

Uma avaliação da utilização de busca local com o método *Reverse Cuthill-McKee*

Sanderson L. Gonzaga de Oliveira e Guilherme Oliveira Chagas¹
Departamento de Ciência da Computação, UFLA, Lavras, MG

Resumo. Com reduções de largura de banda e de *profile* de matrizes, pode-se obter redução no custo de processamento de resolutores iterativos de sistemas de equações lineares (SEs). O método *Reverse Cuthill-McKee* é um algoritmo clássico para redução de largura de banda e de *profile*. Esse método apresenta baixo custo computacional e gera resultados razoáveis. Neste trabalho, é avaliada a busca local *Hill-Climbing* para se melhorar os resultados do método *Reverse Cuthill-McKee*, ao iniciar seu processamento com vértices pseudo-periféricos determinados pelo algoritmo de George-Liu. Não foram observadas melhorias nos resultados com a busca local. Também, constatou-se que o custo de processamento exigido pela busca local não compensa eventuais melhorias nos resultados obtidos nas reduções de largura de banda e de *profile*. Consequentemente, não há redução do custo de processamento na resolução de sistemas de equações lineares pelo método dos gradientes conjugados.

Palavras-chave. Redução de largura de banda, Redução de *profile*, Método dos gradientes conjugados, Matrizes esparsas, Reordenamento de matrizes, Busca local.

1 Introdução

Reduções de largura de banda e de *profile* de matrizes são problemas importantes no contexto de resolução de sistemas de equações lineares na forma $Ax = b$, em que A é a matriz de coeficientes, x é o vetor de incógnitas e b é o vetor de termos independentes. Pode-se obter baixo custo de processamento na resolução de sistemas de equações lineares com as reduções de largura de banda e de *profile* da matriz A de coeficientes [6, 7, p. 94].

Seja $G = (V, A)$ um grafo conexo e não direcionado, em que V e A são conjuntos de vértices e arestas, respectivamente. A largura de banda de $G = (V, A)$ é definida como $\beta(G) = \max[(\forall v \in V)(\forall \{v, u\} \in A) |s(v) - s(u)|]$, em que $s(u)$ e $s(v)$ são as numerações (ou rótulos) dos vértices u e v , respectivamente. O *profile* de $G = (V, A)$ é definido como $profile(G) = \sum_{v \in V} \max[(\forall \{v, u\} \in A) |s(v) - s(u)|]$. Seja $A = [a_{ij}]$ a matriz (simétrica e $n \times n$, em que $n = |V|$) de adjacências do grafo $G = (V, A)$. Equivalentemente, a largura de banda de A é definida como $\beta(A) = \max[(1 \leq i \leq n) (1 \leq j < i) (i - \min(j : (1 \leq j < i) | a_{ij} \neq 0))]$ e *profile* é definido como $profile = \sum_{i=1}^n [i - \min(j : (1 \leq j < i) | a_{ij} \neq 0)]$. Os problemas de minimizações de largura de banda e de *profile* de matrizes A são de

¹sanderson@dcc.ufla.br, guilherme.chagas@computacao.ufla.br

aproximar, ao máximo possível, todos os coeficientes não nulos da diagonal principal de A . Esses são problemas pertencentes à classe NP-difícil [13,14].

Muitas heurísticas para os problemas de reduções de largura de banda e de *profile* de matrizes têm sido propostas desde a década de 1960 [1, 2, 6, 8]. Em revisões sistemáticas realizadas [1, 2, 6, 8], foram identificadas 132 heurísticas para reduções de largura de banda e/ou de *profile* de matrizes. Entre essas heurísticas, pode-se destacar o método *Reverse Cuthill-McKee* (RCM) [5], que é um algoritmo *in-place*, com baixo custo de processamento e que gera reduções de largura de banda e de *profile* razoáveis.

As reduções de largura de banda e de *profile* realizadas pelo método RCM são dependentes do vértice inicial para seu processamento. Um dos algoritmos mais conhecidos para a escolha desse vértice inicial é o algoritmo de George-Liu (GL) [3]. O método RCM-GL [4] é utilizado, por exemplo, no MATLAB (<http://www.mathworks.com/products/matlab>).

