



IMPACTO DA REORDENAÇÃO DE MATRIZES POR NESTED DISSECTION NO DESEMPENHO DO MÉTODO PREDITOR-CORRETOR DE PONTOS INTERIORES

Wellington Barbosa Rodrigues

Faculdade Campo Limpo Paulista, Faccamp,
Campo Limpo Paulista, SP
wellington.rodrigues@sumare.edu.br

Marta Velazco

Faculdade Campo Limpo Paulista, Faccamp,
Campo Limpo Paulista, SP
marta.velazco@gmail.com

Aurelio Ribeiro leite de Oliveira

Instituto de Matemática, Estatística e Computação Científica, IMECC
UNICAMP, Campinas, SP
aurelio@ime.unicamp.br

RESUMO

O cálculo das direções de busca no método preditor-corretor de pontos interiores requer a solução de dois sistemas lineares com a mesma matriz dos coeficientes. Resultados anteriores mostram que a reordenação desta matriz pelo mínimo grau traz vantagens no cálculo das direções pelo método do gradiente conjugado preconditionado quando utilizado um preconditionador obtido pela fatoração incompleta de Cholesky. Este trabalho investiga o impacto da reordenação da matriz dos coeficientes pelo método nested dissection no desempenho do método preditor-corretor.

PALAVRAS CHAVE. Método preditor-corretor. Gradiente Conjugado Precondicionado. Reordenação. Nested Dissection.

Programação Matemática

ABSTRACT

The directions of the predictor-corrector method of interior are computed by solving two linear systems with the same matrix of coefficient. Previous results show that reordering of this matrix by the minimum degree improve the preconditioned conjugate gradient method when a hybrid preconditioner are used. This study investigates the impact of reordering by the nested dissection of coefficient matrix on the performance of the predictor-corrector method.

KEYWORDS. Predictor-Corrector Method. Preconditioned Conjugate Gradient Method. Reorder. Nested Dissection.

Mathematical Programming



1. Introdução

O método primal-dual preditor-corretor calcula as direções de busca por meio da solução de dois sistemas lineares com a mesma matriz dos coeficientes [Wright, 1997]. Estes sistemas podem ser resolvidos pelo método dos gradientes conjugados. A matriz dos coeficientes é mal condicionada o que leva à necessidade do condicionamento da mesma. O condicionador híbrido apresentado em [Bocanegra et al., 2007] foi criado especificamente para as matrizes dos métodos de pontos interiores. O condicionamento desta matriz varia muito durante o processo iterativo até se tornar muito mal condicionada no final do processo de otimização. O condicionador híbrido assume que o processo de otimização ocorre em duas fases. Na primeira fase em que as matrizes não são mal condicionadas é usado um condicionador obtido pela fatoração incompleta de Cholesky com preenchimento controlado. Na segunda fase é utilizado o condicionador separador, específico para as matrizes mal condicionadas dos métodos de pontos interiores. Uma heurística determina a troca fases. Em [Silva et al., 2017], os autores mostraram que a reordenação da matriz dos coeficientes pelo mínimo grau (MG) melhora a qualidade do condicionador híbrido. Neste trabalho, o método de reordenação nested dissection (ND) é utilizado na reordenação e o seu desempenho é avaliado por testes numéricos através da comparação com o desempenho do MG.

Este trabalho está organizado em sete seções descritas a seguir. A seguinte seção descreve o método de pontos interiores preditor-corretor para a solução de problemas de otimização linear. Seguidamente, são apresentados os sistemas lineares resolvidos para calcular as direções, adicionalmente é descrito o condicionador híbrido utilizado no método dos gradientes conjugados condicionado. Seguem os dois métodos de reordenação que serão avaliados: MG e ND. A seção 5 apresenta os resultados obtidos nos testes numéricos com problemas de bibliotecas de domínio público. Finalmente, as conclusões são discutidas.

2. Método preditor-corretor

Considere o problema de programação linear na forma padrão do primal

$$\begin{aligned} &\text{minimizar} && c^t x \\ &\text{s.a} && Ax = b, \\ &&& x \geq 0, \end{aligned} \quad (1)$$

onde $A \in \mathbb{R}^{m \times n}$ tem posto completo, $c, x \in \mathbb{R}^n$ e $b \in \mathbb{R}^m$. Associado ao problema primal é definido o problema dual como segue:

$$\begin{aligned} &\text{maximizar} && b^t y \\ &\text{s.a} && A^t y + z = c, \\ &&& z \geq 0, \end{aligned} \quad (2)$$

em que $y \in \mathbb{R}^m$ é o vetor das variáveis duais e $z \in \mathbb{R}^n$ é o vetor das variáveis de folga. As condições de Karush-Kuhn-Tucker, descritas a seguir garantem a otimalidade das soluções dos problemas (1) e (2):

$$\begin{aligned} Ax - b &= 0, \\ A^t y + z - c &= 0, \\ XZe &= 0, \\ (x, z) &\geq 0, \end{aligned} \quad (3)$$

onde $X = \text{diag}(x)$, $Z = \text{diag}(z)$, e $e \in \mathbb{R}^n$ é um vetor com componentes iguais a um. O método primal-dual preditor-corretor é obtido através da aplicação do método de Newton ao sistema não-linear formado pelas condições de Karush-Kuhn-Tucker (3) do problema de programação linear desconsiderando as condições de não negatividade. A direção de busca é formada por duas componentes: direção preditora a direção corretora, obtidas através da solução de dois sistemas lineares com a mesma matriz dos coeficientes.



