R}^m$  e  $z \in \mathbb{R}^n$  é o vetor dual de folga.

Podemos assumir que os problemas primal (P) e dual (D) satisfazem a condição de ponto interior (CPI), isto é, existe um ponto  $x^0, y^0, z^0$  tal que:

$$Ax^0 = b, \quad x^0 > 0, \quad A^T y^0 + z^0 = c, \quad z^0 > 0.$$

As condições de otimalidade de primeira ordem, também conhecidas como as condições de Karush – Kuhn – Tucker, (KKT), correspondentes ao problema primal (P) e dual (D) são:

$$Ax = b, \quad x \geq 0, \quad (1.1)$$

$$A^T y + z = c, \quad z \geq 0, \quad (1.2)$$

$$XZe = 0, \quad (1.3)$$

onde  $X$  é uma matriz diagonal dada pela componentes do vetor  $x$ , isto é,  $X = \text{diag}(x)$ . O mesmo para a matriz diagonal  $Z = \text{diag}(z)$  e  $e = (1, \dots, 1, \dots, 1) \in \mathbb{R}^n$  é um vetor com componentes de 1's,

A idéia básica do método de pontos interiores primal – dual é substituir a equação (1.3)

pela equação parametrizada dada por  $XZ\mathbf{e} = \mu\mathbf{e}$ . Assim, temos as condições de otimalidade perturbadas de KKT dadas por:

$$Ax = b, \quad x \geq 0, \quad (2.1)$$

$$A^T y + z = c, \quad z \geq 0, \quad (2.2)$$

$$XZ\mathbf{e} = \mu\mathbf{e}, \quad (2.3)$$

em que  $\mu$  é um parâmetro positivo denominado de barreira.

Se a condição (CPI) satisfaz, então o sistema (2) tem uma única solução para  $\mu > 0$ . Pode-se observar que quando  $\mu$  aproxima-se a zero, o sistema (2) aproxima-se ao sistema (1), e assim temos uma solução ótima para os problemas (P) e (D).

Aplicando o método de Newton ao sistema (2) a partir de um ponto interior dado, temos o seguinte sistema linear de equações:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \\ r_c \end{pmatrix} \quad (3)$$

em que:

$$r_p = b - Ax, \quad r_d = c - A^T y - z, \quad r_c = \mu\mathbf{e} - XZ\mathbf{e},$$

e denominadas de folga primal, folga dual e folga de complementaridade, respectivamente.

Se eliminamos  $\Delta z$  da terceira equação do sistema (3), temos o seguinte sistema, denominado de sistema aumentado ou indefinido:

$$\begin{pmatrix} -X^{-1}Z & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_d - X^{-1}r_c \\ r_p \end{pmatrix}, \quad (4.1)$$

$$\Delta z = r_d - A^T \Delta y, \quad (4.2)$$

e se realizamos mais uma substituição, eliminando  $\Delta x$ , temos o seguinte sistema denominado de sistema normal:

$$(AZ^{-1}XA^T) \Delta y = r_p + A(Z^{-1}X r_d + x - \mu Z^{-1}\mathbf{e}), \quad (5.1)$$

$$\Delta z = r_d - A^T \Delta y, \quad (5.2)$$

$$\Delta x = Z^{-1}(r_c - X \Delta z). \quad (5.3)$$

Comparando o sistema de equações normais (5.1) e o sistema indefinido (4.1), pode-se observar que a matriz para o sistema normal é

simétrica e definida positiva, de menor porte e talvez mais densa. A matriz do sistema aumentado (4.1) é simétrica indefinida e usualmente mais esparsa.

Na estratégia preditor – corretor de Mehrotra (1992), o primeiro passo, é determinar uma direção afim de escalonamento (passo preditor). Esta direção resolve um sistema de equações lineares de Newton (3) para  $\mu = 0$ , isto é,  $r_c = -XZ\mathbf{e}$ , e denominar essa direção por  $(\Delta x^a, \Delta y^a, \Delta z^a)$ . A seguir no seguinte passo, novamente resolve-se o sistema de Newton (3), agora sendo o lado direito  $r_p = 0$ ,  $r_d = 0$  e  $r_c = \sigma\mu\mathbf{e} - \Delta X^a \Delta Z^a \mathbf{e}$ , (passo corretor). Aqui,  $\mu$  é o parâmetro de barreira de centralização, e  $\Delta X^a = \text{diag}(\Delta x^a)$ ,  $\Delta Z^a = \text{diag}(\Delta z^a)$ .

