

Um algoritmo pseudo-periférico genérico para a heurística de Snay

Sanderson L. Gonzaga de Oliveira¹

Júnior Assis Barreto Bernardes²

Departamento de Ciência da Computação, UFLA, Lavras, MG

Resumo. A resolução de sistemas de equações lineares na forma $Ax = b$ é fundamental em muitas simulações numéricas na ciência e na engenharia. A redução do *profile* de A pode reduzir o custo de armazenamento e de resolução desses sistemas. Neste trabalho, propõe-se um algoritmo genérico para encontrar vértices pseudo-periféricos para a heurística de Snay. Em testes em oito instâncias da base de matrizes esparsas Harwell-Boeing, verificou-se que o número de vértices pseudo-periféricos selecionados pela heurística de Snay pode ser adequado para instâncias pequenas, mas é insuficiente para se obter bons resultados em instâncias que não são pequenas. Com os testes, mostra-se que é recomendável selecionar até 16% de vértices pseudo-periféricos em relação ao tamanho da instância.

Palavras-chave. Redução de *profile*, Vértices pseudo-periféricos, Renumeração, Reordenação, Matrizes esparsas

1 Introdução

A resolução de sistemas de equações lineares na forma $Ax = b$ é fundamental em muitas simulações numéricas na ciência e na engenharia e, frequentemente, é responsável pela maior parte do custo computacional nesses experimentos. A redução de *profile* pode beneficiar o custo de armazenamento de sistemas de equações lineares, bem como reduzir o custo computacional de métodos diretos e iterativos para resolver sistemas de equações lineares [1].

O *profile* de A pode ser definido como $profile(A) = \sum_{i=1}^n \left(i - \min_{1 \leq j < i} (j \mid a_{ij} \neq 0) \right)$. O problema da minimização de *profile* é um problema NP-difícil [3]. Com isso, dezenas de heurísticas para a redução de *profile* têm sido propostas desde a década de 1960.

A heurística de Snay [5] é uma das possíveis melhores heurísticas, com baixo custo computacional, para a redução de *profile* selecionadas em revisão sistemática da literatura [1]. Essa heurística possui um passo inicial que seleciona 10 vértices pseudo-periféricos para iniciar a renumeração. O objetivo deste trabalho é identificar qual o número de vértices

¹sanderson@dcc.ufla.br

²jassis@posgrad.ufla.br

pseudo-periféricos que devem ser selecionados para se obter o menor valor de *profile* pela heurística de Snay [5].

Na seção 2, é descrita a heurística de Snay [5] e o algoritmo proposto neste trabalho. Na seção 3, são descritas as simulações realizadas para este trabalho e são apresentados os resultados obtidos nos testes. Por fim, na seção 4, são apresentadas as conclusões.

2 Heurística de Snay

A seguir, aborda-se a heurística de Snay, conforme apresentada por Gonzaga de Oliveira e Chagas [4]. Considere o grafo $G = (V, E)$, em que V e E são conjuntos de vértices e de arestas, respectivamente. A heurística de Snay [5] recebe o vértice v inicial e referencia-se esse vértice na primeira entrada da renumeração $S = \{s(1), s(2), \dots, s(|V|)\}$. Em seguida, a cada iteração, entre os vértices candidatos, a heurística de Snay [5] renumera o vértice com o menor grau, ao considerar os vértices adjacentes a vértices já renumerados e também os vértices adjacentes a esses vértices. Nesse caso, para o cálculo do grau de um vértice, são desconsideradas as adjacências a vértices renumerados ou a vértices adjacentes a vértices renumerados. Por fim, a heurística de Snay [5] retorna uma renumeração dos vértices do grafo.

Snay [5] também propôs um algoritmo que retorna 10 vértices pseudo-periféricos iniciais para a heurística. Considere que a distância $d(v, u)$ entre dois vértices $u, v \in V$ é o tamanho do menor caminho entre u e v . O método para a determinação dos candidatos a vértice inicial, mostrado por Snay [5], consiste em considerar, a partir de um vértice aleatório $v \in V$, um conjunto D com cinco vértices mais distantes de v . Considera-se, também, o conjunto Q com cinco vértices mais distantes do vértice $u \in D$, em que u é um dos vértices mais distantes de v . Os conjuntos de vértices D e Q contêm os candidatos a vértice inicial da heurística de Snay [5].

