Problema Sanduíche aplicado às classes de grafos cordais e split

Renan G. da Silva[1], Natália F. da Silva[1]

[1]Departamento de Ciência da Computação – Universidade Federal Rural do Rio de Janeiro (UFRRJ)

renan.gdsilva@gmail.com, natalia-321s@outlook.com

Resumo. *Este trabalho tem como objetivo apresentar e descrever formalmente o problema sanduíche, as classes dos grafos cordais e split e como ambas as classes elas são aplicadas ao problema proposto.*

Abstract. *This work aims to present and describe formally the Sandwich Problem, the chordal and split graph classes and how either classes are applied in purpose problem.*

1. Introdução

Dados dois grafos $G_1 = (V, E_1)$ e $G_2 = (V, E_2)$, o problema sanduíche busca encontrar uma determinada propriedade Π em um grafo $G' = (V, E')$, onde $E_1 \subseteq E' \subseteq E_2$ onde se quer determinar se o grafo G' pertence a uma família ou classe de grafos, como por exemplo se o grafo ele é cordal ou não. O problema sanduíche vem a ser uma generalização do problema de reconhecimento, onde foi introduzido por Golumbic, Kaplan e Shamir [Golumbic et al. 1995]. Nesse trabalho será explicado com maiores detalhes a respeito o problema sanduíche e a aplicação do problema para a classes dos grafos split e os grafos cordais.

O problema sanduíche possui aplicações nas áreas de mapeamento físico do DNA, raciocínio temporal, sincronização de processos paralelos, árvores filogenéticas, sistemas esparsos de equações lineares [Couto 2016].

O trabalho para esta atividade acadêmica tem como objetivo definir formalmente o problema sanduíche, as classes dos grafos cordais e split e o problema sanduíche para essas duas classes. No primeiro capítulo terá, além da descrição do problema brevemente explicado, algumas definições e conceitos básicos que serão discutidos em capítulos posteriores para melhor compreensão, no capítulo 2 serão definidos formalmente as classes dos grafos cordais e split, no capítulo 3 entraremos em detalhes sobre o problema sanduíche e o capítulo 4 com conclusão do trabalho.

1.1. NP-completude

Alguns problemas aplicados ao problema sanduíche são classificados como NP-completos, para os grafos cordais o problema é considerado NP-completo, já para os grafos $split$ existe um algoritmo em tempo polinomial.

Com relação às classes de NP-completude, dizemos que um problema é P caso exista algum algoritmo que resolva um determinado problema Π em tempo polinomial,

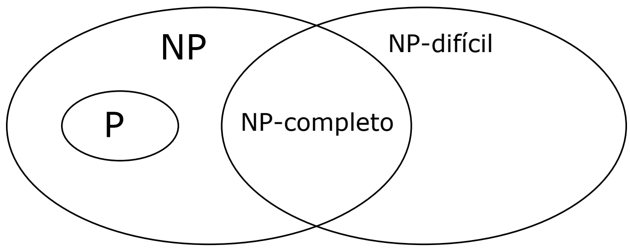

Figura 1. Diagrama das classes de problemas

ou seja, que o tempo de execução para resolvê-lo é proporcial ao tamanho de sua entrada, algoritmos que classificamos como eficientes possuem complexidade na ordem polinomial, como $O(n^2), O(nlog_n), O(n)$. Um problema é considerado NP quando para um determinado problema Π existe uma justificativa SIM para o certificado do problema e um algoritmo complexidade polinomial no passo de reconhecimento, caso não exista, não necessariamente esse problema não pertence a classe NP. No problema do caminho hamiltoniano, por exemplo, uma justificativa para seu problema é encontrar um ciclo no grafo tal que ao percorrê-lo, não se repita o caminho por uma aresta, um reconhecimento desse problema seria encontrar a resposta NÃO para o seu certificado, por exemplo, caso não haja um ciclo hamiltoniano qualquer no grafo, então deverão ser verificados todos os ciclos no grafo para fazer essa verificação, para o problema do ciclo hamiltoniano, não é conhecido até então um algoritmo eficiente para resolver o problema em tempo polinomial.

Os problemas da classe NP-completos são os mais difíceis da classe NP e provavelmente não façam parte da classe dos problemas P, essa classe de problemas são um subconjunto dos problemas NP, na figura 1 podemos observar um diagrama com as classes dos problemas. Os problemas NP-difíceis são problemas tão difíceis quanto os NP-completos, com a diferença que a saída do problema NP-difícil torna-o um problema de otimização, maiores detalhes a respeito dessa classe podem ser consultados em [Szwarcfiter 2018].