Buscas locais são utilizadas para melhorar soluções geradas por heurísticas, a fim de serem encontrados ótimos locais. Uma busca local bastante utilizada é o método *Hill-Climbing* (HC). Não é conhecido pelos autores alguma implementação do método RCM em conjunto com busca local. Neste trabalho, são apresentados resultados de experimentos numéricos realizados com o método RCM-GL e com busca local realizada com uma modificação no método HC para reduções de largura de banda e de *profile* de matrizes. Essa heurística é denominada aqui como RCM-GL-HC. Na seção 2, apresenta-se o método HC implementado. Detalhes sobre os experimentos numéricos e as ferramentas utilizadas são explicados na seção 3. Os resultados das simulações realizadas e as considerações finais são apresentados nas seções 4 e 5, respectivamente.

2 Busca local pelo método *Hill-Climbing*

O método HC implementado neste trabalho é uma modificação do método HC proposto por Lim et al. [12]. Essa modificação foi realizada com o objetivo de se obter uma busca local por HC com custo de processamento baixo, pois o método HC implementado por Lim et al. [12] apresenta custo de processamento alto. Com reduções de largura de banda e de *profile* de matrizes, objetiva-se reduzir o custo de processamento da resolução de SELs. Dessa forma, reduções de largura de banda e de *profile* não podem ter custo de processamento alto. No método HC de Lim et al. [12], são escolhidos vários vértices para se trocar a numeração, dependendo de algumas características como, p.ex., a numeração dos vértices adjacentes e a diferença entre as numerações dos vértices. A modificação avaliada neste trabalho limita a quantidade de vértices no conjunto para trocas de rótulos.

O método HC implementado neste trabalho é mostrado no algoritmo 1. Essa sub-rotina é chamada após a heurística RCM-GL renumerar os vértices do grafo. O algoritmo 1 recebe um grafo $G = (V, A)$ e retorna esse grafo com a numeração alterada se alguma melhora em relação à solução atual é encontrada; caso contrário, o grafo original é retornado. Na linha 2, são selecionadas (pelo menos duas) *arestas críticas* A_c do grafo $G = (V, A)$. Uma aresta $\{u, v\} \in A$ é crítica se $|s(u) - s(v)| = \beta$. Na linha 3, são armazenados a largura de banda e o *profile* originais. O laço de repetição repita-até nas linhas 4-11 é executado até que não se encontre melhoria na largura de banda e no *profile* na nova solução em relação

à solução corrente, de forma a se ter um procedimento com baixo custo de processamento.

Algoritmo 1: Pseudo-código da sub-rotina *Hill-Climbing*, modificada neste trabalho, para reduções de largura de banda e de *profile*, a baixo custo de processamento.

```

Entrada: Grafo  $G = (V, A)$ ;
Saída: Grafo  $G = (V, A)$  renumerado ou o grafo com a numeração original se não houve troca de rótulos dos vértices;
1 início
2    $A_c \leftarrow ArestasCriticas(G)$ ; // seleciona todas as arestas críticas do grafo
3    $\beta \leftarrow G.ObtemLarguraDeBanda()$ ;  $profile \leftarrow G.ObtemProfile()$ ;
4   repita
5      $e_1 \leftarrow Random(A_c)$ ;  $e_2 \leftarrow Random(A_c)$ ; // duas arestas críticas  $e_1 = \{v_1, v_2\}$ ,  $e_2 = \{v_3, v_4\} \in A$  são obtidas
     // troca os rótulos dos vértices das duas arestas, deixando-os com valores próximos entre si
6      $v_1.rotulo \leftarrow \max(v_1.rotulo, v_2.rotulo)$ ;  $v_2.rotulo \leftarrow \max(v_3.rotulo, v_4.rotulo)$ ;
7      $v_3.rotulo \leftarrow \min(v_1.rotulo, v_2.rotulo)$ ;  $v_4.rotulo \leftarrow \min(v_3.rotulo, v_4.rotulo)$ ;
     // calcula os novos largura de banda e profile
8      $novo_\beta \leftarrow G.ObtemLarguraDeBanda()$ ;  $novo\_profile \leftarrow G.ObtemProfile()$ ;
9     se  $((novo_\beta < \beta) \wedge (novo\_profile < profile))$  então
10        $\beta \leftarrow novo_\beta$ ;  $profile \leftarrow novo\_profile$ ;
11   fim-se;
12   até que  $((novo_\beta \neq \beta) \vee (novo\_profile \neq profile))$ ;
13   retorna  $G = (V, A)$ ;
14 fim.

