

Estudo e Implementações do Número de Envoltória nas Convexidades P_3 e Geodésica

Arthur O. B. Lacerda¹, Erika M. M. Coelho²

Instituto de Informática
Universidade Federal de Goiás

email:{cc122301,erikamorais}@inf.ufg.br

Resumo. *Seja G um grafo e $S \subseteq V(G)$. Se não existem vértices fora de S que possuem dois vizinhos em S , então S é um conjunto P_3 -convexo. Se S contém todos os vértices pertencentes aos caminhos mínimos entre quaisquer dois vértices de S , então S é um conjunto geodético-convexo. A envoltória convexa de S , $H(S)$, é o menor conjunto convexo contendo S e o número de envoltória convexa, é o menor S tal que $H(S) = V(G)$. Neste trabalho, estudamos e implementamos algoritmos de tempo polinomial para o cálculo do número de envoltória P_3 e Geodética e caracterizamos o número de envoltória P_3 para grafos *slit*.*

Palavras Chave: Número de envoltória, Convexidade P_3 , Convexidade Geodésica.

1. INTRODUÇÃO

O contexto atual global de comunicação, possui um fluxo massivo de informações, opiniões e ideias. Sendo estes transmitidos por meios reais e virtuais, onde vários setores podem ser afetados fortemente por este contexto. Por exemplo: Mobilização para campanhas de doações, ajudando milhares de pessoas ao redor de todo o mundo; campanhas publicitárias anunciando um novo item no mercado, que gera grande repercussão nos meios de comunicação; vídeos que se tornam populares na internet gerando milhões de visualizações e comentários; empresas que tem a sua reputação comprometida devido a grande divulgação de uma má conduta cometida pela mesma; e até mesmo mudança de cultura e comportamentos locais.

Tudo isso torna-se grande, graças a alta velocidade de disseminação da informação, e também a influência que umas pessoas exercem sobre as outras, se tratando daquelas pessoas que são mais próximas a você, ou que apenas possuem sua admiração e reconhecimento. Seguindo essa lógica, suponhamos um cenário, onde uma notícia, um novo hit de sucesso ou uma nova tendência, tenha sido lançada. Porém apenas algumas pessoas se manifestaram por meio de uma rede social divulgando este conteúdo. Mas quando duas, ou mais pessoas influentes para um certo indivíduo já divulgaram este conteúdo, estas são suficientes para que ele seja influenciado e também o divulgue. Dessa maneira, muitas pessoas podem compartilhar o conteúdo, mesmo que o grupo inicial de influência seja pequeno, devido a propagação da informação.

Esse cenário, tem aplicação computacional relevante, que ilustra um caso de aplicação de grafos. Informalmente falando, um grafo é uma abstração que permite representar o relacionamento entre pares de elementos. Os elementos são chamados vértices e os relacionamentos arestas.

Para obter um modelo de grafo, considere que as pessoas serão representadas por vértices, e as relações de influência entre as pessoas, pelas arestas. Dessa maneira podemos determinar importantes aspectos, utilizando de técnicas em teoria dos grafos. Como no exemplo especificado, dado este conjunto inicial de pessoas que divulgaram a informação, qual seria a quantidade total de influenciados em determinado grupo? Ou até mesmo, determinar quais as pessoas que de-

Revisado pelo orientador

¹Bolsista

²Orientador

vem pertencer ao conjunto inicial, para que essa informação seja espalhada por todos os outros usuários da rede, de tal forma que esse grupo inicial de pessoas, seja o menor possível.

A resposta para esse e outros problemas, se encontra na convexidade em grafos. Para entender melhor, abordaremos rapidamente algumas definições. Um grafo G é um par ordenado $(V(G), E(G))$, onde $V(G)$ é um conjunto finito não vazio cujos elementos são denominados *vértices*, e $E(G)$ é um conjunto de pares de elementos de $V(G)$, os elementos de $E(G)$ são denominados *arestas*. Neste trabalho, utilizamos as definições e notação usuais em Teoria dos Grafos.

Uma *convexidade* sobre $V(G)$ é uma coleção \mathcal{C} de subconjuntos de $V(G)$ tal que $\emptyset, V(G) \in \mathcal{C}$ e \mathcal{C} é fechado sobre interseções. Os conjuntos em \mathcal{C} são chamados *conjuntos convexos* e a *envoltória* convexa $H_{\mathcal{C}}(S)$ em \mathcal{C} de um conjunto $S \subseteq V(G)$ é o menor conjunto convexo em \mathcal{C} contendo S . Se $H_{\mathcal{C}}(S) = V(G)$ em \mathcal{C} , para $S \subseteq V(G)$, dizemos que S é um *conjunto de envoltória*. O *número de envoltória*, $h_{\mathcal{C}}$ em \mathcal{C} , é a cardinalidade do menor conjunto de envoltória de G .

Algumas convexidades, dentre elas a P_3 e a Geodésica, são definidas através de um conjunto \mathcal{P} de caminhos em grafos. Neste caso, um subconjunto $S \subseteq V(G)$ é convexo precisamente quando S contém todos os vértices pertencentes aos caminhos de \mathcal{P} .

Quando \mathcal{P} é o conjunto de todos os caminhos mínimos em G , então \mathcal{C} é a *convexidade geodésica* [9–12]. Já a convexidade utilizada para resolver o problema relatado acima, é obtida considerando \mathcal{P} como o conjunto de todos os caminhos com três vértices, chamada de *convexidade P_3* [3–6]. Denotaremos por $h_{P_3}(G)$ e $h_g(G)$ o número de envoltória de G , respectivamente, nas convexidades P_3 e geodésica.

Determinar se um grafo G tem um conjunto de envoltória de tamanho no máximo k , em ambas as convexidades, é um problema NP-completo [3, 9]. Os problemas NP-completos são considerados problemas super polinomiais e, para problemas nesta classe, não se conhece algoritmo de tempo polinomial para resolvê-los e ainda não se provou que não pode existir tal algoritmo polinomial. Já o cálculo do fecho convexo de um conjunto S , em ambas as convexidades, pode ser feito em tempo polinomial. Para a convexidade P_3 , é possível determinar em tempo polinomial $h_{P_3}(G)$ quando G é um grafo cordal [3], um cografo [7] e quando G é uma floresta [8]. Considerando a convexidade geodésica, é possível determinar em tempo polinomial o número de envoltória para as classes de split e cografos [9].