Existe um problema em aberto envolvendo as classes de algoritmos P e NP onde se quer encontrar a resposta se $P = NP$. Obviamente, ainda não descobriu se existe algum problema da classe NP que seja intratável, ou seja, que não há um algoritmo que o resolva em tempo polinomial. A resposta para esse problema ainda continua em aberto mas que há evidências que direcionam para $P \neq NP$ devido a classe NP possuir muitos problemas, e mesmo assim não foi encontrado um algoritmo em tempo polinomial que os resolva.

1.2. Definições

1.2.1. Esquema de eliminação perfeita

Um vértice v é considerado simplicial se todos os seus vizinhos $N(v)$ induzirem uma clique maximal do grafo original. Por exemplo, o vértice a do grafo da figura 2 não é simplicial, pois seu vizinho b não induz uma clique maximal.

Um conjunto de vértices ordenados $\phi = v_1, v_2, ..., v_n$ é um esquema de eliminação perfeita se cada vértice v_i removido do conjunto é um vértice simplicial do grafo. Por exemplo, os vértices da figura 1 uma vez ordenados num conjunto $\phi = a, b, c, d$, se para cada vértice removido eles forem simpliciais, então dizemos que existe um esquema de eliminação perfeita.

1.2.2. Grafos Perfeitos

Um grafo G é dito perfeito se o número cromático de cada subgrafo induzido for igual ao tamanho da maior clique do subgrafo induzido.

1.2.3. Grafos Livres

Um grafo é dito livre de uma determinada propriedade Π, que podemos usar como exemplo o grafo K_3, se não existe em um grafo $G = (V, E)$ um subgrafo com essa propriedade, caso não exista, chamamos esse grafo de $livres$ de triângulos, ou em termos mais práticos, podemos usar a notação $\{K_3\}$-$free$.

1.3. Conjunto Independente

Um conjunto de vértices é dito independente se todos os vértices do conjunto formam componentes desconexas, ou seja, todos os vértices do grafo são isolados, não tendo entre eles nenhuma vizinhança em comum.

2. Grafos Split e Cordais

2.1. Grafos Split

A classe dos grafos $split$ fazem parte do problema de particionamento de grafos, onde se quer particionar o grafo em um conjunto $\phi = v_1, v_2, ..., v_n$ de vértices com algumas propriedades, essas propriedades podem ser por restrições internas ou externas, um exemplo de restrição interna seria uma propriedade dos grafos $split$, onde uma partição deve ser um conjunto independente e uma restrição externa seriam restrições entre subconjuntos [Couto 2016]. Os grafos split são grafos que fazem parte do problema de particionamento, onde o grafo é particionado em um conjunto independente e uma clique. Brandstädt [Brandstädt 1996] generalizou a classe dos grafos split e introduziu a classe dos $grafos$-(k, l), onde se quer particionar um grafo G em k conjuntos independentes e l cliques, dizemos então que os grafos $split$ são $grafos$-(k, l), onde $k = l = 1$. Brandstädt em [Brandstädt 1996], [Brandstädt and Szymczak 1998], [Brandstädt 2005] provou que, nos casos onde $k \geq 3$ ou $l \geq 3$ o problema de particionamento pertence a classe dos problemas NP-completos e mostrou um algoritmo em tempo polinomial para o reconhecimentos dos $grafos$-$\{(2, 1), (1, 2), (2, 2)\}$.

Os grafos split pertencem à classe dos grafos (k, l), logo, são uma subclasse dos grafos $cordais$ e $perfeitos$. Um grafo é $split$ se e, somente se, for livre de C_4, C_5 e $2K_2$, portanto, essa é uma caracterização dessa classe. Na figura 2 vemos um exemplo de um grafo split, onde o grafo pode ser particionado em um K_5 e um conjunto independente.

2.2. Reconhecimento de Grafos Split

Sabemos que um grafo split é um grafo $\{C_4, C_5, 2K_2\}$-$free$, portanto, essas três propriedades classificamos como grafos proíbidos para a classe $split$, um algoritmo para

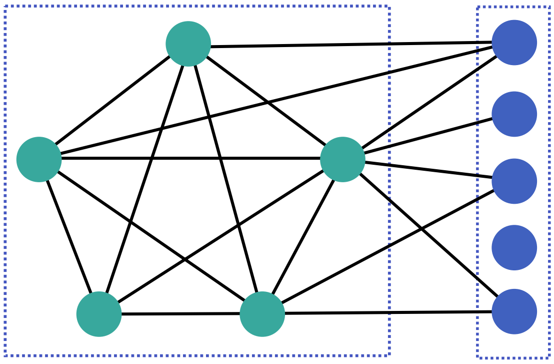

Figura 2. Grafo Split

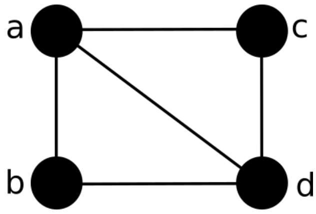

Figura 3. Grafo cordal

reconhecimento dos grafos *split* é verificar se o grafo G é livre dessas três propriedades, caso a afirmação seja verdadeira, então o grafo é *split*.

