1.03.02 – Ciência da Computação / Matemática da Computação

ESTUDO DE EFICIÊNCIA ALGORÍTMICA PARA CONECTIVIDADE DE GRAFOS

Matheus Vyctor Aranda Espíndola[1], Bruno Dias Amaro[2]

1. Estudante da Faculdade de Computação – FACOM - da Universidade Federal de Mato Grosso do Sul
2. Professor do Instituto de Matemática da Universidade Federal de Mato Grosso do Sul/ Orientador

Resumo

Na Teoria Espectral dos Grafos (TEG) – ramo da matemática discreta que focaliza na análise de propriedades de um grafo a partir do espectro de matrizes relacionadas a ele – um problema muito comum é estudar a conectividade de um grafo G, tendo em vista que inúmeras informações estruturais e algébricas podem ser analisadas. Assim, o objetivo do presente trabalho é estudar e comparar a eficiência computacional diante algoritmos que aplicam Teoremas da TEG para computar tal conectividade.

Para tanto, três algoritmos foram implementados na Linguagem de Programação C com diferentes aplicações dos Teoremas da Álgebra Linear. Depois, fez-se um estudo de gasto de memória e tempo, bem como informações que o programa obtem. Ao compará-los, percebe-se que, dependendo do Teorema que era aplicado, o gasto computacional se amenizava consideravelmente, melhorando sua eficiência, o que denota a importância de estudar tal tema, assim como a relação da Álgebra Linear com a Teoria dos Grafos.

Palavras-chave: Matriz Laplaciana; Matriz de Adjacência; custo computacional.

Introdução

A Teoria Espectral dos Grafos (TEG) é um ramo da matemática discreta que tem por objetivo principal analisar as propriedades estruturais e algébricas de um grafo a partir do espectro de matrizes associadas a ele, como as matrizes de adjacência A e laplaciana L. Autovalores e autovetores das respectivas matrizes associadas a um grafo aparecem como relações que formam o autoespaço de grafos [1] [12].

Problemas hodiernos, progressivamente, são não só representados, mas também resolvidos por meio de grafos – o que marca a sua aplicabilidade no cotidiano. Dentre diversos cenários, os conceitos matemáticos de grafos já são vistos na Sociologia [5] e no ramo genético da Biologia [8]. No que se refere a seu estudo estrutural, é relevante destacar a análise da conectividade em sua conformação, tendo em vista que denota inúmeras propriedades. Evidencia-se, portanto, sua grande relevância no meio científico. A importância do estudo vem decorrente dessa conjuntura. Por meio da pesquisa desenvolvida, poder-se-á realizar um estudo de eficiência computacional de algoritmos de conectividade de grafos.

O objetivo do presente trabalho é estudar e comparar a eficiência computacional de algoritmos que analisam conectividade de grafos, implementando-os. Como específico:

- Estudar curvas assintóticas de algoritmos de grafos;
- Implementar e analisar o algoritmo de série de potências;
- Implementar e analisar o algoritmo de *Faddeev-LeVerrier*;
- Implementar e analisar o algoritmo de busca por profundidade;
- Implementar algoritmos que apliquem os teoremas de Teoreia Espectral dos Grafos através da Linguagem de Programação C.

Metodologia

Foram pesquisados, internacionalmente e nacionalmente, artigos e livros os quais abordaram Teoria Espectral dos Grafos (TEG), havendo o embasamento teórico para a elaboração do estudo. Então, os seguintes Teoremas foram utilizados:

Seja G = G(V, E) um grafo simples com n vértices. Denotamos por A = A(G) a matriz de adjacência de G, D = D(G) a matriz diagonal dos graus dos vértices de G e L = L(G) = D − A a matriz laplaciana de G.

Teorema 1 ([2]). Seja um grafo G(V,E) com n vértices e uma matriz de adjacência A associada a ele, temos que o grafo G é conexo se, e somente se, a matriz dada por:

$$A + \mathrm{A}^2 + \ldots + \mathrm{A}^{\mathrm{n}} = \sum_{\mathrm{k}=1}^{\mathrm{n}} \mathrm{A}^{\mathrm{k}}$$

não possuir entrada nula.

Teorema 2 ([1] [2] [4]). Sejam $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_n$ os autovalores de uma matriz laplaciana L associada a um grafo G(V,E). O grafo é considerado conexo se, e somente se, $\lambda_{n-1} \geq 0$

Corolário 1 ([2]). A quantidade de componentes conexas de um grafo G é dada pela multiplicidade de autovalores iguais a 0 da matriz laplaciana L associada a ele.

