UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
CAMPUS DE CHAPECÓ
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ÉVERTON DE ASSIS VIEIRA

GRAFOS d-SNARKS

CHAPECÓ
2019

ÉVERTON DE ASSIS VIEIRA

GRAFOS *d*-SNARKS

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.
Orientador: Emílio Wuerges
Coorientador: Leandro Miranda Zatesko

CHAPECÓ
2019

Vieira, Éverton de Assis

Grafos *d*-Snarks / Éverton de Assis Vieira. – 2019.
30 f.: il.

Orientador: Emílio Wuerges.
Coorientador: Leandro Miranda Zatesko.
Trabalho de conclusão de curso (graduação) – Universidade Federal da Fronteira Sul, curso de Ciência da Computação, Chapecó, SC, 2019.

1. Coloração de grafos e hipergrafos (MSC 05C15). 2. Algoritmos de grafos (MSC 05C85). 3. Teoria dos grafos em relação à Ciência da Computação (MSC 68R10). I. Wuerges, Emílio, orientador. II. Zatesko, Leandro Miranda, coorientador. III. Universidade Federal da Fronteira Sul. IV. Título.

E-mail: eassis.vieira@gmail.com

ÉVERTON DE ASSIS VIEIRA

GRAFOS *d*-SNARKS

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Emílio Wuerges
Coorientador: Leandro Miranda Zatesko

Este trabalho de conclusão de curso foi defendido e aprovado pela banca avaliadora em: 3/12/2019.

BANCA AVALIADORA

Emílio Wuerges – UFFS

Leandro Miranda Zatesko – UTFPR

Marina Groshaus – UTFPR

Alane Marie de Lima – PPGInf/UFPR

Aos meus pais, José e Darci, e a minha esposa, Aira,
por todo amor, carinho e apoio,
dedico este trabalho

AGRADECIMENTOS

Ao professor Leandro Miranda Zatesko, por todos os ensinamentos e conselhos, e pela enorme dedicação empregada no desenvolvimento deste trabalho.

À minha esposa, Aira Aguiar, por estar sempre ao meu lado me apoiando, motivando e dando suporte para alcançar meus objetivos.

À minha família, em especial meu pai e minha mãe, que sempre me apoiaram e incentivaram durante toda minha trajetória na universidade.

Ao professor Emílio Wuerges, à professora Marina Groshaus e a Alane Marie de Lima, que compuseram a banca e contribuíram com diversas sugestões para o trabalho.

RESUMO

Os snarks são grafos com propriedades peculiares. Eles se relacionam com importantes conjecturas como a Conjectura dos Grafos Sobrecarregados. De acordo com a forma como os snarks são definidos, propomos neste trabalho os d-snarks, os quais são uma generalização dos snarks. Neste trabalho, além de definirmos os d-snarks, apresentamos o resultado de um experimento no qual grafos 5-regulares foram testados a fim de encontrar algum 5-snark. Apesar de não termos encontrado 5-snark algum, demonstramos que se os 5-snarks não existem, então $\mathcal{P} = \mathcal{NP}$. Ainda, demonstramos que, a menos que $\mathcal{P} = \mathcal{NP}$, existem muitos 5-snarks, de forma que o número de 5-snarks não pode ser limitado superiormente por um função polinomial.

Palavras-chave: Coloração de grafos e hipergrafos (MSC 05C15). Algoritmos de grafos (MSC 05C85). Teoria dos grafos em relação à Ciência da Computação (MSC 68R10).

ABSTRACT

Snarks are graphs with peculiar properties. They are related to important conjectures such as the Overfull Conjecture. According to how snarks are defined, we propose in this work the d-snarks, which are a snark generalization. In this work, besides defining d-snarks, we present the result of an experiment wherein 5-regular graphs were tested in order to find some 5-snark. Although we have not found any 5-snark, we prove that if 5-snarks do not exist, then $\mathcal{P} = \mathcal{NP}$. Also, we prove that, unless $\mathcal{P} = \mathcal{NP}$, there are many 5-snarks, so that the number of 5-snarks cannot be bounded above by a polynomial function.

Keywords: Coloring of graphs and hypergraphs (MSC 05C15). Graph algorithms (MSC 05C85). Graph theory in relation to Computer Science (MSC 68R10)

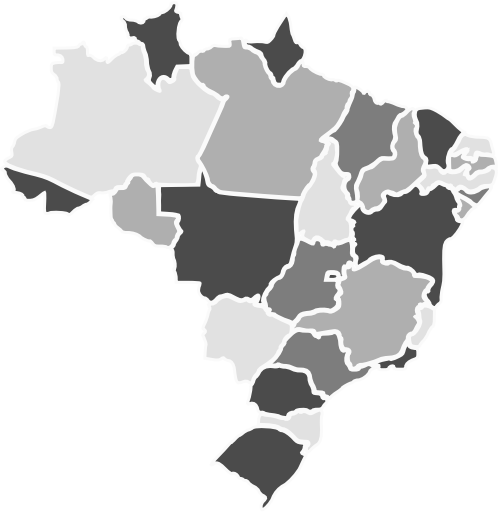
LISTA DE ILUSTRAÇÕES

SUMÁRIO

1 INTRODUÇÃO

Em 1852, Francis Guthrie formulou a *Conjectura das Quatro Cores*. A conjectura (hoje *Teorema das Quatro Cores*) afirma serem suficientes quatro cores para colorir qualquer mapa de forma que regiões adjacentes recebam cores diferentes (conforme Figura 1). A conjectura permaneceu sem prova por mais de um século, sendo demonstrada pela primeira vez em 1977 (APPEL; HAKEN, 1977; APPEL et al., 1977). A prova do Teorema das Quatro Cores foi a primeira grande demonstração matemática a ser realizada com auxílio de um computador, de forma que seria impraticável para um humano verificá-la manualmente (SWART, 1980).

