

Encontrando todos os cliques máximos em um grafo não direcionado (2023)

Bernardo. Guilherme L. P., Co-autor, *Universidade Federal de Roraima*, Ferreira. Kelvin A., Co-autor, *Universidade Federal de Roraima*

1

Resumo— Este artigo tem como objetivo uma revisão de literatura acompanhada de testes em diferentes plataformas relacionadas ao Problema do Clique Máximo. Onde a metodologia usada para chegar nos resultados foi através de comparação entre dois algoritmos semelhantes, um com abordagem em algoritmo guloso e outro com solução baseada em força-bruta em alguns testes de benchmarks de grafos, por meio de diversos testes realizados. Além disso, mostrar suas complexidades, tempos de execução em determinados grafos e discutir sobre os resultados adquiridos.

Principais Termos — computação, cliques, grafos, testes e np-completo.

I. INTRODUÇÃO

O problema do clique é uma questão fundamental na teoria dos grafos e tem aplicações em diversas áreas, como ciência da computação, redes sociais, bioinformática e otimização. Ele está relacionado à identificação de grupos de elementos que estão fortemente interconectados.

Observando isso, o objetivo dos autores é avaliar a solução proposta dos ex-alunos de computação Bruno Claudino e Felipe Derkian[3], de 2018, para o problema do clique. A solução proposta por eles apresenta um algoritmo específico para detectar e enumerar a quantidade de cliques presentes no maior clique de um grafo. Nessa pesquisa, o foco principal não é comparar o algoritmo de Claudino e Derkian com outras soluções disponíveis na praça, mas sim avaliar se ele cumpre efetivamente sua proposta. O objetivo é analisar a eficiência e a eficácia desse algoritmo, verificando sua capacidade de encontrar corretamente todos os cliques máximos em diferentes tipos de grafos. Será realizada uma avaliação experimental, na qual o algoritmo será aplicado a grafos de teste.

Dessa forma, o objetivo central da pesquisa é avaliar a solução proposta por Claudino e Derkian em 2018,

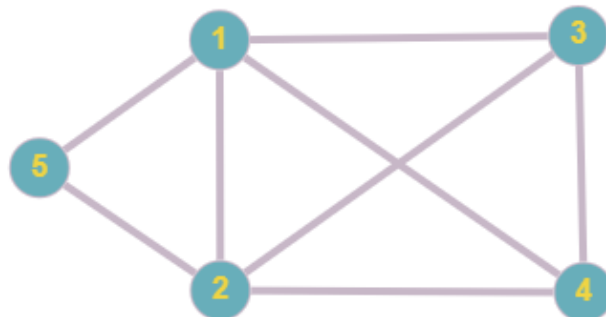
analisando sua eficiência, eficácia e robustez em relação à identificação de cliques máximos em grafos. Essa avaliação permitirá verificar a relevância e a aplicabilidade do algoritmo proposto, bem como identificar possíveis melhorias ou limitações em sua abordagem.

Ao final da pesquisa, espera-se fornecer uma análise crítica da solução proposta por Claudino e Derkian, oferecendo insights sobre seu desempenho em diferentes cenários e contribuindo para o avanço do conhecimento na área do problema do clique. Isso pode orientar futuros pesquisadores na seleção de abordagens adequadas para a identificação de cliques máximos em grafos.

II. REFERENCIAL TEÓRICO

A. O que é um clique?

O termo “Clique” surgiu em 1949 através de um artigo publicado por Luce e Perry[5] na área de Ciências sociais, onde eles correlacionam subgrafos completos com cliques sociais. Em 1970, Harary e Ross[4], em seus trabalhos, propõem uma resolução para o problema do clique onde também definem que um clique em um grafo é um conjunto de vértices onde cada par de vértices está conectado por uma aresta. Em outras palavras, é um subgrafo completo dentro do grafo. O tamanho de um clique é o número de vértices que o compõem. Por exemplo, um clique de tamanho 4 é chamado de 4-clique, e consiste em quatro vértices onde eles estão conectados por uma aresta.



1

Imagem 1: Exemplo de grafo com um 4-clique retirado do livro de CORMEN et al.[7] Algoritmos: Teoria e Prática.