Neste trabalho, estudamos e implementamos o cálculo do número de envoltória convexa P_3 e geodésica para grafos em geral. Estudamos, implementamos e exibimos descrições detalhadas dos algoritmos para o cálculo do número de envoltória P_3 para a classe de grafos cordais, cografos e florestas e para o cálculo de h_g para grafos split e cografos. Motivados pela caracterização e algoritmo para o cálculo de h_g para grafos split, caracterizamos e apresentamos um algoritmo de tempo polinomial para o cálculo de $h_{P_3}(G)$ para G um grafo split. Por fim, comparamos os valores de h_{P_3} e h_g para a classe dos grafos split para verificar a diferença do número de envoltória em diferentes convexidades.

Na seção 2, apresentaremos as definições e termos utilizados neste trabalho para deixar o leitor familiarizado com a linguagem e notação adotada. Em seguida nas seções 3 e 4, abordaremos as convexidades P_3 e geodésica e apresentaremos algoritmos de tempo polinomial para o cálculo do h_{P_3} de grafos cordais, florestas, cografos e split e para o cálculo do h_g para cografos e split. Finalmente, na seção 5, apresentaremos detalhes de implementação utilizados ao longo do desenvolvimento deste trabalho e exibiremos uma tabela comparativa entre h_{P_3} e h_g para grafos split.

2. PRELIMINARES

Apresentaremos nesta seção, definições comuns da teoria dos grafos, que podem ser encontradas em [1], e também definições e notações das classes de grafos que serão discutidas neste artigo.

Um *grafo* G é um par ordenado $(V(G), E(G))$ que consiste em um conjunto finito, não vazio de *vértices* $V(G)$, e um conjunto de subconjuntos de $V(G)$ com cardinalidade dois, chamadas *arestas*, tendo a notação de $E(G)$. Se os vértices u e v , são conectados por uma aresta, esta aresta é denotada por uv . Se a aresta uv existe, então u é adjacente a v . Um grafo *trivial*, é um grafo que possui apenas um vértice.

Um *laço* é uma aresta que conecta um vértice u a ele mesmo. Arestas *paralelas* são arestas que conectam o mesmo conjunto de vértices. Um grafo *simples* é um grafo que não possui arestas paralelas. Um grafo *não direcionado* é um grafo em que se uv existe em $E(G)$, então vu também existe. Neste caso, trataremos a aresta uv e vu como uma só.

O *grafo complementar* de G , denotado \bar{G} , possui o conjunto de vértices $V(G)$ de G , e o conjunto de arestas é composto por todas as arestas que não existem em G , ou seja, se a aresta uv não pertence $E(G)$, então ela pertence a $E(\bar{G})$. Da mesma forma, todas arestas que existem em G , não existem em \bar{G} .

Um *subgrafo* de um grafo G é um grafo H tal que $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$, denotado por $H \subseteq G$. Seja V' um subconjunto não vazio de $V(G)$. O subgrafo de G cujo conjunto de vértices é V' e cujo conjunto de arestas consiste de todas as arestas de G cujas pontas pertencem a V' é chamado de subgrafo de G *induzido* por V' . Denotamos por $G[V']$ e dizemos que $G[V']$ é um *subgrafo induzido* de G .

O grafo $G - v$ é um subgrafo induzido $G[V(G) - v]$. Se H é subgrafo de G , $G - H$ é o grafo obtido pela remoção de todos os vértices de $V(H)$ de G , ou seja $G[V(G) - V(H)]$.

Um *grafo completo* K_n é um grafo simples com n vértices no qual há uma aresta entre cada par de vértices. Uma *clique* de G é um subgrafo induzido de G que é um grafo completo. Um vértice *universal* é um vértice que é adjacente a todos os outros vértices do grafo ao qual ele pertence.

O número de vértices adjacentes a um vértice v é o *grau* de v , denotado por $d(v)$. Um vértice u é dito *isolado* se $d(u) = 0$.

Um subconjunto $S \subseteq V(G)$ é um *conjunto independente* se todos os vértices de S são dois a dois não adjacentes. Um conjunto independente S é *maximal* se S não é subconjunto próprio de nenhum outro conjunto independente de G e é *máximo* se nenhum outro conjunto independente S' de G satisfaz $|S'| > |S|$.

A *vizinhança aberta* ou simplesmente *vizinhança* de um vértice $v \in V(G)$, é dada pelo conjunto de vértices adjacentes a ele, denotada por $N(v)$. A *vizinhança fechada* de um vértice $v \in V(G)$ é o conjunto de vértices adjacentes a v união com v , denotada por $N[v]$. Denotamos por $N_S(v) =$, os vértices adjacentes a v que pertencem a um conjunto S . Um vértice v é dito *simplicial*, se $N(v)$ induz uma clique em G .

Um *caminho* em G é uma sequência finita e não vazia de vértices distintos $W = v_0, v_1, \dots, v_k$, tal que, para $1 \leq i \leq k$, $v_{i-1}, v_i \in E(G)$. Denotamos por P_n um caminho com n vértices. Dizemos que $W = v_0, v_1, \dots, v_k$ é um *ciclo* se v_0, v_1, \dots, v_{k-1} for um caminho e se $v_0 = v_k$. Denotamos por C_n um ciclo com n vértices.

Um grafo é *conexo* se para todo par de vértices $\{u, v\}$ de G , existir um caminho de u até v , caso contrário, o grafo é dito *desconexo*. Cada subgrafo maximal conexo de G é denominado uma *componente* de G . Se G é conexo, e $G - v$ é desconexo, então v é dito *vértice de corte*. Um *conjunto de corte* S , é um conjunto de vértices de um grafo conexo G , onde $G - S$ é desconexo. Um grafo G é k - *conexo* se o tamanho mínimo de um conjunto de corte for pelo menos k .

Uma *floresta* é um grafo acíclico. Uma floresta conexa é uma *árvore*. Os vértices de grau 1 de uma árvore, são denominados *folhas*.

Um *bloco* de um grafo G é um subgrafo maximal 2-conexo de G , e um *bloco final* é um bloco que possui apenas um vertice de corte de G .

Sejam u e v dois vértices de um grafo conexo G . A distância, $d(u, v)$ de u até v é o comprimento (número de arestas) do menor caminho de u até v em G .