Pode-se observar que a característica do método preditor- corretor é resolver dois sistemas lineares de equações, sendo a matriz de coeficientes a mesma para os passos preditor e corretor, mas com diferentes vetores no lado direito e pode-se aproveitar desta característica para usar a decomposição da matriz de coeficientes.

A seguir, apresentamos o método de pontos interiores preditor – corretor, ver Mehrotra (1992), Wright (1997), ou variações desse método, por exemplo consultar os trabalhos de Tits et al. (2005), Salahi e Terlaky (2007), (2008) e Salahi, et al. (2007) dentre outros.

Dados.  $x \in \mathbb{R}^n$ ,  $x > 0$ ,  $y \in \mathbb{R}^m$ ,  $z \in \mathbb{R}^n$ ,  $z > 0$ ,  $\mu = x^T z / n$ .

Parâmetro.  $\beta \in (0, 1)$ .

**Passo 1.** Determinar a direção afim de escalonamento, isto é, resolver:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x^a \\ \Delta y^a \\ \Delta z^a \end{pmatrix} = \begin{pmatrix} b - Ax \\ c - A^T y - z \\ -XZ\mathbf{e} \end{pmatrix},$$

para determinar  $(\Delta x^a, \Delta y^a, \Delta z^a)$  e calcular

$$t_p^a = \arg \max \{t \in [0, 1] / x + t \Delta x^a \geq 0\},$$

$$t_d^a = \arg \max \{t \in [0, 1] / z + t \Delta z^a \geq 0\}.$$

**Passo 2.** Determinar o parâmetro de centro

$$\sigma = (\mu^a / \mu)^3,$$

$$\text{onde } \mu^a = (x + t_p^a \Delta x^a)^T (z + t_d^a \Delta z^a) / n.$$

**Passo 3.** Determinar a direção corretor de centro, isto é, resolver:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x^c \\ \Delta y^c \\ \Delta z^c \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \sigma \mu e - \Delta X^a \Delta Z^a e \end{pmatrix},$$

para determinar  $(\Delta x^c, \Delta y^c, \Delta z^c)$ .

**Passo 4.** Formar a direção de busca total

$$(\Delta x^m, \Delta y^m, \Delta z^m) = (\Delta x^a, \Delta y^a, \Delta z^a) + (\Delta x^c, \Delta y^c, \Delta z^c),$$

e calcular

$$t_{\max}^p = \arg \max \{ t \in [0, 1] / x + t \Delta x^m \geq 0 \},$$

$$t_{\max}^d = \arg \max \{ t \in [0, 1] / z + t \Delta z^m \geq 0 \}.$$

**Passo 5.** Atualização das variáveis: Calcular

$$t_p^m = \min \{ \beta t_{\max}^p, 1 \},$$

$$t_d^m = \min \{ \beta t_{\max}^d, 1 \},$$

e calcular:

$$(x^+, y^+, z^+) = (x, y, z) + (t_p^m \Delta x^m, t_d^m \Delta y^m, t_d^m \Delta z^m),$$

e determinar

$$\mu = (x^+)^T (z^+) / n.$$

Pode-se notar que a principal característica deste método preditor – corretor é que a mesma matriz de coeficiente é usada para os passos preditor e corretor. Isto nos leva significativamente a um menor esforço computacionalmente, portanto apropriado para resolver problema de grande porte. Por outro lado, o passo corretor desse método usa alguma informação encontrada no passo preditor, isto é, de  $\mu^a$  e  $\Delta x^a \Delta z^a$ . Devido a seu superior desempenho computacional comparado com outros métodos de ponto interiores, este método primal – dual preditor – corretor é aplicado em outros problemas de otimização, por exemplo no problema linear estocástico apresentado nos trabalhos de Berkelaar et al. (2002) e Sun e Liu (2006) quando tais problemas lineares são formulados através de cenários, que dependendo do número de cenários, o correspondente problema

linear transforma-se em um problema de grande porte.

