

Uma Família de Novos Algoritmos para Programação Linear Baseada no Algoritmo Von Neumann

Jair da Silva, Aurelio R. L. Oliveira,
E-mail: jair@ime.unicamp.br, aurelio@ime.unicamp.br,

Marta I. Velazco

E-mail: velazco@ime.unicamp.br

Depto de Matemática, Estatística e Computação Científica, UNICAMP, IMECC,
13083-859, Campinas, SP

RESUMO: Neste trabalho, é apresentado uma família de novos algoritmos para programação linear. Esta família surgiu da generalização da idéia apresentada por João Gonçalves, Robert Storer e Jacek Gondzio, para desenvolver o algoritmo de ajustamento pelo par ótimo. Também apresentamos uma nova técnica de abordagem, usando métodos de pontos interiores para resolver um sistema não linear, necessário em cada iteração do algoritmo de ajustamento ótimo para p coordenadas.

1 Introdução

O algoritmo Von Neumann foi apresentado por Dantzig no início dos anos 1990 [1, 2] e mais tarde foi estudado por Epelman e Freund [3, 4]. Este algoritmo possui propriedades interessantes, como simplicidade e convergência inicial rápida, porém, ele não é muito prático para resolver problemas lineares, visto que sua convergência é muito lenta. Gonçalves estudou em sua tese de PhD [5] o algoritmo de Von Neumann e apresentou quatro novos algoritmos baseado neste, sendo que o algoritmo de ajustamento pelo par ótimo foi o que obteve melhor desempenho na prática.

O algoritmo de ajustamento pelo par ótimo herda as melhores propriedades do algoritmo Von Neumann. Embora Gonçalves prove que em termos de convergência o algoritmo de ajustamento pelo par ótimo é superior ao de Von Neumann, ainda assim, ele também não é prático para resolver problemas lineares, visto que sua convergência também é muito lenta.

No trabalho [7], generalizamos a idéia apre-

sentada por João Gonçalves, Robert Storer e Jacek Gondzio em [6], para desenvolver o algoritmo de ajustamento pelo par ótimo.

Ao generalizar a idéia em [6] para o algoritmo de ajustamento do par ótimo, desenvolvemos o algoritmo de ajustamento ótimo para p coordenadas. Na realidade para cada p temos um algoritmo diferente, onde p é limitado pela ordem do problema, assim desenvolvemos uma família de algoritmos. A grande vantagem da família de algoritmos é a sua simplicidade, isto é, em cada iteração dos algoritmos desta família fazemos apenas multiplicação de matrizes e resolvemos um sistema linear com uma matriz definida positiva de ordem pequena, comparada com os problemas que serão de grande porte.

No trabalho [7], temos que em cada iteração do algoritmo de ajustamento ótimo para p coordenadas é necessário resolver um sistema não linear de ordem no máximo $(p+1) \times (p+1)$ sobre algumas condições. Fazemos isto testando todos os casos de possíveis soluções factíveis, mas o número destes possíveis casos cresce exponencialmente com o valor de p , mais precisamente eles são $2^{p+1} - 1$ como veremos adiante. Neste trabalho contornamos este problema resolvendo este sistema não linear usando um método de pontos interiores. A grande vantagem de usar métodos de pontos interiores para resolver o subproblema é que o custo computacional para resolver um problema com uma matriz de ordem 10×10 ou de ordem 100×100 não é muito importante considerando que o problema a ser resolvido é de grande porte, já verificar os possíveis $2^{10} - 1$ casos ou os possíveis $2^{100} - 1$ casos representam custos bem diferentes.

Do ponto de vista computacional, nossa proposta não é resolver os problemas de programação linear até o final e sim usar esta família de algoritmos em conjunto com um método primal-dual de ponto interior in-factível, para acelerar a convergência deste.

2 Descrição do Problema

Consideremos o problema de encontrar uma solução factível do conjunto de restrições lineares:

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

onde $P \in R^{m \times n}$, $x \in R^n$ e $e \in R^n$ é o vetor com todas as coordenadas iguais a um, e as colunas de P tem norma um, isto é, $\|P_j\| = 1$, para $j = 1, \dots, n$. Geometricamente as colunas P_j podem ser vistas como pontos sobre a hipersfera m -dimensional com raio unitário e centro na origem (ver figura 1). O problema acima então pode ser descrito como de atribuir ponderações x_j não negativas às colunas P_j de modo que depois de reescalado seu centro de gravidade seja a origem.