Figura 1 – Mapa do Brasil colorido com 4 cores

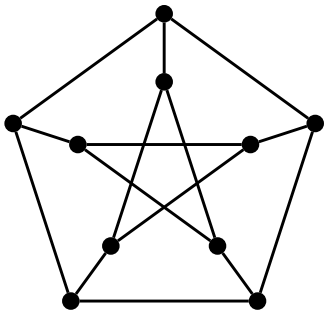


Em estudos sobre o Teorema das Quatro Cores foi encontrada uma classe de grafos peculiares (CHLADNÝ; ŠKOVIERA, 2010 apud ZATESKO, 2018), atualmente conhecidos como *snarks*. A importância dos snarks nesses estudos é dada pelo teorema provado por Tait (1878–1880) que mostra equivalência entre o Teorema das Quatro Cores e a afirmação de que *nenhum snark é planar*[1]. Com a descoberta desta classe de grafos possivelmente origina-se a área de estudos sobre coloração de arestas (ZATESKO, 2018), uma vez que um snark é definido como um grafo cúbico, 2-aresta-conexo e cujas arestas não podem ser coloridas usando somente 3 cores (STIEBITZ et al., 2012).

O primeiro snark descoberto foi o *grafo de Petersen* (PETERSEN, 1898 apud STIEBITZ et al., 2012), em 1898, com 10 vértices (Figura 2). Os próximos snarks encontrados foram os de Blanuša (1946), com 18 vértices cada (Figura 4), e o snark de Descartes (1948), com 210 vértices. Observe que o grafo de Petersen foi encontrado 18 anos após a demonstração do teorema de Tait, e os snarks seguintes foram encontrados aproximadamente 50 anos após a descoberta do primeiro snark.

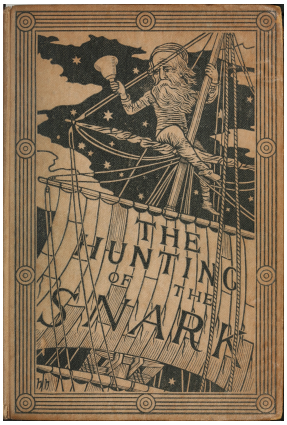
[1] Conceitos teóricos de grafos serão discutidos mais detalhadamente no Capítulo 2.

Figura 2 – Grafo de Petersen



O nome snark foi dado por Gardner (1976) em torno de um século após o Teorema de Tait em razão da estranha criatura ou coisa que é "caçada" no poema *The Hunting of the Snark*, escrito por Lewis Carroll e publicado em 1876 (ZATESKO, 2018).

Figura 3 – Capa da primeira edição de *The Hunting of the Snark*



1.1 COLORAÇÃO DE ARESTAS

Seja G um grafo qualquer e $\mathscr{C}$ um conjunto com k elementos, chamados "cores", uma função que atribui às arestas de G cores de $\mathscr{C}$ de forma que arestas adjacentes recebam cores diferentes é dita uma coloração de arestas de G com k cores. O *índice cromático* de um grafo G, denotado por $\chi'(G)$, é o menor inteiro k para o qual G admite uma coloração de arestas com k cores.

Seja G um grafo e u um vértice de G. Podemos perceber facilmente que o número mínimo de cores necessário para colorir as arestas incidentes a u é igual à quantidade de arestas que incidem a u. Desta forma vemos que o índice cromático de G é limitado inferiormente pelo grau máximo de G, denotado por $\Delta(G)$ (ou simplesmente Δ quando livre de ambiguidade). Um limite superior para o índice cromático de qualquer grafo simples foi estabelecido por Vizing (1964 apud STIEBITZ et al., 2012), que mostrou que qualquer grafo simples pode ser colorido com $\Delta + 1$ cores.

Os grafos podem ser classificados quanto ao seu índice cromático em duas classes. Grafos que possuem índice cromático igual a Δ são chamados de *Classe 1* e grafos que não podem ser coloridos com Δ cores são chamados de *Classe 2*. Um exemplo de grafo *Classe 2* é o grafo de Petersen (Figura 2). Apesar de haver apenas duas possíveis classes, decidir se um grafo é *Classe 1* ou *Classe 2* é um problema $\mathcal{NP}$-difícil (HOLYER, 1981).

A coloração de arestas aplica-se a importante cenários do mundo real; em particular, a aplicações de escalonamento tais como enlaces de redes de sensores (GANDHAM et al., 2005), processos na indústria (WILLIAMSON et al., 1997) e partidas de jogos (BURKE; WERRA et al., 2013).

1.2 ORGANIZAÇÃO DO DOCUMENTO

O restante deste documento está dividido da seguinte maneira: no Capítulo 2 são apresentados conceitos básicos de grafos e resultados anteriores relacionados com os snarks. No Capítulo 3 são apresentados os resultados obtidos neste trabalho. Por fim no Capítulo 4 é feita a conclusão do trabalho e apresentadas sugestões para trabalhos futuros.

2 RESULTADOS ANTERIORES E DEFINIÇÕES PRELIMINARES

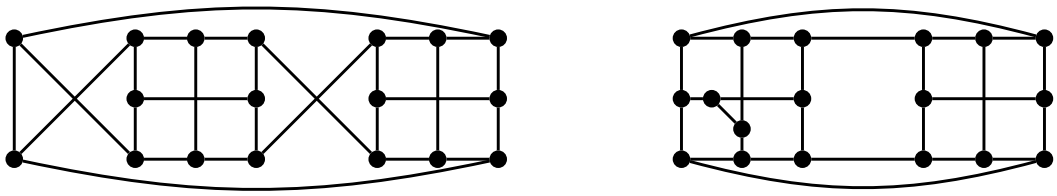
Seja G um grafo simples, conexo e sem laços[1]. O conjunto de *vértices* e *arestas* de G são denotados por $V(G)$ e $E(G)$, respectivamente. Se existe uma aresta $uv \in E(G)$ então os vértices u e v são ditos *vizinhos*. Um grafo H é subgrafo de G se $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$. Se H é um subgrafo de G e possui todas as arestas $uv \in E(G)$ em que $u, v \in V(H)$ então H é dito subgrafo induzido de G. O *grau* de um vértice u é denotado por $d_G(u)$ e é igual ao número de vizinhos de u. Se todos os vértices de G têm grau igual a d então G é dito *d-regular*. Um grafo 3-regular é chamado de *cúbico*. Se G é um grafo conexo com pelo menos 2 vértices e se a remoção de qualquer conjunto $F \subseteq E(G)$ com $|F| < k$ não desconecta G, então G é dito *k-aresta-conexo*. Se podemos desenhar G no plano de forma que não haja interseção entre interiores de arestas, então G é dito *planar*. Se $\{V', V''\}$ é uma partição de $V(G)$, um corte em G é o conjunto $F \subseteq E(G)$ tal que, para toda aresta $uv \in E(G)$ em que $u \in V'$ e $v \in V''$, temos que $uv \in F$.