- **Clique Máximo:** Um clique máximo é um clique de maior tamanho em um determinado grafo. Em outras palavras, é o maior subconjunto de vértices que formam um clique no grafo. O objetivo do problema do clique máximo é encontrar esse clique de maior tamanho.

B. Problema do clique é NP-Completo

Em 1972, Richard M. Karp[6] publicou um artigo contendo 21 algoritmos NP-Completo, onde o problema do clique fazia parte desse montante. Em seu artigo, Karp faz a prova da NP-completude com uma redução do problema SAT para fórmulas na forma normal conjuntiva, utilizando o teorema de Stephen Cook[1] de 1971.

C. Benchmark de grafos e sua importância na avaliação experimental

Benchmarks de grafos avaliam desempenho de algoritmos. Conjuntos de grafos padronizam comparações em domínios diferentes, onde seus resultados auxiliam na identificação de vantagens e limitações. A reprodutibilidade e confiabilidade dos experimentos são garantidas.

Utilizar benchmarks de grafos é extremamente relevante na avaliação de algoritmos pois os conjuntos de grafos de referência são projetados para representar cenários comuns, sua padronização permite avaliar diferentes abordagens comparativamente.

III. METODOLOGIA

A presente pesquisa adotou uma metodologia que envolveu a revisão da solução proposta por Claudino e Derkian em seu trabalho final sobre o problema do clique em 2018, bem como a seleção de benchmarks de grafos para a avaliação experimental.

A revisão foi realizada por meio de uma análise detalhada do artigo original. Esse processo envolveu a compreensão dos algoritmos, bem como a identificação dos principais aspectos, estratégias e técnicas utilizadas na abordagem proposta. Após a revisão da solução, foi selecionado um benchmark de grafos para a avaliação experimental. Para isso, foram considerados os trabalhos de Shaowei Cai[2], que disponibiliza conjuntos de grafos padronizados e amplamente utilizados na comunidade

científica para a realização de testes e avaliação de algoritmos relacionados a problemas de grafos.

A seleção do benchmark de grafos foi baseada na diversidade dos grafos disponíveis, abrangendo diferentes características, tais como tamanho, densidade e complexidade. Foram escolhidos conjuntos de grafos que são comumente utilizados para avaliar algoritmos de clique máximo, garantindo assim a comparabilidade dos resultados obtidos.

Durante a execução dos testes dos algoritmos, foram coletados dados referentes ao tempo de execução, consumo de recursos e qualidade das soluções encontradas. Esses dados foram analisados e comparados com os resultados esperados, permitindo assim uma avaliação crítica da solução proposta em termos de eficiência, eficácia e robustez na identificação de cliques máximos em grafos.

A metodologia adotada nesta pesquisa busca fornecer uma análise objetiva e fundamentada da solução proposta, contribuindo para o avanço do conhecimento na área do problema do clique e auxiliando na identificação de possíveis melhorias ou limitações na abordagem proposta.

A. Algoritmo exato

O algoritmo proposto para encontrar o maior clique em grafos é baseado em uma abordagem de força bruta. Ele utiliza a técnica de busca completa para explorar todas as combinações de vértices e determinar o maior clique presente no grafo através de uma matriz de adjacência.

O funcionamento do algoritmo pode ser resumido da seguinte maneira:

1. Inicialmente, o algoritmo considera cada vértice individual como um clique de tamanho 1.
2. Em seguida, para cada vértice, ele verifica se é possível formar um clique maior adicionando outros vértices que estão conectados a ele por arestas.
3. O algoritmo continua esse processo iterativamente, adicionando vértices aos cliques existentes, até que não seja mais possível adicionar nenhum vértice e todos os cliques máximos tenham sido identificados.
4. Ao final, o algoritmo retorna o maior clique encontrado.

A complexidade do algoritmo no pior caso é de $O(m^n)$, onde n é o número de vértices e m é o número máximo de arestas entre os vértices. Essa complexidade decorre do fato de que o algoritmo precisa avaliar todas as combinações possíveis de vértices para encontrar o maior clique.

Portanto, o algoritmo exato proposto por Claudino e Derkian tem uma complexidade computacional elevada, especialmente para grafos grandes. Em casos de grafos com

um número elevado de vértices e arestas, a execução do algoritmo pode se tornar inviável em termos de tempo de processamento. Nesses casos, pode ser necessário recorrer a abordagens aproximadas ou heurísticas para encontrar soluções em tempo viável.