Teorema 3 ([8] [10] [12]). Seja $c(\lambda) = \det(\lambda \cdot I - M) = \lambda_n + c_1 \cdot \lambda_{n-1} + c_2 \cdot \lambda_{n-2} + \cdots + c_n$ o polinômio característico de uma matriz M. Os coeficientes c_k podem ser encontrados por:

$$c_k = -\frac{Tr(M_k)}{k}, 1 \leq k \leq n$$

$$M_k = A \cdot (M_{k-1} + c_{k-1} \cdot I)\ 2 \leq k \leq n,$$

onde $c_0 = 1,\ M_1 = M$ e I é a matriz identidade.

Teorema 4. A multiplicidade de raizes nulas de um polinômio dado por $p(x) = x_n + c_1 \cdot x_{n-1} + c_2 \cdot x_{n-2} + \cdots + c_n$é igual a n−k se, e somente se, $c_i = 0, \forall\, n-k < i \leq n$.

Conforme foi objetivado, foi feita uma análise de eficiência algorítmica a partir da confecção de algoritmos utilizando os respectivos teoremas. Para isso, implementaram-os por meio da Linguagem de Programação C. Após finalizados, fez-se o estudo de sua eficiência algoritmica, levando em consideração:

- Tempo de execução;
- Gasto de memória;
- Quais informações são obtidas com o algoritmo.

Para encontrar a eficiência do tempo de execução, criou-se uma função f(n), onde n é o tamanho da entrada, ou seja, a quantidade de vértices de um grafo G que será processado.

Buscando encontrar a taxa/ordem de crecimento, ignoraram-se constantes para levar em consideração a operação dominante. Assim, pôde-se analisar sua eficiência assintótica, ou seja, como é seu comportamento dominante quando sua entrada n tende ao infinito, associando-se ao conceito de limite.

Após identificar o termo dominante da expressão que descreve sua complexidade (crescimento assintótico), foi encontrada a função que é um limitante superior assintótico para a expressão, isto é, para instâncias arbitrárias de tamanho n podemos resolver o problema em tempo menor ou igual a $\mathcal{O}$(f(n)).

Resultados e Discussão

Algoritmo 01: somatório de potências da matriz de adjacência A de um grafo G.

O primeiro algoritmo foi implementado aplicando a proposição do Teorema 1 de forma que computasse se um grafo G era ou não conexo. Para isso, o algoritmo realiza a soma das potências de $\mathrm{A}^1, \mathrm{A}^2, \ldots, \mathrm{A}^n$. Com a matriz resultante obtida, o código percorre-a a fim de encontrar uma entrada nula. Se, e somente se, havê-la, o grafo G é desconexo.

Com o algoritmo pronto, fez-se um estudo de eficiência assintótica. Nele, foi possível notar que o algoritmo proposto é caracterizado como $\mathcal{O}$ ($\mathrm{n}^4 \cdot \log_2 n$), tendo em vista que é realizada uma série de somas com n multiplicações de matrizes, onde, cada uma, tem um custo de $\mathcal{O}$ (n^3), e as potências $\mathcal{O}$ ($\log_2 n$).

Depois, executou-o 30 vezes para cada entrada de tamanho n. Nessas execuções, a entrada foi gerada de maneira *randômica*, visando uma amostra ideal. Fez-se uma média aritmética das 30 execuções e uma tabela foi montada, conforme segue:

Ordem(n)	Tempo$(ti_n)(ms)$
5	0.07
10	0.32
25	14.01
50	262.55
75	1554.71
100	5126.99
150	28244.73
200	103426.97
250	331184.60
300	656788.28

Tabela 1. Relação de ordem X tempo – Algoritmo 01

Foi feito, também, um estudo de memória. Nele pôde-se observar que, comparativamente, o gasto de memória do primeiro algoritmo é alto, já que há, constantemente, a criação de matrizes auxiliares; estas que, por sua vez, utilizam variáveis do tipo *long long int*, consumindo muita memória (64 *bits* cada), o que é agravado quando se leva em consideração os armazenamentos serem sequenciais.

Algoritmo 02: análise de conectividade a partir dos autovalores da matriz de laplacina L de um grafo G.

O segundo algoritmo encontra seu L-polinômio característico da matriz Laplaciana L e, depois, computa a conectividade ou não de um grafo G. Posteriormente, contabiliza a quantidade de partes conexas. Para isso, o algoritmo, primeiramente, parte do método de *Faddeev-LeVerrier* (Teorema 3) para encontrar o L-polinômio caracaterístico de L.