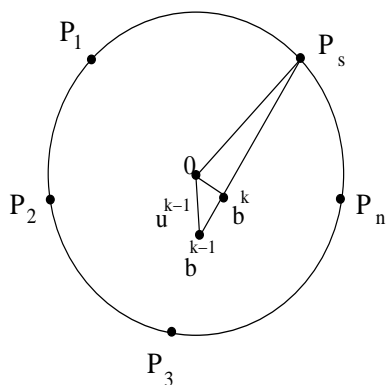


Figura 1: Ilustração do algoritmo Von Neumann

Note que qualquer problema de programação linear pode ser reduzido ao problema (1) ver [7].

3 Algoritmo de Von Neumann

Em 1948, Von Neumann propôs para Dantzig, em comunicação privada um algoritmo para programação linear, que foi divulgado por Dantzig no início dos anos 1990 em [1, 2]. Descrevemos a seguir este algoritmo:

Algoritmo de Von Neumann

Dado: $x^0 \geq 0$, com $e^t x^0 = 1$.

$k=0$

$b^0 = Px^0$.

REPITA

$s^+ = \operatorname{argmin}_{j=1, \dots, n} P_j^t b^{k-1}$.

$v_{k-1} = P_{s^+}^t b^{k-1}$.

Se $v_{k-1} > 0$, então PARE; o problema(1) é in-factível.

$u^{k-1} = \|b^{k-1}\|$

$\lambda = \frac{1-v_{k-1}}{(u^{k-1})^2 - 2v_{k-1} + 1}$

$b^k = \lambda b^{k-1} + (1-\lambda)P_{s^+}$

$x^k = \lambda x^{k-1} + (1-\lambda)e_{s^+}$

$k = k + 1$.

Note que $b^k = Px^k$, para todo $k \geq 0$. No algoritmo acima x^0 é arbitrário por exemplo podemos tomar $x_j^0 = \frac{1}{n}$, para $j = 1, \dots, n$. Na iteração k a coluna P_{s^+} é a coluna que forma maior ângulo com o vetor b^{k-1} .

O critério de parada usado para o algoritmo de Von Neumann em [6] é quando a diferença relativa entre $\|b^{k-1}\|$ e $\|b^k\|$, isto é, $rd = (\|b^{k-1}\| - \|b^k\|) / \|b^{k-1}\|$ é menor que uma certa porcentagem especificada.

4 Algoritmo de Ajustamento pelo Par Ótimo

O algoritmo de ajustamento pelo par ótimo é a generalização do “Weight-Reduction” ver [6]. De um certo modo, podemos dizer que o algoritmo de ajustamento pelo par ótimo, prioriza apenas duas variáveis em cada iteração, porque ele encontra o valor ótimo para duas coordenadas e ajusta o restante das coordenadas em função destes valores. Este algoritmo começa identificando os vetores P_{s^+} e P_{s^-} que tem o maior e menor ângulo com o vetor b^{k-1} , respectivamente. Então em seguida ele encontra os valores $x_{s^+}^k$, $x_{s^-}^k$ e λ onde $x_j^k = \lambda x_j^{k-1}$ para todo $j \neq s^+$ e $j \neq s^-$, que minimiza a distância de b^k a origem satisfazendo a convexidade e as restrições de não negatividade. Este problema de otimização tem a solução facilmente calculada examinando as condições Karush-Kuhn-Tucker (KKT).

Descrevemos o algoritmo de ajustamento pelo par ótimo a seguir:

Algoritmo de Ajustamento pelo par Ótimo

Dado: $x^0 \geq 0$, com $e^t x^0 = 1$.

$k=0$

$b^0 = P x^0$.

REPITA

$$s^+ = \operatorname{argmin}_{j=1, \dots, n} \{P_j^t b^{k-1}\}.$$

$$s^- = \operatorname{argmax}_{j=1, \dots, n} \{P_j^t b^{k-1} \mid x_j > 0\}.$$

$$v_{k-1} = P_{s^+}^t b^{k-1}.$$

Se $v_{k-1} > 0$, então PARE; o problema(1) é infactível.

