LII Simpósio Brasileiro de Pesquisa Operacional
João Pessoa - PB, 29 de setembro a 02 de outubro de 2020

Análise experimental de buscas locais para a construção de inversas reflexivas generalizadas esparsas

Gabriel Ponte
Curso de Engenharia de Computação e Informação
Universidade Federal do Rio de Janeiro
Centro de Tecnologia, Rio de Janeiro, CEP: 21941-909
gabrielponte@poli.ufrj.br

Trabalho de Iniciação Científica
Orientadores: Marcia Fampa (UFRJ), Jon Lee (University of Michigan)

RESUMO

Um grande desafio em diversas áreas científicas, incluindo a área da Pesquisa Operacional, é a resolução de sistemas lineares de grande porte, ou o cálculo de soluções de mínimos quadrados para sistemas inconsistentes. Uma característica importante para tornar este desafio possível é a esparsidade das matrizes envolvidas nos cálculos. A pseudo-inversa M-P (Moore-Penrose) pode ser utilizada para calcular soluções de mínimos quadrados de sistemas inconsistente. Entretanto, independente de uma matriz ser esparsa, sua pseudo-inversa pode ser densa, levando a um alto custo computacional nas suas aplicações. A pseudo-inversa M-P, é caracterizada por quatro propriedades, mas nem todas precisam ser atendidas para satisfazer mínimos quadrados. Assim, [Fampa,Lee(2018)] e [Xu,Fampa,Lee(2019)] propõem procedimentos de buscas locais para a construção de inversas generalizadas esparsas que satisfazem somente algumas dessas propriedades. Propomos abordagens de implementação para essas buscas, visando sua eficiência e escalabilidade. Implementamos e analisamos os algoritmos propostos, considerando experimentos com matrizes aleatórias diversificadas.

PALAVRAS CHAVE. Mínimos Quadrados, Busca local, Pseudo-Inversa de Moore-Penrose.

ABSTRACT

A major challenge in several areas, including the area of Operations Research, is the resolution of large linear systems, or the calculation of least squares solutions for inconsistent systems. An important characteristic to make this challenge possible is the sparsity of the matrices involved in the calculations. The pseudo-inverse M-P (Moore-Penrose) is used to calculate least square solutions of inconsistent systems. However, regardless of whether a matrix is sparse, its pseudo-inverse can be dense, leading to a high computational cost in its applications. The pseudo-inverse M-P is characterized by four properties, but not all need to be met to satisfy least squares. Thus, [Fampa,Lee (2018)], [Xu,Fampa,Lee (2019)] propose local search procedures for the construction of sparse generalized inverses that satisfy only some of these properties. We propose implementation approaches for these local searches, aiming at their efficiency and scalability. We analyze the algorithms proposed, considering experiments with diversified random matrices.

KEYWORDS. Least-squares. Local search. Moore-Penrose pseudoinverse.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

1. Introdução

A pseudo-inversa M-P (Moore-Penrose), descoberta por E.H. Moore e R. Penrose, é usada em importantes aplicações de Álgebra Linear, como por exemplo para encontrar a solução de Mínimos Quadrados de sistemas de equações lineares inconsistentes. Sendo $A = U\Sigma V^\top$, a decomposição em valores singulares de A (veja Golub e Van Loan [1996], por exemplo), a pseudo-inversa M-P de A pode ser definida como $A^\dagger := V\Sigma^\dagger U^\top$, onde $\Sigma^\dagger$ tem a forma da transposta da matriz diagonal Σ e é derivada de Σ utilizando-se os recíprocos de seus elementos não nulos (em outras palavras, dos valores singulares não-nulos de A).