No **passo 1** do método preditor – corretor apresentado acima, o sistema de equações lineares é expresso em um sistema normal descrito anteriormente. Neste caso, resolve-se o seguinte sistema de equações:

$$(AZ^{-1}XA^T) \Delta y^a = r_p + A[ Z^{-1}X r_d + x ] \quad (6.1)$$

$$\Delta z^a = r_d - A^T \Delta y \quad (6.2)$$

$$\Delta x^a = -x - Z^{-1}X \Delta z \quad (6.3)$$

Agora no **passo 3**, o correspondente sistema normal é dado por:

$$(AZ^{-1}XA^T) \Delta y^c = AZ^{-1}[ -\sigma \mu e + \Delta X^a \Delta Z^a e ] \quad (7.1)$$

$$\Delta z^c = -A^T \Delta y^c \quad (7.2)$$

$$\Delta x^c = Z^{-1} [ \sigma \mu e - \Delta X^a \Delta Z^a e ] - Z^{-1} X \Delta z^c. \quad (7.3)$$

Uma decomposição da matriz simétrica definida positiva  $(AZ^{-1}XA^T)$  é o método tradicional de Cholesky, sugere-se consultar o livro de Nocedal e Wright (1996) e o trabalho de Mészáros (2005). A decomposição de Cholesky é dada tal que:

$$(AZ^{-1}XA^T) = R^T R,$$

em que R é uma matriz triangular superior com elementos positivos na diagonal

Existem também outros métodos de decomposição, digamos o método AINV de Benzi et al. (2000) mas que nesse caso a decomposição é da inversa da matriz  $(AZ^{-1}XA^T)$ , isto é:

$$(AZ^{-1}XA^T)^{-1} \approx LP^{-1} L^T,$$

onde L é uma matriz triangular superior com elementos na diagonal igual a 1's, e P é uma matriz diagonal.

Nos casos mencionados anteriormente, usando o método AINV ou de Cholesky, a solução do sistema normal é determinada através da solução de dois sistemas triangulares facilmente calculadas. Por exemplo, se usamos o método de Cholesky, temos que resolver o seguinte sistema:

$$R^T R \Delta y = \psi,$$

para qualquer vetor  $\psi$ , seja dado pelo lado direito de (6.1) ou de (7.1).

A solução de  $\Delta y$  é determinada em duas etapas usando o seguinte esquema:

- i) resolver  $R^T g = \psi$ , para determinar  $g$ .
- ii) resolver  $R \Delta y = g$ , para determinar  $\Delta y$ .

A solução de cada sistema linear de i) ou ii) é facilmente determinada devido ao fato de que são sistemas triangulares. Para maiores detalhes da solução deste tipo especial de sistema triangular, consultar o livro de Nocedal e Wright (1996).

### 3. O PROBLEMA DE FLUXO EM REDE E EXPERIÊNCIA NUMÉRICA

Para ilustrar a eficiência do método de pontos interiores predictor - corretor, consideremos o problema linear de fluxo em rede para um único produto. Este problema é definido por um grafo direcionado  $G = (N, A)$  onde  $N$  é o conjunto de nós com  $\|N\| = n$  e  $A$  é o conjunto de arcos com  $\|A\| = m$ . Para cada nó  $i \in N$ , um escalar  $s_i$  é associado, onde  $s_i$  é a oferta/procura no nó  $i$ , dependendo se seu valor é maior/menor que zero.

A variável  $x_{ij}$  representa o número de unidades de fluxo transportado através do arco  $(i, j)$ ,  $(i, j) \in A$ , isto é, o fluxo do nó  $i$  ao nó  $j$ , e  $c_{ij}$  é o custo de transporte associado a cada arco  $(i, j)$ .

O problema de programação linear geral de fluxo em rede é dado por:

$$\text{minimizar} \quad \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (8.1)$$

$$\text{sujeito a:} \quad \sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{j:(j,i) \in A\}} x_{ji} = s_i, \quad i \in N, \quad (8.2)$$

$$x_{ij} \geq 0, \quad (i,j) \in A. \quad (8.3)$$

A função objetiva (8.1) é uma função linear associada a cada arco. A equação (8.2) é denominada de restrição de conservação de fluxo de rede, que pelo número de nós e arcos pode-se transformar em um problema de grande porte. As restrições de conservação de fluxo são agrupadas na matriz  $A$  que devido a sua estrutura é esparsa. A restrição (8.3) é a condição de não negatividade de fluxo. O problema linear (8) pode ser expresso,

em forma compacta, no formato de programação linear padrão (P).