Ao escolher os vértices mais distantes de um vértice inicial v aleatório, o algoritmo de Snay [5] considera, de maneira implícita, os vértices do(s) último(s) nível(is) da estrutura de nível enraizada em v . Porém, há uma escolha arbitrária de apenas cinco desses vértices, sem estabelecer critérios de desempate caso dois ou mais vértices tenham a mesma distância de v . A mesma arbitrariedade e falta de critério da escolha anterior acontecem tanto na determinação do vértice mais distante de v como na determinação do segundo conjunto de vértices candidatos a vértice inicial da renumeração de vértices do grafo.

Como descrito, o algoritmo de Snay retorna 10 vértices pseudo-periféricos. Esse algoritmo foi generalizado neste presente trabalho de forma a receber o parâmetro ν e retornar um conjunto com $2 \cdot \nu$ vértices. No algoritmo 1, adaptado de Gonzaga de Oliveira e Chagas [4], utiliza-se a estrutura de nível para se determinar os vértices mais distantes a um determinado vértice. O algoritmo 1 recebe um grafo $G = (V, E)$ e o número ν de vértices pseudo-periféricos a serem selecionados para cada um dos conjuntos D e Q , e tem como saída um conjunto de vértices pseudo-periféricos $D \cup Q$. O algoritmo de Snay [5] é um caso especial do algoritmo 1, ao estabelecer $\nu = 5$. Dessa forma, neste presente trabalho, o algoritmo para encontrar vértices pseudo-periféricos de Snay é generalizado. Na condição da linha 2, exige-se que o grafo tenha, pelo menos, $2 \cdot \nu$ vértices. Um vértice aleatório é

atribuído ao vértice v na linha 3. Na linha 4, constrói-se a estrutura de nível enraizada em v . A construção do conjunto de vértices mais distantes a v é iniciada na linha 5. A variável u , inicializada na linha 5, é utilizada para armazenar um vértice mais distante ao vértice v (veja a linha 12). A variável k , inicializada na linha 6, é utilizada para que $|D| = \nu$, pois o(s) último(s) nível(is) da estrutura de nível enraizada em v pode(m) não ter ν vértices. A variável i , também inicializada na linha 6, é utilizada na estrutura de repetição das linhas 7 a 18, em que ν vértices são inseridos em D na linha 9. A estrutura de nível enraizada em u é construída na linha 13. Da mesma forma, ν vértices mais distantes ao vértice u são inseridos em Q , nas linhas 20 a 27. Por simplicidade da apresentação, dois trechos de código similares são mostrados, nas linhas 5 a 18 e nas linhas 19 a 27. Finalmente, o algoritmo retorna $D \cup Q$, na linha 28.

Algoritmo 1: *VerticesPseudoPeriféricosGeneralizadoSnay* (vértices pseudo-periféricos para a heurística de Snay [5]).

Entrada: grafo $G = (V, E)$; número ν de vértices pseudo-periféricos a serem selecionados em cada iteração;