Mesmo se uma dada matriz for esparsa, sua pseudo-inversa M-P pode ser completamente densa, levando, em geral, a um alto custo computacional em suas aplicações, especialmente quando lidamos com matrizes que possuem grandes dimensões. A resolução de sistemas lineares de grande porte é um desafio numérico cada vez mais frequente nas mais diversas aplicações da Pesquisa Operacional. A obtenção da solução de Mínimos Quadrados de sistemas inconsistentes torna-se interessante quando busca-se uma boa solução para problemas, cujas restrições idealmente impostas, não podem de fato ser todas atendidas. Neste contexto, para evitar a realização de cálculos computacionais com matrizes densas de grandes dimensões, consideramos neste trabalho de Iniciação Científica, um método para construir inversas generalizadas com a desejada característica de serem esparsas. O método é baseado no não-atendimento de algumas propriedades da pseudo-inversa M-P, mas garantindo ainda a obtenção da solução de Mínimos Quadrados de sistemas lineares. O trabalho é baseado nos artigos teóricos Fampa e Lee [2018] e Xu et al. [2019], que propõem procedimentos de busca local, e utilizam a minimização da norma-1 (vetorial) para induzir esparsidade.

O objetivo deste trabalho é apresentar os estudos e resultados obtidos durante a Iniciação Científica do discente. O trabalho desenvolvido tem como objetivo tornar a obtenção da solução de Mínimos Quadrados de sistemas lineares inconsistentes mais eficiente através da construção de inversas generalizadas esparsas e estruturadas em blocos. O trabalho do discente compreende a implementação e a análise detalhada dos resultados computacionais obtidos através de experimentos numéricos. Foram também propostos pelo discente diferentes procedimentos de buscas locais, analisadas e comparadas a performance dos mesmos, e conferida, na prática, a proximidade do limite superior teórico apresentado em Fampa e Lee [2018] e Xu et al. [2019], que mede a qualidade das soluções das buscas locais. Finalmente, foram propostas diferentes abordagens para a definição da vizinhança explorada durante a execução das buscas locais, e para a construção de uma solução inicial para os algoritmos, que se mostrou na prática, um grande desafio numérico.

O restante deste trabalho é estruturado da seguinte forma. Na Seção 2, apresentamos os estudos preliminares da literatura. Na Seção 3, apresentamos os algoritmos implementados. Na Seção 4, apresentamos os resultados computacionais. Na Seção 5, apresentamos as considerações finais e perspectivas de trabalho futuro.

2. Estudo preliminar

Nesta seção, apresentamos os estudos realizados nas etapas preliminares da Iniciação Científica. A Subseção 2.1 refere-se à pseudo-inversa M-P e às inversas generalizadas. Na Subseção 2.2, discutimos a modelagem do problema de otimização não-linear estudado e o resultado que possibilita a sua linearização. Por fim, na Subseção 2.3, revisamos a Regra de Cramer, que é de extrema importância para a automatização das buscas locais.

2.1. Compreensão da Pseudoinversa e de suas Propriedades

Para a familiarização do discente com pseudo-inversas e inversas generalizadas, a referência Penrose [1955] foi estudada. Ela apresenta o importante teorema a seguir.

Teorema 1. *Para $A \in \mathbb{R}^{m\times n}$, a pseudo-inversa M-P $(A^\dagger)$ é a única matriz $H \in \mathbb{R}^{n\times m}$ que satisfaz:*

$$AHA = A \tag{P1}$$
$$HAH = H \tag{P2}$$
$$(AH)^\top = AH \tag{P3}$$
$$(HA)^\top = HA \tag{P4}$$

De acordo com Rohde [1964], uma *inversa generalizada* é uma matriz H que satisfaz P1. Essa propriedade é essencial para o nosso trabalho, pois sem ela, a matriz nula, extremamente esparsa, mas sem carregar nenhuma informação sobre a matriz A, iria satisfazer as outras três propriedades. A inversa generalizada é dita *reflexiva* se satisfaz P2. O Teorema 3.14 em Rohde [1964] afirma que: (i) se H é uma inversa generalizada de A, então $\text{posto}(H) \geq \text{posto}(A)$, e (ii) a inversa generalizada H de A é reflexiva se e somente se $\text{posto}(H) = \text{posto}(A)$. Portanto, impor P2 garante que a inversa generalizada tenha o menor posto possível, uma propriedade também bastante desejável, assim como a esparsidade. Finalmente, de acordo com Xu et al. [2019], H é *ah-simétrica* se satisfaz P3. Isso é, ah-simétrica significa que AH é simétrica. Se H é uma inversa generalizada ah-simétrica, então $\hat{x} := Hb$ resolve $\min\{\|Ax - b\|_2 : \ x \in \mathbb{R}^n\}$ (veja Fuentes et al. [2016]; Campbell e Meyer [2009]). Desta forma, observamos que nem todas as propriedades M-P são realmente necessárias para que H resolva este problema chave de Álgebra Linear: obter a solução de Mínimos Quadrados de sistemas de equações lineares.