Com o polinômio obtido, o código percorre-o a fim de apurar a quantidade de autovalores nulos, onde é utilizado o Teorema 5 de modo que veja os coeficientes nulos. Com a quantidade de autovalores iguais a zero obtida, aplica-se o Teorema 2 para decidir a conectividade do grafo G. Posteriormente, é aplicado também o Corolário 1 com a finalidade de computar o seu número de partes conexas [4] [10].

Com o algoritmo pronto, fez-se um estudo de eficiência assintótica. Nele, foi possível notar, conforme visto na bibliografia [4] [10] [2] que o algoritmo proposto é caracterizado como $\mathcal{O}$ (n^4), tendo em vista que são realizadas n multiplicações de matrizes, onde, cada uma, tem um custo de $\mathcal{O}$ (n^3).

Depois, executou-o 30 vezes para cada entrada de tamanho n. Nessas execuções, novamente, a entrada foi gerada de maneira *randômica*. Fez-se uma média das 30 execuções e uma tabela foi montada, conforme segue:

Ordem(n)	Tempo(ti_n)(ms)
5	0.00
10	0.00
25	0.00
50	18.77
75	101.10
100	317.47
150	1707.57
200	6027.93
250	15103.63
300	32102.00

Tabela 2. Relação de ordem X tempo – Algoritmo 02.

Foi feito, também, um estudo de memória. Nele pôde-se observar que, comparativamente, o gasto de memória do segundo algoritmo é médio, já que as matrizes auxiliares são liberadas. No entanto, ainda há a utilização de variáveis do tipo *long long int*, bem como os armazenamentos são sequenciais.

Algoritmo 03: Busca por profundidade.

O terceiro algoritmo para solução de problemas de conectividade em grafos utiliza o conceito de busca por profundidade, ou *Depth-First Search* (DFS). Esse algoritmo utiliza a representação de grafos por meio de listas encadeadas, onde são ligados por nós – possuindo uma relação de pais e filhos. Intuitivamente, o algoritmo começa em um nó raiz e percorre toda a estrutura de um grafo dado. Para percorrê-la, ele utiliza um processo de recursão onde, a cada iteração, é percorrida uma combinatória possível de seus ramos [5] [7].

Tecnicamente, o algoritmo é definido através de uma busca não informada (ou seja, independende de em qual vértice é iniciado, a busca percorrerá todos os caminhos, já que parte de um algoritmo recursivo) que continua através da expansão do primeiro nó filho da árvore que está sendo buscada. A cada ciclo, o algoritmo se aprofunda mais, até o momento que não seja encontrado o alvo ou que ele encontre um nó folha – que não tem filhos.

Nele, há a utilização de *backtracks* de maneira recursiva [9]. É provado [5] que para árvores a sua curva assintótica é de $\mathcal{O}$ (n+m), onde n é a quantidade de vértices e m a quantidade de arestas de um grafo G. Para nosso estudo, sabe-se que n ≥ m e, portanto, o pior caso do algoritmo será de $\mathcal{O}$ (2·n), ou seja, $\mathcal{O}$ (n) [6]. Em relação a memória, o seu uso é considerado, comparativamente, muito baixo, tendo em vista que não há a criação de matrizes auxiliares e, sobretudo, o seu armazenamento ser feito por listas encadeadas, isto é, dinâmicamente.

Comparação: Gráfico comparativo do tempo de execução dos algoritmos 01 e 02.

Com os algoritmos desenvolvidos, executados e seus dados de tempo coletados, fez-se um gráfico comparativo dos algoritmos 1 e 2 – onde se levou em consideração os seus respectivos crescimentos assintóticos. Para tanto, foi utilizado o processo de regressão polinomial de grau 4 – tendo em vista que o estudo de eficiência assintótica realizado mostrou que ambos possuiam uma taxa polinomial de grau 4, possuindo, nessa situação, o menor erro.