Dentro desse problema geral de fluxo em rede, podemos considerar o problema de distribuição de um produto transportado de diferentes centros de produção a diferentes centros de armazenamento a um custo associado de transporte, tal vez dependendo da distancia entre esses centros de produção e armazenamento. Esse produto depois é transportado dos centros de armazenamentos aos centros de distribuição ou consumo, também a um custo associado. Neste problema de distribuição, deseja-se determinar quantas unidades devem ser transportadas dos centros de produção aos centros de armazenamento e de ai aos centros de distribuição a um menor custo total.

Seja  $M$  o conjunto dos centros de produção e  $|M| = m$ ; e  $N$  o conjunto dos centros de armazenamentos e  $|N| = n$ . Seja  $P$  o conjunto dos centros de distribuição e  $|P| = p$ . Denomina-se  $x_{ij}$  o número de unidades do produto transportados do centro de produção  $i \in M$  ao centro de armazenamento  $j \in N$  e  $c_{ij}$  o correspondente custo de transporte por unidade. Seja  $y_{jk}$  o número de unidades do produto transportados do centro de armazenamento  $j \in N$  ao centro de distribuição ou consumo  $k \in P$ , e  $d_{jk}$  o correspondente custo de transporte por unidade. O número de unidades disponíveis em cada centro de produção é  $a_i$ ,  $i \in M$ , e o número de unidades solicitadas em cada centro de distribuição é  $b_k$ ,  $k \in P$ . Suponha-se que o total de unidades produzidas nos centros de produção é igual ao número de unidades solicitadas nos centros de distribuição.

O problema linear de distribuição pode ser formulado na seguinte forma:

$$\text{minimizar} \quad \sum_{i \in M, j \in N} c_{ij} x_{ij} + \sum_{j \in N, k \in P} d_{jk} y_{jk} \quad (9.1)$$

$$\text{sujeito a:} \quad \sum_{j \in N} x_{ij} = a_i, \quad i \in M \quad (9.2)$$

$$\sum_{k \in P} y_{jk} - \sum_{i \in M} x_{ij} = 0, \quad j \in N \quad (9.3)$$

$$\sum_{j \in N} y_{jk} = b_k, \quad k \in P \quad (9.4)$$

$$x_{ij} \geq 0, y_{jk} \geq 0, \quad i \in M, j \in N, k \in P \quad (9.5)$$

Neste caso, a função objetiva (9.1) é uma função linear que minimiza o custo total de transporte. A restrição (9.2) representa a capacidade de produção para cada centro de produção. A condição (9.3) é a condição de equilíbrio para cada centro de armazenamento. Por facilidade, estamos considerando o lado direito da condição (9.3) igual a zero mas pode ser qualquer valor positivo. A restrição (9.4) expressa a capacidade de consumo em cada centro de distribuição. Por último, a restrição (9.5) é a condição de não negatividade de fluxo. O problema de distribuição (9) pode também ser representado como um problema linear geral de fluxo em rede (8).

O método de pontos interiores predictor – corretor foi inteiramente codificado usando a linguagem MATLAB. Todos os experimentos computacionais foram realizados em um micro-computador PC Duron com 1 GB de RAM e 1600 MHz de frequência rodando a plataforma Windows XP.

Redes de diversas dimensões, para o caso geral de fluxo de rede (8), foram analisadas baseadas na rede básica do trabalho de Nagurney (1984), p. 476. Essa rede consiste de 20 nós e 28 arcos e é usada para determinar a solução do problema de equilíbrio de tráfego em rede de transporte quando formulado como um problema variacional. Essa rede básica, para nosso caso, é estendida para formar redes de grande porte. Para isso, foi implementado um programa específico, em MATLAB, para determinar a matriz de restrições lineares e sua dimensão, isto é, o número de nós e o número de arcos.

Para o caso do problema de distribuição, igualmente um programa específico em MATLAB foi implementado, para determinar a correspondente matriz de restrições lineares, usando os dados de número de centros de produção, centros de armazenamento e centros de consumo.