```

Na linha 5, duas arestas críticas, $e_1 = \{v_1, v_2\}$, $e_2 = \{v_3, v_4\} \in A_c$, são obtidas aleatoriamente e removidas de A_c . Os vértices dessas duas arestas têm os rótulos trocados entre si nas linhas 6 e 7: os vértices v_1 e v_2 (v_3 e v_4), da aresta e_1 (e_2), recebem os maiores (menores) rótulos de vértice das arestas e_1 e e_2 (e_1 e e_2), respectivamente. Essa troca dos rótulos é realizada de modo que a largura de banda e *profile* resultantes sejam os menores possíveis. Com isso, a diferença entre rótulos será pequena em vértices adjacentes.

Na linha 8, a largura de banda e o *profile* novos são calculados. O processamento continua se uma melhora em relação à solução corrente é verificada, ou seja, se a largura de banda nova é menor que a largura de banda corrente e o *profile* novo é menor que o *profile* corrente; caso contrário, a busca local é encerrada.

O número de arestas críticas nos grafos é, geralmente, pequeno. Com isso, não há muita variedade de vértices para terem rótulos trocados. Por isso, com maior custo processamento, Lim et al. [12] utilizaram arestas λ -críticas. Uma aresta $\{u, v\} \in A$ é considerada λ -crítica se $|s(u) - s(v)| \geq \beta \cdot \lambda$, para $\lambda \in [0, 1] \subset \mathbb{R}$. Com isso, adotamos, também, vértices λ -críticos para se realizar a troca nas numerações dos vértices com o objetivo de se ter maior variedade de vértices para as trocas de rótulos. A alteração realizada no método HC, para que vértices λ -críticos também possam ser selecionados para trocas de rótulos foi na sub-rotina *ArestasCriticas*(G), executada na linha 2 do Algoritmo 1. Nessa modificação, são selecionadas as arestas λ -críticas do grafo. A heurística com essa modificação é denominada, neste trabalho, de RCM-GL-HC $_\lambda$. Assim, a heurística RCM-GL-HC é um caso especial desse método em que $\lambda = 1$.

3 Detalhes dos testes e ferramentas utilizadas

Foram realizados experimentos numéricos com as heurísticas RCM-GL, RCM-GL-HC e RCM-GL-HC $_\lambda$ em 10 instâncias: cinco instâncias oriundas da base de matrizes esparsas Harwel-Boeing e cinco sistemas de equações lineares, em que quatro desses sistemas são oriundos de discretizações das equações de Laplace [10] e da condução do calor pelo método

dos volumes finitos (MVF) [11]. Especificamente, utilizou-se o método dos gradientes conjugados preconditionado pelo método de Jacobi (MGCPJ) para resolver esses sistemas de equações lineares. Os métodos foram implementados na linguagem C++ e foi utilizado o compilador g++ 4.8.4. Além disso, utilizou-se a biblioteca *GNU Multiple Precision Floating-point computations with correct-Rounding* (MPFR) para se obter maior precisão (256 bits) nos cálculos na resolução dos sistemas de equações lineares pelo MGC.

Utilizou-se uma estação de trabalho com processador Intel® Core™ i5-3337U CPU 1,80GHz, com 3MB de memória *cache* e com 6GB de memória principal DDR3 1.333GHz (Intel, Santa Clara, CA, EUA). Foi utilizado o sistema operacional Ubuntu 14.04 LTS 64 bits com *kernel 3.19.0-39-generic*.

