

Uma Família de Algoritmos Simples Aplicada em Conjunto com Métodos de Pontos Interiores

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 apresentada 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. Este algoritmo, por sua vez, foi desenvolvido tendo como base o algoritmo de Von Neumann. O algoritmo de Von Neumann 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. Do ponto de vista computacional, nossa proposta não é utilizar a família de algoritmos para resolver os problemas de programação linear até encontrar uma solução e sim explorar a sua simplicidade e seu raio de convergência inicial rápido e usá-la em conjunto com um método de pontos interiores primal-dual inactível

Palavras-chave: Programação Linear, Algoritmo de Von Neumann, Algoritmos Simples, Métodos de Pontos Interiores.

1 Introdução

O algoritmo Von Neumann foi apresentado por Dantzig no início dos anos 1990 [2, 3] e mais tarde foi estudado por Epelman e Freund [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 [8], generalizamos a idéia apresentada 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.

Do ponto de vista computacional, nossa proposta não é utilizar a família de algoritmos para resolver os problemas de programação linear até encontrar uma solução e sim explorar a sua simplicidade e seu raio de convergência inicial rápido e usá-la em conjunto com um método de pontos interiores primal-dual inactível [9]. Em cada iteração deste método de pontos interiores é necessário a resolução de um sistema de equações lineares no cálculo da direção de Newton, neste trabalho isso será feito, utilizando uma abordagem iterativa pelo método dos gradientes conjun-

gados condicionado inicialmente pelo condicionador fatoração controlada de Cholesky [1] e posteriormente pelo condicionador separador [7].

No entanto, esta abordagem de condicionamento híbrida nem sempre é robusta e falha em algumas classes de problemas de programação linear. Isso ocorre devido a existência de uma faixa crítica na mudança dos condicionadores, no sentido que, se mudarmos de etapa antes desta faixa ou no início dela, o condicionador separador ainda não está preparado para assumir o processo e assim o método perde desempenho ou não converge. Vamos utilizar a família de algoritmo simples nesta faixa crítica fazendo algumas iterações e devolvendo um ponto melhor para o método de pontos interiores. Com isso prolongamos a vida útil do condicionador fatoração controlada de Cholesky ou melhoramos o desempenho do condicionador separador obtendo uma implementação robusta.

2 Descrição do Problema

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

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

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 negativos às colunas P_j de modo que depois de re-escalado seu centro de gravidade seja a origem.

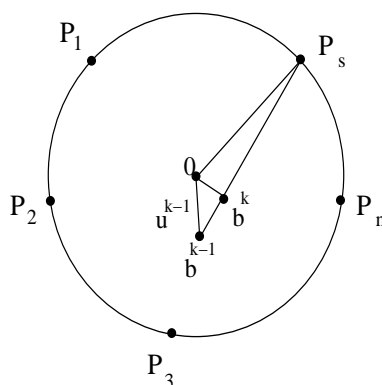


Figura 1: Ilustração do algoritmo Von Neumann

Note que qualquer problema de programação linear pode ser reduzido ao problema (1) ver [5].

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 [2, 3]. Descrevemos a seguir este algoritmo:

Algoritmo de Von Neumann

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

Para $k = 1, 2, 3, \dots$ **faça**

[1] Calcule

$$\begin{aligned} s^+ &= \operatorname{argmin}_{j=1, \dots, n} \{P_j^t b^{k-1}\}, \\ v_{k-1} &= P_{s^+}^t b^{k-1}. \end{aligned}$$

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

[3] Calcule

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

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

[4] Atualize

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

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

onde e_{s^+} é o vetor da base canônica com 1 na s^+ -ésima coordenada.

$$k = k + 1.$$

Fim

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} , 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-Tucher (KKT). Descrevemos o algoritmo de ajustamento pelo par ótimo a seguir:

Algoritmo de Ajustamento pelo par Ótimo

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

Para $k = 1, 2, 3, \dots$ **faça**

[1] Calcule

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

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

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

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

[3] Resolva o problema

$$\text{minimizar } \|\bar{b}\|^2$$

$$\text{s.a. } \lambda_0(1 - x_{s^+}^{k-1} - x_{s^-}^{k-1}) + \lambda_1 + \lambda_2 = 1, \quad (2)$$

$$\lambda_i \geq 0, \text{ para } i = 0, 1, 2,$$

$$\text{onde, } \bar{b} = \lambda_0(b^{k-1} - x_{s^+}^{k-1}P_{s^+} - x_{s^-}^{k-1}P_{s^-}) + \lambda_1P_{s^+} + \lambda_2P_{s^-}.$$

[4] Atualize

$$b^k = \lambda_0(b^{k-1} - x_{s^+}^{k-1}P_{s^+} - x_{s^-}^{k-1}P_{s^-}) + \lambda_1P_{s^+} + \lambda_2P_{s^-},$$

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

$$x_j^k = \begin{cases} \lambda_0 x_j^{k-1}, & j \neq s^+ \text{ e } j \neq s^-, \\ \lambda_1, & j = s^+, \\ \lambda_2, & j = s^-. \end{cases}$$

$$k = k + 1.$$

Fim.

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 qualquer 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 $e^t x^0 = 1$. Calcule $b^0 = P x^0$.

Para $k = 1, 2, 3, \dots$ **faça**

[1] Calcule

$\{P_{\eta_1^+}, \dots, P_{\eta_{s_1}^+}\}$ 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 $s_1 + s_2 = p$.

$v_{k-1} = \min_{i=1, \dots, s_1} P_{\eta_i^+}^t b^{k-1}$.

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

[3] Resolva o problema

minimizar $\|\bar{b}\|^2$

$$\text{s.a } \lambda_0 \left(1 - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1} \right) + \sum_{i=1}^{s_1} \lambda_{\eta_i^+} + \sum_{j=1}^{s_2} \lambda_{\eta_j^-} = 1, \quad (3)$$

$$\lambda_{\eta_i^+} \geq 0, \text{ para } i = 1, \dots, s_1,$$

$$\lambda_{\eta_j^-} \geq 0, \text{ para } j = 1, \dots, s_2,$$

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

[4] Atualize

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

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

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

$$k = k + 1.$$

Fim

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 (3). No caso $p = 2$, que é o algoritmo de ajustamento pelo par ótimo, Gonçalves resolve este subproblema verificando as condições KKT, mais precisamente, ele verifica todas as possíveis soluções factíveis das equações KKT, que neste caso são 7 ver [6]. No caso geral que é o algoritmo para p coordenadas, se resolvermos o subproblema seguindo o mesmo raciocínio, o número de casos possíveis de soluções factíveis cresce exponencialmente com o valor de p e este número de casos é exatamente $2^{p+1} - 1$ ver [8].

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 (3) de outra forma e podemos resolve-lo aplicando métodos de pontos interiores. A grande vantagem de usar métodos de pontos interiores para resolver o subproblema (3), 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 (3), para usar um método de pontos interiores para resolve-lo.

Podemos reescrever $b_k = P_k \lambda_k$, onde $P_k = [w_k P_{\eta_1^+} \dots P_{\eta_{s_1}^+} \dots P_{\eta_1^-} \dots P_{\eta_{s_2}^-}]$, $w_k = b^{k-1} - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} P_{\eta_i^+} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1} P_{\eta_j^-}$ e $\lambda_k = (\lambda_0, \lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \dots, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-})$.

Assim, o subproblema pode ser reescrito como

$$\begin{aligned} & \text{minimizar } \frac{1}{2} \|P_k \lambda_k\|^2 \\ & \text{s.a } a^t \lambda_k = 1, \\ & \quad -\lambda_k \leq 0, \end{aligned} \tag{4}$$

onde $a = (a_1, 1, \dots, 1)$ e $a_1 = 1 - \sum_{i=1}^{s_1} x_{\eta_i^+}^{k-1} - \sum_{j=1}^{s_2} x_{\eta_j^-}^{k-1}$. Portanto as equações KKT do problema

(4) são dadas por

$$\begin{aligned} P_k^t P_k \lambda_k + a r_k - l_k &= 0 \\ l_k^t \lambda_k &= 0 \\ a^t \lambda_k - 1 &= 0, \end{aligned} \tag{5}$$

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 em que aplicamos um método de pontos interiores.

