



# Uma nova Escolha do Parâmetro $p$ para uma Família de Algoritmos Simples em Conjunto com Métodos de Pontos Interiores

Jair da Silva \*

Universidade Federal de Mato Grosso do Sul - Instituto de Matemática - INMA  
09210-170 Campo Grande-MS, Brasil - j.silva@ufms.br

Rúbia Mara de Oliveira Santos

Universidade Federal de Mato Grosso do Sul - Instituto de Matemática - INMA  
09210-170 Campo Grande-MS, Brasil - rubia@dmt.ufms.br

Aurélio R. Leite Oliveira

Universidade Estadual de Campinas - Instituto de Matemática e Computação Científica  
13083-970 Campinas-SP, Brasil - aurelio@ime.unicamp.br

## Resumo

Nesse trabalho apresentamos uma nova heurística para o parâmetro  $p$  de uma família de algoritmos simples. Esta família de algoritmos foi utilizada para acelerar a convergência de uma versão do código PCx que é uma implementação de métodos pontos interiores adicionando uma abordagem iterativa híbrida com dois preconditionares para resolver os sistemas lineares do método de pontos interiores. Como a escolha do parâmetro  $p$  determina a performance dessa família, uma escolha adequada do parâmetro  $p$  é essencial para o sucesso desta abordagem. A família de algoritmos é aplicada na transição dos pré-condicionadores afim de melhorar tanto a velocidade quanto a robustez do código PCx. Experimentos numéricos sobre um conjunto de problemas de programação linear mostraram que a nova heurística do parâmetro  $p$  melhora esta abordagem, reduzindo o número total de iterações de ponto interior e o tempo de execução.

**Palavras Chave.** Heurística, Método de Ponto Interiores, Pré-condicionadores.

Área principal: Programação Matemática

## Abstract

In this work, we present a new heuristic for the parameter  $p$  of a family of simple algorithms. This family of algorithms was used to accelerate the convergence of a version of PCx code which is an implementation of interior point methods by adding a hybrid iterative approach with two preconditioners for solving linear systems of the interior point method. As the choice of the parameter  $p$  determines the performance of this family, an appropriate choice of the parameter  $p$  is essential for the success of this approach. A family of algorithms is applied in the transition of the preconditioners in order to improve both the speed and robustness of the PCx code. Numerical experiments on a set of linear programming problems showed that the new heuristic of the parameter  $p$  improves this approach by reducing the total number of iterations of interior point and runtime.

**Keywords.** Heuristic, Preconditioning, Interior Point Methods.

Main area: Mathematical Programming

---

\*Este trabalho tem apoio financeiro da FAPESP, Brasil

## 1 Introdução

Desde o surgimento dos métodos de pontos interiores para programação linear, códigos computacionais sofisticados baseados nessas ideias vêm se firmando como alternativas eficientes para solução de problemas de grande porte com estruturas genéricas [Adler et al., 1989, Bocanegra et al., 2007, Oliveira and Lyra, 1991, Oliveira and Sorensen, 2005]. Cada iteração de um método de pontos interiores envolve a solução de um ou mais sistemas lineares [Gondzio, 1996, Lustig et al., 1992, Mehrotra, 1992]. Este é o passo mais caro destes métodos do ponto de vista computacional. Usualmente métodos diretos são utilizados para resolver esses sistemas. A abordagem mais utilizada nas implementações existentes é a fatoração de Cholesky no sistema de equações normais [Czyzyk et al., 1999, Gondzio, 1996, Lustig et al., 1992]. No entanto, por limitações de tempo e memória seu uso torna-se proibitivo em muitos problemas de grande porte, fazendo com que abordagens iterativas sejam mais adequadas [Bergamaschi et al., 2007, Bocanegra et al., 2007, Chai and Toh, 2007, Oliveira and Sorensen, 2005]. Na resolução destes sistemas por métodos iterativos a utilização de pré-condicionadores se faz necessária para que uma convergência rápida possa ser obtida. Em particular, pré-condicionadores adaptativos a estes sistemas lineares foram aplicados em conjunto com o método dos gradientes conjugados em [Bocanegra et al., 2007] obtendo excelentes resultados computacionais ao serem integrados ao PCx [Czyzyk et al., 1999]. O preconditionador denominado fatoração controlada de Cholesky desenvolvido em [Campos and Birkett, 1998] é utilizado nas iterações iniciais e o preconditionador separador [Oliveira and Sorensen, 2005] especialmente desenvolvido para as iterações finais é utilizado em uma segunda etapa.