3. Sistemas lineares

O passo de maior custo computacional nos métodos de pontos interiores é o cálculo das direções de busca. Os sistemas lineares que calculam estas direções no primal-dual preditor-corretor são de forma geral:

$$(A\Theta A^t)\Delta y = r, \quad (4)$$

onde $\Theta = Z^{-1}X$ e $X = \text{diag}(x)$, $Z = \text{diag}(z)$.

O sistema (4) é um sistema de equações normais. A matriz A tem posto completo e a matriz Θ é diagonal com todos os elementos positivos, consequentemente a matriz dos coeficientes $A\Theta A^t$ de dimensão m é simétrica e definida positiva. O condicionamento da matriz dos coeficientes varia muito; ela se torna cada vez mais mal condicionada à medida que o método de pontos interiores se aproxima de uma solução ótima. Isto é provocado pela matriz Θ . Os elementos da diagonal são calculados como: $\Theta_i = x_i^k/z_i^k$. Pela condição de complementariedade (condição 3 da Equação (3)), perto da solução ótima, os valores Θ_i serão muito grandes ou muito pequenos tornando a matriz dos coeficientes mal condicionada.

Estes sistemas podem ser resolvidos por métodos diretos ou por métodos iterativos. Na solução por métodos iterativos, o método do gradiente conjugado é o apropriado. O mal condicionamento da matriz dos coeficientes pode provocar problemas de convergência no gradiente conjugado pelo que é necessário o uso de um preconditionador. O preconditionador híbrido [Bocanegra et al., 2007], desenvolvido especificamente para a matriz dos sistemas lineares dos métodos de pontos interiores será utilizado neste estudo.

3.1. Preconditionador híbrido

O preconditionador híbrido foi proposto por [Bocanegra et al., 2007]. Os autores consideram que o processo de otimização ocorre em duas fases e preconditionadores diferentes são usados em cada uma. Na fase inicial, em que a matriz dos coeficientes não é muito mal condicionada é usado o preconditionador fatoração controlada de Cholesky (FCC) proposto por [Campos e Birkett, 1998], obtido a partir da fatoração incompleta de Cholesky da matriz dos coeficientes com preenchimento controlado. Na segunda fase, em que a matriz se torna mal condicionada é usado o preconditionador separador proposto por [Oliveira, 2005], específico para as matrizes mal condicionadas oriundas dos métodos de pontos interiores. Uma heurística determina a troca de fase de preconditionadores [Velazco et al., 2010].

3.2. Fatoração Controlada de Cholesky

A fatoração controlada de Cholesky (FCC) [Campos e Birkett, 1998] é uma fatoração incompleta de Cholesky em que o preenchimento do fator é controlado por um parâmetro η que depende do espaço de memória disponível.

Suponha a matriz dos coeficientes $A\Theta A^t$ fatorada como mostra a Equação (5):

$$(A\Theta A^t) = LL^t = \tilde{L}\tilde{L}^t + R \quad (5)$$

sendo:

L = Fator obtido pela fatoração de Cholesky;

\tilde{L} = Fator obtido pela fatoração incompleta de Cholesky;

R = Matriz de resíduo.

Define-se $E = L - \tilde{L}$. A FCC é baseada na minimização da norma de Frobenius de E , pois quando $\|E\| \approx 0$, implicará $\|R\| \approx 0$;

$$\text{minimizar} \|E\|_F^2 = \sum_{k=1}^{m_j+\eta} |l_{i,k,j} - \tilde{l}_{i,k,j}|^2 + \sum_{k=m_j+\eta+1}^m |l_{i,k,j}|^2 \quad (6)$$



sendo que m_j representa a quantidade de elementos não nulos abaixo da diagonal da coluna j da matriz original $A\Theta A^t$ e η representa o número de elementos não nulos que podem ser adicionados a cada coluna durante a FCC. No primeiro somatório, estão os $m_j + \eta$ elementos não nulos da j -ésima coluna de \tilde{L} . No segundo somatório, estão apenas os elementos de L que não estão em \tilde{L} . Dessa forma, o problema de minimização de E pode ser resolvido usando as heurísticas:

- Aumentar o parâmetro η , permitindo maior preenchimento por coluna. Consequentemente, o primeiro somatório contém mais elementos e o valor total diminui.
- Escolher os $m_j + \eta$ maiores elementos de \tilde{L} em valor absoluto para um η fixo. Assim, os maiores elementos ficam no primeiro somatório e os menores no segundo, produzindo um fator \tilde{L} mais próximo ao original para a quantidade determinada de armazenamento.