Resolva o problema

$$\begin{aligned} & \text{minimizar } \|\bar{b}\|^2 \\ & \text{s.a } \lambda_1(1 - x_{s^+}^{k-1} - x_{s^-}^{k-1}) + \\ & \quad \lambda_2 + \lambda_3 = 1, \\ & \quad \lambda_1 \geq 0, \\ & \quad \lambda_2 \geq 0, \\ & \quad \lambda_3 \geq 0. \end{aligned} \quad (2)$$

onde,

$$\bar{b} = \lambda_1(b^{k-1} - x_{s^+}^{k-1}P_{s^+} - x_{s^-}^{k-1}P_{s^-}) + \lambda_2P_{s^+} + \lambda_3P_{s^-}.$$

A próxima iteração é calculada como

$$\begin{aligned} b^k &= \lambda_1(b^{k-1} - x_{s^+}^{k-1}P_{s^+} - x_{s^-}^{k-1}P_{s^-}) + \\ & \lambda_2P_{s^+} + \lambda_3P_{s^-}, \\ u^k &= \|b^k\|, \end{aligned}$$

$$x_j^k = \begin{cases} \lambda_1 x_j^{k-1}, & j \neq s^+ \text{ e } j \neq s^-, \\ \lambda_2, & j = s^+, \\ \lambda_3, & j = s^-. \end{cases}$$

$$k = k + 1.$$

Como no algoritmo de Von Neumann devemos notar que $b^k = P x^k$, para todo $k \geq 0$. O ponto inicial x^0 é arbitrário por exemplo podemos tomar $x_j^0 = \frac{1}{n}$, para $j = 1, \dots, n$. Na iteração k a coluna P_{s^+} é a coluna que forma maior ângulo com o vetor b^{k-1} e a coluna P_{s^-} é a coluna que forma o menor ângulo com o vetor b^{k-1} tal que $x_{s^-} > 0$.

Para resolver o problema (2), inicialmente eliminamos a variável λ_1 . Temos que

$$\lambda_1 = \frac{1 - \lambda_2 - \lambda_3}{1 - x_{s^+}^{k-1} - x_{s^-}^{k-1}} \quad (3)$$

então substituindo (3) em (2), podemos reescrever o problema como

$$\begin{aligned} & \text{minimizar } \|\bar{b}\|^2 \\ & \text{s.a } 1 - \lambda_2 - \lambda_3 \geq 0, \\ & \quad \lambda_2 \geq 0, \\ & \quad \lambda_3 \geq 0. \end{aligned} \quad (4)$$

$$\text{onde } \bar{b} = \frac{1 - \lambda_2 - \lambda_3}{1 - x_{s^+}^{k-1} - x_{s^-}^{k-1}}(b^{k-1} - x_{s^+}^{k-1}P_{s^+} - x_{s^-}^{k-1}P_{s^-}) + \lambda_2P_{s^+} + \lambda_3P_{s^-}.$$

Definindo $g_1(\lambda_2, \lambda_3) = \lambda_2 + \lambda_3 - 1$, $g_2(\lambda_2, \lambda_3) = -\lambda_2$, $g_3(\lambda_2, \lambda_3) = -\lambda_3$. As restrições do problema (4) podem ser reescritas como $g_1(\lambda_2, \lambda_3) \leq 0$, $g_2(\lambda_2, \lambda_3) \leq 0$ e $g_3(\lambda_2, \lambda_3) \leq 0$. Denotando a função objetivo por $f(\lambda_2, \lambda_3)$ e seja $(\bar{\lambda}_2, \bar{\lambda}_3)$ uma solução factível. Pela convexidade da função objetivo e restrições, se $(\bar{\lambda}_2, \bar{\lambda}_3)$ é uma solução ótima local, ela também é uma solução ótima global. Assim $(\bar{\lambda}_2, \bar{\lambda}_3)$ satisfaz a condição necessária (KKT) dada por

