TÉCNICAS PARA REDUÇÃO DO NÚMERO DE ITERAÇÕES NO MÉTODO DE PONTOS INTERIORES

Lilian F. Berti
Depto de Matemática Aplicada - IMECC - UNICAMP
13083-859 - Campinas - SP
lilian@ime.unicamp.br

Carla T. L. S. Ghidini
Depto de Matemática Aplicada - IMECC - UNICAMP
13083-859 - Campinas - SP
carla@ime.unicamp.br

Aurelio R. L. Oliveira
Depto de Matemática Aplicada - IMECC - UNICAMP
13083-859 - Campinas - SP
aurelio@ime.unicamp.br

RESUMO

Os métodos de pontos interiores têm sido amplamente utilizados para determinar a solução de problemas de programação linear, principalmente, os de grande porte. O método preditor-corretor, dentre todas as variações de métodos de pontos interiores, é um dos que mais se destaca, devido à sua eficiência e convergência rápida. Neste trabalho, com o objetivo de reduzir o número total de iterações do método de pontos interiores e de resolver problemas de grande porte que ainda não foram resolvidos por outras abordagens, consideramos duas técnicas distintas, as quais apresentaram bons resultados quando aplicadas separadamente. Uma delas é utilizar o algoritmo de ajustamento ótimo para p coordenadas para determinar bons pontos iniciais, visto que o ponto inicial influência diretamente no desempenho dos métodos de pontos interiores e a outra é realizar a iteração continuada para diminuir o número de operações realizadas ao resolver os sistemas lineares. As implementações de tais técnicas foram incorporadas ao PCx e os resultados obtidos nos experimentos computacionais realizados em um conjunto diversificado de problemas de programação linear foram satisfatórios.

PALAVRAS-CHAVE: Iteração continuada. Algoritmo de ajustamento ótimo. Método de pontos interiores

Programação Matemática

ABSTRACT

The interior point methods have been widely used to determine the solution of linear programming problems, especially the large problems. The predictor-corrector method, among all interior point methods is one mostly used due to its efficiency and convergence properties. In this work, the main objective is to reduce the total number of iterations of the interior point method and to solve large problems that have not been resolved by other approaches. For this we consider two different techniques, which showed good results when applied separately. One is to use the optimal adjustment algorithm for p coordinates to determine good starting points, since the starting point influence directly the performance of the interior point methods and the other is to use continued iterate to decrease the number of operations performed by solving the linear systems. The implementation of these techniques was incorporated into the PCx. The results

obtained in computational experiments performed on a diverse set of linear programming problems were satisfactory.

KEYWORDS: Continued iteration. Optimal adjustment algorithm. Interior point method

Mathematical Programming

1. Introdução

Os métodos de pontos interiores realizam uma trajetória interior à região de factibilidade formada pelas restrições do problema. No entanto, para que tais métodos sejam aplicados é necessário determinar um ponto inicial interior a esta região. Embora o ponto inicial somente precise manter as condições de não negatividade, o processo de convergência é sensível a este ponto e o desempenho dos métodos de pontos interiores pode melhorar se um bom ponto inicial for utilizado.

O algoritmo de ajustamento ótimo para p coordenadas proposto em Silva (2009) é uma generalização das ideias apresentadas em Gonçalves (2004), para desenvolver o algoritmo de ajustamento pelo par ótimo, o qual, por sua vez, é baseado no algoritmo de Von Neumann. As principais vantagens deste algoritmo são o avanço inicial rápido e simplicidade, visto que em cada iteração é necessário fazer apenas multiplicação de matriz por vetor e resolver um sistema linear com uma matriz definida positiva de ordem pequena. Apesar de que, em termos de convergência, o algoritmo de ajustamento ótimo para p coordenadas seja superior ao de Von Neumann, sua convergência também é lenta. Assim, a nossa proposta é explorar suas características e realizar somente algumas iterações dentro da heurística de Mehrotra (Mehrotra, 1992), a qual determina o ponto inicial para o método dos pontos interiores no software PCx para que pontos iniciais ainda melhores possam ser obtidos.

No método preditor-corretor, em cada iteração, é preciso resolver dois sistemas lineares para determinar a direção preditora-corretora. A resolução desses sistemas corresponde ao passo que requer mais tempo de processamento, devendo assim ser realizada de maneira eficiente. Para resolvê-los a abordagem mais utilizada é a fatoração de Cholesky. No entanto, realizar a fatoração de Cholesky em toda iteração tem um alto custo computacional. Dessa forma, na busca de redução de esforços foi desenvolvida a iteração continuada.