O preenchimento inicial η depende da média da quantidade de elementos não nulos da matriz $A\Theta A^t$.

3.3. Precondicionador Separador

O preconditionador separador foi proposto por [Oliveira, 2005] e neste trabalho foi utilizado uma versão para a matriz $A\Theta A^t$. Considere a matriz $A = [B \ N]P$ ordenada por colunas a partir de uma matriz de permutação $P \in \mathbb{R}^{n \times n}$ tal que $B \in \mathbb{R}^{m \times m}$ é uma matriz não singular e $N \in \mathbb{R}^{m \times n-m}$ são as colunas restantes. Dessa forma, $A\Theta A^t$ pode ser representada como na Equação (7):

$$A\Theta A^t = [B \ N]P^t\Theta P \begin{bmatrix} B^t \\ N^t \end{bmatrix} = [B \ N] \begin{bmatrix} \Theta_B & 0 \\ 0 & \Theta_N \end{bmatrix} \begin{bmatrix} B^t \\ N^t \end{bmatrix} = B\Theta_B B^t + N\Theta_N N^t \quad (7)$$

A Equação (7) pode ser multiplicada por $\Theta_B^{-\frac{1}{2}}B^{-1}$ e pós multiplicada por $(\Theta_B^{-\frac{1}{2}}B^{-1})^t$ resultando na matriz preconditionada T , como mostra a Equação a seguir:

$$T = \Theta_B^{-\frac{1}{2}}B^{-1}(A\Theta A^t)B^{-t}\Theta_B^{-\frac{1}{2}} = I_m + WW^t,$$

sendo $W = \Theta_B^{-\frac{1}{2}}B^{-1}N\Theta_N^{-\frac{1}{2}} \in \mathbb{R}^{m \times n-m}$.

4. Métodos de reordenação

A reordenação de uma matriz esparsa reduz o efeito de preenchimento excessivo que ocorre durante o processo de fatoração na solução de um sistema linear [George et al.]. Obter a reordenação de uma matriz esparsa que provoque o menor preenchimento é um problema NP-Completo. Quando os sistemas são resolvidos pelo método dos gradientes conjugados preconditionado com o preconditionador FCC, o uso de métodos de reordenação permite calcular um preconditionador mais próximo do fator de Cholesky [Silva et al., 2017], mas os métodos de reordenação não tem influência direta na construção do preconditionador separador. Por outro lado, melhorar a eficiência do FCC implica melhorar a eficiência do preconditionador separador.

A matriz $A\Theta A^t$ pode ser reordenada pela permutação de suas linhas e colunas, isto é calculado através de uma matriz de permutação P . A matriz reordenada $PA\Theta A^tP^t$ terá as mesmas linhas e colunas que a matriz original, mas com a ordem modificada. A matriz de permutação P é obtida por uma heurística de ordenação. Neste trabalho apresentaremos duas heurísticas: mínimo grau e nested dissection.

A matriz $A\Theta A^t$ conserva o padrão de esparsidade durante toda a otimização. Isto é, a quantidade e a posição dos elementos não nulos serão as mesmas em todas as iterações do método de pontos interiores, pois embora a matriz Θ varie a cada iteração, todos seus elementos são positivos. Isto é uma vantagem no uso de método de reordenação; somente será necessária uma única reordenação da matriz antes do processo iterativo. Reordenar $A\Theta A^t$ é equivalente a reordenar A por linhas. Nos métodos avaliados neste trabalho, as matrizes são representadas em grafos em que os vértices são as linhas/colunas e as arestas serão os elementos não nulos desconsiderando os elementos da diagonal.



4.1. Mínimo Grau

O método escolhe o vértice de menor grau a cada iteração. O vértice escolhido é eliminado do grafo e arestas são adicionadas entre seus vizinhos não adjacentes. Adicionar novas arestas significa preenchimento adicional depois da fatoração [George e Liu, 1980].

4.2. Nested Dissection

Nesta heurística, o grafo é dividido em três subgrafos: C_1, C_2, S em que S será um grafo separador. Este processo é repetido recursivamente nos subgrafos até chegarem a um único vértice. A reordenação será: primeiro os vértices de C_1 seguindo os de C_2 e finalmente S . Existem duas estratégias para a escolha do grafo separador: por vértice e por arestas. Neste trabalho as duas estratégias foram avaliadas [George, 1973].

5. Resultados numéricos

O estudo será realizado através da comparação do desempenho do primal-dual preditor-corretor usando o MG e o ND com as duas estratégias na reordenação das matrizes. As implementações dos métodos de reordenação serão introduzidos no código PCx-Modificado [Czyzyk et al., 1999], [Bocanegra et al., 2007], [Velazco et al., 2010] e avaliadas em problemas testes. O código PCx-Modificado resolve problemas de programação linear pelo método primal-dual preditor-corretor e os sistemas são resolvido pelo gradiente conjugado preconditionado (GCP) com o preconditionador híbrido [Bocanegra et al., 2007], [Velazco et al., 2010]. Foram utilizadas três versões do PCx-Modificado:

- PCx-Modificado com MG (PCx-MG): utiliza o MG reativado do PCx original [Silva et al., 2017]
- PCx-Modificado com ND (PCx-NDA): utiliza o ND da biblioteca Metis com a estratégia por arestas [Karypis].
- PCx-Modificado com ND (PCx-NDV): utiliza o ND da biblioteca Metis com a estratégia por vértices [Karypis].