Em [Silva, 2009], foi desenvolvido uma família de algoritmos simples para melhorar a desempenho do código PCx. Esta família de algoritmos surgiu ao generalizar a ideia apresentada em [Gonçalves et al., 2009] para desenvolver o algoritmo de ajustamento pelo par ótimo. Ao generalizar a ideia do algoritmo de ajustamento do par ótimo, foi desenvolvido o algoritmo de ajustamento ótimo para  $p$  coordenadas. Na realidade para cada  $p$  tem-se um algoritmo diferente, onde  $p$  é limitado pela ordem do problema, assim foi desenvolvido uma família de algoritmos. A vantagem desta família de algoritmos é a sua simplicidade, isto é, em cada iteração dos algoritmos desta família, fazemos apenas multiplicação de matriz vetor e resolvemos um sistema linear com uma matriz definida positiva de ordem pequena, quando comparada com os problemas que serão de grande porte.

O código PCx foi desenvolvido no Optimization Technology Center at Argonne National Laboratory and Northwestern University por Joe Czyzyk, Sanjay Mehrotra, Michael Wagner e Stephen Wright [Czyzyk et al., 1999] e implementa o método primal-dual preditor-corretor [Mehrotra, 1992] com múltiplas correções [Gondzio, 1996]. Esta abordagem é considerada a mais eficiente das variantes de métodos de pontos interiores. Este código está disponível para download gratuitamente, desde que, não seja para fins comerciais. O código PCx, no qual agregamos a família de algoritmos possui as modificações descritas em [Oliveira and Sorensen, 2005] e [Bocanegra et al., 2007]. Neste código existe uma faixa crítica na troca dos pré-condicionadores, no sentido que se mudarmos de etapa antes desta faixa ou no início dela, o preconditionador separador ainda não está preparado para assumir o trabalho e assim o método não converge. Em [Ghidini et al., 2012], na troca de pré-condicionadores foi realizada uma abordagem de fazer algumas iterações da família de algoritmos, fornecendo assim, uma passagem suave entre estes pré-condicionadores. Esta abordagem se mostrou eficiente para alguns problemas testados, no entanto, o desempenho da família de algoritmos depende da escolha de um parâmetro  $p$ . Em [Ghidini et al., 2012] é proposta uma heurística baseada em experimentos numéricos. Nesse trabalho, propomos uma escolha dinâmica para o parâmetro  $p$  da família de algoritmos simples, que seja

função do número de linhas e colunas do problema a ser resolvido. Por meio de experimentos numéricos em um conjunto de problemas de programação linear, mostramos melhorias significativas sobre a heurística usada em [Ghidini et al., 2012].

## 2 Métodos de Pontos Interiores

### 2.1 O Problema na Forma Padrão

Na descrição deste trabalho consideraremos o problema de programação linear na forma padrão:

$$\begin{aligned} & \text{minimizar} && c^t x \\ & \text{sujeito a} && Ax = b, \quad x \geq 0, \end{aligned}$$

onde  $A$   $m \times n$  é uma matriz de posto completo  $m$  e  $c$ ,  $b$  e  $x$  são vetores colunas de dimensão apropriada. Associado a este problema temos o problema dual:

$$\begin{aligned} & \text{maximizar} && b^t y \\ & \text{sujeito a} && A^t y + z = c, \quad z \geq 0, \end{aligned}$$

onde  $y$  é um vetor coluna de dimensão  $m$  de variáveis livres e  $z$  é o vetor coluna de dimensão  $n$  de variáveis de folga duais. O *gap* dual é dado por  $\gamma = c^t x - b^t y$  que se reduz a  $\gamma = x^t z$  para pontos primais e duais factíveis.