A iteração continuada é acionada no final de cada iteração do método preditor-corretor, após o cálculo da direção preditora-corretora e, com isso, uma nova direção é obtida. Para determinar esta nova direção, algumas componentes da direção anterior são consideradas nulas, aquelas que são responsáveis pelo bloqueio. A fatoração de Cholesky, já calculada na iteração do preditor-corretor é utilizada para resolver os sistemas lineares envolvidos na determinação da nova direção, denominada direção preditora-corretora continuada. Seu cálculo é feito da mesma forma que a direção preditora-corretora. Primeiro, a direção afim escala continuada é determinada, depois a direção de centragem e, por fim, é calculada a correção não linear continuada. Assim, o novo método é realizado em dois níveis. No nível externo é feito o cálculo da fatoração de Cholesky e da direção preditora-corretora tradicional. No nível interno é aplicada a iteração continuada com o objetivo de reduzir o número total de iterações do método de pontos interiores.

Na Seção 2 deste trabalho, são descritos o algoritmo de ajustamento ótimo para p coordenadas, a forma como é resolvido o subproblema que surge a cada iteração deste algoritmo e a heurística de Mehrotra, que determina o ponto inicial do método de pontos interiores e na qual o algoritmo de ajustamento ótimo é aplicado. Na Seção 3, o método de pontos interiores preditor-corretor e a iteração continuada são descritos em detalhes. Na Seção 4, os experimentos computacionais realizados com problemas selecionados de diferentes bibliotecas são apresentados. Finalmente, na Seção 5, estão as conclusões.

2. Algoritmo de ajustamento ótimo para *p* coordenadas

Considere o problema de encontrar uma solução factível para o seguinte conjunto de restrições lineares:

$$\begin{aligned} Ax &= 0, \\ e^t x &= 1, \\ x &\geq 0, \end{aligned} \qquad (1)$$

em que $A \in R^{mxn}$, x e $e \in R^n$, e é um vetor unitário e as colunas de A tem norma um, isto é, $||A_j||=1$, para $j = 1, ..., n$.

Geometricamente, as colunas A_j podem ser vistas como pontos sobre a hiperesfera m-dimensional com raio unitário e centro na origem. Dessa forma, o problema acima pode ser descrito como de atribuir ponderações x_j não negativas às colunas A_j de modo que, depois de reescalado, seu centro de gravidade seja a origem.

Obs.: Todo problema de programação linear pode ser reduzido ao problema (1) (veja Gonçalves, 2004).

O algoritmo Von Neumann, basicamente, consiste em encontrar a coluna A_s de A que forma o maior ângulo com o resíduo b^{k-1}, e então o próximo resíduo é a projeção da origem no segmento de reta ligando b^{k-1} a A_s.

Já o algoritmo de ajustamento pelo par ótimo, desenvolvido por Gonçalves *et al*. (2009) baseia-se na idéia de que o resíduo b^{k-1} pode ser movido de tal forma a aproximar-se da origem 0, aumentando um peso x^j de alguma coluna A_j e reduzindo um peso x^i de certa coluna A_i. Espera-se que o resíduo b^k esteja mais próximo da origem que o resíduo b^{k-1}.

Silva (2009) generalizou a idéia apresentada por Gonçalves *et al*. (2009) ao propor o algoritmo de ajustamento pelo par ótimo e desenvolveu o algoritmo de ajustamento ótimo para p coordenadas, em que p é limitado pela ordem do problema. Este algoritmo também possui como principais vantagens simplicidade e convergência rápida nas iterações iniciais.

O algoritmo de ajustamento ótimo para p coordenadas começa identificando as s_1 e s_2 colunas que fazem o maior e o menor ângulo com o vetor b^{k-1}, respectivamente, em que $s_1 + s_2 = p$ e p é o número de colunas a ser priorizada. Depois, um subproblema de otimização é resolvido e, finalmente, o resíduo e o ponto corrente são atualizados. Aqui, a resolução do subproblema é feita utilizando método de pontos interiores, uma vez que o número de casos a ser considerado, que satisfazem as condições de KKT, cresce exponencialmente com o valor de p.

Algoritmo 1:

Dado: $x^0 \geq 0$, com $e^t x^0 = 1$. Calcular $b^0 = Ax^0$.
Para $k = 1, 2, 3, ...$

1) Calcular: $\{A_{\eta_1^+}, ..., A_{\eta_{s_1}^+}\}$ que formam o maior ângulo com b^{k-1}.

$\{A_{\eta_1^-}, ..., A_{\eta_{s_2}^-}\}$ que formam o menor ângulo com b^{k-1} e tal que $x^{k-1} > 0$,

$i = \eta_1^-, ..., \eta_{s_2}^-$, em que $s_1 + s_2 = p$.

$v^{k-1} = \min_{i=1,..,s_1} A_{\eta_i^+}^t b^{k-1}$.

2) Se $v^{k-1} > 0$, então PARE. O problema (1) é infactível.