Os testes foram realizados em um processador Intel core I3 com 4Gb de memória e sistema operacional Linux Ubuntu 16.04. A comparação considerará: Tempo de reordenação, iterações do GCP, iterações e tempo do método primal-dual preditor-corretor, preenchimento do FCC e troca de fases.

5.1. Problemas testes

Foram utilizados 63 problemas, apresentados na Tabela 1 das bibliotecas: PDS¹, MNET-GEN², KENNINGTON³, QAPLIB⁴, MESZAROS⁵ e NETLIB⁶. Na Tabela, cada problema é caracterizado por: número que o identifica (**Ordem**), nome do problema (**Nome**), densidade da matriz de restrições (**Densidade %**), quantidade de elementos não nulos de $A\Theta A^t$ (**NZAAT**), dimensão da matriz $A\Theta A^t$ (**M**) e biblioteca de origem (**Biblioteca**). Os problemas são organizados considerando a densidade da matriz dos coeficientes, $A\Theta A^t$; dos menos densos aos mais densos. O tamanho e a densidade são referentes ao problema depois do pré-processamento pelo PCx-Modificado.

5.1.1. Tempo de reordenação

O método MG consumiu menos tempo de ordenação em apenas 15 problemas. O ND, usando ambas estratégias consumiu menos tempo para reordenação da matriz em 33 problemas e igualou em 15 problemas. A Tabela 2 mostra os tempos totais obtidos na reordenação da matriz dos 63 problemas.

¹<http://plato.asu.edu/ftp/lptestset/pds/>

²<http://www.di.unipi.it/optimize/Data/MMCF.html>

³<http://www.netlib.org/lp/data/kennington/>

⁴<http://anjos.mgi.polymtl.ca/qaplib/>

⁵http://www.sztaki.hu/~meszaros/public_ftp/lptestset/misc/

⁶<http://www.netlib.org/lp/data/>



Tabela 1: Problemas teste.

Ordem	Problema	Densidade %	NZAAT	M	Biblioteca
1	pds-100	0,0034	787355	152289	PDS
2	pds-90	0,0038	740086	139741	PDS
3	pds-80	0,0043	676093	126109	PDS
4	ken-18	0,0047	289462	78538	KENNINGTON
5	512-256-2	0,0064	705576	104931	MNETGEN
6	512-256-1	0,0066	713835	104124	MNETGEN
7	ken-13	0,0155	77459	22365	KENNINGTON
8	256-256-9	0,0158	674759	65280	MNETGEN
9	256-256-8	0,0162	693445	65328	MNETGEN
10	512-128-11	0,0221	1037581	68462	MNETGEN
11	256-256-10	0,0223	989269	66606	MNETGEN
12	256-256-12	0,0224	988399	66456	MNETGEN
13	256-256-11	0,0226	983752	66024	MNETGEN
14	nemsem2	0,1873	38359	4526	MESZAROS
15	ste36a	0,2041	1564487	27683	QAPLIB
16	cre-a	0,2410	21527	2989	KENNINGTON
17	stocfor2	0,3032	11888	1980	NETLIB
18	pcb3000	0,3076	45639	3852	MESZAROS
19	cre-c	0,3138	17625	2370	KENNINGTON
20	chr25a	0,3755	249324	8149	QAPLIB
21	progas	0,6262	12449	1410	MESZAROS
22	scr20	0,6463	166709	5079	QAPLIB
23	orna1	0,6570	5111	882	MESZAROS
24	orna2	0,6570	5111	882	MESZAROS
25	orna7	0,6570	5111	882	MESZAROS
26	rou20	0,6586	356689	7359	QAPLIB
27	ganges	0,6914	8565	1113	NETLIB
28	nug12	0,7239	56508	2794	QAPLIB
29	els19	0,7284	137825	4350	QAPLIB
30	qap12	0,7708	60174	2794	NETLIB
31	nemscem	0,7919	1817	479	MESZAROS
32	pcb1000	0,8452	17211	1427	MESZAROS
33	nemspnm1	1,0153	50356	2227	MESZAROS
34	scfxm3	1,0272	8600	915	NETLIB
35	nesm	1,1052	4727	654	NETLIB
36	nemspnm2	1,1324	49040	2081	MESZAROS
37	scr15	1,1824	59009	2234	QAPLIB
38	bn1	1,3026	4847	610	NETLIB
39	ship12l	1,4722	5478	610	NETLIB
40	scfxm2	1,5391	5727	610	NETLIB
41	qap08	1,6461	9063	742	NETLIB
42	nug08	1,6841	9272	742	QAPLIB
43	maros	1,8745	8042	655	NETLIB
44	25fv47	1,8749	11642	788	NETLIB
45	nug07	2,1894	4919	474	QAPLIB
46	etamacro	2,2339	2492	334	NETLIB
47	degen3	2,2919	51637	1501	NETLIB
48	model4	2,6386	38887	1214	MESZAROS
49	nug06	3,0587	2398	280	QAPLIB
50	ship04l	3,0634	2612	292	NETLIB
51	scfxm1	3,0680	2854	305	NETLIB
52	boeing1	3,8435	4211	331	NETLIB
53	capri	3,9290	2282	241	NETLIB
54	e226	6,4432	2526	198	NETLIB
55	recipe	6,7627	277	64	NETLIB
56	share1b	7,7089	967	112	NETLIB
57	boeing2	8,7050	1382	126	NETLIB
58	blend	14,2630	719	71	NETLIB
59	forplan	18,0042	2636	121	NETLIB
60	t0331-4l	22,4926	99169	664	MESZAROS
61	kb2	24,0671	445	43	NETLIB
62	nw14	36,4984	1945	73	MESZAROS
63	israel	37,0822	11227	174	NETLIB