B. Algoritmo aproximado

O algoritmo aproximado proposto por Claudino e Derkian, trata-se de um algoritmo com abordagem gulosa, que marca os cliques já identificados em um vetor para não haver necessidade de utilizar técnicas mais complexas como a de Backtracking. A utilização de um algoritmo aproximado para resolver o problema do clique oferece vantagens e limitações em relação ao algoritmo exato.

As vantagens de um algoritmo aproximado são:

1. **Eficiência:** Em comparação com o algoritmo exato, que possui uma complexidade computacional elevada, um algoritmo aproximado pode ser mais eficiente em termos de tempo de execução. Isso é especialmente relevante para grafos grandes, nos quais a busca exaustiva de todas as combinações possíveis pode se tornar impraticável.
2. **Soluções próximas à ótima:** Embora um algoritmo aproximado não garanta a solução exata do maior clique, ele tende a fornecer soluções próximas à ótima. Isso significa que, em muitos casos, o resultado obtido pelo algoritmo aproximado é suficientemente bom para atender às necessidades práticas da aplicação.

No entanto, é importante considerar as limitações de um algoritmo aproximado:

1. **Subotimalidade:** Como o próprio nome sugere, um algoritmo aproximado produz uma solução aproximada, que pode não ser a solução exata do maior clique. Dependendo do contexto, essa subotimalidade pode ser aceitável ou não.
2. **Sensibilidade aos parâmetros:** Algoritmos aproximados podem depender de parâmetros que afetam a qualidade da solução obtida. A escolha adequada desses parâmetros pode ser um desafio e pode exigir ajustes específicos para diferentes tipos de grafos ou conjuntos de dados.

Portanto, a escolha entre um algoritmo exato e um algoritmo aproximado para resolver o problema do clique depende das necessidades e restrições específicas de cada aplicação. Enquanto o algoritmo exato garante a solução ótima, mas com maior custo computacional, o algoritmo aproximado oferece uma solução próxima à ótima com menor tempo de processamento.

C. Testes

Os testes consistem em instanciar os algoritmos em sabatinas de 13 testes, onde após finalizados, são coletados os dados e tirado a média dos dados. Os testes terão como duas bases de dados fornecidas por Cai. A primeira, BROCK, que contém 6 grafos e foi feita pelos pesquisadores Mark Brockington e Joe Culberson, e GEN que contém 5 grafos e foi desenvolvida por Laura Sanchis:

Grafo	quantidade de vértices	tamanho do maior clique	taxa de densidade de arestas	profundidade
brock200_2	200	12	0.50000	1
brock200_4	200	17	0.65000	0
brock400_2	400	29	0.75000	1
brock400_4	400	33	0.75000	1
brock800_2	800	24	0.65000	2
brock800_4	800	26	0.65000	2

Tabela 1: Tabela de informações sobre os 6 grafos do dataset de Brockington.

Grafo	quantidade de Vértices	tamanho do maior clique
gen200_p0.9_44	200	44
gen200_p0.9_55	200	55
gen400_p0.9_55	400	55
gen400_p0.9_65	400	65
gen400_p0.9_75	400	75

Tabela 2: Tabela de informações sobre os 6 grafos do dataset de Laura Sanchis

D. Análise de complexidade

Nesta seção, realizaremos uma análise da complexidade dos algoritmos abordados para o problema do clique, considerando tanto o algoritmo exato quanto o aproximado.

a. Algoritmo aproximado

$$\sum_{v=0}^n (\sum_{i=0}^n + \sum_{i=0}^{m-1} (\sum_{j=i+1}^m) + \sum_{i=0}^m)$$

$$\sum_{v=0}^n (n + \sum_{i=0}^{m-1} (m - (i+1) - 1) + m)$$

$$\sum_{v=0}^n (\sum_{i=0}^{m-1} (m - i) + m)$$

$$\sum_{v=0}^n (n + (m \sum_{i=1}^m - \sum_{i=1}^m i) + m)$$

$$\sum_{v=0}^n (n + m^2 - (m(m+1))/2 + m)$$

$$\sum_{v=0}^n (n + m^2 - (m^2 + m)/2 + m)$$

$$n \sum_{v=0}^n + m^2 \sum_{v=0}^n - (m^2 + m)/2 \sum_{v=0}^n + m \sum_{v=0}^n$$