Xu et al. [2019] apresentam um procedimento de construção de inversas reflexivas generalizadas ah-simétricas estruturadas em blocos. Especificamente, o procedimento é explicado através do Teorema 2, da Definição 1 e do Teorema 3 apresentados abaixo. Estes resultados representam o núcleo deste trabalho. A construção em blocos proposta faz com que as inversas generalizadas: (i) tenham todos os elementos não-nulos confinados em uma escolha seletiva de colunas, (ii) sejam reflexivas, (iii) possuam um número limitado de elementos não-nulos e (iv) tenham a norma-1 com um fator máximo com relação à norma-1 mínima de inversas reflexivas generalizadas ah-simétricas.

Teorema 2 (Xu et al. [2019])**.** *Seja $A \in \mathbb{R}^{m\times n}$, tal que $\text{posto}(A) = r$. Seja T um subconjunto ordenado qualquer de r elementos de $\{1,\ldots,n\}$ e $\hat{A} := A[:,T]$, a submatriz $m \times r$ de A formada pelas colunas em T. Se $\text{posto}(\hat{A}) = r$, defina $\hat{H} := \hat{A}^\dagger = (\hat{A}^\top\hat{A})^{-1}\hat{A}^\top$. Considere $H \in \mathbb{R}^{n\times m}$, como a matriz que possui todas as linhas iguais a zero, com exceção das linhas em T, as quais são dadas por $\hat{H}$. Então, H é uma inversa reflexiva generalizada ah-simétrica de A.*

Definição 1 (Xu et al. [2019])**.** *Seja $A \in \mathbb{R}^{m\times n}$, tal que $\text{posto}(A) = r$. Seja S um subconjunto ordenado de r elementos de $\{1,\ldots,m\}$, tal que as r linhas de A em S são linearmente independentes. Considere T, um subconjunto ordenado de r elementos de $\{1,\ldots,n\}$, e $\epsilon \geq 0$ fixo. Se $|\det(A[S,T])|$ não pode ser aumentado por um fator maior que $1+\epsilon$, ao trocar-se um elemento de T por um elemento no seu complemento, dizemos que $A[S,T]$ é um $(1+\epsilon)$-* maximizador local *para o módulo do determinante, no conjunto das submatrizes $r \times r$ não-singulares de $A[S,:]$.*

Teorema 3 (Xu et al. [2019])**.** *Seja $A \in \mathbb{R}^{m\times n}$, tal que $\text{posto}(A) = r$. Seja S um subconjunto ordenado de r elementos de $\{1,\ldots,m\}$, tal que as r linhas de A em S são linearmente independentes. Considere $\epsilon \geq 0$ e $\tilde{A} := A[S,T]$, um $(1+\epsilon)$-maximizador local para o módulo do determinante no conjunto das submatrizes $r\times r$ não-singulares de $A[S,:]$. Então, a matriz $H \in \mathbb{R}^{n\times m}$ construída como no Teorema 2, com $\hat{A} := A[:,T]$, é uma inversa reflexiva generalizada ah-simétrica de A, que satisfaz: $\|H\|_1 \leq r(1+\epsilon)\|H_{opt}^{ah,r}\|_1$, onde $H_{opt}^{ah,r}$ é a inversa reflexiva generalizada ah-simétrica de A com norma-1 mínima.*

2.2. Uma Formulação como um Problema de programação Linear (PPL)

Nesta subseção, verificamos inicialmente a possibilidade de reformular a propriedade P2, que foi originalmente apresentada como uma equação não linear em H, como uma equação linear, considerando que as propriedades P1 e P3 são ambas satisfeitas. Isso pode ser validado por meio da seguinte propriedade advinda da literatura [Fuentes et al., 2016, Proposição 4.3]:

Proposição 1. *Se H satisfaz P1 e P3, então $AH = AA^{\dagger}$.*

Com o resultado acima, o problema de encontrar a inversa reflexiva generalizada ah-simétrica com norma-1 mínima pode ser formulado como o seguinte PPL:

$$
\begin{array}{rlll}
(P_{123})\ z_{P_{123}} := & \min_{T,H} & \text{Traço}(J^T T)\ , \\
& \text{s.a:} & T - H \geq 0\ ,\ T + H \geq 0\ , \\
& & AHA = A\ ,\ (AH)^{\top} = (AH)\ ,\ HAA^{\dagger} = H\ .
\end{array}
$$

onde $J \in \mathbb{R}^{n\times m}$ tem todos os elementos iguais a um. Notamos que a modelagem acima possibilita a obtenção da inversa reflexiva generalizada ah-simétrica com norma-1 mínima através da resolução de um PPL, a qual será comparada com as soluções das buscas locais implementadas neste trabalho.

2.3. Regra de Cramer

A Regra de Cramer foi fundamental para a melhoria computacional dos algoritmos de busca local implementados. Sua aplicação, em conjunto com a fatoração LU, diminuiu o tempo computacional das iterações, deixando os métodos mais eficientes e possibilitando experimentos com matrizes maiores. Apresentamos a seguir o resultado aplicado em nossos algoritmos.

Observação 1. *Seja $\gamma \in \mathbb{R}^n$ e $A \in \mathbb{R}^{n\times n}$ com $\det(A) \neq 0$. Seja $A_{\gamma/j}$ uma matriz obtida substituindo-se a coluna j de A por γ. Se $\hat{\alpha} \in \mathbb{R}^n$ resolve $A\alpha = \gamma$, então $\det(A_{\gamma/j}) = \hat{\alpha}_j \times \det(A)$.*

3. Métodos Propostos

Nesta seção apresentamos os algoritmos implementados, e discutimos os resultados obtidos com os experimentos numéricos. Na Subseção 3.1, detalhamos os algoritmos para a construção do bloco utilizado para inicializar as buscas locais e na Subseção 3.2, apresentamos as buscas locais.

3.1. Construção do bloco inicial

Aplicamos dois diferentes algoritmos para construir a submatriz $r \times r$ não-singular de A para inicializar as buscas locais. Os algoritmos primeiramente selecionam um "bom" conjunto S de r linhas de A, e então, das linhas escolhidas, selecionamos um "bom" conjunto T de r colunas. No Algoritmo 1, Algoritmo *Greedy*, cada linha (e coluna) é selecionada visando obter o maior aumento do produto dos valores singulares da submatriz parcialmente construída. Na outra versão desse algoritmo descrita no Algoritmo 2, Algoritmo *Greedy Light*, cada linha (e coluna) selecionada é a primeira obtida (com os menores índices), que mantém o produto dos valores singulares das submatrizes parcialmente construídas maiores que uma dada tolerância positiva τ. Na prática, começamos com um valor relativamente grande de τ, e diminuímos iterativamente este valor, caso não seja possível encontrar as r linhas e colunas linearmente independentes.

A partir da submatriz $r \times r$ não-singular de A, obtida com um dos algoritmos apresentados nesta subseção, foi realizada uma busca local com o objetivo de reduzir a norma-1 da matriz H, por meio da substituição de colunas da submatriz, como descrito na próxima subseção.

Entrada: $A \in \mathbb{R}^{m\times n}$, t.q. posto$(A) = r$.
Saída: $S \subset M := \{1,\ldots,m\}, T \subset N := \{1,\ldots,n\}$, t.q. $|S| = |T| = r$, e $A[S,T]$ é não-singular.

$S := \emptyset$;
enquanto $|S| < r$ **faça**
    Escolha $i \in M \setminus S$ t.q. o produto dos $|S|+1$ valores singulares de $A[S \cup \{i\}, N]$ é maximizado;
    $S := S \cup \{i\}$;
$T := \emptyset$;
enquanto $|T| < r$ **faça**
    Escolha $j \in N \setminus T$ t.q. o produto dos $|T|+1$ valores singulares de $A[S, T+j]$ é maximizado;
    $T := T \cup \{j\}$;

Algoritmo 1: Algoritmo *Greedy*.