Saída: conjunto de vértices pseudo-periféricos $D \cup Q$;

```

1 início
2 se (  $|V| < 2 \cdot \nu$  ) então retorna erro;
3  $v \leftarrow \text{VerticeAleatorio}(V)$ ;
4  $\mathcal{L}(v) \leftarrow \text{BuscaEmLargura}(v)$ ; // a estrutura de nível enraizada no vértice  $v$  é construída
5  $D \leftarrow \emptyset$ ;  $u \leftarrow \emptyset$ ; //  $D$  conterá  $\nu$  vértices mais distantes de  $v$  e  $u$  será um desses vértices
6  $k \leftarrow 0$ ;  $i \leftarrow 0$ ;
7 enquanto (  $i < \nu$  ) faça
8     para cada (  $w \in L_{\ell(v)-k}(v)$  ) faça
9          $D \leftarrow D \cup \{w\}$ ;
10         $i \leftarrow i + 1$ ;
11        se (  $u = \emptyset$  ) então
12             $u \leftarrow w$ ; // um vértice  $u$  mais distante de  $v$  é selecionado
13            // a estrutura de nível enraizada no vértice  $u$  é construída
14             $\mathcal{L}(u) \leftarrow \text{BuscaEmLargura}(u)$ ;
15            fim-se;
16            // sai da estrutura de repetição para cada
17            se (  $i \geq \nu$  ) então break;
18        fim-para-cada;
19         $k \leftarrow k + 1$ ;
20    fim-enquanto;
21     $Q \leftarrow \emptyset$ ;  $k \leftarrow 0$ ;  $i \leftarrow 0$ ; //  $Q$  conterá  $\nu$  vértices mais distantes de  $u$ 
22    enquanto (  $i < \nu$  ) faça
23        para cada (  $w \in L_{\ell(u)-k}(u) - D$  ) faça
24             $Q \leftarrow Q \cup \{w\}$ ;
25             $i \leftarrow i + 1$ ;
26            // sai da estrutura de repetição para cada
27            se (  $i \geq \nu$  ) então break;
28        fim-para-cada;
29         $k \leftarrow k + 1$ ;
30    fim-enquanto;
31    retorna  $D \cup Q$ ;
32 fim.
```

No algoritmo 2, adaptado de Gonzaga de Oliveira e Chagas [4], avalia-se o *profile* da matriz associada ao grafo $G = (V, E)$ para cada renumeração de vértices iniciada por cada vértice pseudo-periférico retornado pelo algoritmo 1. O algoritmo recebe o grafo $G = (V, E)$ e a metade do número de vértices pseudo-periféricos a serem selecionados pelo

algoritmo 1, e tem como saída uma renumeração dos vértices de V .

Algoritmo 2: calcula o *profile* para cada vértice pseudo-periférico da heurística de Snay.

Entrada: grafo $G = (V, E)$; número de vértices pseudo-periféricos ν a serem selecionados em cada iteração;
Saída: renumeração S para V ;

```

1 início
2   se (  $2 \cdot \nu > |V|$  ) então  $\nu \leftarrow \lfloor |V|/2 \rfloor$ ;
3    $C \leftarrow VerticesPseudoPerifericosSnay(G, \nu)$ ; // algoritmo 1
4    $menorProfile \leftarrow +\infty$ ;  $melhorOrdem \leftarrow \emptyset$ ;
5   para cada (  $w \in C$  ) faça
6      $S \leftarrow HeuristicaSnay(G, w)$ ; // heurística mostrada em [4]
7      $profile \leftarrow CalculaProfile(G, S)$ ;
8     se (  $profile < menorProfile$  ) então
9        $menorProfile \leftarrow profile$ ;  $melhorOrdem \leftarrow S$ ;
10    fim-se;
11  fim-para-cada;
12  retorna  $melhorOrdem$ ;
13 fim.
```

Na condição da linha 2, limita-se o número de vértices pseudo-periféricos a serem retornados: $2 \cdot \nu$. Na linha 3, $2 \cdot \nu$ vértices pseudo-periféricos são inseridos em C , que é o conjunto de vértices pseudo-periféricos retornados pelo algoritmo 1. As variáveis *menorProfile* e *melhorOrdem* são inicializadas na linha 4. Na estrutura de repetição das linhas 5 a 11, para cada vértice $w \in C$, avalia-se a renumeração com w como o vértice inicial na linha 6 e o *profile* dessa renumeração na linha 7. Atualizam-se as variáveis *menorProfile* e *melhorOrdem* nas linhas 8 a 10. Finalmente, a melhor renumeração é retornada na linha 12.

3 Descrição e resultados dos testes

A heurística de Snay [5] foi implementada na linguagem de programação C++. Para avaliar a relação entre o número de vértices pseudo-periféricos selecionados pelo passo inicial da heurística de Snay [5] e o *profile* obtido, foram utilizadas oito instâncias da base de matrizes esparsas *Harwell-Boeing* (<http://math.nist.gov/MatrixMarket/data/Harwell-Boeing>) [2], mostradas na tabela 1. Foi utilizada uma máquina com processador Intel® Core™ i3-4005U (3MB Cache, CPU 1.8GHz×4, 4GB RAM DDR3 1333MHz) nos testes.