Complexidade no *pior caso* = $O(m^n)$. Onde m é o total de vértices e n são os tamanhos dos subgrafos.

b. Algoritmo exato

$$\sum_{v=0}^n (\sum_{i=0}^n + \sum_{i=0}^{m-1} (\sum_{j=i+1}^m) + \sum_{i=0}^m)$$

$$\sum_{v=0}^n (n + \sum_{i=0}^{m-1} (m - (i+1) - 1) + 2m)$$

$$\sum_{v=0}^n (\sum_{i=0}^{m-1} (m - i) + 2m)$$

$$\sum_{v=0}^n (n + (m \sum_{i=1}^m - \sum_{i=1}^m i) + 2m)$$

$$\sum_{v=0}^n (n + m^2 - (m(m+1))/2 + 2m)$$

$$n \sum_{v=0}^n + m^2 \sum_{v=0}^n - (m^2 + m)/2 \sum_{v=0}^n + 2m \sum_{v=0}^n$$

Complexidade no *pior caso* = $O(m^n)$

IV. RESULTADOS E DISCUSSÃO

Nesta seção, apresentaremos os resultados obtidos na avaliação da solução proposta por Claudino e Derkian, comparando-os. Também discutiremos as limitações identificadas na solução e possíveis melhorias a serem consideradas.

Durante a avaliação experimental, aplicamos o algoritmo proposto nos grafos de teste disponibilizados no benchmark de grafos, conforme mencionado anteriormente. Observamos resultados que revelam algumas imprecisões na detecção dos cliques máximos em relação aos valores reais esperados.

DataSets	T. exec. (alg. aprox.)	T. exec (alg exato)	Acurácia	% de erros	> clique encontrado	maior clique (esperado)
brock20_0_2	0.000873	0.000983	100%	0%	10	12
brock20_0_4	0.000820	0.001442	100%	0%	14	17
brock40_0_2	0.003751	0.006357	100%	0%	20	29
brock40_0_4	0.003916	0.006513	100%	15,38 %	20	33
brock80_0_2	0.011577	0.019997	100%	23,07 %	17	24
brock80_0_4	0.011852	0.019980	100%	0%	18	26
gen200_p0.9_44	0.001708	0.002764	100%	23,07 %	32	44
gen200_p0.9_55	0.001756	0.002813	97,14 %	0%	35	55
gen400_p0.9_55	0.006545	0.011759	100%	30,76 923%	39	55
gen400_p0.9_65	0.007464	0.011759	100%	38,46 154%	40	65
gen400_p0.9_75	0.007350	0.013043	100%	15,38 462%	42	75

Tabela 3: Tabela contendo todos os dados obtidos nos testes com os benchmarks de grafos

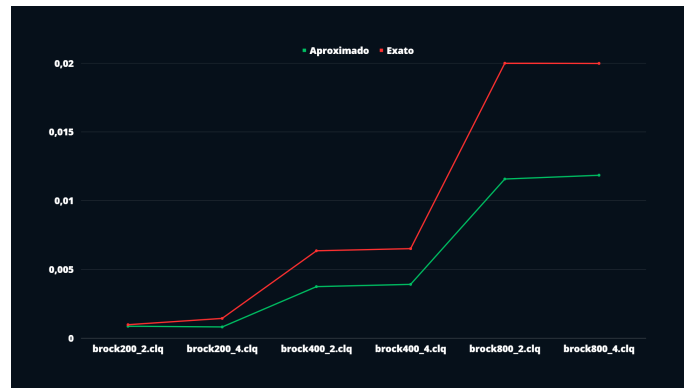


Gráfico 1: Gráfico de comparação de tempo de execução entre o algoritmo aproximado e o exato, em diferentes grafos de testes do dataset brock



Gráfico 2: Gráfico de comparação de tempo de execução entre o algoritmo aproximado e o exato, em diferentes grafos de testes do dataset gen.



Gráfico 3: Gráfico de comparação de cliques entre o algoritmo aproximado e o exato, em diferentes grafos de testes do dataset gen.