Entrada: $A \in \mathbb{R}^{m\times n}$, t.q. posto$(A) = r, \tau > 0$.
Saída: $S \subset M := \{1,\ldots,m\}, T \subset N := \{1,\ldots,n\}$, t.q. $|S| = |T| = r$, e $A[S,T]$ é não-singular.

$S := \emptyset$;
enquanto $|S| < r$ **faça**
    Escolha o menor $i \in M \setminus S$ t.q. o produto dos $|S|+1$ valores singulares de $A[S \cup \{i\}, N]$ é $> \tau$;
    $S := S \cup \{i\}$;
$T := \emptyset$;
enquanto $|T| < r$ **faça**
    Escolha o menor $j \in N \setminus T$ t.q. o produto dos $|T|+1$ valores singulares de $A[S, T \cup \{j\}]$ é $> \tau$;
    $T := T \cup \{j\}$;

Algoritmo 2: Algoritmo *Greedy Light*.

Entrada: $A \in \mathbb{R}^{m\times n}$, com posto$(A) = r$.
$S \subset M := \{1,\ldots,m\}, T \subset N := \{1,\ldots,n\}$, tal que $|S| = |T| = r$, e $A[S,T]$ é não-singular.
Saída: conjunto T possivelmente melhorado.

$\tilde{A} := A[S,T]$;
$\bar{N} := N \setminus T$;
$C := A[S,\bar{N}]$;
$[L,U] := LU(\tilde{A})$ (Calcule a fatoração LU de $\tilde{A}$);
$cont = true$;
enquanto $(cont)$ **faça**
    $cont = false$;
    para $\ell = 1,\ldots,n-r$ **faça**
        Resolva $Ly = C[S,\ell]$, com solução $\hat{y}$;
        Resolva $U\alpha = \hat{y}$, com solução $\hat{\alpha}$;
        $[\hat{\alpha}^{\ell}_{\max}, \hat{\jmath}] := \max_i\{|\hat{\alpha}_i|\}$;
        se $|\hat{\alpha}| \not\leq (1,\ldots,1)^{\top}$ **então**
            $\hat{\jmath} := \min\{j : |\hat{\alpha}_j| > 1\}$ para 'FI(det)', ou $\hat{\jmath} := \text{argmax}_j\{|\hat{\alpha}_j|\}$ para 'FI^{+}(det)';
            $aux := \tilde{A}[S,\hat{\jmath}]$;
            $\tilde{A}[S,\hat{\jmath}] := C[S,\ell]$;
            $C[S,\ell] := aux$;
            $T := T \cup \{\bar{N}(\ell)\} \setminus \{T(\hat{\jmath})\}$;
            $\bar{N} := \bar{N} \setminus \{\bar{N}(\ell)\} \cup \{T(\hat{\jmath})\}$;
            $[L,U] := LU(\tilde{A})$ (fatoração LU melhorada da iteração anterior);
            $cont = true$;

Algoritmo 3: 'FI(det)' ('FI^{+}(det)') para inversas generalizadas ah-simétricas

3.2. Procedimentos de Busca Local

No Algoritmo 3, apresentamos os procedimentos de busca local, First Improvement (FI(det)) e First Improvement Plus (FI$^+$(det)), que consideram como critério para a melhoria de uma dada solução, o aumento no módulo do determinante da submatriz $r \times r$ não-singular de matriz A. Baseado no Teorema 3, os procedimentos começam por um conjunto S de r linhas e um conjunto T de r colunas de A, tal que $A[S,T]$ é não-singular. Notamos que no Algoritmo 3, só as colunas de $A[S,T]$ são consideradas para realizar a troca no intuito de aumentar o determinante. Do resultado do Teorema 3, vemos que *qualquer* conjunto S de r linhas linearmente independentes pode ser utilizado para encontrar o maximizador local do módulo do determinante no conjunto das submatrizes $r \times r$ não-singulares de $A[S,:]$. As submatrizes utilizadas para inicializar a busca local são encontradas pelos Algoritmos 1 e 2, apresentados na Seção 3.1. No ciclo do Algoritmo 3 (linhas 6-20), uma coluna de $A[S, N \setminus T]$ substitui uma coluna de $A[S,T]$ se a troca fizer o módulo do determinante aumentar. Para avaliar o quanto o determinante varia quando cada coluna de $A[S,T]$ é trocada por uma dada coluna γ de $A[S, N \setminus T]$, usa-se o resultado apresentado na Observação 1.