2.1 SNARKS

Os snarks, como mencionado no Capítulo 1, são grafos cúbicos, 2-aresta-conexos e *Classe 2*, sendo que o primeiro (e por muito tempo único) snark encontrado foi o grafo de Petersen (Figura 2). Este também é o menor snark existente (ISAACS, 1975) e, apesar de ser creditado a Petersen, já havia aparecido em trabalhos anteriores (KEMPE, 1886).

Os snarks descobertos por Blanuša (1946) (Figura 4) são dois diferentes resultados obtidos através da aplicação da operação chamada de *dot product* sobre duas cópias do grafo de Petersen (SKUPIEŃ, 2007).

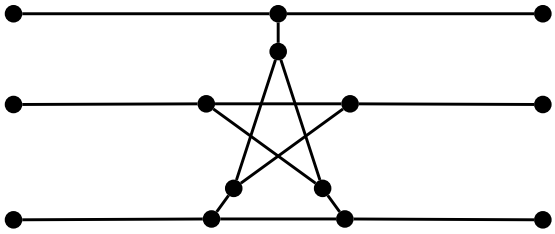
Figura 4 – Snarks de Blanuša



O snark descoberto por Descartes (1948) também é construído sobre o grafo de Petersen. Cada vértice do grafo é substituído por um *eneágono* e estes são unidos por *links* que são criados também sobre o grafo de Petersen. A Figura 5 mostra como são os links utilizados na construção do grafo. Um eneágono é conectado a outro eneágono se os vértices que foram substituídos pelos eneágonos são vizinhos no grafo original.

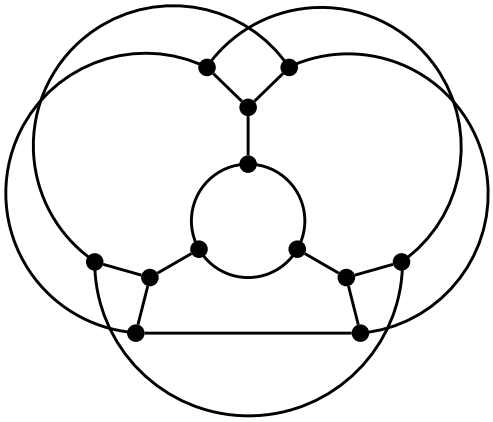
[1] Um laço é uma aresta que conecta um vértice a ele próprio.

Figura 5 – *Link* utilizado por Descartes (1948) na construção de um snark



A primeira família infinita de snarks foi mostrada por Isaacs (1975). Para gerar estes snarks também é utilizada a operação *dot product* que realiza a união de dois snarks para gerar um outro snark. Utilizando essa técnica, o snark gerado pode ser utilizado juntamente com outro snark ou com uma cópia dele próprio para gerar outro snark. Essa técnica foi utilizada anteriormente para gerar os snarks de Blanuša (ISAACS, 1975), conforme mencionado anteriormente. Com esse procedimento provou-se a existência de infinitos snarks. Os snarks de Isaacs são conhecidos como *snarks flores* (SKUPIEŃ, 2007). A Figura 6 mostra o snark flor conhecido como J_3.

Figura 6 – Snark flor J_3



Em um trabalho mais recente Skupień (2007) mostrou que o número de snarks de ordem n para $n > 0$ par é limitado inferiormente pela função exponencial $2^{(n-84)/18}$.

2.2 A CONJECTURA DOS GRAFOS SOBRECARREGADOS

Um grafo de ordem n é dito sobrecarregado se ele possui quantidade de arestas estritamente maior do que $\Delta\lfloor n/2 \rfloor$. Um Δ-*subgrafo* de um grafo G é um subgrafo H de G tal que $\Delta(H) = \Delta(G)$. Se um grafo G possui um Δ-subgrafo H sobrecarregado então G é dito *subgrafo-sobrecarregado*, ou simplesmente *SO* (do inglês, *subgraph-overfull*). Ser subgrafo-sobrecarregado é condição suficiente para um grafo ser *Classe 2* (BEINEKE; WILSON, 1973).

Podemos notar que grafos de ordem par não podem ser sobrecarregados, uma vez que podem ter no máximo $\Delta n/2$ arestas. Mais precisamente, um grafo G é sobre-

carregado se e somente se possui ordem ímpar e $\sum_{u \in V(G)}(\Delta(G) - d_G(u)) \leqslant \Delta(G) - 2$ (NIESSEN, 1994).

Inicialmente a conjectura foi proposta para grafos com $\Delta \geqslant n/2$ (CHETWYND; HILTON, 1984). Dois anos depois a conjectura foi reformulada para grafos com $\Delta \geqslant n/3$ (CHETWYND; HILTON, 1986); um contraexemplo foi encontrado: o grafo P^* (Figura 7), que é obtido a partir do grafo de Petersen (Figura 2) com a remoção de qualquer vértice. Tanto o grafo de Petersen, que possui $\Delta = (n-1)/3$, quanto o P^* são exemplos de grafos *Classe 2* que não são *SO* (NASERASR; ŠKREKOVSKI, 2003; ZATESKO, 2018).