A direção afim nos métodos de pontos interiores primais-duais é dada por [Monteiro et al., 1990, Wright, 1996]:

$$\begin{pmatrix} 0 & I & A^t \\ Z & X & 0 \\ A & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta \tilde{x} \\ \Delta \tilde{z} \\ \Delta \tilde{y} \end{pmatrix} = \begin{pmatrix} r_d \\ r_a \\ r_p \end{pmatrix} \quad (1)$$

onde  $X = \text{diag}(x)$ ,  $Z = \text{diag}(z)$  e os resíduos são dados por  $r_p = b - Ax$ ,  $r_d = c - A^t y - z$ ,  $r_a = -XZe$  e  $e$  representa o vetor de uns.

Eliminando as variáveis  $\Delta \tilde{z}$  de (1) obtemos o sistema aumentado:

$$\begin{pmatrix} -D & A^t \\ A & O \end{pmatrix} \begin{pmatrix} \Delta \tilde{x} \\ \Delta \tilde{y} \end{pmatrix} = \begin{pmatrix} r_1 \\ r_p \end{pmatrix} \quad (2)$$

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

A forma mais utilizada para resolver (2) consiste em reduzir o sistema através de eliminação das variáveis  $\Delta \tilde{x}$  a um sistema simétrico definido positivo com a matriz de equações normais  $AD^{-1}A^t$  e aplicar então a fatoração de Cholesky [Adler et al., 1989, Czyzyk et al., 1999, Gondzio, 1996].

### 2.2 pré-condicionadores Adaptativos para Solução dos Sistemas Lineares

Em [Oliveira and Sorensen, 2005] é desenvolvido o preconditionador separador

$$D_B^{\frac{1}{2}} B^{-1} (A D^{-1} A^t) B^{-t} D_B^{\frac{1}{2}} = I + D_B^{\frac{1}{2}} B^{-1} N D_N^{-1} N^t B^{-t} D_B^{\frac{1}{2}}. \quad (3)$$

onde

$$PDP^t = \begin{pmatrix} D_N & 0 \\ 0 & D_B \end{pmatrix} \quad (4)$$

e  $P$  é uma matriz de permutação das colunas de  $A$ .

Este preconditionador é específico para os sistemas lineares oriundos dos métodos de pontos interiores e quando aplicado ao método dos gradientes conjugados evita o cálculo da matriz de equações normais e sua fatoração. Este preconditionador apresenta muito bons resultados próximo a solução do problema com uma escolha adequada de  $P$  mas não é eficiente nas primeiras iterações. Em [Oliveira and Sorensen, 2005] é utilizado um preconditionador diagonal para as iterações iniciais dos métodos de pontos interiores.

Por outro lado, a fatoração controlada de Cholesky [Campos and Birkett, 1998] apresenta bons resultados em outras aplicações com uma fatoração incompleta que apresenta um alto grau de esparsidade.

Considere  $B = LL^t = \tilde{L}\tilde{L}^t + R$  onde  $R$  é a matriz resto da fatoração incompleta de Cholesky  $\tilde{L}$  e a matriz preconditionada  $\tilde{L}^{-1}B\tilde{L}^{-t} = (\tilde{L}^{-1}L)(\tilde{L}^{-t}L)^t$ . Definindo  $E = L - \tilde{L}$  temos  $\tilde{L}^{-1}B\tilde{L}^{-t} = (I + \tilde{L}^{-1}E)(I + \tilde{L}^{-t}E)^t$ .

O número de iterações do método dos gradientes conjugados está diretamente relacionado com a norma de  $R = LE^t + E\tilde{L}^t$  [Duff and Meurant, 1989]. A fatoração controlada de Cholesky foi desenvolvida buscando minimizar

$$\|E\|_F = \sum_{j=1}^m c_j \quad \text{onde, } c_j = \sum_{i=1}^m |l_{ij} - \tilde{l}_{ij}|^2.$$

$$\text{Considere: } c_j = \sum_{k=1}^{m_j+\eta} |l_{ikj} - \tilde{l}_{ikj}|^2 + \sum_{k=m_j+\eta+1}^m |l_{ikj}|^2, \text{ onde}$$

$m_j$  representa o número de elementos não nulos abaixo da diagonal da coluna  $j$  de  $B$ ;  
 $\eta$  representa o número adicional de elementos não nulos permitido na fatoração incompleta.