5.2 Propriedades Teóricas do Novo Método

Em [8], 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 demonstrado em [8].

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.*

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

Nos experimentos o algoritmo de ajustamento ótimo para p coordenadas com $p = 4$, $p = 10$ e $p = 20$ foi adaptado ao código PCx com as modificações dadas em [9] e fizemos dois experimentos. Os problemas utilizados foram 8 da coleção QAPLIB e 12 da coleção NETLIB.

No decorrer desta seção vamos utilizar as seguintes notações nas tabelas: (MPI*) indica o número de iterações que o PCx modificado necessita para detectar que o problema não converge, itMelh) é onde realizamos iterações com o algoritmo de ajustamento ótimo para p coordenadas, número de iterações auxiliares em cada iteração melhorada (itAux) é a quantidade de iterações realizadas com o algoritmo de ajustamento ótimo para p coordenadas, (MPI _{p}) indica o número total de iterações do PCx modificado auxiliado pelo algoritmo simples para $p = 2, 10$ e 20 , (MPI) indica o número de iterações que o PCx modificado necessita para detectar que o problema converge, (Itgc) indica o número de iterações internas do método dos gradientes conjugados nas direções preditora e corretora após o auxílio do algoritmo simples, (T) indica o tempo total para a solução dos problemas em segundos, e (.) indica a iteração da mudança de fase.

No primeiro experimento, para 8 problemas onde PCx modificado (PCxM) não consegue obter uma solução, o experimento foi realizado da seguinte forma: nos problemas onde o método

necessita de segunda fase foram realizadas algumas iterações com algoritmo simples com $p = 2$, $p = 10$ e $p = 20$ na faixa crítica da mudança de fase e devolvido um ponto melhorado para o método prosseguir. Para os problemas que o método não necessita de segunda fase, na iteração onde o gap ou a infactibilidade primal ou dual aumenta, realizamos iterações com o algoritmo de ajustamento ótimo para p coordenadas com $p = 2$, $p = 10$ e $p = 20$ e devolvemos um ponto melhorado para o método de pontos interiores prosseguir. Apresentamos na tabela 1 uma comparação dos resultados obtidos com e sem o auxílio do algoritmo para p coordenadas. Os experimentos foram realizados sempre procurando fazer o mínimo de iterações do algoritmo simples.

Tabela 1: Problemas que não convergem na abordagem híbrida

Nome	MPI*	itMelh	itAux	MPI ₂	T	MPI ₁₀	T	MPI ₂₀	T
nug05	7(3)	4	50	-	-	6(3)	0.10	6(3)	0.14
nug05-3rd	7(3)	3	50	-	-	9(3)	6.14	7(3)	3.72
nug06-3rd	9(4)	2 a 4	50	-	-	-	-	12(-)	26.9
finnis	25(-)	16	100	-	-	34	0.80	27	0.73
agg2	26(-)	13	50	-	-	31	1.70	29	1.15
agg3	21(-)	12	100	47	2.9	39	1.45	36	1.85
scrs8	23(-)	2 a 8	50	-	-	-	-	38	2.04
bnl2	37(-)	11 a 12	300	-	-	47	14.1	40	25.1

Com esta abordagem conseguimos obter uma solução para os 8 problemas onde o PCx modificado não conseguia obter uma solução. Os resultados da tabela indicam que o algoritmo simples com $p = 20$ é superior ao algoritmo simples com $p = 2$ e $p = 10$, ou seja, aumentando o valor de p o desempenho do algoritmo melhora para esta classe de problemas. Em [9] para que os problemas NUG05, NUG05-3rd e NUG06-3rd convergissem, o parâmetro η foi alterado e um novo parâmetro foi calculado explorando a estrutura dos problemas, já em nossa abordagem o parâmetro η original foi mantido.

No segundo experimento para 12 problemas onde o PCx modificado converge e necessita de segunda fase, a estratégia utilizada foi fazer algumas iterações do algoritmo simples com $p = 20$ na faixa crítica da mudança de fase e devolver um ponto melhorado para o método de pontos interiores prosseguir.