Um *isomorfismo* dos grafos G e H , é uma bijeção $f : V(G) \rightarrow V(H)$, de tal forma que para todos vértices u e v de G , são adjacentes se e somente se $f(u)$ e $f(v)$ são adjacentes em H .

Considere C uma convexidade sobre o grafo G e \mathcal{P} um conjunto de caminhos em G . O *intervalo fechado* entre os vértices u e v , denotado por $I[u, v]$, possui u, v e todos os vértices que pertencem a um caminho \mathcal{P} entre eles. Para $S \subseteq V(G)$, denotamos o intervalo de S por $I[S] = \bigcup_{u, v \in S} I[u, v]$. Seja $I^0[S] = S$, $I^1[S] = I[S]$ e $I^k[S] = I[I^{k-1}[S]]$, para $k \geq 2$. A partir de algum termo p , a sequência será constante e quando $I^p[S] = I[I^{p-1}[S]]$ obtém-se a envoltória convexa de S . Assim, temos uma forma iterativa de calcular $H(S)$.

3. CONVEXIDADE P_3

Nesta sessão, apresentaremos os algoritmos de tempo polinomial para o cálculo do número de envoltória convexa para a convexidade P_3 para a classe de grafos cordais, cografos e florestas. Iniciaremos, apresentando um algoritmo para cálculo no número de envoltória h_{P_3} para grafos gerais.

Tendo um grafo G e um subconjunto S , para encontrar a envoltória convexa, faremos n iterações de I sobre S , até que $I^n[S] = I^{n-1}[S]$.

Dado o grafo G , representado na Figura 1(a), e $S = \{c, d, f\}$. Temos que, $S \subseteq H(S)$. Como g possui c e f em sua vizinhança, então $g \in I[S]$ como mostrado na Figura 1(b). Como b possui c e g em sua vizinhança, então $I^2[S] = \{b, c, d, f, g\}$, como mostrado na Figura 1(c). Finalmente, como $I^2[S] = I^3[S]$, então $H(S) = \{b, c, d, g, g\}$.

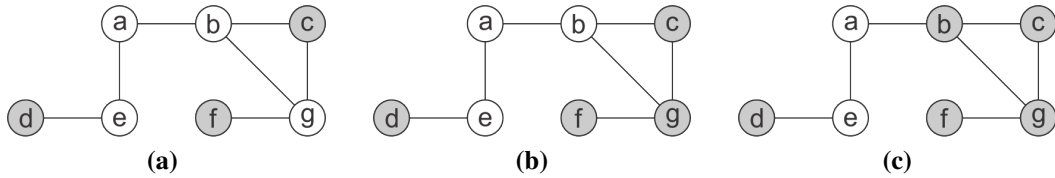


Figura 1: Cálculo da envoltória convexa P_3 de S no grafo G : (a) Estágio inicial, apenas S marcado. (b) $I[S]$. (c) $I^2[S] = H(S)$.

Seguimos então para o algoritmo do cálculo da envoltória convexa P_3 para grafos gerais.

Seja um grafo G e um subconjunto $S \subseteq V(G)$. Para todo $v \in V(G)$ será atribuído um rótulo $l(v)$, onde $l(v) = 0$ indica que o vértice está contaminado e $l(v) > 0$ indica que o vértice v não está contaminado. Também utilizaremos um conjunto H que irá conter todos os vértices contaminados.

Inicialize todos os vértices $u \in S$ com $l(u) = 0$ e todos os vértices $v \in G - S$ com $l(v) = 2$. Inclua S em H .

Iterativamente realize os passos a seguir, até que não seja possível incluir mais nenhum vértice novo em H . Ao final retorne H .

1. Para todo $u \in H$ faça;

- se $uv \in E(G)$, decemente $l(v)$ em uma unidade, se $l(v) = 0$, inclua-o em H .

Com este procedimento, podemos então calcular o $H(S)$, para qualquer subconjunto $S \in V(G)$. Utilizando o procedimento acima para todo subconjunto de vértices de $S \in V(G)$ e selecionando a cardinalidade do menor S tal que $H(S) = V(G)$, encontramos o número de envoltória h_{P_3} de G .

Observe que o número de subconjuntos de $V(G)$ é combinatório e logo o algoritmo acima é NP-completo. Tendo isto em vista, abordaremos então algoritmos de tempo polinomial de classes de grafos que possuem caracterizações definidas para a convexidade P_3 .

3.1. O número de envoltória para Grafos Cordais

De acordo com [3], podemos determinar o número de envoltória de um grafo cordal conexo em tempo quadrático. O algoritmo utiliza o conceito de bloco e articulações, utilizando-se do fato de que a união do conjunto de envoltória P_3 mínimo de cada bloco resulta no conjunto da envoltória P_3 mínimo do grafo. Mas não se pode simplesmente calcular os conjuntos da envoltória P_3 mínimo de cada bloco separadamente, pois um bloco pode ser contaminado somente com vértices de outros blocos. Por exemplo na Figura 2(a), para que todo bloco tivesse seu número de envoltória calculado separadamente, necessitaríamos de incluir os vértices a, c, d, e, g, h para que todos os blocos fossem contaminados. Porém o número de envoltória do grafo inteiro é 4, veja 2(b), com os vértices a, c, d e h formando um conjunto de envoltória mínimo.

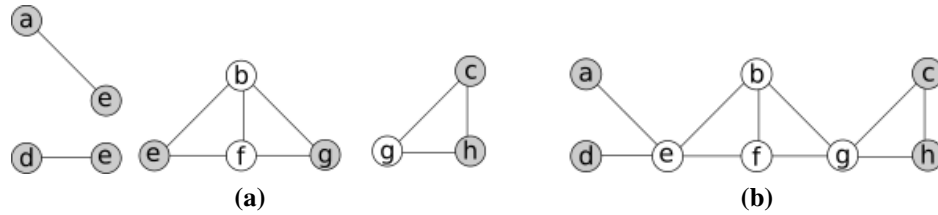


Figura 2: (a) Conjunto de envoltória mínimo de cada bloco de G . (b) Conjunto de envoltória mínimo de G .