Tabela 2: Tempo total de reordenação usando as três heurística

Heurística	Tempo Total (s)
MG	6617,628
NDA	15,028
NDV	16,108



Tabela 3: Tempo de reordenação nos problemas com densidade menor que 0,21%

Ordem	Tempo Ordenação - NDA (s)	Tempo Ordenação - NDV (s)	Tempo Ordenação - MG (s)
1	3,124	3,292	323,412
2	2,936	3,104	314,584
3	2,692	2,888	409,696
4	0,708	0,772	0,664
5	1,480	1,656	694,496
6	1,436	1,524	602,708
7	0,200	0,236	0,088
8	0	0	556,572
9	0	0,004	606,088
10	0	0	743,980
11	0,004	0	786,508
12	0	0,004	683,036
13	0	0	863,108
14	0	0	0,012
15	1,400	1,456	29,240
Total	13,98	14,938	6614,192

A diferença no tempo de reordenação é mais significativa nos problemas com matrizes mais esparsas, especificamente nos primeiros 15 problemas. À medida que aumenta a densidade da matriz dos coeficientes $A\Theta A^t$, o desempenho do MG melhora. A Tabela 3 mostra os tempos de reordenação obtidos nos problemas com densidade menor que 0,21%.

Nos 15 problemas em que o MG consumiu menos tempo de reordenação a redução foi de apenas 0,228 segundos comparado com o NDA e 0,4 segundos comparado com o NDV. Este resultado é confirmado no perfil de desempenho [Dolan, 2002] mostrado na Figura 1 em que o tempo de reordenação é considerado como métrica.

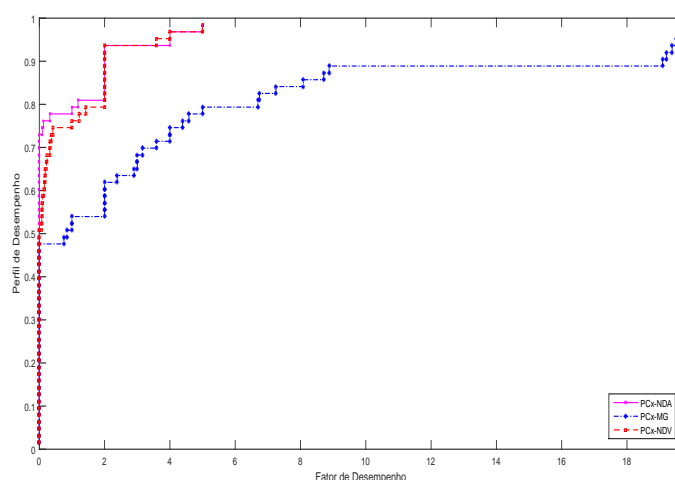


Figura 1: Perfil de desempenho das reordenações MG e ND.



5.1.2. Gradiente Conjugado e Métodos de Pontos Interiores

A Tabela 4 mostra para cada uma das versões do PCx-Modificado: o número total de iterações do GCP (**GCP-NDA**, **GCP-NDV**, **GCP-MG**) para solucionar os dois sistemas lineares e o número de iterações (**PCx-NDA(i)**, **PCx-NDV(i)**, **PCx-MG(i)**) e tempo (**PCx-NDA(s)**, **PCx-NDV(s)**, **PCx-MG(s)**) do método primal-dual preditor-corretor. O GCP teve melhor desempenho quando usado o PCx-NDV. Com o PCx-NDV foram necessárias 860256 iterações do GCP (14103 iterações em media por problema), com o PCx-NDA foram necessárias 930589 iterações (15256 iterações em media por problema) e com o PCx-MG, 902366 iterações (14792 iterações em media por problema). O PCx-MG realizou quase 5% a mais de iterações. Apesar do aumento no número de iterações com o PCx-MG, o desempenho foi melhor em 40 problemas.