Tabela 2: PCxM \times PCxMA₂₀

nome	MPI	Itgc	T	itMelh	itAux	MPI ₂₀	Itgc	T
e226	32(8)	5202	0.64	7	50	20(12)	655	0.50
maros	40(9)	28282	8.62	9	30	26(9)	15778	7.14
25fv47	29(17)	10231	6.25	17	50	28(17)	6814	5.30
scr15	24(11)	4860	30.63	13	100	23(11)	3997	28.7
chr22b	29(16)	2880	80.0	15	100	28(16)	10855	99.80
chr25a	29(15)	12698	181.4	14-15	50	28(15)	17362	219.2
nug07	11(4)	465	0.72	6	30	10(4)	729	0.79
nug12M	20(7)	9372	232.8	8	50	20(7)	7351	209.7
scr20	21(15)	5207	244.6	16	20	21(15)	4284	241.4
stocfor2	21(15)	2025	3.98	17	30	21(15)	747	3.78
degen3	16(8)	1426	19.91	9	50	16(8)	1126	19.2
degen2	12(5)	598	0.71	5	30	12(5)	409	0.69

Ao usarmos o algoritmo simples para 20 coordenadas em conjunto com o PCx modificado, conseguimos reduzir o número de iterações deste, para uma classe de problemas e para outra classe conseguimos reduzir o número de iterações internas dos gradientes conjugados, tornando o método mais robusto para estas classes de problemas.

7 Conclusões

Neste trabalho apresentamos uma nova família de algoritmos para programação linear. A grande vantagem desta família de algoritmos é a sua simplicidade e seu raio de convergência inicial rápido. Ao usarmos a família de algoritmos simples em conjunto com um método de pontos interiores ineficaz em uma abordagem onde o método não converge, foi possível obter convergência para um número maior de problemas. Para a maioria dos problemas a quantidade de iterações auxiliares utilizada do algoritmo simples foi pequena, e tendo em vista que, o seu custo por iteração é baixo o tempo computacional utilizado pelo novo algoritmo não é significativo em relação a uma iteração dos métodos de pontos interiores. Para outra classe de problemas onde o método de pontos interiores ineficaz necessita da segunda fase, com a abordagem utilizada foi possível reduzir o número de iterações do método para alguns problemas ou reduzir o número de iterações internas do método dos gradientes conjugados para outros problemas.

Os resultados numéricos indicam que ao usarmos a família de algoritmos simples em conjunto com um método de pontos interiores ineficaz melhoramos sua robustez e portanto seu desempenho. Em trabalhos futuros pretendemos utilizar a família de algoritmos simples para encontrar um ponto inicial para métodos de pontos interiores ineficazes.

8 Agradecimentos

Ao CNPq e a Fapesp pelo suporte fornecido.

Referências

- [1] F. F. Campos and N. R. C. Birkett. An efficient solver for multi-right hand side linear systems based on the CCCG(η) method with applications to implicit time-dependent partial differential equations. *SIAM J. Sci. Comput.*, 19(1):126–138, 1998.
- [2] G. B. Dantzig. Converting a converging algorithm into a polynomially bounded algorithm. Technical report, Stanford University, SOL 91-5, 1991.
- [3] 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, Stanford University, SOL 92-5, 1992.
- [4] M. Epelman 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, 2004.
- [6] J. P. M. Gonçalves, R. H. Storer, and J. Gondzio. A family of linear programming algorithms based on an algorithm by von neumann. *Aceito para publicação na revista Optimization Methods and Software*, 2007.
- [7] A. R. L. Oliveira and D. C. Sorensen. A new class of preconditioners for large-scale linear systems from interior point methods for linear programming. *Linear Algebra and Its Applications*, 394:1–24, 2005.
- [8] J. Silva. *Uma Família de Algoritmos para Programação Linear Baseada no Algoritmo de Von Neumann*. PhD thesis, IMECC – UNICAMP, Campinas SP, Março, 2009.
- [9] M. I. Velazco, A. R. L. Oliveira, and F. F. Campos. A note on hybrid preconditioners for large-scale normal equations arising from interior-point methods. *Aceito para publicação na revista Optimization Methods and Software*, 2008.