Para a determinação do conjunto de envoltória P_3 mínimo de G , o algoritmo utiliza um rótulo $l(v)$ para cada vértice v de G . O rótulo $l(v)$ representa a quantidade de vizinhos contaminados que v precisa para ser contaminado. Por exemplo, se $l(k) = 1$, então o vértice k precisa de mais um vizinho contaminado para que ele também fique contaminado. Esse rótulo é utilizado para determinar, se necessário, quais vértices de um certo bloco B serão incluídos no conjunto de envoltória mínimo.

Porém, na descrição do Algoritmo 1, a escolha dos vértices de cada bloco, que fará parte do conjunto de envoltória mínimo, é feita de maneira implícita entre as linhas 8 e 21. Para fazer essa escolha, calcula-se um conjunto mínimo C_B em cada bloco, onde a envoltória desse conjunto resulta na contaminação de todo o bloco.

Na nossa versão não será necessário o cálculo de C_B . Levaremos em consideração que um bloco de um grafo cordal necessita de 2 vértices contaminados para que o mesmo esteja totalmente contaminado, pois cada bloco é um grafo cordal 2-conexo. Portanto, vamos verificar cada caso em que deve ser adicionado um vértice ao conjunto solução S através da rotulação l dos vértices de cada bloco.

Considere B um bloco final de G e u uma articulação de G em B . Note que u faz parte de outros blocos de G .

Se não houverem vértices com rótulo 0 ou 1, Figura 3, então seleciona-se um vértice $v \in N(u)$, e adicione-o ao conjunto solução S . Como B é bloco final, ele não voltará a ser analisado, e sua única conexão com o resto do grafo é através da articulação u . Portanto, um vizinho de

u foi adicionado a solução, e assim u necessitará apenas de um novo vértice vizinho para que todo esse bloco se contamine. Decrementa-se o rótulo de u em uma unidade.

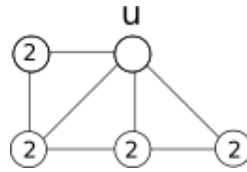


Figura 3: Todos vértices adjacentes a u possuem rótulo 2.

Se houver algum vértice com rótulo igual a 0, devem ser analisados alguns casos. No primeiro caso, devem se analisar todos os vértices $v \in V(B)$, em que $l(v) = 0$, e verificar se existe algum outro vértice $w \neq v$ em B , com $l(w) = 0$ e $d_{v,w} \leq 2$, ou $l(w) = 1$ e $d_{v,w} = 1$. Já que um vértice de rótulo 0 já está contaminado, se houver mais um vértice de rótulo 0 de distância menor ou igual a dois dele, os dois já serão suficientes para contaminar esse bloco, veja Figura 4(a) e (b). Da mesma maneira, se v for vizinho de um vértice de rótulo 1, v irá contaminar o de rótulo 1, pois ele só precisava de mais um vizinho contaminado para se contaminar, e dessa maneira os dois também serão suficientes para contaminar todo o bloco, Figura 4(c).

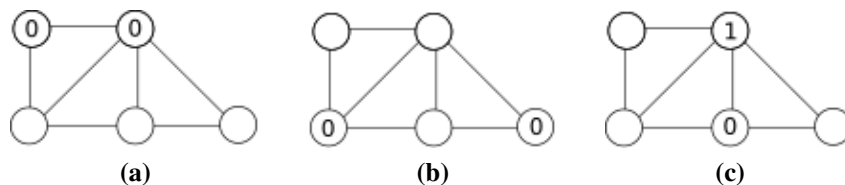


Figura 4: Situações em que o bloco já se encontra contaminado: (a) dois vértices adjacentes com rótulo 0. (b) dois vértices com 2 de distância entre eles com rótulo 0. (c) dois vértices adjacentes, um com rótulo 0, outro com rótulo 1.

Passamos então para o segundo caso. Se o primeiro caso falhar, deve-se verificar se existe $v \in B$ tal que $d_{v,u} = 1$ e $l(v) = 0$. Se isso ocorrer, o vértice u dependerá de um vizinho de outro bloco para que ele seja contaminado, logo atribuiremos $l(u) = 1$. Se existir $v \in B$ tal que $d_{v,u} = 2$ e $l(v) = 0$, o rótulo de u não irá mudar, pois se u for contaminado, todo o bloco será contaminado posteriormente. A Figura 5(a) e (b) ilustram estes casos respectivamente.

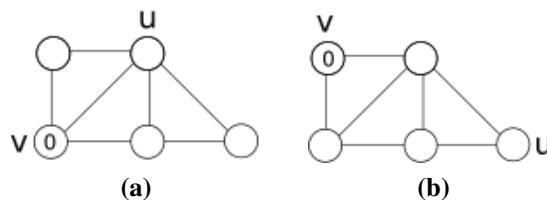


Figura 5: (a) $d_{v,u} = 1$ e $l(v) = 0$, atribua $l(u) = 1$. (b) $d_{v,u} = 2$ e $l(v) = 0$, não altere $l(u)$.

Se houver algum vértice $v \in V(B)$ com $l(v) = 1$ e $d_{v,u} = 1$, o rótulo da articulação deverá ser mantido, pois a contaminação do bloco ficará dependente do vértice u ser contaminado em algum próximo bloco. Neste caso, Figura 6, quando for contaminado, ele passará a ser o vértice que v precisava para ser contaminado e contaminará todo o bloco.

Por fim, no último caso, deveremos então selecionar um vértice w adjacente a algum $v \in V(B)$ com $l(v) = 0$ ou $l(v) = 1$ e adicioná-lo ao conjunto S . Dessa maneira, haverá dois vértices

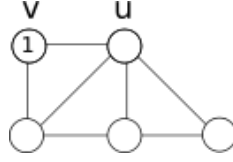


Figura 6: $d_{v,u} = 1$ e $l(v) = 1$, não altere $l(u)$.

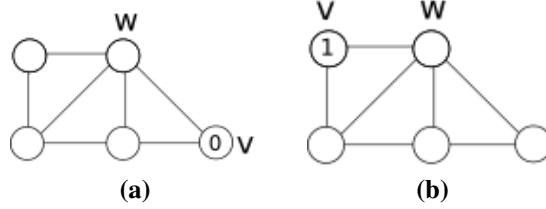


Figura 7: Adiciona-se w no conjunto S , atribua $l(u) = 0$. (a) $d_{w,v} = 1$ e $l(v) = 0$. (b) $d_{w,v} = 1$ e $l(v) = 1$.

vizinhos contaminados, e assim, todo o bloco também será contaminado e o rótulo de u passa a ser zero. Este caso é ilustrado nas Figuras 7(a) e (b).