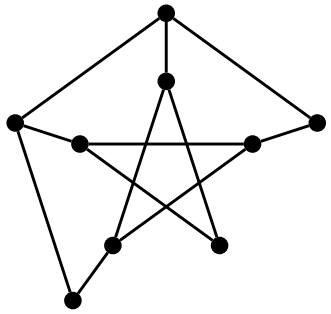


Figura 7 – Grafo P^*

A Conjectura dos Grafos Sobrecarregados é importante para coloração de arestas pelo fato de estabelecer uma equivalência entre a propriedade de um grafo ser *Classe 2* e uma propriedade que pode ser verificada em tempo polinomial, que é a propriedade de ser *SO* (PADBERG; RAO, 1982). A importância desta conjectura para os grafos d-snarks, a classe de grafos que introduzimos neste trabalho (conforme Capítulo 3), é vista uma vez que os grafos snarks limitam o escopo da conjectura no domínio dos grafos cúbicos. Assim, estudar grafos com grau arbitrário é importante para tentarmos entender a estrutura combinatória do problema de coloração de arestas.

Conjectura 2.1 (Conjectura dos Grafos Sobrecarregados (CHETWYND; HILTON, 1984, 1986; HILTON; JOHNSON, 1987)). *Um grafo G com n vértices e grau máximo $\Delta > n/3$ é Classe 2 se e somente se G é subgrafo-sobrecarregado.*

3 RESULTADOS OBTIDOS

Como resultado deste trabalho foi proposta a definição dos grafos d-snarks. Apresentamos: o Teorema 3.5 e o Corolário 3.6, nos quais mostramos que os grafos 5-snarks devem existir, e não somente existir, mas existir muitos; a Observação 3.2, na qual mostramos que os d-snarks não podem ser *SO*; por fim, apresentamos os resultados obtidos com o experimento.

Definição 3.1. *Um d-snark é um grafo d-regular $(d-1)$-aresta-conexo e Classe 2, para todo $d \geqslant 3$ ímpar.*

Observe que a definição dos grafos d-snarks para valor de $d = 3$ é igual à definição dos snarks.

Observação 3.2. *Nenhum d-snark pode ser SO.*

Demonstração. Se um d-snark G é *SO*, ele possui um subgrafo sobrecarregado H com $\Delta(H) = d$. Uma vez que os grafos d-snarks são grafos d-regulares com d ímpar, qualquer d-snark consequentemente possui ordem par, e por esse motivo $V(H) \neq V(G)$. Além disso, podemos supor sem perda de generalidade que H é induzido por $V(H)$. Logo, temos que $s := \sum_{u \in V(H)}(\Delta(H) - d_H(u))$ é o número de arestas $uv \in E(G)$ em um corte entre H e o complemento de H em G, já que G é um grafo regular. Desta forma (conforme explicado na Seção 2.2) se H é sobrecarregado então $s \leqslant (d-2)$, porém $s \geqslant (d-1)$ por definição, o que é uma contradição. □

É importante notar que sendo os d-snarks exemplos de grafos que são *Classe 2* e não são *SO*, a menos que a Conjectura dos Grafos Sobrecarregados não se sustente, um d-snark deve ter pelo menos $3d + 1$ vértices.

3.1 5-SNARKS

Ainda sem conhecer nenhum 5-snark foi possível demonstrar com o Teorema 3.5 que sua não-existência implica em $\mathcal{P} = \mathcal{NP}$. E não somente eles devem existir, mas deve existir muitos, no sentido de que a quantidade de 5-snarks de uma determinada ordem n não pode ser limitada superiormente por uma função polinomial $p(n)$, a menos que $\mathcal{P} = \mathcal{NP}$, conforme o Corolário 3.6.

No intuito de demonstrar o Teorema 3.5 será apresentada primeiramente a demonstração do Teorema 3.4, o qual mostra que o problema de coloração de arestas para grafos 5-regulares 4-aresta-conexo é $\mathcal{NP}$-completo. Para isto segue a descrição do problema COLORAÇÃO DE ARESTAS.

COLORAÇÃO DE ARESTAS:

Entrada: um grafo G;
Problema: o índice cromático de G é igual a Δ?

O problema COLORAÇÃO DE ARESTAS é um problema $\mathcal{NP}$-completo (conforme explicado na Seção 1.1) mesmo quando restrito a grafos d-regulares para qualquer $d \geqslant 3$ (LEVEN; GALIL, 1983). Para a demonstração do Teorema 3.4 será considerada a seguinte restrição do problema COLORAÇÃO DE ARESTAS.

COLORAÇÃO DE ARESTAS(d-regular $(d-1)$-aresta-conexo):

Entrada: um grafo d-regular $(d-1)$-aresta-conexo G;
Problema: o índice cromático de G é igual a d?

A demonstração do Teorema 3.4 é feita reduzindo o problema COLORAÇÃO DE ARESTAS(5-regular) para a sua restrição COLORAÇÃO DE ARESTAS(5-regular 4-aresta-conexo). Para fazer essa redução usando uma tradicional *redução Karp* é necessário apresentar um algoritmo de tempo polinomial, que recebe como entrada um grafo 5-regular G, e retorna um grafo 5-regular 4-aresta-conexo $G^{\ddagger}$, de forma que $\chi'(G^{\ddagger}) = 5$[1] se e somente se $\chi'(G) = 5$. Porém, para a redução deste problema será utilizada uma *redução Turing* por oráculo, como a redução utilizada por Cook na prova da $\mathcal{NP}$-completude do problema da *satisfazibilidade Booleana* (COOK, 1971). No contexto do nosso problema essa redução pode ser vista como uma redução de tempo polinomial que recebe como entrada um grafo 5-regular G e retorna não somente um, mas possivelmente vários grafos 5-regulares 4-aresta-conexos de forma que todos possuem índice cromático igual a 5 cores se e somente se $\chi'(G) = 5$, o que pode ser visto de forma equivalente a um grafo 5-regular $G^{\ddagger}$ possivelmente desconexo, no qual seus componentes são todos 4-aresta-conexos e $\chi'(G^{\ddagger}) = 5$ se somente se $\chi'(G) = 5$.

A seguir é apresentado o *Lema da Paridade*, o qual é importante para a redução mencionada anteriormente.

Lema 3.3 (Lema da Paridade (ISAACS, 1975)). *Se G é um grafo d-regular d-aresta-colorível e $F \subseteq E(G)$ é um corte em G, então, para qualquer d-aresta-coloração de G com o conjunto de cores $\{1,\ldots,d\}$, então*

$$f_1 \equiv f_2 \equiv \cdots \equiv f_d \pmod 2,$$

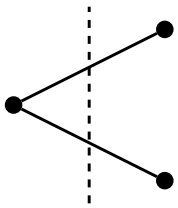
sendo que f_α é o número de arestas em F coloridas com a cor $\alpha \in \{1,\ldots,d\}$.