$$\begin{aligned} \nabla f(\bar{\lambda}_2, \bar{\lambda}_3) + \sum_{i=1}^3 \mu_i \nabla g_i(\bar{\lambda}_2, \bar{\lambda}_3) &= 0, \\ \mu_i g_i(\bar{\lambda}_2, \bar{\lambda}_3) &= 0, \text{ para } i = 1, 2, 3, \\ \mu_i &\geq 0, \text{ para } i = 1, 2, 3, \end{aligned}$$

onde os escalares μ_i são os multiplicadores de Lagrange. Da convexidade da função objetivo e restrições, as condições KKT são também suficientes. Resolvemos o problema (4) selecionando uma solução factível entre todas as possibilidades que satisfazem a condição KKT. Isto é feito analisando os seguintes casos:

$$(a) \lambda_2 = \lambda_3 = 0;$$

$$(b) \lambda_2 = 0, 0 < \lambda_3 < 1;$$

$$(c) \lambda_2 = 0, \lambda_3 = 1;$$

$$(d) 0 < \lambda_2 < 1, \lambda_3 = 0;$$

$$(e) 0 < \lambda_2 < 1, 0 < \lambda_3 < 1, \lambda_2 + \lambda_3 - 1 = 0;$$

$$(f) \lambda_2 = 1, \lambda_3 = 0;$$

$$(g) 0 < \lambda_2 < 1, 0 < \lambda_3 < 1, \lambda_2 + \lambda_3 - 1 \neq 0;$$

Para cada caso acima substituímos os valores conhecidos na equação KKT e resolvemos o sistema linear resultante.

5 Algoritmo de Ajustamento Ótimo para p Coordenadas

O algoritmo de ajustamento pelo par ótimo construído por Gonçalves em sua tese prioriza duas coordenadas em cada iteração. Vamos nos referir a este, como o algoritmo para 2 variáveis. Utilizando a mesma idéia contida neste algoritmo podemos generalizá-lo e construir o algoritmo para p variáveis. A idéia central utilizada no algoritmo para 2 variáveis para dar prioridade as duas coordenadas é resolver o subproblema (2). Este subproblema pode ser generalizado, e ao invés de utilizarmos duas colunas para formular o problema, podemos utilizar

quaisquer quantidades de colunas e assim dar importância a quantas variáveis desejarmos.

A maneira como escolhemos as variáveis para dar prioridade é livre e podemos escolhe-las, de acordo com o problema que iremos resolver. Uma escolha natural se vamos construir um algoritmo para p variáveis, é tomarmos $p/2$ colunas que fazem o maior ângulo com o vetor b^k e as outras $p/2$ colunas são as que fazem o menor ângulo com o vetor b^k , se p for ímpar colocamos uma coluna a mais para o conjunto de vetores que formam o maior ângulo com o vetor b^k por exemplo.

O algoritmo de ajustamento ótimo para p coordenadas segue as mesmas linhas do algoritmo de ajustamento pelo par ótimo:

Algoritmo de Ajustamento Ótimo para p Coordenadas

Dado: $x^0 \geq 0$, com $e^t x^0 = 1$.

$k=0$

$b^0 = P x^0$.

REPITA

$\{P_{\eta_1^+}, \dots, P_{\eta_{s_1}^+}\}$ que fazem o maior ângulo com o vetor b^{k-1} .
 $\{P_{\eta_1^-}, \dots, P_{\eta_{s_2}^-}\}$ que fazem o menor ângulo com o vetor b^{k-1} e tal que $x_i^{k-1} > 0, i = \eta_1^-, \dots, \eta_{s_2}^-$, onde $s_1 + s_2 = p$.
 $v_{k-1} = \max_{i=1, \dots, s_1} P_{\eta_i^+}^t b^{k-1}$.

Se $v_{k-1} > 0$, então PARE; o problema(1) é infactível.