4 Resultados e análise

Nesta seção, são apresentados os resultados das simulações realizadas. Na Tabela 1, são apresentados os resultados das heurísticas RCM-GL, RCM-GL-HC e RCM-GL-HC $_{\lambda}$ (para $\lambda = 0,1; 0,3; 0,6; 0,9$) ao serem aplicados em cinco instâncias da base de matrizes esparsas Harwell-Boeing. Cada heurística foi executada cinco vezes e os dados apresentados nessa tabela são referentes à média dessas execuções. Observa-se que não houve melhoria nos resultados referentes à largura de banda e ao *profile* com as execuções dos métodos *Hill Climbing* (HC) e HC $_{\lambda}$ (para $\lambda = 0,1; 0,3; 0,6; 0,9$) ao serem executados após o método RCM-GL. Observa-se na Tabela 1 que não compensou aumentar o número de possíveis vértices para terem rótulos trocados. Independentemente do valor de λ , os resultados, em relação à largura de banda e ao *profile*, foram os mesmos obtidos pelas heurísticas RCM-GL e RCM-GL-HC. Quanto menor o valor de λ , mais vértices são selecionados e, com isso, maior o custo de processamento para manipular esses vértices. Dessa forma, como esperado, a heurística RCM-GL-HC (RCM-GL-HC $_{\lambda}$) apresentou custo de processamento maior que a heurística RCM-GL (RCM-GL-HC) em todas as simulações.

Tabela 1: Resultados do método RCM-GL [4] em conjunto com o método *Hill Climbing* (HC) e com o método HC $_{\lambda}$ (para $\lambda = 0,1; 0,3; 0,6; 0,9$) avaliados em relação à largura de banda, ao *profile* e ao tempo de execução em simulações com matrizes simétricas.

Instância	n	β			<i>profile</i>			tempo (s)						
		-	GL	GL-HC	GL-HC $_{\lambda}$	-	GL	GL-HC	GL-HC $_{\lambda}$	GL	GL-HC	$\lambda = 0,9$	$\lambda = 0,6$	$\lambda = 0,3$
Awh85	85	39		13		1153		617	0,00012	0,00021	0,00017	0,00034	0,00037	0,00080
Dwt_234	234	48		24		1765		1359	0,00015	0,00024	0,00034	0,00026	0,00049	0,00075
Nos1	237	4		4		780		467	0,00019	0,00031	0,00047	0,00051	0,00108	0,00175
Can_838	838	837		118		207200		33859	0,00152	0,00174	0,00269	0,02005	0,04095	0,07046
Sherman1	1000	100		59		34740		25948	0,00075	0,00081	0,00179	0,03042	0,05238	0,05956

Em vez de \wedge , utilizamos também \vee na condição composta na linha 9 no Algoritmo 1 para que, em relação à largura de banda e ao *profile*, houvesse melhoria de um e possivelmente piora de outro resultado. Essa alteração faz com que aumente o tempo de execução da busca local. Foram realizadas cinco execuções em cada instância mostrada na Tabela 1 com essa variação. Todavia, não foi observada melhora nos resultados ao final do procedimento. Em algumas execuções, foi observado na instância Nos1 que um dos resultados (largura de banda ou *profile*) piorou e o outro foi melhorado. Entretanto, ao final da execução, encontrou-se o mesmo resultado obtido pelo método RCM-GL. Na Tabela 2,

são apresentados resultados, referentes a uma execução, de experimentos numéricos realizados na resolução de sistemas de equações lineares pelo MGCPJ. As heurísticas RCM-GL e RCM-GL-HC resultaram na mesma renumeração dos vértices, pois a busca local pelo método Hill Climbing não alterou a solução gerada pela heurística RCM-GL. Mesmo assim, houve diferentes tempos de execução do MGCPJ. Foram utilizados, nesses experimentos numéricos, cinco instâncias de sistemas de equações lineares: duas instâncias oriundas da discretização da equação da condução do calor pelo MVF [11]; duas instâncias oriundas da discretização da equação de Laplace pelo MVF [10], em que a instância composta por 1.874 incógnitas foi numerada originalmente na sequência dada pela curva de Sierpiński modificada [9, 15] e a instância composta por 34.238 incógnitas foi numerada originalmente de forma aleatória; uma instância intitulada *Pres.Poisson* pertence à base de matrizes esparsas *The University of Florida Sparse Matrix Collection*, e também é oriunda de problema de dinâmica de fluidos computacional.

Tabela 2: Resultados de execuções do MGCPJ aplicado a SELs com renumeração dos vértices pelo método RCM-GL [4] em conjunto com o método HC avaliado neste trabalho.