O problema de minimização pode ser abordado por heurísticas:

- Aumentando o parâmetro  $\eta$  permitindo maior preenchimento.
- Escolhendo os  $m_j + \eta$  maiores elementos de  $\tilde{L}$  em valor absoluto para  $\eta$  fixo. Desta forma, os maiores elementos ficam na primeiro somatório diminuindo  $\|E\|_F$ .

Estes pré-condicionadores foram agregados ao código PCx resultando em uma abordagem eficiente que obteve melhores resultados onde a fatoração de Cholesky falha ou é mais lenta em diversos problemas, principalmente de grande porte, em que a fatoração é densa.

### 3 Forma Padrão dos Problemas que a Família de Algoritmos Simples Resolve

A família de algoritmos simples [Silva, 2009] resolve o problema de encontrar uma solução factível do conjunto de restrições lineares:

$$\begin{aligned} P\bar{x} &= 0, \\ e^t \bar{x} &= 1, \\ \bar{x} &\geq 0, \end{aligned} \quad (5)$$

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

Todo problema de programação linear pode ser reduzido ao problema (5), ver [Silva, 2009] suplemento online. Quando reduzimos o problema de programação linear ao problema (5) a estrutura da matriz  $P$  torna-se bastante esparsa.

O algoritmo geral da família de algoritmos simples, resolve o problema (5) da seguinte maneira

Dado:  $e^t \bar{x}^0 = 1$ ,  $\bar{x}^0 \geq 0$ , arbitrário (por exemplo  $\bar{x}_j = \frac{1}{n}$ ).

Calcule  $rr^0 = P\bar{x}^0$ .

Para  $k=1, 2, \dots$  faça

[1] Calcule

$\{P_{\eta_1^+}, \dots, P_{\eta_{s_1}^+}\}$  que fazem o maior ângulo com o vetor  $rr^{k-1}$ .

$\{P_{\eta_1^-}, \dots, P_{\eta_{s_2}^-}\}$  que fazem o menor ângulo com o vetor  $rr^{k-1}$ ,

e tal que  $\bar{x}_i^{k-1} > 0, i = \eta_1^-, \dots, \eta_{s_2}^-$ , onde  $s_1 + s_2 = p$ .

$v_{k-1} = \text{máximo}_{i=1, \dots, s_1} P_{\eta_i^+}^t rr^{k-1}$ .

[2] Se  $v_{k-1} > 0$ , então PARE; o problema(5) é infactível.

[3] Resolva o subproblema

$$\begin{aligned} &\text{minimizar } \frac{1}{2} \|W\lambda\|^2 \\ &\text{s.a } a^t \lambda = 1, \\ &\quad -\lambda \leq 0, \end{aligned} \quad (6)$$

$$\begin{aligned} W &= \left[ \bar{w} P_{\eta_1^+} \dots P_{\eta_{s_1}^+} P_{\eta_1^-} \dots P_{\eta_{s_2}^-} \right], \\ \bar{w} &= rr^{k-1} - \sum_{i=1}^{s_1} \bar{x}_{\eta_i^+}^{k-1} P_{\eta_i^+} - \sum_{j=1}^{s_2} \bar{x}_{\eta_j^-}^{k-1} P_{\eta_j^-}, \\ \lambda &= \left( \lambda_0, \lambda_{\eta_1^+}, \dots, \lambda_{\eta_{s_1}^+}, \dots, \lambda_{\eta_1^-}, \dots, \lambda_{\eta_{s_2}^-} \right), \\ a &= (a_1, 1, \dots, 1), a_1 = 1 - \sum_{i=1}^{s_1} \bar{x}_{\eta_i^+}^{k-1} - \sum_{j=1}^{s_2} \bar{x}_{\eta_j^-}^{k-1}. \end{aligned} \quad (7)$$