3) Resolver o subproblema:

$$\begin{aligned} \text{Min} \quad & \|b\|^2 \\ \text{s.a.} \quad & \lambda_0\left(1-\sum_{i=1}^{s_1} x^{k-1}_{\eta_i^+}-\sum_{j=1}^{s_2} x^{k-1}_{\eta_j^-}\right)+\sum_{i=1}^{s_1}\lambda_{\eta_i^+}+\sum_{j=1}^{s_2}\lambda_{\eta_j^-}=1 \qquad (2) \\ & \lambda_{\eta_i^+}\geq 0, \text{ para } i=1,\ldots,s_1, \\ & \lambda_{\eta_j^-}\geq 0, \text{ para } j=1,\ldots,s_2. \end{aligned}$$

em que, $$b=\lambda_0\left(b^{k-1}-\sum_{i=1}^{s_1} x^{k-1}_{\eta_i^+}A_{\eta_i^+}-\sum_{j=1}^{s_2} x^{k-1}_{\eta_j^-}A_{\eta_j^-}\right)+\sum_{i=1}^{s_1}\lambda_{\eta_i^+}A_{\eta_i^+}+\sum_{j=1}^{s_2}\lambda_{\eta_j^-}A_{\eta_j^-}$$

4) Atualizar:

$$b^k=\lambda_0\left(b^{k-1}-\sum_{i=1}^{s_1} x^{k-1}_{\eta_i^+}A_{\eta_i^+}-\sum_{j=1}^{s_2} x^{k-1}_{\eta_j^-}A_{\eta_j^-}\right)+\sum_{i=1}^{s_1}\lambda_{\eta_i^+}A_{\eta_i^+}+\sum_{j=1}^{s_2}\lambda_{\eta_j^-}A_{\eta_j^-}$$

$$u^k=\|b^k\|$$

$$x_j^k=\begin{cases} \lambda_0 x_j^{k-1} & j\notin\{\eta_1^+,\ldots,\eta_{s_1}^+,\eta_1^-,\ldots,\eta_{s_2}^-\} \\ \lambda_{\eta_i^+}, & j=\eta_i^+;\, i=1,\ldots,s_1 \\ \lambda_{\eta_j^-}, & j=\eta_j^-;\, i=1,\ldots,s_2. \end{cases}$$

$$k=k+1.$$

2.1 Solução do subproblema

Em cada iteração do algoritmo de ajustamento ótimo para p coordenadas é necessário resolver o subproblema (2). No caso $p = 2$, que é o algoritmo de ajustamento pelo par ótimo, o subproblema é resolvido verificando todas as possíveis soluções factíveis das condições de KKT, sendo um total de 7. No caso geral, que é o algoritmo para p coordenadas, o número de casos possíveis de soluções factíveis cresce exponencialmente com o valor de p. Este número é, exatamente, $2^{(p+1)}-1$ e isso torna a implementação do algoritmo inviável para valores razoavelmente grandes de p. Para contornar esta dificuldade, o subproblema (2) é abordado de uma forma diferente e, então, o método de pontos interiores seguidor de caminho é usado para resolvê-lo. A grande vantagem da abordagem é que o custo computacional para resolver um problema de grande porte não é significativo. Para mais detalhes sobre como resolver o subproblema (2) veja Silva (2009).

2.2. Heurística de Mehrotra

A heurística de Mehrotra, utilizada para determinar um ponto inicial para o método de pontos interiores no código do PCx, consiste nos seguintes passos:

Algoritmo 2:

1) Resolver mínimos quadrados para calcular os pontos:

$$\tilde{y}=\left(AA^t\right)^{-1}Ac, \quad \tilde{z}=c-A^t\tilde{y}, \quad \tilde{x}=A^t\left(AA^t\right)^{-1}b.$$

2) Encontrar os valores δ_x e δ_z tais que $\tilde{x}+\delta_x$ e $\tilde{z}+\delta_z$ sejam não-negativos:

$$\delta_x=\max(-1.5\min\{\tilde{x}_i\},0),$$
$$\delta_z=\max(-1.5\min\{\tilde{z}_i\},0).$$

3) Determinar δ_x e δ_z tais que os pontos x^0 e z^0 sejam centralizados:

$$\widetilde{\delta}_x = \delta_x + \frac{(\widetilde{x}+\delta_x e)^t(\widetilde{z}+\delta_z e)}{2\sum_{i=1}^{n}(\widetilde{z}_i+\delta_z)}, \quad \widetilde{\delta}_z = \delta_z + \frac{(\widetilde{x}+\delta_x e)^t(\widetilde{z}+\delta_z e)}{2\sum_{i=1}^{n}(\widetilde{x}_i+\delta_x)}.$$

4) Calcular os pontos iniciais:

$$y^0 = \widetilde{y}, \quad z^0 = \widetilde{z} + \widetilde{\delta}_z e, \quad x^0 = \widetilde{x} + \widetilde{\delta}_x e.$$

Algumas iterações do algoritmo de ajustamento ótimo para p coordenadas são realizadas antes da etapa de centralização (Passo 2). Isto porque, se o algoritmo for utilizado depois de centralizar os pontos, estes são melhorados, porém a centralidade, que é importante para os métodos de pontos interiores, pode ser perdida. O ponto inicial para o algoritmo de ajustamento ótimo para p coordenadas é o ponto determinado ao resolver mínimos quadrados no Passo 1.