Resolva o problema

$$\begin{aligned} & \text{minimizar } \|\bar{b}\|^2 \\ \text{s.a } & \lambda_0 \left(1 - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1} \right) + \\ & + \sum_{i=1}^{s_1} \lambda_{\eta_i^+} + \sum_{j=1}^{s_2} \lambda_{\eta_j^-} = 1, \\ & \lambda_{\eta_i^+} \geq 0, \text{ para } i = 1, \dots, s_1, \\ & \lambda_{\eta_j^-} \geq 0, \text{ para } j = 1, \dots, s_2, \end{aligned} \quad (5)$$

onde,

$$\bar{b} = \lambda_0 \left(b^{k-1} - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} P_{\eta_i^+} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1} P_{\eta_j^-} \right) + \sum_{i=1}^{s_1} \lambda_{\eta_i^+} P_{\eta_i^+} + \sum_{j=1}^{s_2} \lambda_{\eta_j^-} P_{\eta_j^-}.$$

A próxima iteração é calculada como

$$\begin{aligned} b^k &= \lambda_0 \left(b^{k-1} - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} P_{\eta_i^+} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1} P_{\eta_j^-} \right) \\ &+ \sum_{i=1}^{s_1} \lambda_{\eta_i^+} P_{\eta_i^+} + \sum_{j=1}^{s_2} \lambda_{\eta_j^-} P_{\eta_j^-}, \\ u^k &= \|b^k\|, \\ x_j^k &= \begin{cases} \lambda_0 x_j^{k-1}, & j \notin \{\eta_1^+, \dots, \eta_{s_1}^+, \eta_1^-, \dots, \eta_{s_2}^-\}, \\ \lambda_{\eta_i^+}, & j = \eta_i^+, i = 1, \dots, s_1, \\ \lambda_{\eta_j^-}, & j = \eta_j^-, j = 1, \dots, s_2. \end{cases} \\ k &= k + 1. \end{aligned}$$

5.1 Resolução do Subproblema Usando Métodos de Pontos Interiores

Em cada iteração do algoritmo de ajustamento ótimo para p coordenadas, é necessário resolver o subproblema (5). Este subproblema é resolvido encontrando uma solução no octante positivo, sujeito a factibilidade para um sistema linear de ordem no máximo $(p+1) \times (p+1)$. Em [7] resolvemos este subproblema verificando todos os casos de possíveis soluções factíveis. O inconveniente que surge naturalmente, ao resolver o subproblema desta forma é que o número de casos possíveis que temos que verificar cresce exponencialmente com o valor p .

De fato em [7], para resolver o subproblema (5) primeiramente eliminamos a variável λ_0 do problema. Assim temos que

$$\lambda_0 = \frac{1 - \sum_{i=1}^{s_1} \lambda_{\eta_i^+} - \sum_{j=1}^{s_2} \lambda_{\eta_j^-}}{1 - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1}} \quad (6)$$

então substituindo (6) em (5) podemos reescrever o problema como

$$\begin{aligned} & \text{minimizar } \|\bar{b}\|^2 \\ \text{s.a } & 1 - \sum_{i=1}^{s_1} \lambda_{\eta_i^+} - \sum_{j=1}^{s_2} \lambda_{\eta_j^-} \geq 0, \\ & \lambda_{\eta_i^+} \geq 0 \text{ para } i = 1, \dots, s_1, \\ & \lambda_{\eta_j^-} \geq 0 \text{ para } j = 1, \dots, s_2, \end{aligned} \quad (7)$$

$$\bar{b} = \lambda_0 \left(b^{k-1} - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} P_{\eta_i^+} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1} P_{\eta_j^-} \right) + \sum_{i=1}^{s_1} \lambda_{\eta_i^+} P_{\eta_i^+} + \sum_{j=1}^{s_2} \lambda_{\eta_j^-} P_{\eta_j^-}.$$

Definindo

$$g_0(\lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-}) = \sum_{i=1}^{s_1} \lambda_{\eta_i^+} + \sum_{j=1}^{s_2} \lambda_{\eta_j^-} - 1,$$

$$g_i(\lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-}) = -\lambda_{\eta_i^+},$$

para $i = 1, \dots, s_1$ e

$$h_j(\lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-}) = -\lambda_{\eta_j^-},$$

para $j = 1, \dots, s_2$.