4. Resultados Numéricos

Nesta seção apresentamos resultados obtidos com experimentos numéricos realizados com 450 instâncias de teste geradas aleatoriamente, com diferentes dimensões, postos e densidades, utilizando a função *sprand* do Matlab. Essa função gera aleatoriamente uma matriz A com dimensão $m \times n$, densidade aproximada d e valores singulares dados por um vetor rc. Os r elementos positivos de rc foram selecionados sendo $M \times (\rho^1, \rho^2, \ldots, \rho^r)$, tal que $M = 2$ e $\rho = (1/M)^{(2/(r+1))}$. Na Subseção 4.1, utilizamos um conjunto de 90 instâncias da categoria 'Pequenos'. Na Subseção 4.2, utilizamos um conjunto de 360 instâncias da categoria 'Grandes'.

4.1. Comparação com o Limite Superior

Inicialmente, consideramos os experimentos feitos com 90 instâncias da categoria 'Pequenos', com o objetivo de analisar as razões $\|H\|_1 / z_{P_{123}}$, entre a norma-1 das matrizes H encontradas pelas duas buscas locais e da inversa reflexiva generalizada ah-simétrica com norma-1 mínima, a qual é obtida coma solução do PPL P_{123}, visto na Subseção 2.2. Com isto, objetivamos verificar quão próximas essas razões estão do limite superior dado pelo Teorema 3. Observamos que o parâmetro ϵ, mencionado no teorema, pode ser considerado nulo na prática. Ele é considerado no teorema, apenas para garantir a complexidade polinomial das buscas locais (ver Xu et al. [2019]).

Os algoritmos propostos foram implementados em Matlab R2019b. O PPL foi resolvido com o Gurobi v.8, chamado através de um programa em Python (Spyder 3.3.2). Os experimentos foram executados em um computador de 16-core (Windows Server 2012 R2): dois processadores Intel Xeon CPU E5-2667 v4, 3.20GHz, com 8 cores cada, e 128GB de memória ram.

Não apresentamos aqui, resultados referentes à construção da solução inicial pelo Algoritmo 1 (*Greedy*). Apesar dele ter apresentado soluções iniciais melhores, demandou um alto esforço computacional, sendo um método muito caro computacionalmente e inutilizável na prática, a não ser para instâncias muito pequenas para serem representativas. Ademais, percebeu-se que o Algoritmo 2 (*Greedy Light*) leva a um tempo computacional total consideravelmente menor (aproximadamente 7 vezes menor na categoria 'Pequenos'), além de levar a resultados obtidos pelas buscas locais, similares aos obtidos com o Algoritmo 1. Desta forma, os resultados a seguir consideram a solução inicial construída pelo Algoritmo 2.

Na Tabela 1, apresentamos as médias das razões para matrizes com mesmas dimensões (m(=n)), postos (r) e densidades (d). Pelo Teorema 3, sabe-se que essas razões não podem ser maiores do que r. Observamos que as buscas locais ficam longe de atingir o limite máximo teórico, apesar dele ser comprovadamente alcançado no pior caso (ver Xu et al. [2019]). Apresentamos

também a média dos tempos computacionais para encontrar as inversas generalizadas. No caso das buscas locais, o tempo total para achar a inversa generalizada é dado pela soma do tempo de geração do bloco inicial pelo Algoritmo *Greedy Light* (GL) com o tempo da busca local (FI(det) / FI$^+$(det)). Podemos ver na Tabela 1 que os dois procedimentos de busca local convergem para