Para mais detalhes sobre a heurística de Mehrotra veja Mehrotra (1992) e Czyzyk *et al*. (1999) e sobre o algoritmo de ajustamento ótimo para p coordenadas e sua implementação veja Ghidini *et al*. (2011).

3. Método preditor-corretor e iteração continuada

Considere o problema de programação linear na forma padrão primal:

$$\begin{array}{ll} \text{Min} & c^t x \\ \text{s.a.} & Ax = b \\ & x \geq 0 \end{array} \tag{3}$$

em que $A \in \mathrm{R}^{mxn}$, posto(A) = m, x e $c \in \mathrm{R}^n$, e $b \in \mathrm{R}^m$.

O problema dual associado na forma padrão é o seguinte:

$$\begin{array}{ll} \text{Max} & b^t y \\ \text{s.a.} & A^t y + z = c \\ & z \geq 0 \end{array} \tag{4}$$

em que $y \in \mathrm{R}^m$ representa o vetor das variáveis duais livres e $z \in \mathrm{R}^n$ representa as variáveis de folga duais.

As condições de otimalidade de primeira ordem (KKT) do problema (3) e (4) são:

$$\begin{cases} Ax - b = 0 \\ A^t y + z - c = 0 \\ XZe = 0 \\ (x, z) \geq 0 \end{cases} \tag{5}$$

sendo X = diag(x), Z = diag(z) e $e \in \mathrm{R}^n$ vetor unitário.

3.1 Método Preditor-Corretor

Os métodos de pontos interiores do tipo primal-dual consistem em aplicar o método de Newton às condições de otimalidade (5) do PPL, partindo de um ponto interior e mantendo interior a cada iteração. O método preditor corretor desenvolvido por Mehrotra (1992) consiste em utilizar uma direção composta por três componentes: direção afim-escala (direção de

Newton), direção de centragem e direção de correção não linear. (Monteiro, Adler, e Resende, 1990). Então, aplicando o método de Newton às condições de otimalidade obtém-se:

$$\begin{cases} A\tilde{d}x = r_p \\ A^t\tilde{d}y + \tilde{d}z = r_d \\ Z\tilde{d}x + X\tilde{d}z = r_a \end{cases} \tag{6}$$

em que

$$\begin{bmatrix} r_p \\ r_d \\ r_a \end{bmatrix} = \begin{bmatrix} b - Ax \\ c - A^t y - z \\ -XZe \end{bmatrix} \tag{7}$$

Eliminando variáveis podemos resolver o sistema da seguinte forma:

$$\begin{aligned} \tilde{d}y &= \left(ADA^t\right)^{-1}\left[r_p + AD\left(r_d + X^{-1}r_a\right)\right] \\ \tilde{d}x &= D\left(A^t\tilde{d}y - r_d + X^{-1}r_a\right) \\ \tilde{d}z &= X^{-1}\left(r_a - Z\tilde{d}x\right) \end{aligned} \tag{8}$$

em que $D = XZ^{-1}$, a matriz ADA^t é simétrica e definida positiva, pois A tem posto completo e D é uma matriz diagonal definida positiva.

Como $x = x + \alpha_p\tilde{d}x > 0$, $z = z + \alpha_d\tilde{d}z > 0$, $\alpha_p \in (0,1]$ e $\alpha_d \in (0,1]$ então

$$\alpha_p = \min\left\{1, \tau\min\left\{-\frac{x_i}{\tilde{d}x_i} \mid \tilde{d}x_i < 0\right\}\right\},\ \alpha_d = \min\left\{1, \tau\min\left\{-\frac{z_i}{\tilde{d}z_i} \mid \tilde{d}z_i < 0\right\}\right\} \text{ para } \tau \in (0,1) \tag{9}$$

No método primal dual afim escala (Monteiro *et al*. 1990) os produtos x_iz_i podem convergir para zero com velocidades diferentes, assim o método pode falhar ou progredir lentamente. Para evitar isto, o parâmetro μ é criado nas condições de otimalidade de forma que $x_iz_i = \mu$. Este parâmetro é atualizado a cada iteração e tende a zero à medida que o método aproxima da solução (método seguidor de caminho). No método preditor-corretor, μ é determinado de acordo com progresso da direção afim escala, de modo que se ocorrer melhoria suficiente, então μ é pequeno, caso contrário μ é uma perturbação maior.