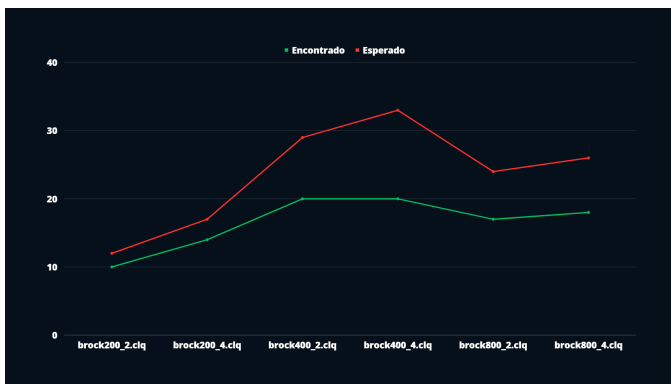


Gráfico 4: Gráfico de comparação de cliques entre o algoritmo aproximado e o exato, em diferentes grafos de testes do dataset brock.

Conforme mostrado nos gráficos 1 e 2, podemos observar que o resultado da comparação entre os dois algoritmos se mostrou condizente com o esperado, suas linhas de desempenho relembram suas funções de custo. Porém, nos gráficos 3 e 4 notou-se uma diferença negativa na detecção do clique máximo, que vai se agravando conforme há mais

cliques nos grafos. Por exemplo, ao analisar o dataset brock200_2.clq, composto por 200 vértices e densidade de 0.50000 de arestas, identificamos um clique máximo de 12 vértices. No entanto, ao aplicarmos o algoritmo, observamos uma diferença de 2 vértices abaixo do valor esperado, obtendo um clique máximo de apenas 10 vértices. De maneira similar, ao analisar o dataset brock200_4.clq, com 200 vértices e densidade de 0.65000 de arestas, identificamos um clique máximo de 17 vértices. No entanto, o algoritmo apresentou uma diferença de 3 vértices abaixo do valor real, encontrando um clique máximo de apenas 14 vértices.

Esses resultados indicam que a solução proposta possui algumas limitações em termos de precisão na detecção dos cliques máximos, especialmente em grafos com maior densidade de arestas. Essas imprecisões podem comprometer a validade dos resultados obtidos e a correta identificação dos cliques máximos. É importante ressaltar que há casos em que a diferença diminui, indicando que pode haver algo afetando a otimicidade do algoritmo, porém, durante os testes não foi detectado a fonte específica que possa estar influenciando nesse resultado, demandando mais testes em trabalhos futuros.

Uma possível melhoria para a solução proposta é a investigação e implementação de mecanismos que visem aprimorar a precisão na detecção dos cliques máximos. Isso pode envolver a análise de diferentes critérios de seleção de vértices, a utilização de técnicas de poda mais eficientes ou a exploração de algoritmos alternativos. Além disso, é importante ressaltar que a avaliação dos algoritmos propostos foi limitada apenas aos benchmarks de grafos selecionados. Seria interessante realizar comparações com outros algoritmos estabelecidos pela comunidade científica, a fim de ter uma visão mais abrangente do desempenho e da eficácia da solução proposta.

Portanto, concluímos que a solução proposta por Bruno Claudino e Felipe Derkian apresenta limitações em termos de precisão na detecção de cliques máximos, especialmente em grafos densos. Recomendamos a investigação de melhorias na solução e a realização de comparações com outros algoritmos para uma análise mais completa e abrangente dos resultados. Essas contribuições podem levar ao aprimoramento da solução proposta e ao avanço do conhecimento na área do problema do clique.

V. CONCLUSÃO

Ao longo deste artigo, abordamos o problema do clique na teoria dos grafos, destacando sua importância e aplicações práticas em diversas áreas. O objetivo principal da pesquisa foi avaliar a solução proposta, que apresenta um algoritmo para encontrar cliques máximos em grafos. Realizamos uma revisão dessa solução e selecionamos um benchmark de grafos consolidado para executar os testes experimentais.

Com base na análise dos resultados, concluímos que o algoritmo proposto apresenta resultados fracos na identificação de cliques máximos em diferentes tipos de grafos. Esse resultado mostra que encontramos algumas limitações e possíveis melhorias, como aprimorar a eficiência computacional em grafos de grande escala.