Foi atribuído um conjunto de valores para o parâmetro de entrada ν ($1 \leq \nu \leq \lfloor |V|/2 \rfloor$) do algoritmo 1 e foi verificado qual o menor valor de *profile* encontrado no algoritmo 2. Na figura 1, são apresentados os gráficos dos *profiles* e custos computacionais obtidos pela heurística de Snay [5] nas instâncias *BCSPWR02*, *PLAT362*, *662_BUS* e *NOS3*, em relação ao número de vértices pseudo-periféricos ($2 \cdot \nu$) selecionados no primeiro passo da heurística (algoritmo 1). Nessa figura, é possível perceber que, para que se possa obter o menor valor de *profile* nessas quatro instâncias pela heurística de Snay [5], é necessário estabelecer, respectivamente, pelo menos, $\nu = 3$, $\nu = 20$, $\nu = 30$ e $\nu = 48$ no algoritmo 1, lembrando que o algoritmo 1 retorna $2 \cdot \nu$ vértices pseudo-periféricos. Esses valores representam, aproximadamente, 12%, 11%, 9% e 10% da quantidade total de vértices das

instâncias *BCSPWR02*, *PLAT362*, *662_BUS* e *NOS3*, respectivamente.

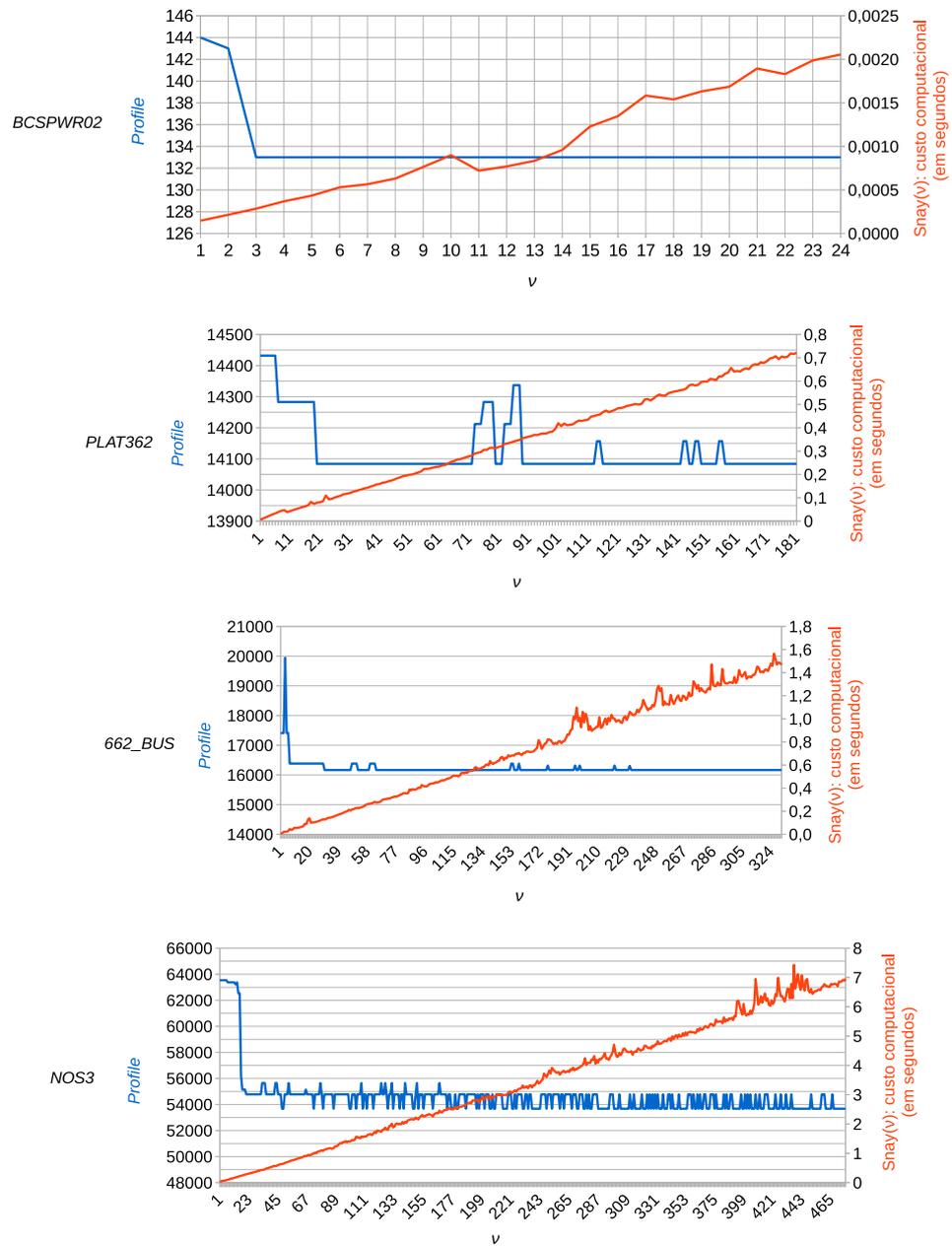


Figura 1: *Profiles* e custos computacionais obtidos pela heurística de Snay [5] em quatro instâncias, em relação à quantidade de vértices selecionados no algoritmo 1.

Esses resultados são sumarizados na tabela 1, em que são apresentadas as oito instâncias testadas, seus tamanhos, o valor de *profile* inicial ($profile_0$), o valor de ν mínimo

utilizado para se encontrar o *profile* mínimo obtido, o percentual de $2 \cdot \nu$ em relação ao tamanho da instância, o menor *profile* obtido pelas heurísticas $\text{Snay}(\nu)$ e Snay [5]. Na tabela 1, também são mostrados o tempo t de ambas as