As três versões do PCx utilizadas não conseguiram a convergência em dois problemas cada uma. O tempo total do PCx-NDA foi de 64283,88 segundos, do PCx-NDV foi de 54489,18 segundos e do PCx-MG de 67806,45 segundos. Na comparação entre o PCx-NDV e o PCx-MG considerando o tempo de solução, o PCx-NDV diminuiu o tempo em 20 problemas, igualou em 9 e piorou em 31 problemas. Consequentemente, o desempenho do PCx-NDV foi pior apesar da redução do tempo total. O PCx-NDV resolveu os 61 problemas em menos iterações do método primal-dual preditor-corretor. Teve melhor desempenho em 13 problemas e igualou em 39 enquanto que o PCx-MG diminuiu em apenas 9 problemas.

5.1.3. Preenchimento do FCC

Quando os sistemas são resolvidos pelo GCP com um preconditionador obtido a partir de uma fatoração incompleta, a reordenação ajuda na convergência do GCP [Silva et al., 2017]. O reordenamento da matriz permite criar um preconditionador FCC onde a norma de E (Equação (6)) será menor. O parâmetro η determina o preenchimento a mais que será permitido por coluna no preconditionador FCC. A Tabela 5 mostra a quantidade de elementos não nulos no preconditionador FCC depois do primeiro aumento de η usando o PCx-NDA (**NZFCC-NDA**), o PCx-NDV (**NZFCC-NDV**) e o PCx-MG (**NZFCC-MG**). O símbolo - significa que não houve aumento do preenchimento do preconditionador.

A Tabela 5 mostra que depois do primeiro aumento do valor de η o preenchimento do FCC foi menor usando o PCx-MG em 38 problemas comparado com o PCx-NDA e na mesma quantidade comparado com o PCx-NDV; pelo que a reordenação da matriz usando ND não foi melhor nesses casos. Maior preenchimento do FCC melhora a convergência do GCP, mas encarece a construção do mesmo. Para valores razoáveis de η , o cálculo do FCC é barato e requer menos memória e menos tempo computacional para a sua construção que o preconditionador separador.

5.1.4. Troca de fases

À medida que a matriz do sistema fica mal-condicionada o preconditionador FCC perde a eficiência aumentando a quantidade de iterações do GCP. Neste caso é feita a troca de fases e o sistema será preconditionado pelo preconditionador separador. Em alguns problemas não ocorre troca de preconditionadores e o FCC é utilizado durante todo o processo de otimização. O preconditionador separador foi construído para as matrizes mal condicionadas dos métodos de pontos interiores pelo que quando a troca ocorre muito cedo o preconditionador separador não terá um bom desempenho.

A iteração da troca de preconditionadores dá uma medida de desempenho do preconditionador FCC:

- Antecipação da Troca: O FCC não está funcionando bem e o GCP está realizando muitas iterações;
- Manteve a Troca: O desempenho do FCC é igual comparado com o PCx-MG;
- Atrasou a Troca: O FCC está preconditionando a matriz de forma eficiente.



Tabela 4: Total de iterações do GCP e tempo e total de iterações do método preditor-corretor.(*)-
O método não convergiu