As restrições do problema (7) podem ser reescritas como

$$g_i(\lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-}) \leq 0, \text{ para } i = 0, \dots, s_1 \text{ e}$$

$$h_j(\lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-}) \leq 0, \text{ para } j = 1, \dots, s_2.$$

Denotando a função objetivo por $f(\lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-})$ então a condição necessária (KKT) é dada por

$$\begin{aligned} & \nabla f(\lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-}) + \\ & + \sum_{i=0}^{s_1} \mu_{g_i} \nabla g_i(\lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-}) + \\ & + \sum_{j=1}^{s_2} \mu_{h_j} \nabla h_j(\lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-}) = 0, \\ & \text{com } \mu_{g_i} g_i(\lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-}) = 0, \\ & \text{para } i = 0, \dots, s_1, \\ & \mu_{h_j} h_j(\lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-}) = 0, \\ & \text{para } j = 1, \dots, s_2, \\ & \mu_{g_i} \geq 0, \text{ para } i = 0, \dots, s_1 \\ & \mu_{h_j} \geq 0, \text{ para } j = 1, \dots, s_2. \end{aligned} \quad (8)$$

Então resolvemos o problema (7) selecionando uma solução factível entre todas as possibilidades que satisfazem a condição KKT. Fazemos isto, analisando todos os casos possíveis de valores para as variáveis $\lambda_{\eta_i^+}$, para $i = 1, \dots, s_1$ e $\lambda_{\eta_j^-}$, para $j = 1, \dots, s_2$. Os casos a considerar são:

1) Caso $\lambda_{\eta_i^+} = 0$, $i = 1, \dots, s_1$, e $\lambda_{\eta_j^-} = 0$, $j = 1, \dots, s_2$.

2) Caso um $\lambda_k \neq 0$, com $k \in \{\eta_1^+, \dots, \eta_{s_1}^+, \eta_1^-, \dots, \eta_{s_2}^-\}$ e o restante igual a zero, o que nos fornece uma combinação C_1^p , mas temos que considerar o caso em que $\lambda_k = 1$, e $\lambda_k \neq 1$, assim temos $2C_1^p$ possibilidades.

3) Caso dois $\lambda_k \neq 0$, com $k \in \{\eta_1^+, \dots, \eta_{s_1}^+, \eta_1^-, \dots, \eta_{s_2}^-\}$ e o restante igual a zero, o que nos fornece uma combinação de C_2^p , mas novamente temos que considerar o caso em que $\lambda_k = 1$, e $\lambda_k \neq 1$, assim

temos $2C_2^p$ possibilidades.

Continuando com o raciocínio acima, concluímos que, o número total de casos quando queremos modificar p coordenadas é de

$$1 + 2C_1^p + 2C_2^p + 2C_3^p + \dots + 2C_p^p = 2^{p+1} - 1. \quad (9)$$

A equação (9), mostra que o número de casos que temos que considerar cresce exponencialmente com o valor de p . Este fato torna inviável a programação de um código do algoritmo de ajustamento ótimo para p coordenadas para valores razoavelmente altos de p . Com a finalidade de contornar este problema, abordamos o subproblema (5) de outra forma e podemos resolvê-lo aplicando métodos de pontos interiores. A grande vantagem de usar métodos de pontos interiores para resolver o subproblema (5), é que o custo computacional para resolver um problema de grande porte com uma matriz de ordem 10×10 ou de ordem 100×100 não é significativo, além disso, a programação de um código para o algoritmo de ajustamento ótimo para p coordenadas fica bem mais fácil.

A seguir descreveremos como reformulamos o subproblema (5), para usar um método de pontos interiores para resolvê-lo.

Podemos reescrever $b_k = P_k \lambda_k$, onde

$$\begin{aligned} P_k &= [w_k P_{\eta_1^+} \dots P_{\eta_{s_1}^+} \dots P_{\eta_1^-} \dots P_{\eta_{s_2}^-}], \\ w_k &= b^{k-1} - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} P_{\eta_i^+} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1} P_{\eta_j^-}, \\ \lambda_k &= (\lambda_0, \lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \dots, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-}). \end{aligned} \quad (10)$$