heurísticas, em segundos, $\%_{\nu} = \lfloor \frac{2 \cdot \nu}{\text{Tamanho}} \rfloor * 100$ e $\%_r = \lfloor \frac{\text{profile}}{\text{profile}_0} \rfloor * 100$, que é um fator de redução de *profile* proporcionado pelas heurísticas (note que o objetivo é minimizar $\%_r$). Resultados com maior de redução de *profile* foram obtidos com a heurística $\text{Snay}(\nu)$ do que com a heurística de Snay [5] original, a um custo computacional maior. Em particular, houve aumento do *profile* na instância *NOS3*. As reduções de *profiles* e os custos computacionais das heurísticas $\text{Snay}(\nu)$ e Snay [5] também podem ser comparadas na figura 2.

Tabela 1: Resultados da heurística de Snay [5] em conjunto com o algoritmo de Snay para encontrar vértices pseudo-periféricos e com o algoritmo genérico para encontrar $2 \cdot \nu$ vértices pseudo-periféricos proposto neste trabalho. Mostra-se o menor valor de ν estabelecido no algoritmo 1 para encontrar o menor *profile* em oito instâncias da base de matrizes esparsas Harwell-Boeing.

Instância	Tamanho	$profile_0$	$\text{Snay}(\nu)$					Snay [5]		
			ν	$\%_{\nu}$	$profile$	$\%_r$	$t(s)$	$profile$	$\%_r$	$t(s)$
<i>BCSPWR01</i>	39	292	2	10	85	29	0,0002	85	29	0,0003
<i>BCSPWR02</i>	49	377	3	12	133	35	0,0003	133	35	0,0004
<i>PLAT362</i>	362	45261	20	11	14084	31	0,0783	14283	32	0,0279
<i>DWT_503</i>	503	35914	13	5	23744	66	0,0785	23803	66	0,0306
<i>DWT_592</i>	592	28805	19	6	19335	67	0,1067	19360	67	0,0376
<i>662.BUS</i>	662	45165	30	9	16164	36	0,1272	17411	39	0,0225
<i>NOS3</i>	960	39101	48	10	53674	137	0,6324	63546	163	0,0740
<i>DWT_992</i>	992	262306	78	16	54024	21	1,1380	62260	24	0,0741