2.3. Grafos Cordais

Dado um grafo $G = (V, E)$ é cordal quando o grafo possui todos os seus ciclos com pelo menos tamanho 4 que possuam uma corda em vértices não consecutivos com uma distância ímpar entre os vértices. Grafos cordais também são conhecidos como grafos $triangularizados$. Nas figuras 3 e 4 estão alguns exemplos de grafos cordal e não cordal, observamos que o grafo da figura 3 possui uma corda, que seria o vértice $\overline{ad}$ e o grafo não cordal da figura 4 não possui as arestas $\overline{ac}$ e $\overline{bd}$ no ciclo induzido a, b, c, d, a.

Os grafos cordais são uma subclasse dos grafos perfeitos. Um grafo é dito perfeito quando $\chi(G) = \omega(G)$.

Assim como os $grafos$-(k, l), os grafos cordais também podem ser chamados de (k, l)-$cordais$ com a mesma propriedade do problema de particionamento, onde se quer particionar um grafo em k conjuntos independentes e l cliques.

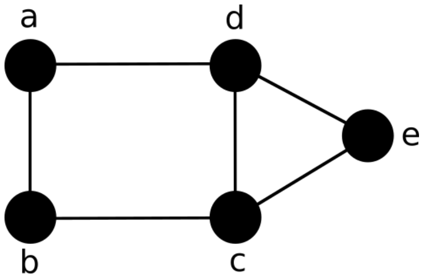

Figura 4. Grafo não cordal

Há também a definição de grafo $cordal$-(k,l), onde o grafo possui a mesma propriedade do problema de particionamento proposto por Brandstadt, de se particionar o grafo em k conjuntos independentes e l cliques, ou seja, é um $grafo$-(k,l) e ser um grafo $cordal$.

A propriedade ser cordal é uma propriedade hereditária, portanto, qualquer subgrafo induzido de um grafo $cordal$ também será um grafo $cordal$.

2.3.1. Reconhecimento de Grafos Cordais

O reconhecimento dos grafos cordais pode ser feito através do algoritmo de busca lexicográfica (LEX-BFS). O LEX-BFS é um algoritmo que se assemelha a busca em largura, porém com critérios de ordenação para exploração dos vértices de um grafo, maiores detalhes a respeito desse algoritmo podem ser conferidos no livro do Szwarcfiter [Szwarcfiter 2018].

Lemma 2.1 *Seja $G = (V, E)$ um grafo cordal aplicado ao algoritmo de busca em largura lexicográfica. Então a sequência S de vértices v ordenados decrescentemente segundo $largura(v)$ é um esquema de eliminação perfeita*

O algoritmo de busca lexicográfica ($lex - BFS$) ordena todos os vértices do grafo em ordem descrecente, os grafos cordais possuem esquema de eliminação perfeita, portanto, executando o algoritmo $lex - BFS$ sobre um grafo cordal G, obtemos os vértices $\phi = v_1, v_2, ..., v_n$ em ordem descrescente, após obter a lista de vértices ordenada, deverá ser verificado se os vértices induzem uma clique, ou seja, se o vértice v_i é simplicial, caso a afirmação seja verdadeira, então os vértices do conjunto retornado ϕ é um esquema de eliminação perfeita. Esse algoritmo de reconhecimento de grafos cordais possui a complexidade $O(nm)$.

3. Problemas Sanduíche

O problema sanduíche é uma generelização do problema de reconhecimento de classes de grafos, onde se quer determinar se, dados dois grafos G_1 e G_2, sendo G_2 um supergrafo de G_1, se existe algum grafo G' que possua a mesma propriedade dos grafos G_1 e G_2, uma propriedade Π pode ser, por exemplo, se um grafo pertence a classe dos grafos cordais ou threshold.

O problema foi introduzido por [Golumbic et al. 1995] formalmente da seguinte forma:

PROBLEMA SANDUÍCHE PARA UMA PROPRIEDADE Π

$Entradas$: Grafos $G_1 = (V, E_1)$ e $G_2 = (V, E_2)$, onde $G_2 > G_1$, $E_2 \subseteq E_1$.

$Saida$: Existe algum grafo G_s, onde $E_1 \subseteq E_s \subseteq E_2$ que satisfaça a propriedade Π?

Quando a entrada para o problema sanduíche onde $G_1 = G_2$, o problema torna-se um problema de reconhecimento, pois ambos os grafos possuem o mesmo conjunto de arestas, portanto, não é do interesse tratar esse caso específico.