Em vista do Lema da Paridade (Lema 3.3) segue que se G é um grafo d-regular e possui um corte $F \subseteq E(G)$, tal que o número de arestas em F é ímpar e estritamente

[1] Na Seção 1.1 foi dito que os grafos podem ser classificados como *Classe 1* ou *Classe 2*; porém, essa classificação não será utilizada nesta demonstração pois essa redução pode gerar grafos não simples, que por consequência podem necessitar de mais que $\Delta+1$ cores (VIZING, 1964 apud STIEBITZ et al., 2012).

menor que d, então $\chi'(G) > d$. Além disso, pelo Lema da Paridade segue também que, se o valor de d é estritamente maior que 2 e G possui um corte $F \subseteq E(G)$ com 2 arestas uv e $u'v'$, estando u e u' do mesmo lado do corte e sendo $u = u'$ (conforme Figura 8), então $\chi'(G) > d$. Por esse motivo a redução utilizada na demonstração do Teorema 3.4 assume que a entrada da redução não será um grafo com corte com 1 ou 3 arestas, ou um grafo com corte com 2 arestas da forma mencionada acima, já que para essas instâncias do problema a resposta é *não*. Apesar destas restrições do problema COLORAÇÃO DE ARESTAS(5-regular) o problema continua tão difícil quanto o problema original, pois decidir se um grafo possui um corte com 1 ou 3 arestas, ou se possui um corte com duas arestas adjacentes, pode ser feito em tempo polinomial (FORD; FULKERSON, 1956; EDMONDS; KARP, 1972).

Figura 8 – Exemplo de cortes com 2 arestas desconsiderados na redução



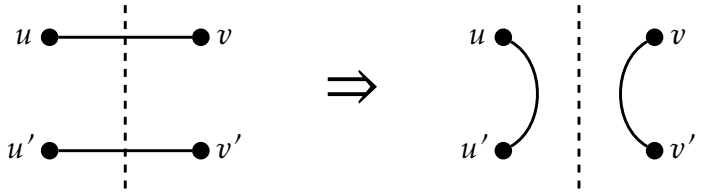
Teorema 3.4. COLORAÇÃO DE ARESTAS(5-regular, 4-aresta-conexo) *é $\mathcal{NP}$-completo.*

Demonstração. Seja G um grafo 5-regular sem cortes com 1 ou 3 arestas, e sem cortes com 2 arestas adjacentes. Devemos mostrar como construir em tempo polinomial um grafo 5-regular $G^{\ddagger}$ cujos componentes conexos são todos 4-aresta-conexos de maneira que $\chi'(G^{\ddagger}) = 5$ se e somente se $\chi'(G) = 5$. Se G é 4-aresta-conexo, então a redução apenas retorna $G^{\ddagger} \coloneqq G$. Se G não é 4-aresta-conexo, então todos os seus cortes com cardinalidade menor que 4 possuem 0 ou 2 arestas. Enquanto algum componente conexo de G possuir um corte F com 2 arestas uv e $u'v'$, estando u e u' do mesmo lado do corte e sendo $u \neq u'$, substituímos as arestas uv e $u'v'$ pelas arestas uu' e vv' (o que pode gerar arestas paralelas, porém isso não é um problema para a redução), conforme Figura 9. O resultado é um grafo 5-regular $G^{\ddagger}$ cujo seus componentes são todos 4-aresta-conexos. Podemos observar que encontrar os cortes com 2 arestas de forma iterativa pode ser feito em tempo polinomial. Além disso, são considerados apenas os cortes com 2 arestas em um mesmo componente conexo de G, e desta forma o número de cortes decrementa a cada corte considerado; sendo assim, a redução para.

Demonstramos a seguir que $\chi'(G^{\ddagger}) = 5$ se e somente se $\chi'(G) = 5$.

Se $\chi'(G) = 5$, pelo Lema 3.3 é garantido que para todo corte $F = \{uv, u'v'\}$ considerado na iteração da redução as arestas uv e $u'v'$ possuem a mesma cor. Podemos, então, transferir essa cor para as arestas uu' e vv' e consequentemente construir uma

Figura 9 – Ação da redução sobre um corte com 2 arestas



Fonte – Elaborado pelo autor

coloração de arestas com 5 cores para $G^{\ddagger}$, já que todos os cortes com 2 arestas foram considerados na redução.

Reciprocamente, se $\chi'(G^{\ddagger}) = 5$, consideremos, um de cada vez, todos os cortes $F = \{uv, u'v'\}$ tomados na iteração da redução, transferindo as cores das arestas uu' e vv', criadas pela redução, para as arestas uv e $u'v'$, e consequentemente construindo uma coloração de arestas com 5 cores para G. Claramente esse procedimento funciona se as arestas uu' e $u'v'$ possuem a mesma cor, porém, caso elas não possuam a mesma cor, podemos permutar as cores em um dos lados do corte para obtermos a mesma cor nas duas arestas, uma vez que o corte foi desconectado pela redução. □

Teorema 3.5. *Se 5-snarks não existem, então $\mathcal{P} = \mathcal{NP}$.*

Demonstração. Uma vez que os 5-snarks são por definição as instâncias negativas do problema COLORAÇÃO DE ARESTAS(5-regular 4-aresta-conexo), ou seja as instâncias do problema para as quais a resposta é *não*, se eles não existem, esse problema pode ser decidido com um algoritmo de tempo constante, o que pelo Teorema 3.4 implica que $\mathcal{P} = \mathcal{NP}$. □

O Corolário 3.6 segue do Teorema de Fortune–Mahaney sobre a complexidade computacional de linguagens esparsas (FORTUNE, 1979; MAHANEY, 1982). O Teorema de Fortune–Mahaney afirma que se qualquer linguagem esparsa é $\mathcal{NP}$-completa ou co$\mathcal{NP}$-completa, então $\mathcal{P} = \mathcal{NP}$. Uma linguagem formal L é dita *esparsa* se existe uma função polinomial $p(n)$ tal que, o número de palavras de tamanho n pertencentes à linguagem L é limitado superiormente por $p(n)$.