Resta ainda a análise do último bloco final. Seja B este bloco. Note que não haverão mais vértices de corte, mas a análise é semelhante a dos casos anteriores, com pequenas modificações. Como não ocorrem mais articulações, pendências que eram deixadas para a articulação não podem mais ocorrer e todos os vértices do bloco devem terminar contaminados ao fim da execução dessa etapa.

Caso existam apenas vértices com rótulo 2, então nesse bloco, devem ser selecionados dois vértices $v, w \in V(B)$, $d_{v,w} \leq 2 \cap v \neq w$ e adicioná-los a solução S .

Caso exista algum vértice com rótulo 0, devem se analisar todos os vértices $v \in V(B)$, em que $l(v) = 0$. Deve-se verificar se existe algum outro vértice $w \in V(B)$, onde $(l(w) = 0 \cap d_{v,w} \leq 2 \cap v \neq w) \cup (l(w) = 1 \cap d_{v,w} = 1)$. Caso exista, nenhum vértice será adicionado a solução. Se o caso anterior não for satisfeito, então deve-se selecionar um vértice $w \in N(v)$, para algum $v \in V(B)/l(v) = 0$, e adicioná-lo a solução S .

Caso exista zero vértices com rótulo 0 e algum vértice com rótulo 1, então, selecione um vértice $w \in N(v)$, para algum $v \in V(B)/l(v) = 1$, e adicione w a solução S .

Desta forma concluímos o procedimento para encontrar o número de envoltória. Vale ressaltar que para a implementação do algoritmo, ao verificar os rótulos dos vértices, podemos analisar apenas aqueles vértices que são ou já foram vértices de corte, pois apenas eles mudarão seu rótulo durante a execução do algoritmo e influenciarão em cada um dos casos.

3.2. O número de envoltória para cografos

Um *cografo* é um grafo que não possui P_4 como subgrafo induzido. Para todo cografo G , tem-se que G é conexo se, e somente se, \bar{G} é desconexo. Nesse sentido, o algoritmo, disponível em [7], determina o número de envoltória na convexidade P_3 em cografos utilizando as componentes conexas de \bar{G} .

O algoritmo primeiramente testa a quantidade de vértices universais em G . Em termos de implementação, basta verificar os vértices v nos quais $d(v) = |V(G)| - 1$. A quantidade de tais vértices é denotada por u . Se $u \neq 1$, então, $h_{P_3}(G) = 2$, pois dois vértices universais em um conjunto S seriam suficientes para contaminar todo o grafo e $H_{P_3}(S) = V(G)$.

Porém, caso $u = 1$, então é necessário calcular o número t de componentes conexas de \bar{G} . Nossa implementação usou busca em largura para encontrar o valor de t . Se $t = 1$, então G é trivial e portanto $h_{P_3}(G) = 1$. Se $t \geq 3$, um conjunto S contendo vértice universal e qualquer

outro vértice satisfaz $H_{P_3}(S) = G$ e assim $h_{P_3}(G) = 2$.

Finalmente, se $t = 2$, teremos dois subgrafos em \tilde{G} . Considere \tilde{G}_1 a componente que possui apenas o vértice universal de G e \tilde{G}_2 a outra componente. O número de envoltória desse grafo será o número de componentes conexas em G_2 , que pode ser facilmente calculado utilizando uma busca em largura.

3.3. O número de envoltória para florestas

O número de envoltória pode ser determinado em tempo polinomial para florestas pelo algoritmo apresentado em [8].

Seja G uma floresta. O algoritmo inicialmente cria um conjunto S e um conjunto S' , ambos vazios, e a todos os vértices $v \in V(G)$, é atribuído um rótulo $l(v) = 0$.

A cada etapa do algoritmo, um vértice de G é excluído. As etapas a seguir são verificadas sequencialmente. Ao final, verifica-se se G não possui mais vértices. Se G possuir, as etapas abaixo são novamente executadas. Caso contrário, a execução é finalizada e o algoritmo retorna $|S|$ como o número de envoltória P_3 de G .

1. Iterativamente, enquanto houver, remove-se todas as folhas $f \in V(G)$, das quais $l(f) = 1$;
2. Iterativamente, enquanto houver, remove-se todos os vértices isolados $i \in V(G)$, dos quais $l(i) \geq 2$;
3. Seleciona-se um vértice $v \in V(G)$, tal que ou v é vértice isolado, ou v é folha e $l(v) = 0$;
4. Adiciona v em S e em S' ;
5. Iterativamente, realize os dois passos seguintes, até que $|S'| = 0$:
 - (a) selecionar qualquer $v' \in S'$
 - (b) para todo $w \in N_G(v')$, incrementa-se uma unidade em $l(w)$ e se $l(w) = 2$, inclui-se w em S' .
 - (c) remove v' de S' e de G .

3.4. O número de envoltória para grafos Split

Um grafo *split* é um grafo em os vértices podem ser particionados em uma clique e um conjunto independente. Considere que $G = ((C, I), E)$ é um grafo split onde o subgrafo induzido de G pelos vértices em C formam uma clique, e o subgrafo induzido de G pelos vértices em I formam um conjunto independente. Considere $\beta(G)$ o conjunto formado pelos vértices de I com grau 1. Ainda, considere os seguintes conjuntos: $I_1 = \{v \in I | d(v) = 1\}$ e $\mathcal{F}_i = \{v \in C | |N_{I_1}(v)| \geq i\}$. Apresentamos abaixo uma caracterização para o número de envoltória P_3 para grafos split.

Teorema 3.1. *Seja $G = (C, I, E)$ um grafo split. Então,*

$$h_{P_3}(G) = \begin{cases} 2, & \text{se } \nexists v \in V(G), \text{ com } d(v) = 1, \text{ ou se } G = P_3 \\ \beta(G), & \text{se } \exists u \in \mathcal{F}_1 \text{ e } \exists v \in \mathcal{F}_2, \\ \beta(G) + 1, & \text{caso contrário.} \end{cases}$$

Demonstração. Considere que não existam vértices em I com grau 1 e tome $S = \{u, v | u, v \in C\}$. Como todo $w \in C$ é adjacente a u e v , temos que $C \subseteq H_{P_3}(S)$. Ainda, todo $z \in I$, satisfaz $d(z) \geq 2$, logo $I \subseteq H(S)$. Portanto $H_{P_3}(S) = V(G)$.