Ordem	GCP-NDA(I)	PCx-NDA(I)	PCx-NDA(s)	GCP-NDV(I)	PCx-NDV(I)	PCx-NDV(s)	GCP-MG(I)	PCx-MG(I)	PCx-MG(s)
1	47004	85	2992,36	46124	85	3204,52	49285	88	4428,58
2	40420	80	2741,80	45791	81	2863,05	42984	81	3163,68
3	36905	83	2017,40	37927	84	2178,98	36944	83	2692,81
4	27978	39	2089,36	35399	39	1737,83	36458	41	2440,82
5	*	*	*	9595	48	2104,30	23674	49	3387,92
6	23766	53	1763,04	12862	57	2801,96	16021	54	3124,90
7	20461	29	267,29	19035	30	306,75	16019	30	194,90
8	51242	51	5753,34	51242	51	5681,44	40407	40	2772,14
9	41427	50	5124,75	37302	48	4100,46	31550	48	2940,90
10	71269	51	9384,48	70604	49	8833,64	64961	49	8519,54
11	73987	61	4790,90	56632	59	4035,28	75549	64	6898,15
12	61239	49	7975,22	61023	50	5831,64	52736	51	7335,32
13	63725	57	7344,32	63848	62	8689,06	59831	61	7481,93
14	7103	44	14,55	7103	44	14,58	5208	43	9,70
15	49099	38	9890,73	*	*	*	49010	38	10244,54
16	5017	27	3,24	4264	27	3,38	4484	28	3,22
17	6749	21	2,18	6977	21	2,38	4652	21	1,58
18	15062	28	64,70	15062	28	62,26	14483	28	38,34
19	5636	29	3,88	4906	28	2,92	5281	28	3,62
20	16089	28	87,64	18210	27	98,48	18603	28	160,54
21	5146	16	1,76	4863	16	1,71	4632	16	1,57
22	17705	21	102,27	16680	21	103,74	16819	21	99,70
23	2746	25	0,42	2518	25	0,41	2488	25	0,39
24	2355	19	0,36	2345	20	0,38	2151	19	0,35
25	3416	20	0,49	2963	19	0,45	2790	19	0,42
26	16056	24	1154,68	16014	24	1148,46	14910	24	1169,06
27	2946	19	1,36	2748	19	1,36	3115	19	1,69
28	22157	20	147,88	19525	20	126,56	19307	20	134,62
29	24941	18	87,77	23371	31	78,16	25248	31	78,76
30	19741	20	113,93	17933	20	119,71	17626	20	110,23
31	1916	22	0,16	1948	22	0,16	1526	22	0,13
32	7607	25	9,19	7733	25	8,91	6487	25	5,40
33	*	*	*	*	*	*	28697	54	34,51
34	2412	20	0,74	2403	20	0,75	3638	25	0,86
35	11712	32	3,10	11058	31	2,98	7818	33	2,38
36	23491	45	28,04	23491	45	28,40	22816	45	25,73
37	14797	24	16,55	13675	24	15,55	13837	24	15,74
38	16179	42	2,22	15697	42	2,12	4549	42	0,73
39	235	17	0,09	262	17	0,10	256	17	0,10
40	1387	20	0,34	1357	20	0,34	482	20	0,14
41	3856	10	1,46	4202	10	1,52	4030	10	1,47
42	2996	10	1,22	3051	9	1,08	3237	9	1,19
43	5305	20	1,11	5305	20	1,09	3767	20	0,66
44	5529	26	1,69	5553	26	1,69	5488	26	1,53
45	1979	11	0,42	1989	11	0,43	*	*	*
46	1131	30	0,21	1131	30	0,21	983	27	0,21
47	9497	16	11,47	9429	16	11,60	9364	16	7,92
48	20100	40	14,85	20100	40	14,92	12311	39	9,08
49	651	6	0,10	579	6	0,10	567	6	0,10
50	105	18	0,03	104	18	0,04	104	18	0,03
51	1051	17	0,14	1051	17	0,14	398	17	0,05
52	908	20	0,11	915	20	0,12	1034	20	0,13
53	823	20	0,08	937	19	0,09	577	20	0,06
54	1375	18	0,12	1375	18	0,12	1014	18	0,07
55	60	12	0,00	60	12	0,00	60	12	0,00
56	41	19	0,01	41	19	0,01	212	19	0,02
57	560	14	0,03	568	14	0,03	543	14	0,03
58	279	10	0,01	330	10	0,01	348	10	0,01
59	736	23	0,11	736	23	0,11	*	*	*
60	10171	42	252,03	9996	42	246,23	9402	42	244,80
61	199	13	0,00	200	13	0,00	218	13	0,00
62	986	48	16,14	986	48	15,93	735	47	13,14
63	1128	21	0,54	1128	21	0,55	642	21	0,31
TOTAL	930589	1816	64284,41	860256	1841	54489,18	902366	1898	67806,45



Tabela 5: Elementos não nulos do FCC depois do primeiro aumento do η .(-)- Não houve aumento do preenchimento.

Ordem	NZFCC-NDA	NZFCC-NDV	NZFCC-MG
1	-	-	-
2	-	-	-
3	-	-	-
4	630285	631175	590095
5	1309767	1308860	1258884
6	1336877	1337113	1285375
7	166743	167342	158215
8	674759	674759	674759
9	693445	693445	693445
10	672910	672910	659521
11	696787	696787	688593
12	696643	696643	688195
13	693356	693356	684666
14	78879	78879	-
15	996835	996829	996885
16	29906	29555	29743
17	23254	23718	22389
18	39107	39107	39792
19	25607	25659	25488
20	82145	82122	82088
21	24769	24850	24280
22	62633	62655	62632
23	6510	7857	7216
24	7893	7857	7216
25	7893	7857	7216
26	205754	205783	205722
27	21275	21275	17473
28	27099	27120	27122
29	48750	48729	48717
30	28060	28090	28120
31	2798	2821	2535
32	13585	13585	13671
33	22578	22578	22606
34	19857	19857	13458
35	11865	11776	11116
36	20924	20924	20962
37	22179	22203	22186
38	12265	12265	8521
39	-	-	-
40	13277	13277	8858
41	7226	7240	7211
42	7392	7397	7417
43	6429	6429	6411
44	8110	8110	8095
45	4919	4919	4919
46	-	-	5860
47	21155	21155	20940
48	14041	14041	14024
49	6206	6269	6114
50	-	-	-
51	6581	6581	4254
52	3381	3379	3381
53	4185	4202	3724
54	2024	2024	2019
55	-	-	-
56	-	-	1413
57	1382	1382	1382
58	719	719	719
59	1241	1241	1239
60	85566	85550	85572
61	445	445	445
62	748	748	742
63	7661	7661	7659



Tabela 6: Troca de fases.