Corolário 3.6. *Se existe uma função polinomial $p(n)$ que limita superiormente a quantidade 5-snarks de ordem n, então $\mathcal{P} = \mathcal{NP}$.*

Demonstração. Conforme mostra o Teorema 3.4 o problema COLORAÇÃO DE ARESTAS(5-regular, 4-aresta-conexo) é um problema $\mathcal{NP}$-completo, o que implica que a linguagem dos 5-snarks é co$\mathcal{NP}$-completa, uma vez que os 5-snarks são as instâncias negativas do problema. Consequentemente, se a linguagem dos 5-snarks é esparsa então $\mathcal{P} = \mathcal{NP}$. □

É importante salientar que o Teorema 3.5 é utilizado apenas para evidenciar que os 5-snarks devem existir. A existência dos 5-snarks não implica em $\mathcal{P} \neq \mathcal{NP}$.

3.2 EXPERIMENTO

Durante a execução deste trabalho foi realizado um experimento para tentar encontrar algum 5-snark. O experimento baseia-se num teste que foi realizado para grafos 5-regulares de ordem entre 6 e 18 inclusive. Para cada grafo 5-regular G considerado, testamos se G era um 5-snark. Para os grafos de ordem entre 6 e 16 inclusive, foi possível verificar todos os grafos existentes. Já para grafos com 18 vértices, apenas uma parte foi verificada. Segue a tabela que apresenta a exata quantidade de grafos analisados para cada ordem.

Tabela 1 – Quantidade de grafos analisados por ordem

Ordem	Grafos analisados
6	1
8	3
10	60
12	7 848
14	3 459 383
16	2 585 136 675
18	90 882 962 449

Para grafos com 18 vértices não foi possível analisar todos os grafos 5-regulares devido à quantidade de grafos existentes, porém, foi possível verificar todos os grafos 5-regulares com 18 vértices e cintura (i.e tamanho do menor ciclo) no mínimo 4, os quais totalizaram 406 824 grafos dentre os mais de 90 bilhões de grafos com 18 vértices que analisamos. Dentre os grafos analisados nenhum 5-snark foi encontrado.

Para a geração dos grafos foi utilizado o programa *genreg* (MERINGER, 1999), o qual dado um valor n e um valor d gera todos os grafos d-regulares de ordem n. Também é possível fornecer um terceiro valor g para o programa, o que irá limitar a geração dos grafos para grafos com cintura no mínimo g. O programa possibilita também realizar a geração dos grafos de forma fracionada (recurso que foi utilizado para geração dos grafos 5-regulares com 18 vértices, que foram analisados no experimento), fornecendo dois valores i e j, sendo $i \leqslant j$, é possível fracionar a geração dos grafos em j partes, gerando apenas a i-ésima parte dos grafos.

Na análise dos grafos foram utilizados outros dois programas (os códigos-fonte dos programas estão listados no Apêndice A). O programa *parser* foi utilizado para tratar a saída gerada pelo programa genreg, pois nesta saída algumas informações não são úteis para a análise do grafo (por exemplo, uma lista de automorfismos para o

grafo). Além da lista de automorfismos, o programa parser remove informações que não são necessárias para montar o grafo e analisá-lo. A Figura 10 mostra a saída do programa genreg antes e depois de ser tratada. A frente de "Graph" é dado o índice do grafo, a frente do termo "Taillenweite" é dado a cintura do grafo gerado, e a frente do termo "Ordnung" é o tamanho do grupo de automorfismos do grafo gerado.

Figura 10 – Ação do programa parser sobre a saída do programa genreg

```
Graph 1:

1 : 2 3 4 5 6
2 : 1 3 4 5 6
3 : 1 2 4 5 6
4 : 1 2 3 5 6
5 : 1 2 3 4 6
6 : 1 2 3 4 5
Taillenweite: 3

5 : 1 2 3 4 6 5
4 : 1 2 3 5 4 6
4 : 1 2 3 6 5 4
3 : 1 2 4 3 5 6
3 : 1 2 5 4 3 6
3 : 1 2 6 4 5 3
2 : 1 3 2 4 5 6
2 : 1 4 3 2 5 6
2 : 1 5 3 4 2 6
2 : 1 6 3 4 5 2
1 : 2 1 3 4 5 6
1 : 3 2 1 4 5 6
1 : 4 2 3 1 5 6
1 : 5 2 3 4 1 6
1 : 6 2 3 4 5 1
Ordnung: 720
```

```
1
2 3 4 5 6
1 3 4 5 6
1 2 4 5 6
1 2 3 5 6
1 2 3 4 6
1 2 3 4 5
3
```

O programa *snarks* foi utilizado para verificar se os grafos gerados são $(d-1)$-aresta-conexos e *Classe 2*, ou seja, se são d-snarks. O programa faz a leitura do grafo e verifica primeiramente se o grafo é $(d-1)$-aresta-conexo, uma vez que a verificação dessa propriedade pode ser feita em tempo polinomial. Caso seja verificado que o grafo é $(d-1)$-aresta-conexo, o programa irá tentar encontrar uma d-aresta-coloração para o grafo através de um algoritmo de *backtracking*, de tempo exponencial. Caso não encontre, isto significa que o grafo é *Classe 2* e consequentemente um d-snark. A saída do programa são os d-snarks encontrados. Além disso o programa escreve alguns arquivos de registro. Nos arquivos de registro, a cada 1 milhão de grafos analisados, é escrito o total de grafos que foram analisados até o momento, e há quanto tempo

o programa está executando. Quando o programa encerra é escrito no arquivo de registro que o programa encerrou.

O experimento foi realizado utilizando 30 computadores. Os computadores utilizavam sistema operacional baseado no *Linux* e como configuração dispunham de um processador *Core I7* e capacidade de memória de 8GB.

