



ARTHUR ARANTES FARIA

REDUÇÃO DE LARGURA DE BANDA EM MATRIZES

LAVRAS – MG

2018

ARTHUR ARANTES FARIA

REDUÇÃO DE LARGURA DE BANDA EM MATRIZES

Relatório sobre redução de largura de banda em matrizes com algoritmo de George Liu, apresentado na disciplina de Projeto e Análise de Algoritmos da Universidade Federal de Lavras como trabalho prático para obtenção de nota na disciplina.

LAVRAS – MG

2018

SUMÁRIO

1	Introdução	1
2	Materiais e métodos	2
2.1	Conceitos Básicos	2
2.1.1	Largura de banda	2
2.1.2	Banda	2
2.2	George Liu	2
3	Experimentos	4
4	Conclusão	6
	REFERÊNCIAS	7

1 INTRODUÇÃO

O Problema de redução de largura de banda em matrizes é um problema computacional amplamente pesquisado na literatura, por ser um subproblema de diversos outras questões computacionais relacionados a matrizes. A dificuldade de tais problemas pode ser reduzida, fornecendo condições desejadas às matrizes, por meio da resolução do problema em questão.

A banda de uma matriz é igual a banda diagonal que compreende as diagonais que contenham elementos não nulos mais distantes da diagonal principal. O objetivo da redução de banda é então diminuir a distância das diagonais mencionadas mais afastadas da diagonal principal da matriz, tanto a direita quanto a esquerda. O problema é definido a partir de uma matriz M quadrada esparsa.

Os sistemas de equações lineares considerados tem a seguinte forma $Ax = b$, a qual A é uma matriz $n \times n$ esparsa, simétrica e positiva-definida. Considera-se que uma matriz A é positiva-definida quando, para qualquer vetor x , tem-se, $x^T Ax > 0$. A matriz $A = [a_{ij}]$ é positiva definida, se e somente se todos os valores são positivos.

Skiena (2008) retrata que um exemplo de utilização, a qual essa tarefa é positiva, é o cálculo de inversões, determinantes e redução Gaussiana, que pode ser aplicada em $O(n^2)$ em matrizes com largura de banda b , ponto significativo quando comparado ao algoritmo geral de complexidade $O(n^3)$ quando $b < n$. Papadimitriou (1976) expõe que o problema é completo em NP, assim como as suas variantes conhecidas, ainda que cada linha da matriz possua no máximo 3 elementos não nulos (considera-se m_{ij} , com $i \neq j$). De acordo com Garey et al. (1978) isso aumenta o interesse teórico no problema.

Devido às características do problema, é recomendado que métodos heurísticos sejam adotados para obter uma solução, já que métodos exatos encontram dificuldades, principalmente de tempo, para problemas de tamanhos moderados a grandes (considerando que existem $n!$ permutações de linhas e $n!$ permutações de colunas, em que n denota a dimensão da matriz). Em algumas heurísticas, o vértice inicial de renumeração influencia na qualidade da solução.

O presente trabalho propõe uma comparação na resolução do problema de redução de banda de matrizes, comparando o algoritmo de George-Liu a outro. A estrutura do trabalho é composta da seguinte forma: na seção seguinte são apresentados os materiais e métodos utilizados; a Seção 3 são expostos os experimentos realizados e por fim na última seção são apresentados as conclusões obtidas pelo desenvolvimento do mesmo.

2 MATERIAIS E MÉTODOS

2.1 Conceitos Básicos

Essa seção, contém de conceitos básicos para a compreensão do presente trabalho. Suas definições foram vistas e extraídas do Capítulo 1 do livro de Oliveira e Chagas (2014).

2.1.1 Largura de banda

Seja A uma matriz simétrica $n \times n$, com coeficientes a_{ij} e $1 \leq i, j \leq n$. A largura de banda da i – sima linha é $\beta(A)$ é a maior distância de coeficiente não nulo da matriz triangular inferior até a diagonal principal, considerando-se todas as n linhas da matriz. A definição anterior significa que a largura de banda de uma matriz simétrica A é o número de subdiagonais da matriz triangular inferior.

2.1.2 Banda

A banda de A é definida como: $\text{banda}(A) = (1 \leq i \leq n)(1 \leq j \leq i)(i, j) | (0 < i - j \leq \beta(A))$. Isso significa que a banda de uma matrix é formada pelas diagonais da matriz triangular inferior que contenham coeficientes não nulos. Note que a diagonal principal não compõe a banda. Ainda, note que só se considera a matriz triangular inferior porque A é simétrica.