	$\|H\|_1/z_{P_{123}}$		Time(sec)			
m, r, d	FI(det)	FI$^+$(det)	P_{123}	GL	FI(det)	FI$^+$(det)
50,05, 0.25	1.017	1.017	1.244	0.005	0.001	0.001
50,25, 0.25	1.050	1.050	5.808	0.033	0.001	0.001
80,08, 0.25	1.073	1.073	3.805	0.016	0.001	0.001
80,40, 0.25	1.101	1.101	50.280	0.248	0.003	0.002
100,10, 0.25	1.066	1.066	8.054	0.048	0.003	0.003
100,50, 0.25	1.094	1.094	334.604	0.546	0.005	0.005
50,05, 0.50	1.022	1.022	1.367	0.003	0.001	0.001
50,25,0.50	1.071	1.071	10.986	0.037	0.001	0.001
80,08,0.50	1.027	1.030	4.875	0.016	0.001	0.001
80,40,0.50	1.088	1.088	247.588	0.240	0.003	0.002
100,10,0.50	1.053	1.073	12.130	0.026	0.002	0.003
100,50,0.50	1.112	1.116	1538.981	0.507	0.004	0.003
50,05,1.00	1.075	1.056	1.592	0.004	0.001	0.001
50,25,1.00	1.155	1.155	17.013	0.060	0.002	0.001
80,08,1.00	1.070	1.079	8.306	0.014	0.001	0.002
80,40,1.00	1.227	1.241	374.945	0.243	0.005	0.004
100,10,1.00	1.124	1.084	29.103	0.032	0.003	0.003
100,50,1.00	1.337	1.321	3342.032	0.534	0.006	0.004

Tabela 1: Buscas Locais para inversa generalizada vs. P_{123}

soluções similares na maioria dos experimentos. Além disso, o tempo para resolver P_{123} aumenta rapidamente à medida que as dimensões se tornam maiores, e enfatizam a importância das buscas locais para obter-se inversas generalizadas esparsas.

4.2. Eficiência das buscas locais

Realizamos também experimentos com 360 instâncias da categoria 'Grandes' a fim de comparar as buscas locais. As médias dos resultados para cada grupo de 30 instâncias com a mesma configuração são apresentadas na Tabela 2. Apresentamos as configurações das instâncias os valores médios das porcentagens de melhoria da solução das buscas locais com relação a solução inicial, dos tempos computacionais e dos números de iterações. Comparando os resultados com aqueles obtidos para as instâncias da categoria 'Pequenos', vemos que mesmo as matrizes tendo dimensões maiores, a qualidade dos resultados das buscas locais são similares. Elas obtiveram porcentagens de melhoria na norma-1 de H que variam de 60% a 91%, aumentando com a dimensão, o posto e a densidade. Entendemos que quanto maiores, maior é o espaço para modificação da matriz inicial.

	Melhoria		Tempo(seg)			Iterações	
m, r, d	FI(det)	FI$^+$(det)	GL	FI(det)	FI$^+$(det)	FI(det)	FI$^+$(det)
1000,050,0.25	0.605	0.605	18.749	0.065	0.057	53.333	49.767
1000,100,0.25	0.640	0.640	64.786	0.223	0.196	94.133	82.133
2000,100,0.25	0.683	0.683	265.938	0.377	0.322	120.633	110.200
2000,200,0.25	0.753	0.753	1346.207	0.967	0.693	212.233	181.73
1000,050,0.50	0.667	0.668	17.328	0.065	0.058	61.867	55.833
1000,100,0.50	0.739	0.739	76.082	0.229	0.198	113.767	95.733
2000,100,0.50	0.740	0.740	299.308	0.371	0.317	133.500	119.067
2000,200,0.50	0.831	0.831	1417.383	1.018	0.735	255.067	210.933
1000,050,1.00	0.826	0.827	19.087	0.064	0.065	97.733	82.933
1000,100,1.00	0.867	0.867	82.487	0.276	0.241	185.800	140.467
2000,100,1.00	0.880	0.879	336.089	0.388	0.382	196.600	169.600
2000,200,1.00	0.913	0.913	1772.662	1.177	0.866	401.100	295.433