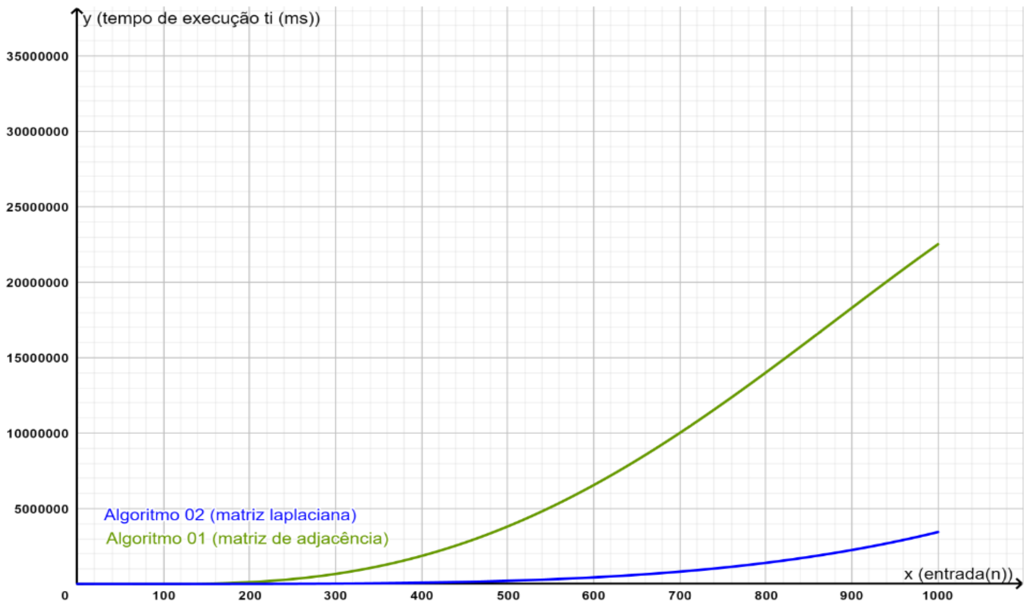

Gráfico 1. Gráfico comparativo o tempo de execução dos algoritmos 01 e 02.

Conclusões

O desenvolvimento do presente estudo possibilitou uma análise de eficiência de algoritmos computacionais que aplicam os conceitos e teoremas vistos em Teoria Espectral dos Grafos para a resolução de problemas de conexidade em grafos, bem como uma reflexão acerca da comparação entre os algoritmos implementados. Tendo em vista o estudo realizado e usando-o como base, concluiu-se que há uma significativa diferença em eficiência algoritma entre eles, principalmente em memória e tempo de execução. Em uma comparação gráfica, o algoritmo 2 – que utiliza a matriz laplaciana L – apresenta um melhor desempenho assintótico. Já o terceiro algoritmo – que implementa conceitos da Ciência de Computação por meio da Busca por Profundidade – motrou-se com grande eficácia, resolvendo o problema de forma recursiva e signitificamente rápida. Dada a importância do tema, faz-se necessário a continuidade de pesquisas relacionadas não só à Teoria Espectral de Grafos, mas também à conexão das propriedades de grafos com a Álbegra Linear e, também, dos conceitos da Ciências de Computação na resolução de problemas práticos.

Referências bibliográficas

[1] AMARO, B. D. A soma dos maiores autovalores da matriz laplaciana sem sinal em fam´Ilias de grafos. Tese de Doutorado, IMECC/UNICAMP, 2014.

[2] ABREU, N. M. M., DEL-VECCHIO, R. P. e VINAGRE, C. T. M. Teoria espectral de grafos - uma introdução. IIIº Colóquio de Matemática da Região Sul, 2014.

[3] BALDAS, M. T. M. L. Reconhecimento e traçado de grafos planares. Tese de Doutorado, COPPE / UFRJ, 1995.

[4] EVEN, S. Graph algorithms. Cambridge University Pressv, 2012.

[5] LOZADA, L. A. P. A-graph: Uma ferramenta computacional de suporte para o ensinoaprendizado da disciplina teoria dos grafos e seus algoritmos. Anais dos Workshops do CBIE 2014, 2014.

[6] NETTO, P. O. A. Grafos: Teoria, modelos, algoritmos. Edgard Blucherl, 2014.

[7] OLIVEIRA, C. C. B. e NAGAN, N. Configurações de redes de distribuição de energia elétrica com múltiplos objetivos e incertezas através de procedimentos heurísticos. Universidade de São Paulo, 1997.

[8] PERNET, C. e STORJOHANN, A. Faster algorithms for the characteristic polynomial. Symbolic Computation Group, 2007.

[9] PIMENTEL, M. G. e SAMPAIO, F. F. Comunicografia - uma metodologia para análise de processos de interação que se desenvolvem nas ferramentas de comunicação textual da internet utilizadas no contexto de educação a distância. revista brasileira de informática na educação. Revista Brasileira de Inform´atica na educação, 2002.

[10] REHMAN, R. e IPSEN, I. C. F. La buddes's mehotd for computing characteristic polynomials. arXiv:1104.3769v1, 2011.

[11] ROSSETTO, D. R. Algoritmos de busca de caminhos em grafos aplicados aos problemas: Alinhamento de proteínas e quebra-cabeca de 15 peças, Universidade Federal de Santa Catarina - Curso de Pós-Graduação em Matemática e Computação Científica, 2007.

[12] YU, B e KITAMOTO, T. The CHACM method for computing the characteristic polynomial of a polynomial matrix. Ieice Trans. Fundamentals, Vol E83-A, nº7, 2000.