[4] Atualize

$$\begin{aligned} rr^k &= \lambda_0 \left( rr^{k-1} - \sum_{i=1}^{s_1} \bar{x}_{\eta_i^+}^{k-1} P_{\eta_i^+} - \sum_{j=1}^{s_2} \bar{x}_{\eta_j^-}^{k-1} P_{\eta_j^-} \right) \\ &\quad + \sum_{i=1}^{s_1} \lambda_{\eta_i^+} P_{\eta_i^+} + \sum_{j=1}^{s_2} \lambda_{\eta_j^-} P_{\eta_j^-}, \\ \bar{x}_j^k &= \begin{cases} \lambda_0 \bar{x}_j^{k-1}, & j \notin \{\eta_1^+, \dots, \eta_{s_1}^+, \eta_1^-, \dots, \eta_{s_2}^-\}, \\ \lambda_{\eta_i^+}, & j = \eta_i^+, i = 1, \dots, s_1, \\ \lambda_{\eta_j^-}, & j = \eta_j^-, j = 1, \dots, s_2. \end{cases} \\ k &= k + 1. \end{aligned}$$

O critério de parada usado para o algoritmo, é quando  $\|rr^k - rr^{k-1}\|/\|rr^{k-1}\|$  é menor que uma certa tolerância especificada. No passo [2] do algoritmo, se  $v_{k-1} > 0$  então todas as colunas  $P_j$  da matriz  $P$ , estão do mesmo lado do hiperplano que passa pela origem e é perpendicular ao vetor  $rr^{k-1}$ . Isto significa que não podemos encontrar nenhuma combinação convexa das colunas da matriz  $P$ , resultando na origem. Neste caso, o problema é infactível.

Resolvemos o subproblema usando métodos de pontos interiores. As equações KKT do problema (6) são dadas por

$$\begin{aligned} W^t W \lambda + a l - \mu &= 0 \\ \mu^t \lambda &= 0 \\ a^t \lambda - 1 &= 0, \end{aligned} \tag{8}$$

com  $0 \leq \mu, \lambda$ , onde  $l$  e  $\mu$  são os multiplicadores de Lagrange de igualdade e desigualdade respectivamente e a matriz  $W^t W$  é de ordem  $(p+1) \times (p+1)$ . Estas são as equações onde aplicamos o método tradicional de pontos interiores seguidor de caminhos.

Em [Silva, 2009], provou-se que o desempenho do novo método é superior ao algoritmo de von Neumann. Além disso, demonstrou-se que se  $p_2 > p_1$ , então o algoritmo de ajustamento ótimo para  $p_2$  coordenadas tem um desempenho superior ao algoritmo de ajustamento ótimo para  $p_1$  coordenadas. Por outro lado, não é aconselhável escolher um valor muito grande de  $p$  uma vez que existe um custo de construção e atualização da matriz  $W$  em (8). Tal custo é insignificante para valores pequenos de  $p$ . No entanto, torna-se perceptível para valores maiores.

## 4 Nova Heurística

Para que a família de algoritmos trabalhe de maneira eficiente a escolha do parâmetro  $p$  é essencial. Em [Ghidini et al., 2012] é proposta uma heurística para a escolha de  $p$  levando em consideração o tamanho do problema, no entanto, ela foi construída baseado em experimentos numéricos. Nossa proposta é que o parâmetro  $p$  seja uma função do número de linhas e colunas da matriz do problema de programação linear a ser resolvido. Em cada iteração do método de pontos interiores precisamos resolver um sistema linear com uma matriz definida positiva de ordem  $n$  e quando resolvemos este sistema por Cholesky o custo computacional é  $O(n^2)$ . Então propomos uma função para  $p$  que nos forneça uma percentagem fixa da média geométrica entre o número de linhas e colunas da matriz de restrição do problema primal, ou seja,  $p = c\sqrt{m.n}$ . Queremos que o valor de  $p$  cresça de maneira mais suave que o custo computacional  $O(n^2)$  para resolver o sistema linear. A constante  $c$  é uma constante de ajuste e em todos os nossos experimentos essa constante é tomada como 2%. Este valor foi determinado por experimentos.