Sejam:

$$\tilde{\gamma} = \left(x + \alpha_p\tilde{d}x\right)^t\left(z + \alpha_d\tilde{d}z\right) \quad \text{e} \quad \gamma = x^t z \tag{10}$$

representando o *gap* de dualidade (Wright, 1997). Definimos,

$$\mu = \begin{cases} \left(\gamma/\sqrt{n}\right)(\gamma/n), & \text{se } \gamma < 1 \\ \left(\tilde{\gamma}/\gamma\right)^{\kappa}(\gamma/n) & \text{caso contrário} \end{cases} \tag{11}$$

Quando $\alpha_p = \alpha_d = 1$, temos um ponto que satisfaz as equações de igualdade da factibilidade primal e dual, $r_p = r_d = 0$ e na restrição de complementaridade, $r_a = -\tilde{D}_x\tilde{D}_z e$, sendo $\tilde{D}_x = \text{diag}(\tilde{d}x)$ e $\tilde{D}_z = \text{diag}(\tilde{d}z)$.

Fazendo a correção não linear acima e introduzindo a perturbação, temos o sistema,

$$\begin{cases} A\bar{d}x = 0 \\ A^t\bar{d}y + \bar{d}z = 0 \\ Z\bar{d}x + X\bar{d}z = \mu e - \tilde{D}_x\tilde{D}_z e. \end{cases} \tag{12}$$

Desse modo, a direção preditora-corretora d é dada por $d = \tilde{d} + \bar{d}$ e determinada resolvendo o sistema:

$$\begin{cases} Adx = r_p \\ A^t dy + dz = r_d \\ Zdx + Xdz = r_s \end{cases} \tag{13}$$

em que

$$r_s = r_a + \mu e - \tilde{D}_x\tilde{D}_z e. \tag{14}$$

A solução de (13), obtida de forma similar ao sistema (6) é a seguinte:

$$\begin{aligned} dy &= \left(ADA^t\right)^{-1}\left[r_p + AD\left(r_d + X^{-1}r_s\right)\right] \\ dx &= D\left(A^t dy - r_d + X^{-1}r_s\right) \\ dz &= X^{-1}\left(r_s - Zdx\right) \end{aligned} \tag{15}$$

3.2 Iteração continuada

No método preditor-corretor, a etapa crítica de esforço computacional consiste no cálculo da fatoração de Cholesky da matriz ADA^t presente nos dois sistemas lineares que devem ser resolvidos a cada iteração. Dessa maneira, com o objetivo de reduzir esforços, mais especificamente o número total de fatorações, foi desenvolvida da iteração continuada, a qual é aplicada após o cálculo da direção d obtido por (15).

Na iteração continuada, é feita uma busca pelas componentes responsáveis pelo bloqueio nas direções (dx, dz), ou seja, de forma a encontrar i e j, tais que:

$$\begin{aligned} i &= \arg\min_t \left\{-x_t / dx_t \mid dx_t < 0\right\}, \\ j &= \arg\min_t \left\{-z_t / dz_t \mid dz_t < 0\right\}. \end{aligned} \tag{16}$$

Como pode não existir variável de bloqueio em alguma ou em ambas direções, os seguintes casos são considerados: (i) as componentes i e j bloqueiam, (ii) somente a componente i bloqueia, (iii) somente a componente j bloqueia e (iv) nenhuma componente bloqueia.

- **Caso (i):** As componentes i e j bloqueiam

Com as componentes que bloqueiam determinadas, na direção afim escala continuada $\hat{d}$ a ser calculada, deve-se ter $\hat{dx}_i = 0$ e $\hat{dz}_j = 0$. Além disso, a direção $\hat{d}$, aproximadamente, deve satisfazer o sistema:

$$\begin{cases} A\hat{dx} \cong r_p \\ A^t\hat{dy} + \hat{dz} \cong r_d \\ Z\hat{dx} + X\hat{dz} \cong r_a \end{cases} \tag{17}$$

De forma similar a ideia de Dikin (1967) ao desenvolver o método primal afim escala, para encontrar a solução deste sistema, um problema auxiliar é usado para determinar $\hat{dx}$, de forma a minimizar a alteração na direção dx. Considere $\bar{dx} = dx$ a solução do sistema (15). Assim, o problema auxiliar é o seguinte:

$$\begin{array}{ll} \text{Min} & \frac{1}{2}\left\| D^{-1/2}\hat{dx} - D^{-1/2}\bar{dx}\right\|^2 \\ & A\hat{dx} = r_p \\ \text{s.a.} & \hat{dx}_i = \beta_a \\ & \hat{dx}_j = \beta_b \end{array} \qquad (18)$$

em que $\beta_a = 0$ e $\beta_b = -x_j$ e cuja solução é:

$$\hat{dx} = \bar{dx} - DA^t v - \alpha d_{ii} e_i - \gamma d_{jj} e_j .$$

sendo $v = -\alpha d_{ii}\left(ADA^t\right)^{-1} A_i - \gamma d_{jj}\left(ADA^t\right)^{-1} A_j$,