2.2 George Liu

Esse algoritmo foi proposto como melhoria do algoritmo de Gibbs, Poole Jr e Stockmeyer (1976), George e Liu (1979) propuseram quatro modificações a esse algoritmo. A primeira modificação, é citada no artigo como *short circuiting*, consiste em terminar a montagem das estruturas de nível enraizadas dos vértices folhas, assim que encontrar um nível com largura (de nível) maior do que a previamente encontrada. A segunda modificação é iniciar o algoritmo com um vértice qualquer em vez de um com grau mínimo. A terceira e quarta modificações são denominadas *shrinking*, pois ambas são técnicas para escolher um vértice do último nível da estrutura (de nível) enraizada. A terceira modificação é montar a estrutura de nível enraizada apenas para o vértice folha que possuir o grau mínimo. A última proposta foi escolher um vértice folha qualquer para a montagem da estrutura de nível. Após os experimentos dele, o algoritmo final proposto por George e Liu (1979) é a combinação da segunda e da terceira propostas de melhoria.

George e Liu (1979) mostra que o algoritmo pode ser simplificado em alguns passos. Na parte de inicialização, um vértice qualquer $v \in V$ é escolhido para ser o novo vértice inicial. Logo, a estrutura de nível $\mathcal{L}(v)$ é gerada. Seguindo, escolhe-se um vértice de grau mínimo $u \in L_{\ell(v)}(v)$. É gerado $\mathcal{L}(u)$ e verifica-se se $\ell(v) < \ell(u)$. Caso a condição for satisfeita, então, o vértice u passa a ser o vértice v e o processo é repetido. Em um caso contrário, o vértice pseudo-periferico será v .

Na Figura 2.1 mostra-se o pseudocódigo do algoritmo. Nele, inicializa-se a variável v com um vértice arbitrário, logo, é construída a estrutura de nível enraizada $\mathcal{L}(v)$ utilizando o algoritmo de busca em largura. Na estrutura de repetição *repita* das linhas 4 a 11, constrói-se a estrutura de nível do nó folha u de grau mínimo de $\mathcal{L}(v)$. Se a excentricidade de u for maior que a excentricidade de v , então, atribui-se u a v e repete-se o processo. O algoritmo retorna o vértice v quando $\ell(v)$ for maior ou igual que a excentricidade do seu vértice folha de grau mínimo.

Figura 2.1 – Algoritmo de GeorgeLiu

Algoritmo 1: George-Liu.	
<hr/>	
Entrada: grafo $G = (V, E)$;	
Saída: vértice pseudo-periférico $v \in V$;	
1	início
2	$v \leftarrow EscolheVertice(V)$;
	// constrói-se estrutura de nível enraizada
3	$\mathcal{L}(v) \leftarrow BuscaEmLargura(v)$;
4	repita
5	$u \leftarrow VerticeComGrauMinimo(L_{\ell(v)}(v))$;
	// constrói-se estrutura de nível enraizada
6	$\mathcal{L}(u) \leftarrow BuscaEmLargura(u)$;
7	se ($\ell(u) > \ell(v)$) então
8	$v \leftarrow u$;
9	$\mathcal{L}(v) \leftarrow \mathcal{L}(u)$;
10	fim-se ;
11	até que ($u \neq v$) ;
12	retorna v ;
13	fim.

3 EXPERIMENTOS

Neste capítulo será exposto os resultados dos experimentos utilizados para resolução do problema de redução de largura de banda. Nele também será apresentado resultados utilizando outro algoritmo, a fim de comparar qual entre eles pode ser o melhor para escolher pseudo-vértices iniciais,

Foi utilizado como métrica de performance o tempo de relógio de execução de cada algoritmo, os testes foram executados em um computador pessoal com as seguintes configurações: Processador Intel® Core™ i7-6700HQ *cache* de 6M, *clock* até 3,50 GHz e 16gb ram 2133 MHz DDR4.