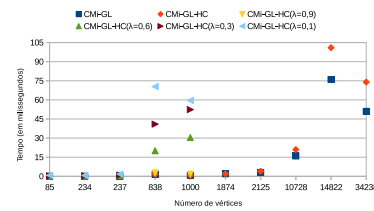
Instância (n)	Heurística	β	<i>profile</i>	t (s)	MGCPJ[t(s)]	<i>Speedup</i>
Eq. do calor (MVF) (2125)	-	2120	1818156	-	3	-
	GL	114	129907	0,003	3	0,95
	GL-HC	114	129907	0,004	3	0,94
Eq. do calor (MVF) (10728)	-	10626	45314579	-	37	-
	GL	271	1575766	0,016	35	1,06
	GL-HC	271	1575766	0,021	35	1,06
<i>Pres.Poisson</i> (14822)	-	12583	9789525	-	404	-
	GL	401	3158771	0,076	441	0,92
	GL-HC	401	3158771	0,101	442	0,91
Eq. de Laplace (MVF) (1874)	-	1873	57765	-	2	-
	GL	42	51807	0,002	2	0,94
	GL-HC	42	51807	0,002	2	0,94
Eq. Lap. aleat. (MVF) (34238)	-	34059	357518296	-	171	-
	GL	192	3413888	0,051	188	0,91
	GL-HC	192	3413888	0,074	188	0,91

Observa-se, na Tabela 2, que os resultados em relação às reduções de largura de banda e de *profile* das heurísticas RCM-GL e RCM-GL-HC foram iguais. Isso porque, como nas simulações mostradas na Tabela 1, a busca local não influenciou nos resultados. Em nenhuma das iterações do método HC houve melhora em relação à solução corrente. O método RCM [5] numera os vértices na sequência dada pela busca em largura, ao ordenar os vértices de um nível em ordem ascendente de grau. Ao numerar os vértices de uma aresta crítica $e_1 = \{v_1, v_2\} \in A_c$ pelos maiores rótulos em relação à segunda aresta $e_2 = \{v_3, v_4\} \in A_c$, é possível que os vértices $v_1, v_2 \in V$ já tenham rótulos maiores que os vértices $v_3, v_4 \in V$. Como esperado, a heurística RCM-GL-HC foi mais lenta que a heurística RCM-GL. Na Figura 1, são mostrados os tempos de execução, em milissegundos, obtidos ao aplicar as heurísticas RCM-GL, RCM-GL-HC e RCM-GL-HC $_{\lambda}$ (para $\lambda = 0,1; 0,3; 0,6; 0,9$) para reduções de largura e de *profile* de matrizes em 10 matrizes simétricas.

5 Considerações finais

Como pode ser observado nos resultados apresentados neste trabalho, o método de busca local *Hill Climbing* não proporcionou resultados satisfatórios quando aplicado em conjunto com o método *Reverse Cuthill-McKee* com vértice inicial (do processamento)

Figura 1: Tempos de execução, em milissegundos, obtidos ao aplicar heurísticas para reduções de largura e de *profile* de matrizes em 10 matrizes simétricas.



dados pelo algoritmo de George-Liu [4], em nenhuma das simulações realizadas: a busca local não alterou a renumeração dos vértices obtida pela heurística RCM-GL. Não comprou utilizar essa busca local, pois os resultados não foram melhorados e, também, há o custo de processamento (adicional) da busca local. Mesmo com a utilização de arestas λ -críticas, com valores de λ variados, a busca local por HC não alterou as soluções e, ainda, apresentou custo de processamento ainda maior que o método HC com apenas arestas críticas. Com as arestas λ -críticas, aumentou-se o número de vértices para trocar a numeração, mas os vértices ainda são escolhidos aleatoriamente. Com isso, essa abordagem também não se mostrou eficiente, pois depende da escolha de vértices com numerações adequadas. Uma alternativa seria utilizar o método HC proposto por Lim et al. [12]; mas isso aumentaria ainda mais o custo de processamento e não compensaria na redução do custo de processamento na resolução de sistemas de equações lineares (veja a Tabela 2).