Um mesmo ponto inicial é usado em todos os experimentos computacionais do método predictor – corretor, e dado por:

$$x^0 = 10e, y^0 = -0,25, z^0 = e, e = (1, \dots, 1, \dots, 1) \in \mathbb{R}^n.$$

o qual não necessariamente é viável, mas existem outros métodos que poderiam ser usa-

dos para obter uma boa escolha de outro ponto inicial e talvez poderia melhorar o desempenho computacional. Para essa escolha, pede-se consultar o trabalho de Mehrotra (1992). Por outro lado, o valor de  $\beta = 0,99995$  é usado para determinar o respectivo tamanho de passo.

O critério de parada do método em estudo é estabelecido em termos da proximidade do valor da função objetiva no ponto atual e no ponto anterior. Exige-se também que a diferença dos valores das funções objetivas do problema primal e do dual seja menor ou igual a uma tolerância  $\varepsilon$  pré-estabelecida, isto é que:

$$\frac{|c^T x - b^T y|}{1 + |b^T y|} \leq \varepsilon.$$

Também poderíamos adotar outros critérios de parada.

Os custos da função objetiva foram obtidos usando uma função aleatória uniformemente distribuída, **rand** da plataforma MATLAB. Para resolver o correspondente sistema normal usamos a decomposição de Cholesky para a matriz  $(AZ^{-1}XA^T)$ , isto é, usando a função **chol** de MATLAB temos que  $(AZ^{-1}XA^T) = R^T R$ , e assim podemos determinar o correspondente  $\Delta y^a$  em (6.1) ou  $\Delta y^c$  em (7.1).

O comando **chol** do MATLAB usado para determinar a solução do sistema linear de equações  $(AZ^{-1}XA^T) \Delta y = \psi$ , para qualquer vetor  $\psi$ , é realizado da seguinte forma:

$$R = \mathbf{chol} (AZ^{-1}XA^T)$$

o qual determina a matriz triangular superior R, e

$$\Delta y = R \setminus (R' \setminus \psi)$$

o qual determina  $\Delta y$  a solução do correspondente sistema linear, sendo A' e R' representando no ambiente MATLAB a matriz  $A^T$  e  $R^T$  respectivamente.

A seguir, apresentamos os resultados computacionais para redes de diferentes dimensões, isto é, quando estendemos a rede básica de transporte de Nagurney (1984) a um número maior de arcos e nós.

**Tabela 1. Resultados computacionais da rede Nagurney.**

rede	arco=605 nós=320	arco=800 nós=420	arco=1190 nós=620	arco=1580 nós=820	arco=2390 nós=1230
<b>f.o.</b>	10500	11700	13100	18500	18900
<b>dual</b>	10500	11700	13100	18500	18900
<b>iter</b>	9	8	8	9	8
<b><math>\mu</math></b>	1,2130e-07	1,0343e-07	2,3284e-10	5,8190e-10	2,7217e-09
<b>tpo</b>	5,570728	9,891849	29,582768	76,646394	239,857102

onde:

**f.o.** :significa o valor da função objetiva do correspondente problema primal.

**dual**: representa o valor da função objetiva do correspondente problema dual.

**iter**: é o número de iterações realizados.

**$\mu$** : é o parâmetro de barreira.

**tpo**: significa o tempo de processamento (em

segundos) determinado pelo ambiente Matlab, usando as funções **tic** e **toc**. Não inclui o tempo necessário para determinar a matriz correspondente as restrições de conservação de fluxo do problema.

A seguir, temos os resultados computacionais para a rede do problema de distribuição para diferentes dimensões.

**Tabela 2. Resultados computacionais da rede de distribuição.**

rede	arco=1801 nós=90	arco=1751 nós=95	arco=1951 nós=95	arco=2101 nós=100	arco=2276 nós=100	arco=3201 nós=120
<b>m</b>	25	20	25	20	25	30
<b>n</b>	30	25	30	30	35	40
<b>p</b>	35	50	40	50	40	50
<b>f.o.</b>	1757,1	1980	1912,5	1830	1818,8	1140
<b>dual</b>	1757,1	1980	1912,5	1830	1818,7	1140
<b>iter</b>	9	8	9	9	10	8
<b><math>\mu</math></b>	1,7216e-11	8,8871e-08	3,7821e-09	1,5019e-08	4,2414e-08	1,5024e-09
<b>tpo</b>	11,342601	10,019547	13,516223	15,600158	20,105602	34,709601

onde neste caso temos que:

**m**: representa o número de centros de produção.

**n**: é o número de centros de armazenamentos.

**p**: significa o número de centros de consumo.

As outras variáveis, **f.o.**, **dual**, **iter**,  **$\mu$**  e **tpo.**, representam as mesmas definições dadas no problema da rede Nagurney.