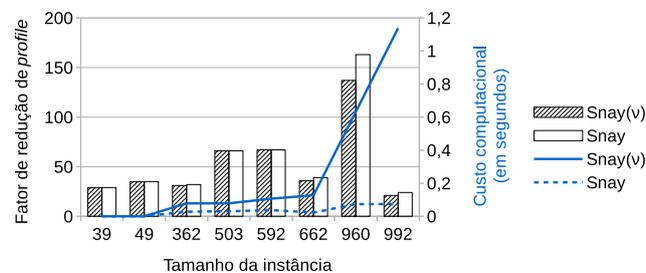


Figura 2: Fatores de redução de *profiles* e custos computacionais obtidos pelas heurísticas Snay [5] e $\text{Snay}(\nu)$ em oito instâncias da base de matrizes esparsas Harwell-Boeing.

4 Conclusão

A partir dos resultados obtidos com as simulações realizadas para este trabalho, é possível perceber que a seleção de 10 vértices pseudo-periféricos para a heurística de Snay

[5] é razoável nos casos em que o tamanho da instância é pequeno. Porém, quando a instância não é pequena, é necessária a escolha de um número maior de vértices pseudo-periféricos para que o menor valor de *profile* possa ser encontrado. Possivelmente, atribuir o parâmetro ν com 16% (veja a entrada para $\%_{\nu}$ na instância *DWT_992* na tabela 1) do tamanho da instância resulte no menor (ou bastante próximo ao menor) *profile* que pode ser obtido pela heurística. Com isso, o usuário poderá atribuir um percentual menor que esse para obter um menor custo computacional da heurística. Certamente, a quantidade de testes mostrada neste trabalho é pequena, mas pode ser uma amostra para investigações exploratórias em instâncias específicas do usuário.

O custo computacional da heurística de Snay(ν) é linear em relação ao parâmetro ν , já que a segunda etapa (algoritmo 2) da heurística de Snay [5] é executada para cada vértice pseudo-periférico encontrado na primeira etapa (algoritmo 1). Por sua vez, o custo computacional da heurística de Snay [5] depende, principalmente, do vértice inicial da renumeração (ou seja, vértice pseudo-periférico selecionado no algoritmo 1). Em mais detalhes, o custo computacional da heurística de Snay [5] depende principalmente da quantidade de vértices candidatos (para serem numerados) em cada iteração. Pretende-se apresentar a análise de complexidade da heurística de Snay [5] em trabalho futuro.

Também como sequência deste trabalho, pretende-se implementar as heurísticas selecionadas por Bernardes e Gonzaga de Oliveira [1] e compará-las com a heurística de Snay executada após o algoritmo genérico para encontrar vértices periféricos, com ν estabelecido após investigações exploratórias no conjunto de instâncias a serem testadas. O objetivo dessa revisão é descobrir a heurística no estado da arte em redução de *profile* de matrizes.

Agradecimentos

Este trabalho foi realizado com o apoio da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (Fapemig). Agradecemos também ao revisor anônimo os comentários e as sugestões valiosas para a melhoria deste trabalho.

REFERÊNCIAS

- [1] J. A. B. Bernardes and S. L. G. de Oliveira. A systematic review of heuristics for profile reduction of symmetric matrices, *Procedia Computer Science*, 51:221–230, 2015.
- [2] I. S. Duff, R. G. Grimes, and J. G. Lewis. Sparse matrix test problems. *ACM Transactions on Mathematical Software*, 15(1):1–14, 1989.
- [3] Y. X. Lin and J. J. Yuan. Profile minimization problem for matrices and graphs. *Acta Mathematicae Applicatae Sinica*, 10(1):107–122, 1994.
- [4] S. L. G. de Oliveira e G. O. Chagas. *Introdução a heurísticas para redução de largura de banda de matrizes*. SBMAC, São Carlos, 2014.
- [5] R. A. Snay. Reducing the profile of sparse symmetric matrices. *Bulletin Geodesique*, 50(4):341–352, 1976.