Para futuras pesquisas, sugerimos explorar novas abordagens algorítmicas para a identificação de cliques máximos, como algoritmos heurísticos ou técnicas de programação inteira. Além disso, é importante investigar o desempenho desses algoritmos em cenários específicos, como redes sociais ou sistemas biológicos, a fim de fornecer soluções mais eficientes e adaptadas a diferentes contextos.

Em suma, esta pesquisa contribui para o avanço do conhecimento na área do problema do clique, oferecendo uma análise crítica da solução proposta e sugerindo caminhos promissores para futuras pesquisas nessa área. Esperamos que essas descobertas inspirem e guiem pesquisadores interessados em lidar com o problema do clique e suas aplicações práticas.

VI. REPOSITÓRIO

Link para acesso ao repositório do projeto:

https://github.com/GuilhermeBn198/GuilhermeLucasKelvinAraujo_ProblemaDoClique_AA_RR_2023

VII. REFERÊNCIAS

- [1] COOK, Stephen. **The complexity of theorem proving procedures.** In: Proceedings of the Third Annual ACM Symposium on Theory of Computing. [S.l.: s.n.], 1971. p. 151-158.
- [2] CAI, Shaowei. **Datasets Combinatorial Optimization on Graphs.** 2010. Disponível em: <<https://lcs.ios.ac.cn/~caisw/graphs.html>>. Acesso em: 20/06/2023
- [3] DERKIAN, Felipe; CLAUDINO, Bruno C. **Trabalho Final de Análise de Algoritmos - O Problema do Clique.** 2018. Disponível em: <https://github.com/felipekian/Felipe.Derkian-Bruno.Cesar_finalProject_AA_RR_2018>. Acesso em: 22 jun. 2023.
- [4] HARARY, Frank; ROSS, Ivan C. "A procedure for clique detection using the group matrix." *Sociometry*, v. 20, n. 3, p. 205-215, 1957. Disponível em: <<https://www.jstor.org/stable/2785673>>. doi: 10.2307/2785673.
- [5] LUCE, R. Duncan; PERRY, Albert D. **A method of matrix analysis of group structure.** *Psychometrika*, v. 14, n. 2, p. 95-116, 1949. Disponível em: <https://www.jstor.org/stable/18152948>. doi: 10.1007/BF02289146.
- [6] KARP, M. Richard. **Reducibility among combinatorial problems.** In: MILLER, R. E.; THATCHER, J. W. (Eds.). *Complexity of Computer Computations.* New York: Plenum, [s.d.], p. 85-103.
- [7] CORMEN, Thomas H. et al. **Algoritmos: Teoria e Prática.** Tradução de Arlete Simille Marques. Rio de Janeiro: Elsevier, 2012.

VIII. AUTORES



Guilherme L. P. Bernardo (Graduando de Ciência da Computação, UFRR) Nascido em Boa Vista, Roraima, Brasil. Técnico de informática pelo Instituto Federal de Roraima, em Boa vista, Roraima, Brasil; pelo Instituto Federal de Roraima, no ano de 2018. Graduando

em Ciência da computação pela Universidade Federal de Roraima, em Boa Vista, Roraima, Brasil.

Sua área de estudo é voltada à gerência de softwares, análise de código e análise de dados.

Já foi estagiário de Suporte em TI pela Companhia de Águas e Esgotos de Roraima. Atual bolsista do PIBIC-CNPq no projeto MAP2CHECK, desenvolvedor back-end freelancer trabalhando para a DevOps(Maranhão), diretor de eventos da Associação Atlética Acadêmica Nexus e presidente do centro acadêmico de ciência da computação da UFRR. Interessado na área de verificação de software e análise de dados.



Kelvin A. Frreira (Graduando de Ciência da Computação, pela Universidade Federal de Roraima - UFRR). Nascido em Roraima, em maio de 2001. Possui formação técnica em Informática e na área de Eletrônica.

Seu campo de estudo é voltado para a área de Análise de Dados, Eletrônica, Sistemas IOT e áreas afins.

Atualmente, Kelvin trabalhou como Desenvolvedor Front-End Júnior no Ministério Público do Estado de Roraima - MPRR. Anteriormente, atuou como Analista de Dados Júnior na TRACTIAN.

Além disso, ele é secretário da Associação Atlética Acadêmica Nexus e também secretário do Centro Acadêmico de Ciência da Computação - CACC/UFRR.