Agora suponha que existam vértices $v \in I$ com $d(v) = 1$. Sabemos que todos tais vértices devem pertencer a qualquer conjunto envoltório de G , então $h_{P_3}(G) \geq \beta(G)$. Suponha que

existam vértices $u \in \mathcal{F}_1$ e $v \in \mathcal{F}_2$. Considere $S = I_1$. Observe que $u, v \in I[S]$. Como C é uma clique $C \subseteq I^2[S]$. Como I é um conjunto independente $\{w \in I | d(w) \geq 2\} \subseteq I^2[S]$. Como todo vértice de I com grau um está em S temos que $H_{P_3}(S) = V(G)$. Assim, $h_{P_3}(G) \leq \beta(G)$. Logo, $h_{P_3}(G) = \beta(G)$.

Por fim, suponha que exista $u \in \mathcal{F}_1$ e não exista $v \in \mathcal{F}_2$, ou que, exista $u \in \mathcal{F}_1$ e não exista $v \in \mathcal{F}_2$. Se $S = I_1$, então $H_{P_3}(S) \subset S$. Portanto, $h_{P_3}(G) \geq \beta(G) + 1$.

Considere que $u \in \mathcal{F}_1$ e não exista $v \in \mathcal{F}_2$. Seja $w \in C$ tal que $w \neq u$ e $S = I_1 \cup \{w\}$. Note que $I[S] = S \cup \{u\}$. Como C é uma clique $C \subseteq I^2[S]$. Como I é um conjunto independente $\{x \in I | d(x) \geq 2\} \subseteq I^3[S]$. Como todo vértice de I com grau um está em S temos que $H_{P_3}(S) = V(G)$. Logo $h_{P_3}(G) \leq \beta(G) + 1$.

Agora, considere que exista $u \in \mathcal{F}_1$ e não exista $v \in \mathcal{F}_2$. Seja $w \in C$ tal que $w \neq v$ e $S = I_1 \cup \{w\}$. Note que $I[S] = S \cup \{v\}$. Como C é uma clique $C \subseteq I^2[S]$. Como I é um conjunto independente $\{x \in I | d(x) \geq 2\} \subseteq I^3[S]$. Como todo vértice de I com grau um está em S temos que $H_{P_3}(S) = V(G)$. Logo $h_{P_3}(G) \leq \beta(G) + 1$. E concluímos a demonstração. \square

Portanto para construir o algoritmo, primeiramente, verificaremos se o grafo G é um grafo P_3 , se sim, retorne 2. Também identificaremos todos os vértices de grau 1. Se não existir nenhum vértice nessa condição, retorne 2.

Caso exista algum vértice de grau 1, ou seja $\beta(G) \geq 1$, então visitaremos iterativamente todos os vértices vizinhos a estes, rotulando-os como \mathcal{F}_1 ou \mathcal{F}_2 da seguinte forma: Se o vértice não pertence nem a \mathcal{F}_1 , nem a \mathcal{F}_2 , então inclua-o em \mathcal{F}_1 . Se o vértice já pertence a \mathcal{F}_1 , remova-o de \mathcal{F}_1 e inclua-o em \mathcal{F}_2 . Se o vértice já pertence a \mathcal{F}_2 , não altere nada.

Durante esta rotulação, verificaremos se em algum momento, $|\mathcal{F}_1| > 0$ e $|\mathcal{F}_2| > 0$. Se sim, retorne $\beta(G)$.

Caso chegue ao fim do procedimento, e a condição acima não seja satisfeita, simplesmente retorne $\beta(G) + 1$

4. CONVEXIDADE GEODÉSICA

Apresentaremos agora, nossas implementações de algoritmos que envolvem o cálculo do número de envoltória convexa na convexidade geodésica.

Como na convexidade P_3 , a envoltória convexa, pode ser calculada por etapas de iteração do intervalo fechado I . Faremos n iterações de I sobre S , até que $I^n[S] = I^{n-1}[S]$. Então, teremos que $I^{n-1}[S]$ é a envoltória convexa de S , ou $H(S) = I^{n-1}[S]$.

Como ilustração, seja o grafo G , representado na Figura 8, e tome $S = \{a, c, f\}$. Sabemos que $S \in H(S)$, como ilustra a Figura 8(a). Na primeira iteração, como g pertence ao menor caminho entre c, f e b pertence ao menor caminho entre a, c , temos que $b, g \in I[S]$, Figura 8(b). Na próxima iteração, $e \in I^2[S]$, pois e pertence a um dos menores caminhos entre b e f , Figura 8(c). Como nenhum vértice será adicionado em $I^3[S]$, ou seja, $I^3[S] = I^2[S]$, então $H(S) = I^2[S]$.

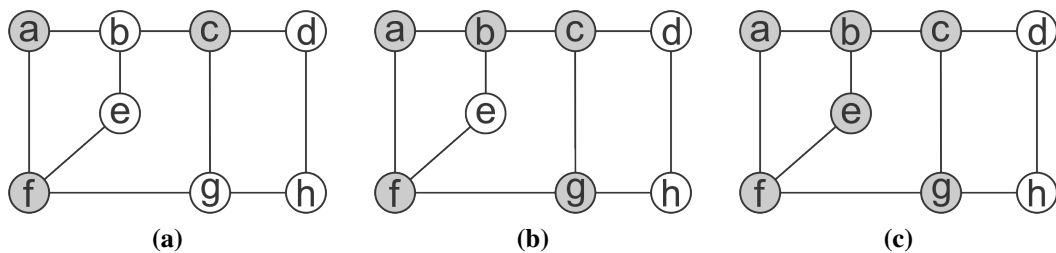


Figura 8: Cálculo da envoltória convexa geodésica de S no grafo G : (a) Estágio inicial, apenas S marcado. (b) $I[S]$. (c) $I^2[S] = H(S)$.