Como pode observar-se das tabelas acima, o valor da função objetiva do problema primal é praticamente igual ao valor da função objetiva do problema dual. Por outro lado, o correspondente

valor do parâmetro de barreira  **$\mu$**  é bastante pequeno. Podemos afirmar, pelos resultados computacionais, que o método de ponto interior preditor- corretor mostra a sua eficiência para resolver o problema linear de fluxo de rede de diferentes dimensões.

#### 4. CONCLUSÕES

Este artigo apresentou o método de pontos interiores primal – dual preditor – corretor para resolver o problema de programação line-

ar de grande porte como é o caso do problema de fluxo em rede. O método de Cholesky, representado pelo comando **chol** do ambiente MATLAB, é usado na solução de mais de um sistema linear, que tem a mesma matriz de coeficientes para diferentes lados direitos, quando é aplicado tanto o passo preditor e o passo corretor respectivamente.

Os resultados computacionais realizados em algumas redes de diferentes dimensões confirmam a eficiência do método de pontos interiores primal – dual preditor – corretor.

## 5. BIBLIOGRAFIA

- AHUJA, A., MAGNANTI, T. e ORLIN, J., Network Flows : Theory, Algorithms and Applications, *Prentice-Hall*, Inc. New Jersey. 1993.
- BENZI, M., CULLUM, J. e TUMA, M., (2000), Robust approximate inverse pre-conditioning for the conjugate gradient method, *SIAM Journal of Scientific Computing*, 22, 1318-1332.
- BERKELAAR, A., DERT, C., OLDENKAMP, B. e ZHANG, S., (2002), A primal-dual decomposition-based interior point approach to two-stage stochastic linear programming, *Operations Research*, 50, 904 -915.
- GONDZIO, J., (1996), Multiple centrality corrections in a primal-dual method for linear programming, *Computational Optimization and Applications*, 6, 137 – 156.
- GUARDIA, L. e MELLO, J., (2007), Experiências computacionais com modelos de fluxo para múltiplos produtos com funções de custo não lineares e não separáveis, *Investigação Operacional*, 27, 53 -65.
- KARMAKAR, N., (1984), A polynomial-time algorithm for linear programming, *Combinatorica*, 4, 373 -395.
- MEHROTRA, S., (1992), On the implementation of a primal-dual interior point” , *SIAM Journal on Optimization*, 2, 575 – 601.
- MÉSZÁROS, C. (2005), The Cholesky factorization in interior point methods, *Computers and Mathematics with Applications*, 50, 1157 – 1166.
- NAGURNEY, A., (1984), Comparative test of multimodal traffic equilibrium methods, *Transportation Research*, 18B, 469 – 485.
- NOCEDAL, J. e WRIGHT, S., *Numerical Optimization*, Springer-Verlag, New York, 1999.
- SUN, J. e LIU, X., (2006), Scenario formulation of stochastic linear programs and the homogeneous self-dual interior-point method, *INFORMS Journal on Computing*, 18, 444 - 454.
- SALAHY, M. e TERLAKY, T., (2007), Adaptive large-neighborhood self-regular predictor-corrector interior-point methods for linear optimization, *Journal of Optimization Theory and Applications*, 132, 143 – 160.
- SALAHY, M. e TERLAKY, T., (2008), Mehrotra – type predictor – corrector algorithm revisited, *Optimization, Methods and Software*, 23, 259 -273.
- SALAHY, M., PENG, J. e TERLAKY, T., (2007), On Mehrotra – type predictor – corrector algorithms, *SIAM Journal on Optimization* , 18, 1377 – 1397.
- TITS, A., ABSIL, P. e WOESSNER, W., (2006), Constraint reduction for linear programs with many inequality constraints, *SIAM Journal on Optimization*, 17, 119 – 146.
- VANDERBEI, R., *Linear Programming: Foundations and extensions*, 2<sup>nd</sup>. Ed. Kluwer Academic, Boston, 2001.
- WRIGHT, S., *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.