## 5 Resultados Computacionais

O principal objetivo dos nossos experimentos computacionais é comparar o desempenho de uma versão do código PCx [6] que incorporou uma abordagem híbrida de pré-condicionamento e que foi adicionada o algoritmo de ajustamento ótimo para  $p$  coordenadas para ser utilizado na fase de troca dos pré-condicionadores com a heurística dado em [Ghidini et al., 2012], com a mesma versão adicionada a nova heurística proposta.

Ambas as versões do PCx são implementadas em C, exceto a fatoração controlada de Cholesky, que é implementada em Fortran. Utilizamos os compiladores gcc e gfortran.

Todos os experimentos foram realizados em uma máquina com processador Intel Core 2 Quad Q9550 2.83 GHz e com 4GB de RAM, com sistema operacional Linux.

## 5.1 Problemas Testes

Nos experimentos utilizamos um subconjunto de 23 problemas de programação linear dos 40 apresentados em [Ghidini et al., 2012]. O conjunto de problemas é dividido em 13 problemas da coleção Netlib [collection LP test sets, ] e 10 problemas da coleção QAPLIB [Burkard et al., 1991]. Estes problemas são apresentados na tabela 1 , onde o número de linhas e colunas são referentes aos problemas após a fase de pré-processamento.

Tabela 1: Problemas Testes

Problema	Linha	Coluna	Coleção	Problema	Linha	Coluna	Coleção
25fv47	788	1843	NETLIB	stocfor2	1980	2868	NETLIB
bandm	240	395	NETLIB	els19	4350	13186	QAPLIB
blend	71	111	NETLIB	chr25a	8149	15325	QAPLIB
boeing2	125	264	NETLIB	chr22b	5587	10417	QAPLIB
bore3d	81	138	NETLIB	nug05	210	225	QAPLIB
degen2	444	757	NETLIB	nug06	372	486	QAPLIB
degen3	1503	2604	NETLIB	nug07	602	931	QAPLIB
etamacro	334	669	NETLIB	nug08	912	1632	QAPLIB
forplan	121	447	NETLIB	nug12	3192	8856	QAPLIB
israel	174	316	NETLIB	scr15	2234	6210	QAPLIB
kb2	43	68	NETLIB	scr20	5079	15980	QAPLIB
maros	655	1437	NETLIB				

## 5.2 Critério de Parada

O número de iterações do algoritmo de ajustamento ótimo para  $p$  coordenadas a ser rodadas é um importante parâmetro a ser determinado, uma vez que influencia diretamente o desempenho do PCx. Este algoritmo alcança melhores resultados quando a solução é determinada com boa precisão. No entanto, em alguns casos, o número de iterações necessárias para a convergência do algoritmo de ajustamento ótimo para  $p$  coordenadas pode ser muito grande, tornando-se impraticável seu uso. Assim, devemos estabelecer um número máximo de iterações. Nosso critério de parada para o algoritmo de ajustamento ótimo para  $p$  coordenadas é o seguinte: número máximo de iterações (100) ou o erro relativo da norma residual menor do que um dada tolerância ( $10^{-4}$ )(O que ocorrer primeiro).

## 5.3 Resultados

Na Tabela 2 comparamos o número total de iterações de ponto interior (coluna : iterações ) e o tempo total em segundos ( coluna : Tempo ) das duas versões do PCx com a abordagem híbrida de pré-condicionadores e com as duas heurísticas . PCxMod-NH é a versão que usa o algoritmo de ajustamento ótimo para  $p$  coordenadas na troca de pré-condicionadores com a nova heurística e PCxMod-H é a versão que usa o algoritmo de ajustamento ótimo para  $p$  coordenadas na troca de pré-condicionadores com a heurística adotada em [Ghidini et al., 2012]. Coluna: p-NH mostra o valor de  $p$  usado pelo algoritmo de ajustamento ótimo com a nova heurística. Coluna: p-H mostra o valor de  $p$  usado pelo algoritmo de ajustamento ótimo com heurística dada em [Ghidini et al., 2012]. Coluna:



ItAuxNH mostra o número de iterações realizadas pelo algoritmo de ajustamento ótimo para p coordenadas com a nova heurística. ItAuxH mostra o número de iterações realizadas pelo algoritmo de ajustamento ótimo para p coordenadas com a heurística dada em [Ghidini et al., 2012]. Os valores em negrito na Tabelas 3, mostram o menor número de iteração, ou o menor tempo de execução entre as duas abordagens .