Segue o algoritmo para o calculo da envoltória convexa de um conjunto S . Dado um grafo G e um subconjunto S , utilizaremos um conjunto H para armazenar a envoltória convexa e uma fila Q para auxiliar no controle de análise:

1. Inclua todos vértices de S em H e inclua-os na fila Q .
2. Desenfileire u de Q .
3. Faça uma busca em largura a partir de u , obtendo o vetor L . Este vetor armazenará a distância dos vértices com relação a u .
4. Para cada $v \in H$, faça:
 - (a) Se $l(v) = 0$ ou $l(v) = 1$, avance para o próximo $v \in H$ e repita este passo. Se não houver próximo, volte para o passo 2.
 - (b) Para cada $w \in N(v)$, se $l(w) = l(v) + 1$, então insira w em H e enfileire-o em Q .
5. Se $Q \neq \emptyset$ volte para o passo 2. Senão, retorne o conjunto H .

Ao termino do procedimento, $H(S) = H$. Para efetuar o cálculo do número de envoltória, basta executar o algoritmo acima para todas as combinações de vértices. Inicia-se dos conjuntos com menor quantidade, até os de maior quantidade. Se em algum momento $H = V(G)$, retorne $|S|$ como resposta. Entretanto, esse algoritmo é NP-completo e do ponto de vista de tempo computacional, inviável na prática.

Nas próximas subseções, apresentaremos algoritmos de tempo polinomial para o cálculo de $h_g(G)$ para cografos e grafos split.

4.1. O número de envoltória para cografos

Dado um cografo G , é possível calcular seu número de envoltória em tempo polinomial, conforme [9]:

Teorema 4.1. *Seja G um cografo conexo e k o número de componentes não triviais de \bar{G} . Então:*

- a) *se $k = 0$, então $h_g(G) = n$;*
- b) *se $k = 1$, então $h_g(G) = h_g(G_1)$, onde G_1 é um subgrafo de G induzido pelos vértices da única componente não trivial em \bar{G} ;*
- c) *se $k \geq 2$, então $h_g(G) = 2$;*

Proposição 4.1. *Se G é um cografo desconexo, então $h_g(G)$ é igual a soma de todos os números de envoltória de seus componenetes.*

A partir dessa informação, implementamos o algoritmo que calcula $h_g(G)$, da seguinte maneira:

1. Seleciona-se primeiramente algum vértice arbitrário v e aplica-se busca em largura a partir de v . Se a busca visitar todos os vértices, então o cografo é conexo. Nesse caso, será chamada a função para calcular o número de envoltória de todo o grafo. Se a busca não visitar todos os vértices, então o grafo é desconexo. Nesse caso, deve-se calcular o número de envoltória da componente dos vértices visitados e aplicar uma nova busca em largura a partir de vértices remanescentes para calcular os números de envoltória de cada componente. Ao fim do processamento, soma-se todos os números de envoltória calculados nessa etapa recursivamente.

2. Cálculo do número de envoltória para cada componente conexa: Primeiramente gera-se o grafo complementar \bar{G} e adota-se inicialmente que todos os vértices de \bar{G} já foram visitados. Supondo que no melhor caso, G seja um grafo completo. Após isso, verifica se existem vértices que possuem arestas em \bar{G} . Se não houver nenhum, então $k = 0$, e retorne que $h_g(G) = |V(G)|$. Caso contrário, marca-se todos os vértices com arestas em \bar{G} como não visitados, incluindo-os em G_1 . Com isso, sabemos que $k \geq 1$.
3. Se $k \geq 1$, executaremos uma busca em largura em \bar{G} a partir de vértices ainda não visitados em G_1 . Se na busca ainda sobraem vértices a serem visitados, então a componente dos vértices de G_1 em \bar{G} forma uma componente desconexa, o que significa que $k \geq 2$. Dessa maneira, retorna-se 2 como resposta para o cálculo do número de envoltória dessa componente como resposta da etapa 2. Caso contrário, se todos os vértices forem visitados, então retornaremos à etapa 1 recursivamente, porém usando como entrada o subgrafo induzido G_1 .

4.2. O número de envoltória para grafos split

Seja $G = ((C, I), E)$ um grafo split onde o subgrafo induzido de G pelos vértices em C formam uma clique, e o subgrafo induzido de G pelos vértices em I formam um conjunto independente. Por [9], podemos determinar o número de envoltória geodésica de grafos split em tempo polinomial. Assim, dado um grafo split G , executaremos o algoritmo abaixo.

1. Escolha o vértice v de menor grau, inclua-o no conjunto I , e inclua todos os vértices de sua vizinhança em C .
2. Se $d(v) = |V(G)| - |I|$, significa que $v \in N(w)$ para todo $w \in C$. Neste caso, retorne $|I|$ como resposta.
3. Senão, execute o passo 1 até que cada vértice $u \in V(G)$, ou pertença a I ou a C .
4. Se $H(I) = |V(G)|$, então retorne $|I|$ como resposta. Senão, retorne $|I| + 1$.

5. IMPLEMENTAÇÕES

Durante o processo de pesquisa, todos os algoritmos descritos neste trabalho foram implementados. Apresentaremos nesta seção, estruturas e alguns conceitos de programação que foram utilizados e uma comparação entre h_{P_3} e h_g para grafos split.

Para os algoritmos de cálculo de h_{P_3} e h_g para cografos, e de cálculo de h_{P_3} para florestas, foi utilizada a matriz de adjacência, como forma de representação. Uma *matriz de adjacência* consiste em uma matriz A , $N \times N$, sendo N o número de vértices do grafo, tal que o valor de A_{ij} é 1 se $(i, j) \in E(G)$ e 0 caso contrário.

Para os demais algoritmos, foi utilizada a lista de adjacência para a representação dos grafos. Uma *lista de adjacência* consiste em um arranjo Adj de $|V|$ listas, uma para cada vértice em V . Para cada $u \in V(G)$, a lista de adjacências $Adj[u]$ contém todos os vértices v tais que existe uma aresta $(u, v) \in E(G)$.

Também foi utilizado nas implementações, o algoritmo clássico de *Busca em Largura* [2]. Em resumo, este algoritmo calcula a distância (menor número de arestas) desde um vértice s até todos os vértices acessíveis em G , descobrindo todos os vértices à distância k a partir de s , antes de descobrir quaisquer vértices à distância $k + 1$.

A fim de comparação, geramos grafos splits não livres de isomorfismo, com 2 até 11 vértices e executamos os algoritmos de determinação de número de envoltória P_3 e geodésica em todos.

Tabela 1: N: Número de vértices de G, Q: Quantidade de grafos split gerados, MP: Média do número de envoltória P_3 , MG: Média do número de envoltória geodésico.