Tabela 2: Resultados dos Procedimentos de Busca Local

5. Considerações Finais

Essa Iniciação Científica foi realizada por um ano sob a orientação da Profa. Marcia Fampa, da UFRJ, e do Prof. Jon Lee, da Universidade de Michigan (UM), e com a colaboração do aluno de doutorado Luze Xu, também da UM. O trabalho faz parte de um projeto de pesquisa

dos professores, e foi apresentado pelo Prof. Jon Lee em *CRM/DIMACS Workshop on Mixed-Integer Nonlinear Programming*, em Montreal (http://dimacs.rutgers.edu/events/details?eID=1491) e em seminário na Universidade de Michigan (https://ioe.engin.umich.edu/events/73226-18179646/). Foi também apresentado pelo discente na Semana de Integração Acadêmica (SIAc) na Universidade Federal do Rio de Janeiro, tendo recebido Menção Honrosa. O trabalho completo, disponível em https://arxiv.org/abs/2001.03732, foi submetido ao Journal of Global Optimization.

Com a pesquisa desenvolvida, conseguimos demonstrar que os procedimentos de busca local apresentados em Fampa e Lee [2018]; Xu et al. [2019] podem ser implementados com sucesso para construir inversas reflexivas generalizadas ah-simétricas que são esparsas e possuem blocos estruturados. Concluímos também que a razão entre a norma-1 da matriz obtida pelas buscas locais e a norma-1 mínima dentre as inversas reflexivas generalizadas ah-simétricas é muito menor nas instâncias de teste, do que o demonstrado na teoria. Além disso, vemos que os procedimentos de busca local são bastante robustos. Com relação a capacidade de lidar com instâncias cada vez maiores, é necessário estar atento à economia do custo computacional e aos erros numéricos. Nesse contexto, o cálculo das soluções iniciais provaram-se ser uma dificuldade prática surpreendente.

Comparando os dois procedimentos de busca local, concluímos que eles convergem para soluções de qualidade similar. Em geral, o procedimento 'FI$^+$(det)' obteve tempos um pouco melhores do que 'FI(det)'. Observamos que o tempo computacional para resolver os problemas de programação linear é muito maior do que o tempo de execução das buscas locais. À medida que aumentamos as dimensões, postos e densidades das instâncias, o tempo para resolver o problema de programação linear cresce muito rapidamente. O tempo de execução das buscas locais foi radicalmente reduzido com o uso dos resultados apresentados na Observação 1 (Regra de Cramer), que indicam a mudança no determinante das matrizes após a troca de linhas e colunas a cada iteração. Uma implementação ingênua, que recalculasse os determinantes do zero, não permitiria a aplicação dos algoritmos nas instâncias da categoria 'Grandes'.

O próximo passo para esta pesquisa, que está em andamento, é o desenvolvimento de algoritmos que incorporam a restrição P2 iterativamente e medem o trade-off entre a esparsidade e o posto de H. Finalmente, pretendemos aplicar nossos resultados na obtenção da solução de Mínimos Quadrados de sistemas lineares advindos de problemas de Pesquisa Operacional e Otimização.

Referências

Campbell, S. L. e Meyer, C. D. (2009). *Generalized inverses of linear transformations*. SIAM.

Fampa, M. e Lee, J. (2018). On sparse reflexive generalized inverses. *Oper. Res. Lett.*, 46:605–610.

Fuentes, V., Fampa, M., e Lee, J. (2016). Sparse pseudoinverses via LP and SDP relaxations of Moore-Penrose. In *CLAIO 2016*, p. 343–350.

Golub, G. e Van Loan, C. (1996). *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA.

Penrose, R. (1955). A generalized inverse for matrices. *Proc. Cambridge Philos. Soc.*, 51:406–413.

Rohde, C. (1964). *Contributions to the theory, computation and application of generalized inverses*. PhD thesis, University of North Carolina, Raleigh, N.C.

Xu, L., Fampa, M., e Lee, J. Approximate 1-norm minimization and minimum-rank structured sparsity for various generalized inverses via local search. https://arxiv.org/abs/1903.05744, 2019.