$$\alpha = \frac{\bar{dx}_i - \beta_a + \gamma d_{ii} d_{jj} A_i^t \left(ADA^t\right)^{-1} A_j}{d_{ii}\left(1 - d_{ii} A_i^t \left(ADA^t\right)^{-1} A_i\right)},$$

$$\gamma = \frac{\left(\bar{dx}_j - \beta_b\right)\left(1 - d_{jj} A_i^t \left(ADA^t\right)^{-1} A_i\right) + \left(\bar{dx}_i - \beta_a\right) d_{jj} A_i^t \left(ADA^t\right)^{-1} A_j}{d_{jj}\left[\left(1 - d_{jj} A_j^t \left(ADA^t\right)^{-1} A_j\right)\left(1 - d_{ii} A_i^t \left(ADA^t\right)^{-1} A_i\right) - d_{ii} d_{jj}\left(A_i^t \left(ADA^t\right)^{-1} A_j\right)^2\right]}.$$

Pelo sistema (17) obtém-se:

$$\begin{array}{l} \hat{dz} = X^{-1}\left(r_a - Z\hat{dx}\right) \\ \hat{dy} = \left(ADA^t\right)^{-1} AD\left(r_d - \hat{dz}\right) \end{array}$$

Para determinar a direção preditora-corretora continuada, fazemos o cálculo de μ, a correção não linear e, além disso, deve-se ter $dx_i = 0$ e $dz_j = 0$. Assim, a direção deve satisfazer aproximadamente o sistema (13), o qual é resolvido de forma similar ao sistema (17), sendo utilizado o problema auxiliar:

$$\begin{array}{ll} \text{Min} & \frac{1}{2}\left\| D^{-1/2} dx - D^{-1/2}\bar{dx}\right\|^2 \\ & Adx = r_p \\ \text{s.a.} & dx_i = \beta_a \\ & dx_j = \beta_b \end{array} \qquad (19)$$

em que $\beta_a = 0$ e $\beta_b = -x_j + \dfrac{\mu - \hat{dx}_j dz_j}{z_j}$ e cuja solução é:

$$dx = \bar{dx} - DA^t v - \alpha d_{ii} e_i - \gamma d_{jj} e_j .$$

As demais direções são dadas por:

$$\begin{array}{l} dz = X^{-1}\left(r_s - Zdx\right) \\ dy = \left(ADA^t\right)^{-1} AD\left(r_d - dz\right) \end{array}$$

Para determinar a direção d não é necessário calcular novamente os sistemas lineares que envolvem a matriz ADA^t no cálculo de v. Assim, é necessário resolver somente um sistema linear com a matriz ADA^t para determinar dy.

Obs: Os casos (ii) e (iii) são situações particulares do caso (i) e as expressões são determinadas de modo análogo. Para detalhes, veja Berti (2012). A iteração continuada não é realizada se ocorre o caso (iv), pois não existe componente de bloqueio, e quando as componentes i e j são iguais.

Algoritmo 3:

1) Calcule o tamanho dos passos primal α_p e dual α_d por (9).
2) Atualize o ponto: $x = x + \alpha_p dx, \quad y = y + \alpha_d dy, \quad z = z + \alpha_d dz.$
3) Atualize os resíduos.
4) Encontre as componentes de bloqueio por (16).
5) Se $i = j$ ou não existem i e j, siga para próxima iteração.
6) Calcule $\hat{dx}$.
7) Calcule $\hat{dy}$ e $\hat{dz}$.
8) Calcule μ por (11) e r_s por (14).
9) Calcule dx.
10) Calcule dy e dz.
11) Calcule o tamanho do passo: $\alpha = \min\{\alpha_p, \alpha_d\}$.
12) Atualize o ponto: $x = x + \alpha dx, \quad y = y + \alpha dy, \quad z = z + \alpha dz.$

4. Experimentos Computacionais

Os experimentos computacionais foram realizados em um Intel Core i7, 2.93GHz, 16GB RAM, HD 1TB, Linux 64Bits, utilizando os compiladores gcc e gfortran.

O algoritmo de ajustamento ótimo para p coordenadas e a iteração continuada foram implementados em linguagem C e incorporados ao código do software PCx.

Para analisar o desempenho do PCx com a abordagem proposta foram resolvidos 43 problemas, alguns com acesso livre na internet (Netlib, Qaplib e Kennington) e outros problemas gentilmente cedidos por Gonçalves. Somente problemas cujo número de colunas é maior que cinco mil foram considerados.

O critério de parada para o algoritmo de ajustamento ótimo para p coordenadas é o número máximo de iterações (100) ou o erro relativo da norma residual menor que 10^{-4}. Aquele que ocorrer primeiro.