Analisando a Tabela 2 temos que em 86,9% dos problemas testados, o PCxMod-NH ganha em termos de tempo de execução em obter a solução ótima. Também temos que o número total de iterações foi reduzida em 30,4% dos casos. Embora o número total de iterações não diminuiu em vários problemas usando a nova heurística, o ponto melhorado reduziu o tempo de execução mostrando que a nova heurística oferece um ponto melhor. Na Tabela 3, podemos ver que existem problemas tais como bandm, degen2, degen3, ch25a, chr22b, scr15 entre outros, onde o número de iterações foi maior ou igual para a versão PCxMod-NH, mas o tempo total de execução foi menor. Isso acontece porque o número de iterações do gradiente conjugado realizada durante a resolução desses problemas é menor. Em outras palavras, utilizando o ponto melhorado dado pela nova heurística na troca de pré-condicionadores, produzimos sistemas lineares mais fáceis de resolver.

Problema	P-Nh	P-H	ItAuxNh	ItAux-H	Iterações		Tempo	
					PCxmod-Nh	PCxmod-H	PCxmod-Nh	PCxmod-H
25fv47	3	2	100	100	27	38	3.35	11.06
bandm	2	4	100	100	17	17	0.17	0.26
blend	2	4	100	100	12	13	0.02	0.03
boeing2	2	4	5	11	15	15	0.04	0.06
bore3d	2	4	100	100	34	34	0.17	0.35
degen2	2	4	100	100	12	12	0.43	0.65
degen3	4	4	100	100	16	16	13.30	18.83
etamacro	2	4	5	46	29	48	0.28	0.60
forplan	2	4	7	5	31	31	0.26	0.38
israel	2	4	3	9	22	22	0.40	0.59
kb2	2	4	7	100	13	15	0.01	0.02
maros	2	4	9	15	29	39	5.92	7.74
stocfor2	5	4	70	28	22	22	4.86	4.40
els19	16	8	64	100	40	49	277.28	395.08
chr25a	23	20	100	90	28	28	114.64	111.43
chr22b	16	8	100	100	33	33	49.68	67.81
nug05-	2	4	100	100	6	6	0.06	0.06
nug06	2	4	1	1	6	6	0.12	0.17
nug07	2	4	100	100	11	11	0.64	0.84
nug08	3	4	1	1	10	10	1.52	2.09
nug12	10	8	1	1	20	20	151.71	199.82
scr15	8	4	22	20	25	24	21.78	28.11
scr20	19	20	100	100	28	29	329.68	351.20



## 6 Conclusões

Neste trabalho, apresentamos uma nova heurística para o parâmetro  $p$  de uma família de algoritmos simples. As principais vantagens dessa família são simplicidade e convergência inicial rápida. Esta família de algoritmos foi utilizada para acelerar a convergência de métodos de pontos interiores em [Ghidini et al., 2012], produzindo resultados significativos. No entanto, para que a família de algoritmos trabalhe de maneira eficiente a escolha do parâmetro  $p$  é essencial. Em [Ghidini et al., 2012] é utilizado uma heurística baseada em experimentos numéricos. Nossa proposta é que uma nova heurística para o parâmetro seja  $p = c\sqrt{m \cdot n}$ , onde  $c$  é uma constante de ajuste e tomamos esta constante com 2%, em nossos experimentos. Mostramos que nossa heurística é superior a heurística apresentada em [Ghidini et al., 2012] para um subconjunto de 23 dos problemas apresentado também em [Ghidini et al., 2012]. A nova heurística obteve em 86,9% desses problemas um menor tempo de execução para obter a solução ótima. Também obteve em 30,4% dos problemas uma diminuição no número de iterações do método de pontos interiores.