4 CONCLUSÃO E TRABALHOS FUTUROS

Apesar de nenhum 5-snark ter sido encontrado com o experimento, ao demonstrarmos que sua não-existência implica em $\mathcal{P} = \mathcal{NP}$ mostramos uma forte evidência de sua existência. Além disso foi possível demonstrar que os d-snarks, assim como os já bem conhecidos snarks, são exemplos de grafos *Classe 2* que não são *SO*, o que mostra a importância de se estudar esta classe de grafos, uma vez que a Conjectura dos Grafos Sobrecarregados se mantém sem demonstração por mais de 30 anos. Com o experimento foram verificados que são *Classe 1* todos os grafos 5-regulares com até 16 vértices e mais de 90 bilhões com 18 vértices. Este resultado é importante, ainda que nenhum 5-snark tenha sido encontrado, pois a existência de 5-snarks de ordem menor que 16 implicaria que a Conjectura dos Grafos Sobrecarregados não se mantém para grafos 5-regulares.

Para trabalhos futuros sugere-se estudar a existência de relações entre os snarks e os 5-snarks, ou de forma mais geral a existência de relações entre os d-snarks e os $(d + 2)$-snarks (por exemplo, se é possível construir um $(d + 2)$-snark a partir de um d-snark).

REFERÊNCIAS

APPEL, K.; HAKEN, W. Every planar map is four colorable. Part I: Discharging. *Illinois J. Math.*, v. 21, p. 429–490, 3 1977.

APPEL, K. et al. Every planar map is four colorable. Part II: Reducibility. *Illinois J. Math.*, v. 21, p. 491–567, 3 1977.

BEINEKE, L. W.; WILSON, R. J. On the edge-chromatic number of a graph. *Discrete Math.*, v. 5, p. 15–20, 1973.

BLANUŠA, D. Problem četiriju boja. *Glasnik. Mat. Fiz. Astr. Ser. II*, v. 1, p. 31–42, 1946.

BURKE, Edmund; WERRA, Dominique de et al. Applications to timetabling. In: HANDBOOK of graph theory. [S.l.]: Chapman e Hall/CRC, 2013. p. 530–562.

CHETWYND, A. G.; HILTON, A. J. W. Star multigraphs with three vertices of maximum degree. *Math. Proc. Cambridge Philos. Soc.*, v. 100, p. 303–317, 1986.

_____. The chromatic index of graphs of even order with many edges. *J. Graph Theory*, v. 8, p. 463–470, 1984.

CHLADNÝ, M.; ŠKOVIERA, M. Factorisation of Snarks. *Electron. J. Combin.*, v. 17, #R32, 2010.

COOK, S. A. The complexity of theorem-proving procedures. In: PROC. 3rd Annual ACM Symposium on Theory of Computing (STOC '71). New York: ACM, 1971. p. 151–158.

DESCARTES, B. Network-colourings. *The Mathematical Gazette*, London, v. 32, p. 67–69, 1948.

EDMONDS, J.; KARP, R. M. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, v. 19, p. 248–264, 2 1972.

FORD, L. R.; FULKERSON, D. R. Maximal flow through a network. *Can. J. Math.*, v. 8, p. 399–404, 1956.

FORTUNE, S. A note on sparse complete sets. *SIAM J. Comput.*, v. 5, p. 431–433, 3 1979.

GANDHAM, S. et al. Link scheduling in sensor networks: distributed edge coloring revisited. In: PROC. 24th INFOCOM. [S.l.: s.n.], 2005. p. 2492–2501.

GARDNER, M. Mathematical Games. *Scientific American*, v. 4, n. 234, p. 126–130, 1976.

HILTON, A. J. W.; JOHNSON, P. D. Graphs which are vertex-critical with respect to the edge-chromatic number. *Math. Proc. Cambridge Philos. Soc.*, v. 102, p. 103–112, 1987.

HOLYER, I. The $\mathcal{NP}$-completeness of edge-colouring. *SIAM J. Comput.*, v. 10, n. 4, p. 718–720, 1981.

ISAACS, R. Infinite families of non-trivial trivalent graphs which are not Tait-colorable. *Amer. Math. Monthly*, v. 82, p. 221–239, 3 1975.

KEMPE, A. B. A memoir on the theory of mathematical form. *Philos. Trans. Royal Soc.*, v. 177, p. 1–70, 1886.

LEVEN, D.; GALIL, Z. $\mathcal{NP}$-completeness of finding the chromatic index of regular graphs. *J. Algorithms*, v. 4, p. 35–44, 1983.

MAHANEY, S. R. Sparse complete sets of $\mathcal{NP}$: Solution of a conjecture by Berman and Hartmanis. *J. Comput. Syst. Sci.*, v. 25, p. 130–143, 1982.

MERINGER, Markus. Fast generation of regular graphs and construction of cages. *Journal of Graph Theory*, Wiley Online Library, v. 30, n. 2, p. 137–146, 1999.

NASERASR, R.; ŠKREKOVSKI, R. The Petersen graph is not 3-edge-colourable — a new proof. *Discrete Math.*, v. 268, p. 325–326, 2003.

NIESSEN, T. How to find overfull subgraphs in graphs with large maximum degree. *Discrete Appl. Math.*, v. 51, p. 117–125, 1994.

PADBERG, M. W.; RAO, M. R. Odd minimum cut-sets and b-matching. *Math. Oper. Res.*, v. 7, p. 67–80, 1982.

PETERSEN, J. Sur le théorème de Tait. *L'Intermédiaire des Mathématiciens*, v. 5, p. 225–227, 1898.

SKUPIEŃ, Zdzisław. Exponentially many hypohamiltonian snarks. *Electronic Notes in Discrete Mathematics*, Elsevier, v. 28, p. 417–424, 2007.

STIEBITZ, M. et al. *Graph Edge Coloring: Vizing's Theorem and Goldberg's Conjecture*. [S.l.]: Wiley, 2012.

SWART, E. R. The philosophical implications of the four-color problem. *Amer. Math. Monthly*, Mathematical Association of America, v. 87, n. 9, p. 697–702, 1980.

TAIT, P. G. On the colouring of maps. *Proc. Roy. Soc. Edinburgh Sect. A*, v. 10, p. 501–503, 729, 1878–1880.