Assim, o subproblema pode ser reescrito como

$$\begin{aligned} & \text{minimizar } \frac{1}{2} \|P_k \lambda_k\|^2 \\ & \text{s.a } a^t \lambda_k = 1, \\ & \quad -\lambda_k \leq 0, \end{aligned} \quad (11)$$

onde $a = (a_1, 1, \dots, 1)$ e

$$a_1 = 1 - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1}. \quad (12)$$

Portanto as equações KKT do problema (11) são dadas por

$$\begin{aligned} P_k^t P_k \lambda_k + a r_k - l_k &= 0 \\ l_k^t \lambda_k &= 0 \\ a^t \lambda_k - 1 &= 0, \end{aligned} \quad (13)$$

com $0 \leq l_k, \lambda_k$, onde r_k e l_k são os multiplicadores de Lagrange de igualdade e desigualdade respectivamente.

Estas são as equações aonde aplicamos um método de pontos interiores.

5.2 Propriedades Teóricas do Novo Método

Em [7], foi demonstrado que o desempenho do novo método é superior em relação ao algoritmo de Von Neumann. Também mostrou-se que se $p_2 \geq p_1$ então o algoritmo de ajustamento ótimo para p_2 coordenadas possui um desempenho superior em relação ao algoritmo de ajustamento ótimo para p_1 coordenadas. A seguir apresentamos os teoremas demonstrados em [7].

Teorema 1 *O decréscimo em $\|b^k\|$ obtido por uma iteração do algoritmo de ajustamento ótimo para p coordenadas, com $1 \leq p \leq n$, onde n é dimensão das colunas de P , no pior caso é igual ao obtido por uma iteração do algoritmo Von Neumann.*

O teorema acima garante, que a família de algoritmos convergem no pior caso com a mesma taxa de convergência do algoritmo de Von Neumann. No caso $p = 2$, Gonçalves prova em [6], que este algoritmo tem um desempenho superior ao algoritmo de Von Neumann.

Teorema 2 *O decréscimo em $\|b^k\|$ obtido por uma iteração do algoritmo de ajustamento ótimo para p_2 coordenadas, no pior caso é igual ao obtido por uma iteração do algoritmo de ajustamento ótimo para p_1 coordenadas com $p_1 \leq p_2 \leq n$, onde n é a dimensão das colunas de P .*

6 Resultados Numéricos

Fizemos dois experimentos. No primeiro usamos os problemas scsd8, degen3 e 25fv47 da biblioteca Netlib. Neste implementamos o caso geral do algoritmo de ajustamento ótimo para p coordenadas no Matlab. Realizamos 100 iterações deste algoritmo para os casos $p = 4$, $p = 10$ e $p = 20$ nos problemas scsd8, degen3 e 25fv47 e comparamos o desempenho destes. Os resultado são mostrados nas três tabelas abaixo, onde $\|b^0\|$ é o resíduo inicial, $\|b^{100}\|$

é o resíduo na centésima iteração e \bar{m} , \bar{n} são o número de linhas e colunas do problema respectivamente. As notações são as seguintes: iteração (it) e redução relativa do resíduo (desemp).

Tabela 1: Problema scsd8.

scsd8	$\bar{m} = 397$	$\bar{n} = 2750$
it=100	$\ b^0\ $	$\ b^{100}\ $
$p = 4$	0,3332	0,0176
$p = 10$	0,3332	0,0120
$p = 20$	0,3332	0,0078
algoritmos	desemp(%)	
$p = 4 \times p=10$	47%	
$p = 4 \times p=20$	126 %	
$p = 10 \times p=20$	54%	

Tabela 2: Problema degen3.

degen3	$\bar{m} = 1503$	$\bar{n} = 2604$
it=100	$\ b^0\ $	$\ b^{100}\ $
$p = 4$	0,0651	0,0082
$p = 10$	0,0651	0,0060
$p = 20$	0,0651	0,0055
algoritmos	desemp(%)	
$p = 4 \times p=10$	37%	
$p = 4 \times p=20$	49 %	
$p = 10 \times p=20$	9%	

Tabela 3: Problema 25fv47.