## Referências

- [Adler et al., 1989] Adler, I., Resende, M. G. C., Veiga, G., and Karmarkar, N. (1989). An implementation of Karmarkar's algorithm for linear programming. *Mathematical Programming*, 44:297–335.
- [Bergamaschi et al., 2007] Bergamaschi, L., Gondzio, J., Venturin, M., and Zilli, G. (2007). Inexact constraint preconditioners for linear systems arising in interior point methods. *Computational Optimization and Applications*, 36(1–2):137–147.
- [Bocanegra et al., 2007] Bocanegra, S., Campos, F. F., and Oliveira, A. R. L. (2007). Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods. *Computational Optimization and Applications*, 36(1–2):149–164. Special issue on “Linear Algebra Issues arising in Interior Point Methods”.
- [Burkard et al., 1991] Burkard, R. S., Karisch, S., and Rendl, F. (1991). Qaplib - a quadratic assignment problem library. *European Journal of Operations Research*, pages 55:115–119.
- [Campos and Birkett, 1998] Campos, F. F. and Birkett, N. R. C. (1998). An efficient solver for multi-right hand side linear systems based on the CCCG( $\eta$ ) method with applications to implicit time-dependent partial differential equations. *SIAM J. Sci. Comput.*, 19(1):126–138.
- [Chai and Toh, 2007] Chai, J. S. and Toh, K. C. (2007). Preconditioning and iterative solution of symmetric indefinite linear systems arising from interior point methods for linear programming. *Computational Optimization and Applications*, 36(1–2):221–247.
- [collection LP test sets, ] collection LP test sets, N. Netlib lp repository. *Online at <http://www.netlib.org/lp/data>*.
- [Czyzyk et al., 1999] Czyzyk, J., Mehrotra, S., Wagner, M., and Wright, S. J. (1999). PCx an interior point code for linear programming. *Optimization Methods & Software*, 11-2(1-4):397–430.
- [Duff and Meurant, 1989] Duff, I. S. and Meurant, G. A. (1989). The effect of ordering on preconditioned conjugate gradients. *BIT*, 29(4):635–657.

- [Ghidini et al., 2012] Ghidini, C. T. L. S., Oliveira, A. R. L., Silva, J., and Velazco, M. (2012). Combining a hybrid preconditioner and a optimal adjustment algorithm to accelerate the convergence of interior point methods. *Linear Algebra and its Applications*, 436:1–18.
- [Gonçalves et al., 2009] Gonçalves, J. P. M., Storer, R. H., and Gondzio, J. (2009). A family of linear programming algorithms based on an algorithm by von neumann. *Optimization Methods and Software*.
- [Gondzio, 1996] Gondzio, J. (1996). Multiple centrality corrections in a primal-dual method for linear programming. *Computational Optimization and Applications*, 6:137–156.
- [Lustig et al., 1992] Lustig, I. J., Marsten, R. E., and Shanno, D. F. (1992). On implementing Mehrotra’s predictor-corrector interior point method for linear programming. *SIAM Journal on Optimization*, 2:435–449.
- [Mehrotra, 1992] Mehrotra, S. (1992). On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601.
- [Monteiro et al., 1990] Monteiro, R. D. C., Adler, I., and Resende, M. G. C. (1990). A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension. *Mathematics of Operations Research*, 15:191–214.
- [Oliveira and Lyra, 1991] Oliveira, A. R. L. and Lyra, C. (1991). Implementação de um método de pontos interiores para programação linear. *SBA: Controle & Automação*, 3(2):370–382.
- [Oliveira and Sorensen, 2005] Oliveira, A. R. L. and Sorensen, D. C. (2005). 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.
- [Silva, 2009] Silva, J. (2009). *Uma Família de Algoritmos para Programação Linear Baseada no Algoritmo de Von Neumann*. PhD thesis, IMECC – UNICAMP, Campinas SP.
- [Wright, 1996] Wright, S. J. (1996). *Primal–Dual Interior–Point Methods*. SIAM Publications, SIAM, Philadelphia, PA, USA.