Para a escolha do valor de p foi utilizado o seguinte critério:

$$
\begin{array}{rcccll}
0 & < & m & \leq & 100 & \Rightarrow \quad p = 2 \\
100 & < & m & \leq & 2000 & \Rightarrow \quad p = 4 \\
2000 & < & m & \leq & 15000 & \Rightarrow \quad p = 8 \\
15000 & < & m & \leq & 30000 & \Rightarrow \quad p = 10 \\
30000 & < & m & \leq & 90000 & \Rightarrow \quad p = 20 \\
90000 & < & m & \leq & 150000 & \Rightarrow \quad p = 40 \\
150000 & < & m & & & \Rightarrow \quad p = 60
\end{array}
$$

Na Tabela 1, comparamos o número total de iterações realizadas pelo PCx original (coluna PCx) e pela versão do PCx com o algoritmo de ajustamento ótimo e a iteração continuada incorporados (coluna PCx_Mod). Nas colunas 2 e 3 desta tabela estão as dimensões (número de linhas e colunas) dos problemas após o pré-processamento. Nas colunas p e *Itaux* estão os valores

de *p* e o número de iterações do algoritmo de ajustamento ótimo. A coluna *k* traz as iterações em que a iteração continuada foi aplicada e a última coluna, mostra a origem dos problemas.

Problema	Linha	Coluna	*p*	*Itaux*	*k*	PCx	PCx_Mod	Coleção
80bau3b	2140	11066	8	2	5 a 10	**36**	42	Netlib
dfl001	5984	12143	4	10	5 a 10	56	**45**	Netlib
fit2d	25	10524	2	2	5 a 10	22	**21**	Netlib
fit2p	3000	13525	4	10	5 a 10	20	**19**	Netlib
maros-r7	2152	7440	4	2	1 a 5	16	**15**	Netlib
pilot87	1971	6373	4	2	5 a 10	33	**32**	Netlib
stocfor3	15362	22228	10	2	1 a 5	30	30	Netlib
woodw	708	5364	4	10	5 a 10	30	**28**	Netlib
cre-a	2994	6692	8	2	5 a 10	23	23	Kennington
cre-b	5336	36382	8	2	5 a 10	41	**39**	Kennington
cre-c	2375	5421	8	2	5 a 10	25	**23**	Kennington
cre-d	4102	28601	8	10	6 a 8	41	**36**	Kennington
ken11	10085	16740	8	2	6 a 8	20	20	Kennington
ken13	22534	36561	10	2	6 a 8	**23**	24	Kennington
ken18	78862	128434	20	2	6 a 8	29	**28**	Kennington
osa-07	1081	25030	4	2	6 a 8	22	**20**	Kennington
osa-14	2300	54760	8	2	1 a 5	25	**20**	Kennington
osa-30	4313	104337	8	4	5 a 10	24	**23**	Kennington
osa-60	10243	243209	8	5	1 a 5	33	**23**	Kennington
pds-06	9156	28472	8	10	1 a 5	**34**	36	Kennington
pds-10	15648	48780	10	10	6 a 8	**39**	40	Kennington
pds-20	38722	106180	20	2	6 a 8	55	55	Kennington
bl	5729	12462	8	2	6 a 8	32	**31**	Gonçalves
bl2	5729	12462	8	2	6 a 8	37	**34**	Gonçalves
co5	4849	10787	8	2	5 a 10	47	47	Gonçalves
co9	9090	19997	8	10	5 a 10	*	**51**	Gonçalves
cq9	8004	19317	8	2	5 a 10	49	**47**	Gonçalves
ex05	825	7797	2	10	1 a 5	32	32	Gonçalves
ex06	832	7895	4	2	1 a 5	63	**59**	Gonçalves
ex09	1821	18184	4	10	6 a 8	37	**35**	Gonçalves
ge	9150	14990	8	2	1 a 10	43	**42**	Gonçalves
nl	6665	14680	8	2	6 a 8	37	**36**	Gonçalves
els19	8149	15325	8	2	5 a 10	26	**25**	Qaplib
chr25a	5587	10417	4	10	6 a 8	28	**25**	Qaplib
chr22b	4350	13186	4	10	5 a 10	27	**16**	Qaplib
kra30a	18059	85725	8	2	5 a 10	26	26	Qaplib
kra30b	18059	85725	8	2	6 a 8	**27**	29	Qaplib
rou20	7359	37640	8	2	1 a 5	**16**	17	Qaplib
scr15	2234	6210	8	2	6 a 8	21	**20**	Qaplib
scr20	5079	15980	8	2	5 a 10	20	**19**	Qaplib
ste36a	27683	131076	20	4	5 a 10	32	**30**	Qaplib
ste36b	27683	131076	20	4	5 a 10	31	**30**	Qaplib
ste36c	27683	131076	20	4	5 a 10	31	**29**	Qaplib

Tabela 1: Comparação do desempenho de PCx e PCxMod
* significa que o método falhou

O uso da iteração continuada e do algoritmo de ajustamento ótimo para p coordenadas dentro da heurística de Mehrotra fizeram reduzir o número total de iterações do método de pontos interiores em 70% dos problemas testados. Nos problemas dfl001 e chr22b, a nova versão do PCx realizou 11 iterações a menos, o que é um valor bem significativo. Em aproximadamente 14% dos problemas o número de iterações aumentou e em cerca de 16% não houve alteração. O maior aumento do número de iterações foi de 6 iterações.