VIZING, V. G. On an estimate of the chromatic class of a p-graph (em Russo). *Diskret. Analiz.*, v. 3, p. 25–30, 1964.

WILLIAMSON, D. P. et al. Short shop schedules. *Oper. Res.*, v. 45, n. 2, p. 288–294, 1997.

ZATESKO, Leandro Miranda. *Novel procedures for graph edge-colouring*. 2018. Tese (Doutorado).

APÊNDICE A – PROGRAMAS UTILIZADOS NO EXPERIMENTO

A.1 CÓDIGOS-FONTE

A.1.1 Arquivo parser.cc

```
#include <cstdio>
#include <cstdlib>
#include <cinttypes>
#include <cstdint>

#define MAX 1123

int main(int argc, char **argv) {
  int girth, u, n = atoi(argv[1]), d = atoi(argv[2]);
  uint64_t id;
  char s[MAX];
  while (scanf("\nGraph %" SCNu64 ":", &id) != EOF) {
    printf("%" PRIu64 "\n", id);
    for (int i = 0; i < n; i++) {
      scanf("%d : ", &u);
      for (int j = 0; j < d; j++) {
        scanf("%d", &u); printf("%d ", u);
      }
      printf("\n");
    }
    scanf("\nTaillenweite: %d\n", &girth);
    printf("%d\n", girth);
    fflush(stdout);
    while (fgets(s, MAX, stdin) && s[0] != '\n');
  }
  return 0;
}
```

A.1.2 Arquivo snarks.cc

```
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <cstdint>
```

```
#include <cinttypes>
#include <cerrno>
#include <ctime>
#include <queue>
using namespace std;

#define MAX 1123
#define INF 112345678

FILE *outfile, *logfile;
char outfile_name[MAX], logfile_name[MAX];
int lg[MAX][MAX], res[MAX][MAX], phi[MAX][MAX];
int dist[MAX], p[MAX], n, d;
time_t init_time;
uint64_t count;

int bfs(int t) {
  int u, v, i;
  queue<int> q;
  for (u = 0; u < n; u++) dist[u] = INF;
  q.push(0); dist[0] = 0;
  while (!q.empty()) {
    u = q.front(); q.pop();
    if (u == t) return 1;
    for (i = 0; i < d; i++) {
      v = lg[u][i];
      if (dist[v] == INF && res[u][v] > 0) {
        dist[v] = dist[u] + 1;
        p[v] = u; q.push(v);
      }
    }
  }
  return 0;
}

int ffek(int t) {
  int f = 0, u;
  while (bfs(t)) {
    for (u = t; u; u = p[u]) {
```

```
      res[p[u]][u] -= 1; res[u][p[u]] += 1;
    }
    f++;
  }
  return f;
}

int lambda(void) {
  int l = INF, u, i, j;
  for (u = 1; u < n; u++) {
    for (i = 0; i < n; i++)
      for (j = 0; j < d; j++) {
        res[i][lg[i][j]] = 1;
      }
    l = min(l, ffek(u));
  }
  return l - 1;
}

int missing(int alpha, int u) {
  for (int i = 0; i < d; i++)
    if (phi[u][i] == alpha) return 0;
  return 1;
}

int backtrack(int u, int i) {
  int alpha, v = lg[u][i], j;
  if (u == n) return 1;
  if (i == d) return backtrack(u + 1, 0);
  if (phi[u][i]) return backtrack(u, i + 1);
  for (j = 0; lg[v][j] != u && j < d; j++);
  for (alpha = 1; alpha <= d; alpha++)
    if (missing(alpha, u) && missing(alpha, v)) {
      phi[u][i] = phi[v][j] = alpha;
      if (backtrack(u, i + 1)) return 1;
      phi[u][i] = phi[v][j] = 0;
    }
  return 0;
}
```

```
void print_graph(int girth) {
  outfile = fopen(outfile_name, "a");
  if (!outfile) {
    fprintf(stderr, "Failed to open out file");
    return;
  }
  fprintf(outfile, "Graph: %" PRIu64 "\n", count);
  for (int i = 0; i < n; i++) {
    for (int j = 0; j < d; j++) {
      fprintf(outfile, "%d ", lg[i][j]);
    }
    fprintf(outfile, "\n");
  }
  fprintf(outfile, "Girth: %d\n\n", girth);
  fclose(outfile);
}


void write_log() {
  logfile = fopen(logfile_name, "w");
  if (!logfile) {
    fprintf(stderr, "Failed to open log file");
    return;
  }
  int diff = difftime(time(NULL), init_time);
  fprintf(logfile,
      "timestamp: %dhrs%dmins%dsecs\n%" PRIu64
      " graphs were checked\n",
      (diff / 3600), (diff / 60) % 60, diff % 60, count);
  fclose(logfile);
}

void completed() {
  logfile = fopen(logfile_name, "a");
  if (!logfile) {
    fprintf(stderr, "Failed to open log file");
    return;
  }
```

```
  fprintf(logfile, "Program execution completed\n");
  fclose(logfile);
}

int main(int argc, char **argv) {
  int u, i, j, girth;
  n = atoi(argv[1]); d = atoi(argv[2]);
  if (argc >= 4) {
    sprintf(outfile_name,
      "%d_%d-snarks_%s#%s.out", n, d, argv[3], argv[4]);
    sprintf(logfile_name,
      "%d_%d-snarks_%s#%s.log", n, d, argv[3], argv[4]);
  } else {
    sprintf(outfile_name, "%d_%d-snarks.out", n, d);
    sprintf(logfile_name, "%d_%d-snarks.log", n, d);
  }
  time(&init_time);
  while (scanf("%" SCNu64, &count) != EOF) {
    if (!(count % 1000000L)) write_log();
    for (i = 0; i < n; i++)
      for (j = 0; j < d; j++) {
        scanf("%d", &u); lg[i][j] = --u;
      }
    scanf("%d", &girth);
    if (lambda() >= (d - 1)) {
      memset(phi, 0, sizeof(phi));
      if (!backtrack(0, 0)) {
        print_graph(girth);
      }
    }
  }
  write_log();
  completed();
  return 0;
}
```