As bases de dados são matrizes coletadas do Suite Sparse Matrix Collection (DAVIS; HU, 2011), que atendem todos os requisitos previamente citados como necessários. Cada uma como um tamanho diferente de outra. Vale mencionar, que até a data de entrega deste relatório, o presente algoritmo, mesmo utilizando de um meio eficiente para armazenar os dados da matrix, denominado Compressed Row Storage (CRS), o mesmo não apresentou resultado para matriz vezes maiores que as testadas.

Toda programação foi desenvolvida utilizando a linguagem de programação C++. Fazendo uso da ferramenta de versionamento de código, Github, a implementação está disponível publicamente no link: <<https://github.com/aarantes23/cuddly-lamp/tree/master/tp1>>

Por fim, a tabela a seguir mostra os resultados para a execução dos algoritmos nas bases de dados da coluna inicial. O resultado do tempo de execução de cada um corresponde em uma media simples de 20 execuções de cada algoritmo.

Tabela 3.1 – Resultados da execução do algoritmo para cada base

Base	Linhas	Colunas	Instancias	Largura Inicial	George Liu		Diagonal Dominante	
					Tempo de execução	Largura Final	Tempo de execução	Largura Final
1138_bus	1138	1138	2596	387	0.75597	356	0.024904	465
494_bus	494	494	1080	61	0.112782	45	0.003988	61
662_bus	662	662	1568	502	0.191143	497	0.007975	576
685_bus	685	685	1967	598	0.279776	482	0.010002	207
bp_0	822	822	3276	810	0.089758	803	0.015924	810
bp_1000	822	822	4661	817	0.144615	743	0.02294	817
bp_1200	822	822	4726	817	0.141596	724	0.021944	817
bp_1400	822	822	4790	817	0.139641	769	0.022959	817
bp_1600	822	822	4841	817	0.122707	770	0.022938	817
bp_200	822	822	3802	809	0.113981	809	0.019983	809
bp_400	822	822	4028	808	0.113016	810	0.019972	808
bp_600	822	822	4172	808	0.157024	709	0.019947	808
bp_800	822	822	4534	817	0.124083	658	0.023912	817
Goodwin_010	1182	1182	32282	1180	1.062183	1179	0.141979	1180
Goodwin_013	1965	1965	56069	1963	3.613688	1963	0.419533	1963

4 CONCLUSÃO

Após todo o estudo dos métodos descritos no Capítulo 2 e com os testes realizados expostos no Capítulo 3 esta seção apresenta as considerações finais do trabalho, fazendo um relacionamento dos resultados com os métodos e materiais citados anteriormente.

Neste trabalho, com todo o cenário disponível, incluindo a implementação desenvolvida e os testes executados, pode-se concluir que com o algoritmo de George e Liu (1979) obtém uma melhor (*menor*) largura de banda final, quando comparado a utilização de nenhum algoritmo e na maioria das vezes sendo ainda melhor que o outro algoritmo implementado também.

Acreditasse que com algumas alterações e diversos testes, a implementação possa ser executada para bases maiores, podendo assim, comparar com o tempo de outros trabalhos publicados na literatura.

REFERÊNCIAS

- DAVIS, T. A.; HU, Y. The university of florida sparse matrix collection. **ACM Transactions on Mathematical Software (TOMS)**, ACM, v. 38, n. 1, p. 1, 2011.
- GAREY, M. R. et al. Complexity results for bandwidth minimization. **SIAM Journal on Applied Mathematics**, SIAM, v. 34, n. 3, p. 477–495, 1978.
- GEORGE, A.; LIU, J. W. An implementation of a pseudoperipheral node finder. **ACM Transactions on Mathematical Software (TOMS)**, ACM, v. 5, n. 3, p. 284–295, 1979.
- GIBBS, N. E.; POOLE JR, W. G.; STOCKMEYER, P. K. An algorithm for reducing the bandwidth and profile of a sparse matrix. **SIAM Journal on Numerical Analysis**, SIAM, v. 13, n. 2, p. 236–250, 1976.
- OLIVEIRA, S. G. de; CHAGAS, G. Introdução a heurísticas para redução de largura de banda de matrizes. **SBMAC, São Carlos**, 2014.
- PAPADIMITRIOU, C. H. The np-completeness of the bandwidth minimization problem. **Computing**, Springer, v. 16, n. 3, p. 263–270, 1976.
- SKIENA, S. S. Section 15.2: Topological sorting. **The Algorithm Design Manual (2nd ed.)**, Springer-Verlag, London, p. 481–483, 2008.