A busca local não alterou as soluções geradas, provavelmente, por causa da forma como a troca dos rótulos dos vértices foi realizada. Isso porque não foram analisadas as demais adjacências dos vértices em que os rótulos foram trocados. Isso faz com que a largura de banda ou o *profile*, na maioria das vezes, aumente, fazendo com que o método HC termine após realizar poucas iterações. No método HC proposto por Lim et al. [12], são analisadas, entre outras características, as adjacências dos vértices. Porém, isso aumenta significativamente o custo de execução do método. Com as simulações mostradas neste trabalho, entendemos que dificilmente uma busca local melhorará os resultados de reduções de largura de banda e de *profile* de matrizes de forma a reduzir o custo de processamento na resolução de sistemas de equações lineares, principalmente por causa do custo de processamento para se realizar a busca local. Apesar disso, como trabalho futuro, pretende-se testar outras técnicas de busca local, como a abordagem *k-opt* em conjunto com heurísticas para reduções de largura de banda e de *profile* de matrizes esparsas [6].

Agradecimentos

Este trabalho foi realizado com o suporte da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG).

Referências

- [1] J. A. B. Bernardes and S. L. Gonzaga de Oliveira. A systematic review of heuristics for profile reduction of symmetric matrices. In *International Conference on Computational Science, ICCS 2015*, volume 51, pages 221–230, Iceland, 2015. Elsevier.

- [2] G. O. Chagas and S. L. Gonzaga de Oliveira. Metaheuristic-based heuristics for symmetric-matrix bandwidth reduction: A systematic review. In *International Conference on Computational Science*, pages 211–220, Iceland, 2015. Elsevier. 51.
- [3] A. George and J. W. H. Liu. An implementation of a pseudoperipheral node finder. *ACM Transactions on Mathematical Software*, 5(3):284–295, 1979.
- [4] A. George and J. W. H. Liu. *Computer solution of large sparse positive definite systems*. Prentice-Hall, Englewood Cliffs, 1981.
- [5] J. A. George. *Computer implementation of the finite element method*. PhD thesis, Computer Science Department, Stanford University, CA, USA, 1971.
- [6] S. L. Gonzaga de Oliveira, J. A. B. Bernardes, and G. O. Chagas. An evaluation of low-cost heuristics for matrix bandwidth and profile reductions (doi=10.1007/s40314-016-0394-9). *Computational & Applied Mathematics*, 2016.
- [7] S. L. Gonzaga de Oliveira and G. O. CHAGAS. *Introdução a heurísticas para redução de largura de banda de matrizes*. SBMAC, São Carlos-SP, 2014. Notas em Matemática Aplicada.
- [8] S. L. Gonzaga de Oliveira and G. O. Chagas. A systematic review of heuristics for symmetric-matrix bandwidth reduction: methods not based on metaheuristics. In *SBPO*, Pernambuco, Brazil, Agosto 2015. Sobrapo.
- [9] S. L. Gonzaga de Oliveira and M. Kischinhevsky. Sierpiński curve for total ordering of a graph-based adaptive simplicial-mesh refinement for finite volume discretizations. In *Congresso Nacional de Matemática Aplicada e Computacional - CNMAC*, pages 581–585, Belém, Brazil, 2008.
- [10] S. L. Gonzaga de Oliveira, M. Kischinhevsky, and J. M. R. S. Tavares. Novel graph-based adaptive triangular mesh refinement for finite-volume discretizations. *Computer Modeling in Engineering & Science - CMES*, 95(2):119–141, 2013.
- [11] S. L. Gonzaga de Oliveira, F. S. Oliveira, and G. O. Chagas. A novel approach to the weighted laplacian formulation applied to 2-D Delaunay triangulations. In *Proceedings of the 15th ICCSA*, volume 9155 of *LNCS*, pages 502–515. Springer, 2015.
- [12] A. Lim, B. Rodrigues, and F. Xiao. Heuristics for matrix bandwidth reduction. *European Journal of Operational Research*, 174:69–91, 2006.
- [13] Y. X. Lin and J. J. Yuan. Profile minimization problem for matrices and graphs. *Acta Mathematicae Applicatae Sinica*, 10(1):107–122, 1994.
- [14] C.H. Papadimitriou. The NP-completeness of bandwidth minimization problem. *Computing Journal*, 16:177–192, 1976.
- [15] L. Velho, L. H. Figueiredo, and J. Gomes. Hierarchical generalized triangle strips. *The Visual Computer*, 15(1):21–35, 1999.