Um resultado importante é que o problema co9 foi resolvido somente pelo Pcx_Mod, mostrando que o código ficou mais robusto com a abordagem proposta.

Vale ressaltar que o tempo total necessário para obter uma solução para o algoritmo de ajustamento ótimo para p coordenadas não é significativo em relação ao tempo total de resolução dos problemas e que o esforço de cada iteração continuada é dominado pela solução de no máximo quatro sistemas lineares com uma matriz já fatorada.

5. Conclusões

Neste trabalho, o algoritmo de ajustamento ótimo para p coordenadas e a iteração continuada foram utilizados em conjunto com o método de pontos interiores com o objetivo de reduzir o número total de iterações necessárias para resolver até a otimalidade os problemas de programação linear.

Ao incorporar o algoritmo de ajustamento ótimo para p coordenadas após o Passo 1 da heurística de Mehrotra e realizar algumas iterações, pontos iniciais melhores foram determinados. Além disso, houve uma redução do esforço computacional ao aplicar a iteração continuada em algumas iterações do método de pontos interiores, uma vez que melhores direções foram determinadas.

Os experimentos computacionais em um conjunto diversificado de problemas mostraram a superioridade da abordagem proposta, a qual reduziu o número total de iterações em torno de 70% dos casos, principalmente, os de maiores dimensões.

Vale ressaltar que a nova versão do PCx tornou-se mais robusta desde que um dos problemas testados foi resolvido somente por esta versão.

Agradecimentos

Ao CNPq, CAPES e FAPESP pelo apoio financeiro.

Referências

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.

Berti, L., Iteração continuada aplicada ao método de pontos interiores, Dissertação, IMECC – UNICAMP, Campinas SP, 2012.

Chvátal, V.V., *Linear Programming*, W. H. Freeman and Company, New York, USA,1983.

Czyzyk, J., Mehrotra, S., Wagner, M., Wright, S.J. (1999), PCx an interior point code for linear programming. *Optimization Methods & Software*, 11-2, 397–430.

Dantzig, G.B., Converting a converging algorithm into a polynomially bounded algorithm. *Tech. rep., Stanford University*, SOL 91-5, 1991.

Dantzig, G.B., An ϵ-precise feasible solution to a linear program with a convexity constraint in $1/\epsilon^2$ iterations independent of problem size. *Tech. rep., Stanford University*, SOL 92-5, 1992.

Dikin, I.I. (1967), Iterative solution of problems of linear and quadratic programming, *Soviets Math. Doklady* ,8, 674-675.

Epelman, M., Freund, R.M. (2000), Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system. *Mathematical Programming*, 88, 451–485.

Ghidini, C. T. L. S., Oliveira, A. R. L., Silva, J., (2011) Optimal adjustment algorithm for *p* coordinates and the starting point in interior point methods. *American Journal of Operations Research*, 01, 191-202.

Gonçalves, J.P.M., A family of linear programming algorithms based on the von Neumann algorithm. PhD thesis, Lehigh University, Bethlehem, 2004.

Gonçalves, J.P.M., Storer, R.H., Gondzio, J. (2009), A family of linear programming algorithms based on an algorithm by von Neumann. *Optimization Methods and Software*, 24, 461–478.

Mehrotra, S. (1992), Implementations of affine scaling methods: Approximate solutions of systems of linear equations using preconditioned conjugate gradient methods, *ORSA Journal on Computing*, 4, 103–118.

Mehrotra, S. (1992), On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2, 575–601.

Monteiro, R. D. C., Adler, I., Resende, M. G. C. (1990), A polynomial-time primal-dual afine scaling algorithm for linear and convex quadratic programming and its power series extension, *Mathematics of Operations Research* , 15, 191-214.

Oliveira, A. R. L., Lyra, C., Correia, P.B. (1988), Implementação computacional de algoritmo polinomial de programação linear: aplicação ao planejamento de operação de sistemas hidrotérmicos, *Anais do VII Congresso Brasileiro de Automática – CBA*, 928-929.

Silva, J., Uma família de algoritmos para programação linear baseada no algoritmo de Von Neumann. Tese, IMECC – UNICAMP, Campinas SP, 2009.

Wright, S.J, *Primal-Dual Interior-Point Methods*, SIAM Publications, SIAM, Philadelphia, PA, USA, 1997.