N	Q	MP	MG
3	4	2	2.25
4	17	2.294118	2.705882
5	92	2.358696	3.141304
6	681	2.383260	3.519824
7	7044	2.377910	3.880040
8	102049	2.348391	4.264010
9	2069164	2.305921	4.681883
10	58716761	2.256771	5.126866
11	2333572244	2.206924	5.589888

Calculamos a média do número de envoltória para cada número de vértices e disponibilizamos na Tabela 1.

O tempo de processamento para gerar e calcular o valor de todos os números de envoltória entre 3 à 7 vértices foi inferior a 1 segundo somando todos os tempos de execução dos mesmos. O processamento para 8 vértices demorou 4 segundos e para 9 vértices, demoraram-se 29 segundos. Para 10 vértices, já foram consumidos 30 minutos de tempo de execução e para 11 vértices, o processamento terminou após 15 horas. Em uma tentativa de aumentar a amostra para 12 vértices, deixamos uma máquina rodando por 4 dias seguidos e não obtivemos uma resposta, e o processamento teve de ser interrompido pela falta de tempo para realização de testes finais.

Enfatizamos que a geração dos grafos, não livre de isomorfismo, foi o responsável pelo alto tempo gasto em processamento descrito acima. Pois, os algoritmos para cálculo de h_{P_3} e h_h são de tempo polinomial.

Observando a Tabela 1, podemos notar que a média do número de envoltória P_3 foi menor em todos os casos. Além disso, ela sempre se manteve próxima de 2. Enquanto que a média para h_g se manteve em crescimento com o aumento do número de vértices. Isso se deve ao fato de que a convexidade P_3 terá valor 2 para todos os grafos que não possuírem vértices de grau 1. E mesmo se possuírem, no máximo o número de envoltória será a quantidade de vértices de grau 1, mais 1. Já o número de envoltória para convexidade geodésica, sempre vai ser no mínimo o número de vértices do conjunto independente, e ainda em outros casos, inclui a quantidade de vértices simpliciais presentes no grafo.

6. CONCLUSÃO

Neste trabalho, estudamos as definições básicas de grafos e de convexidade em grafos. Estudamos e implementamos algoritmos de cálculo da envoltória P_3 e geodésica para grafos gerais. Ainda, estudamos e implementamos algoritmos de tempo polinomial, já descritos na literatura, para cálculo de h_{P_3} e h_g . Porém, como os artigos onde se encontram os algoritmos os apresentam em linguagem com maior nível de formalidade, abordamos os algoritmos de maneira mais descritivas. Com isso disponibilizamos um conteúdo mais didático ao leitor que busca à implementação destes algoritmos em futuros estudos.

A fim de traçar uma comparação, entre os valores do número de envoltória em diferentes convexidades, encontramos e demonstramos uma caracterização para grafos split na convexidade P_3 .

Na ausência de uma base de dados para grafos split, fizemos uma geração destes grafos

com até 11 vértices. Realizamos testes a fim de comparar o número de envoltória em ambas as convexidades e observou-se que os valores do número de envoltória na convexidade geodésica tende a ser maior que o número de envoltória na convexidade P_3 para grafos split.

Em relação a subseção 3.1, detalhamos o algoritmo descrito em [3] para grafos cordais. Dividimos o algoritmo em vários casos distintos de comportamento iterativo para cada etapa de execução. Com isso, facilitamos o entendimento para o leitor interessado em implementar o algoritmo.

Como trabalhos futuros sugerimos a comparação entre h_{P_3} e h_g para cografos e um estudo do número de envoltória em outras convexidades.

7. AGRADECIMENTOS

Agradecemos a Universidade Federal de Goiás (UFG) pelo auxílio financeiro que possibilitou a realização deste trabalho.

REFERÊNCIAS

- [1] J. A. Bondy, U. S. R Murty, Graph Theory, *Graduate Texts in Mathematics* (2008).
- [2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. Algoritmos: Teoria e Prática. Editora Campus, 2002.
- [3] C. C. Centeno, M. C. Dourado, L.D. Penso, D. Rautenbach, J. L. Szwarcfiter, Irreversible conversion of graphs, *Theoretical Computer Science* 412 (2011) 3693 - 3700.
- [4] C. C. Centeno, S. Dantas, M. C. Dourado, D. Rautenbach, J. L. Szwarcfiter, Convex Partitions of Graphs induced by Path of Order Three, *Discrete Mathematics and Theoretical Computer Science*. 12 (2010) 175 - 184.
- [5] C. C. Centeno, M. C. Dourado, L. D. Penso, V. G. Pereira de Sá, D. Rautenbach, Immediate versus eventual conversion: comparing the geodetic number and the hull number in P_3 -convexity, *Proceedings of WG 2012. Lecture Notes in Computer Science. Heidelberg: Springer-Verlag*. 7551 (2012) 262 - 273.
- [6] D. B. Parker, R. F. Westhoff, M. J. Wolf, On two-path convexity in multipartite tournaments. *European J. Combin* 29 (2008) 641 - 651.
- [7] C. C. Centeno, M. C. Dourado, J. L. Szwarcfiter, A CONVEXIDADE P_3 PARA GRAFOS NÃO DIRECIONADOS, *COPPE UFRJ*
- [8] J. P. A. Dreyer, F. S. Roberts, Irreversible k-Threshold processes: Graph-Theoretical Threshold Models of the Spread of Disease and of Opinion, *Discrete Appl. Math* 157 (1998) 307-316
- [9] M. C. Dourado, J. G. Gimbel, J. Kratochvíl, F. Protti, J. L. Szwarcfiter, On the computation of the hull number of a graph, *Theoretical Computer Science* 309 (2009) 5668 - 567444
- [10] J. Cáceres, C. Hernando, M. Mora, I. M. Pelayo, M. L. Puertas, C. Seara, On geodetic sets formed by boundary vertices. *Discrete Math*. 306 (2011) 193 - 217.
- [11] M. G. Everett, S. B. Seidman, The hull number of a graph. *Discrete Math*. 57 (1985) 217 - 223.
- [12] M. C. Dourado, F. Protti, D. Rautenbach, J. L. Szwarcfiter, On the hull number of triangle-free graphs, *SIAM J. Discrete Math*. 23, (2010) 2163 - 2172.