Heurística	Adiantou	Manteve	Atrasou
PCx-NDA	16	33	7
PCx-NDV	18	39	3

A Tabela 6 mostra o número de problemas para os quais a troca de condicionador foi antecipada, foi mantida ou foi atrasada utilizando o ND com as duas estratégias. A comparação foi realizada com o PCx-MG.

5.1.5. Resumo dos resultados

O resultados obtidos são resumidos a seguir, separados por biblioteca de origem:

- **PDS:** Foram estudados três problemas grandes e muito esparsos (maior que 99,996%). O condicionador FCC funciona durante todo o processo de otimização. Nestes problemas, o ND ordenou em até 100 vezes mais rápido que o MG, diminuiu as iterações do GCP e diminuiu o tempo total do preditor-corretor.
- **MNETGEN:** Problemas com esparsidade muito alta (maior que 99,97% e menor que 99,9936%). Nestes problemas, o ND ordenou muito mais rápido que o MG, gerou um preenchimento maior no condicionador FCC e os resultados da quantidade de iterações do GCP e do tempo do preditor-corretor foram diversos.
- **KENNINGTON:** Foram estudados quatro problemas com esparsidade alta. Nestes problemas, o MG ordenou mais rápido que o ND.
- **QAPLIB:** Problemas menos esparsos, com esparsidade de até 97%. O ND ordenou a matriz dos coeficientes em menos tempo computacional e gerou um preenchimento similar ao MG. Na maioria dos problemas, o tempo computacional do método preditor-corretor foi diminuído.
- **MESZAROS:** Foram estudados 13 problemas com esparsidade variada, menor que 99,8%. O ND ordenou a maioria dos problemas em menor ou igual tempo computacional. O preenchimento obtido com o ND foi similar ao obtido com o MG. Em todos os problemas aumentaram as iterações do GCP e consequentemente o tempo total de otimização quando usada ordenação ND.
- **NETLIB:** Foram estudados 25 problemas com esparsidade variada, menor que 99,7%. O ND ordenou os problemas em menor ou igual tempo computacional e o preenchimento obtido foi maior ou igual ao obtido pelo MG. Os resultados obtidos em número de iterações do GCP, do preditor-corretor e do tempo de otimização foram similares.

6. Conclusões

Neste trabalho foi estudado o impacto da reordenação das matrizes oriundas do método preditor-corretor de pontos interiores pelo ND. Foram testadas as duas estratégias de busca do grafo separador no ND e o desempenho das heurísticas foi comparado com o desempenho do método MG em 63 problemas de seis bibliotecas de domínio público.

O tempo de reordenação das matrizes pelo ND foi mais rápido que o tempo do MG. Esta diferença é mais acentuada em problemas de maior porte e mais esparsos. Não entanto, a reordenação pelo ND provocou em geral maior preenchimento no condicionador FCC. O desempenho do GCP e o tempo computacional do método preditor-corretor tiveram resultados diferentes nas bibliotecas de problemas estudadas. O ND não teve um bom desempenho na biblioteca MESZAROS, mas parece ser o mais indicado nas bibliotecas PDS e QAPLIB.



Agradecimentos

Os autores agradecem à Faculdade Campo Limpo Paulista (Faccamp) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Referências

- Bocanegra, S., Campos, F., e Oliveira, A. (2007). Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods. *Computational Optimization and Applications*, 36:149–167.
- Campos, F. F. e Birkett, N. R. C. (1998). An efficient solver for multi-right-hand-side linear systems based on the cccg (?) method with applications to implicit time-dependent partial differential equations. *SIAM Journal on Scientific Computing*, 19(1):126–138.
- Czyzyk, J., Mehrotra, M., S. and Wagner, e Wright, S. (1999). Pcx an interior point code for linear programming. *Optimization Methods and Software*, 11(2):397–430.
- Dolan, J. J., E. D. and Moré (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91:201–213.
- George, A. (1973). Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363.
- George, A. e Liu, J. (1980). A fast implementation of the minimum degree algorithm using quotient graphs. *ACM Transactions on Mathematical Software*, 6(23):337–358.
- George, A., Liu, J., e Ng, E. Computer solution of sparse linear systems. http://web.engr.illinois.edu/~heath/courses/cs598mh/george_liu.pdf. Acessado: 2017.
- Karypis, G. Metis. a software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, version 5.0. <http://glaros.dtc.umn.edu/gkhome/metis/metis/download>. Acessado: 2016.
- Oliveira, D. C., A. R. L.; Sorensen (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, D., Velazco, M., e Oliveira, A. (2017). Influence of matrix reordering on the performance of iterative methods for solving linear systems arising from interior point methods for linear programming. *Mathematical Methods of Operations Research*, 85(1):97–112.
- Velazco, M., Oliveira, A., e Campos, F. (2010). A note on hybrid preconditioners for large-scale normal equations arising from interior point methods. *Optimization Methods and Software*, 25(2):321–332.
- Wright, S. (1997). *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, USA.