Dizemos que o conjunto de vértices $E(\bar{G_2})$ são o conjunto de vértices proibidos para G', $E(G_1)$ sendo o conjunto de arestas forçadas, ou seja, que obrigatoriamente o grafo encontrado deverá conter esse conjunto de arestas e o conjunto $E(G_2)$ de arestas opcionais [Couto et al. 2012].

Existe algoritmo que resolva em tempo polinomial o problema sanduíche para a classe dos grafos split, para a classe dos grafos (k, l), que são uma generalização do problema de particionamento, o problema torna-se NP-completo para $k \geq 3$ ou $l \geq 3$, assim como seu problema de reconhecimento [Golumbic et al. 1995].

Há também outros problemas sanduíches que são resolvidos em tempo polinomial, que são os casos dos grafos $threshold$ e os $cografos$, como mostrado por Kaplan, Golumbic e Shamir [Golumbic et al. 1995]. Na classe dos grafos P_4-$sparse$, Dantas, Klein, Mello e Morgana [Klein et al. 2009] apresentaram um algoritmo para resolver problema em tempo polinomial, os grafos P_4-$sparse$ pertencem a classe dos $cografos$, essa classe possui um algoritmo de reconhecimento em tempo polinomial utilizando o método de decomposição modular. Sua complexidade para o problema sanduíche é $O(\mid V \mid^2 (\mid V \mid + \mid E^1 \mid + \mid \bar{E}^2 \mid))$, como a proposta desse trabalho não é tratar especificamente desse problema, o algoritmo pode ser consultado no artigo citado.

Alguns problemas sanduíches aplicados às classes de grafos que pertencem a classe dos grafos perfeitos estão na classe dos problemas NP-completos como a classe dos grafos $cordais$, fortemente cordais e permutação [Golumbic et al. 1995].

Classes de grafos que tenham seu problema de reconhecimento na classe dos problemas NP-completos, o seu problema sanduíche correspondente também pertencerá a essa classe de problemas [Couto 2016].

3.1. Grafos Cordais e Split

O problema sanduíche para grafos split possui um algoritmo em tempo polinomial com a complexidade de $O(\mid V \mid + \mid E^1 \mid + \mid \bar{E}^2 \mid)$ já proposto por Golumbic, Kaplan e Shamir [Golumbic et al. 1995].

É conhecido que os grafos cordais são uma subclasse dos grafos $perfeitos$, alguns grafos que são subclasse dos grafos $perfeitos$ tem seu respectivo problema sanduíche na classe dos problemas NP-$completos$, grafos como de permutação, comparabilidade também pertencem a essa classe [Golumbic et al. 1995]. Alguns trabalhos tem sido desenvolvidos com o intuito de encontrar algoritmos eficientes e prova de NP-$completude$ para algumas classes específicas dos grafos $cordais$, como por exemplo o trabalho desenvolvido por Couto [Couto et al. 2012], onde se mostrou que os grafos $(2, 1)$-$cordais$ pertencem a classe dos problemas NP-$completos$ com o intuito de se desenvolver um algoritmo para o problema sanduíche, nesse trabalho, as ferramentas utilizadas para essa prova através das definições dos grafos cordais não foram satisfatórias e foram utilizados recursos da classe dos grafos $fortementecordais$ para esse objetivo.

4. Conclusão

O objetivo desse trabalho foi uma breve introdução a respeito do problema sanduíche, bem como a aplicação do problema para a classe dos grafos cordais e split, foi mostrado também que existe problemas sanduíche que pertencem a classe dos problemas

NP-completo onde se existe um esforço pela comunidade acadêmica de ciência da computação e matemática para encontrar algoritmos eficientes para esses problemas citados, trabalhos os quais foram citados como fonte.

5. Referências Bibliográficas

Referências

Brandstädt, A. (1996). Partitions of graphs into one or two independent sets and cliques. pages 47–54.

Brandstädt, A. (2005). *Corrigendum.*

Brandstädt, A. and Szymczak, V. B. L. (1998). The complexity of some problems related to graph 3-colorability. discrete applied mathematics. pages 89(1–3):59–73.

Couto, F. V. D. (2016). Complexidade dos problemas sanduíche e probe para subclasses de grafos-(k,l).

Couto, F. V. D., Faria, L., and Klein, S. (2012). A complexidade do problema sanduíche para grafos fortemente cordais(2,1).

Golumbic, M., Kaplan, G., and Shamir, R. (1995). Graph sandwich problems. pages 449–473.

Klein, S., Mello, C. P., Morgana, A., and Dantas, S. (2009). The graph sandwich problem for p4-sparse graphs. page 3664–3673.

Szwarcfiter, J. L. (2018). *Teoria Computacional de Grafos: os Algoritmos*. Rio de Janeiro.