25fv47	$\bar{m} = 798$	$\bar{n} = 1854$
it=100	$\ b^0\ $	$\ b^{100}\ $
$p=4$	0,0396	0,0079
$p=10$	0,0396	0,0043
$p=20$	0,0396	0,0038
algoritmos	desemp(%)	
$p=4 \times p = 10$	84%	
$p=4 \times p = 20$	108 %	
$p=10 \times p = 20$	13%	

Os três experimentos mostram, que aumentando o valor de p o desempenho do algoritmo melhora, ou seja, que conseguimos reduzir mais rapidamente o resíduo $\|b^k\|$ do problema auxiliar. Na realidade os três experimentos comprovam o Teorema 2.

No segundo experimento usamos o problema 25fv47, e adaptamos o algoritmo de ajustamento ótimo para p coordenadas com $p = 4$

ao PCx com modificações ver [8]. O PCx é um código de um método de ponto interior infactível aberto. Na iteração 24 tomamos o ponto (x^{24}, y^{24}, z^{24}) como ponto inicial do novo algoritmo. Realizamos 100 iterações e um ponto melhorado é devolvido para o PCx prosseguir da iteração 25 em diante. Um experimento similar, foi feito com o problema degen3, na iteração 12. A Tabela 4 mostra os ganhos obtidos. As notações são as seguintes: iteração (it) e iterações interna do gradiente conjugado (itGC).

Tabela 4: Ganhos por Iteração.

25fv47	PCx	p=4	
it	itGC	itGC	ganho(%)
25	365	260	19 %
26	35	21	40 %
27	102	48	53 %
TOTAL	502	329	35%
degen3	PCx	p = 4	
it	itGC	itGC	ganho (%)
13	22	37	-41%
14	92	73	21%
15	139	73	48%
TOTAL	253	183	28 %

Ao usarmos o algoritmo de ajustamento para 4 coordenadas reduzimos o número de iterações internas do gradiente conjugado para o cálculo das direções nas iterações precedentes do PCx, melhorando o desempenho computacional.

7 Conclusões

Apresentamos uma nova técnica de abordagem, usando métodos de pontos interiores para resolver um sistema não linear, necessário em cada iteração do algoritmo de ajustamento ótimo para p coordenadas. Os resultados numéricos mostram que na prática aumentando o valor de p o desempenho do algoritmo também melhora comprovando o Teorema 2. Os resultados numéricos também mostram que, a idéia de usar o algoritmo de ajustamento ótimo para p coordenadas, para melhorar o desempenho de um método primal-dual de ponto interior infactível é bastante promissora. Em trabalhos futuros pretendemos implementar, o caso geral do algoritmo de ajustamento ótimo para p coordenadas, integrado ao PCx e melhorar o

ponto sistematicamente em cada iteração dos métodos de pontos interiores.

8 Agradecimentos

Ao CNPq pelo suporte fornecido.

Referências

- [1] G. B. Dantzig. Converting a converging algorithm into a polynomially bounded algorithm. Technical Report SOL 91-5, Stanford University, 1991
- [2] G. B. Dantzig. An ϵ -precise feasible solution to a linear program with a convexity constraint in $\frac{1}{\epsilon^2}$ iterations independent of problem size. Technical Report SOL 92-5, Stanford University, 1992.
- [3] M. Epeleman and R. M. Freund. Condition number complexity of an elementary algorithm for resolving a conic linear system. Working Paper OR 319-97, Massachusetts Institute of Technology, 1997.
- [4] M. Epeleman and R. M. Freund. Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system. *Mathematical Programming*, 88:451-485, 2000.
- [5] J. P. M. Gonçalves. A Family of Linear Programming Algorithms Based on the Von Neumann Algorithm. PhD thesis, Lehigh University, Bethlehem, PA, 2004.
- [6] J. P. M. Gonçalves, R. H. Storer and J. Gondzio. New Algorithms for Linear Programming Based on an Algorithm by Von Neumann. Technical Report, Lehigh University, 2005.
- [7] J. Silva, A. R. L. Oliveira, M. I. Velazco. Um Algoritmo Novo Baseado no Algoritmo Von Neumann para Programação Linear. Submetido para publicação na revista TEMA.
- [8] S. Bocanegra, F. F. Campos, A. R. L. Oliveira. Using a Hybrid Preconditioner for Solving Large-Scale Linear Systems arising from Interior Point Methods, *Computational Optimization and Applications*, Vol 36, pp. 149-164, 2007.