

André Breda Carneiro

Identificação dos Snarks Fluxo-Críticos de Ordem Pequena

Sorocaba, SP

29 de Abril de 2016

André Breda Carneiro

Identificação dos Snarks Fluxo-Críticos de Ordem Pequena

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação (PPGCCS) da Universidade Federal de São Carlos como parte dos requisitos exigidos para a obtenção do título de Mestre em Ciência da Computação. Área de concentração: Sistemas de Informação.

Universidade Federal de São Carlos – UFSCar

Centro de Ciências em Gestão e Tecnologia – CCGT

Programa de Pós-Graduação em Ciência da Computação – PPGCCS

Orientadora: Prof^ª. Dr^ª. Cândida Nunes da Silva

Sorocaba, SP

29 de Abril de 2016

Carneiro, André Breda

Identificação dos Snarks Fluxo-Críticos de Ordem Pequena / André Breda
Carneiro. -- 2016.
89 f. : 30 cm.

Dissertação (mestrado)-Universidade Federal de São Carlos, campus
Sorocaba, Sorocaba

Orientador: Cândida Nunes da Silva

Banca examinadora: Alberto Alexandre Assis, José de Oliveira Guimarães
Bibliografia

1. Snark. 2. Conjectura dos 5-Fluxos. 3. k-fluxo-crítico. I. Orientador. II.
Universidade Federal de São Carlos. III. Título.



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências em Gestão e Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato André Breda Carneiro, realizada em 29/04/2016:

Cândida Nunes da Silva

Profa. Dra. Cândida Nunes da Silva
UFSCar

AA

Prof. Dr. Alberto Alexandre Assis Miranda
IFNMG

José de Oliveira Guimarães

Prof. Dr. José de Oliveira Guimarães
UFSCar

À minha família e à minha namorada

Agradecimentos

Agradeço,

aos meus pais Sérgio e Lincra pelo suporte e motivação nessa jornada de estudos;

ao meu irmão Daniel pelos momentos de distração que tanto precisei;

à minha namorada Maria Rafaela pela paciência, compreensão e ajuda nos momentos difíceis;

à minha professora Cândida pela orientação, pelos puxões de orelha quando precisei e pelos elogios para me motivar durante todo o desenvolvimento deste trabalho.

*“Just the place for a Snark!” the Bellman cried,
As he landed his crew with care;
Supporting each man on the top of the tide
By a finger entwined in his hair.*

*“Just the place for a Snark! I have said it twice:
That alone should encourage the crew.
Just the place for a Snark! I have said it thrice:
What I tell you three times is true.”
(The Hunting of the Snark, Lewis Carroll)*

Resumo

O tema de pesquisa deste projeto são os grafos k -fluxo-críticos, grafos que não admitem k -fluxo, mas que após a contração de um par de vértices, adjacentes ou não, passam a admitir um k -fluxo. Dentre estes, nos concentraremos no estudo de *snarks*, que são grafos cúbicos que não admitem 3-coloração de arestas, e tampouco 4-fluxo, dado que Tutte demonstrou que um grafo cúbico admite 3-coloração de arestas se e somente se admite 4-fluxo. Diversas conjecturas famosas podem ser reduzidas a *snarks*, fato que motiva muito estudo da estrutura de tais grafos. A Conjectura dos 5-Fluxos de Tutte, a qual afirma que todo grafo 2-aresta-conexo admite um 5-fluxo é uma destas. Em 2013, Brinkmann, Goedgebeur, Hägglund e Markström conseguiram gerar computacionalmente todos os *snarks* com até 36 vértices. Silva, Pesci e Lucchesi observaram que todo *snark* 4-fluxo-crítico admite 5-fluxo, e que os *snarks* não 4-fluxo-críticos têm um *snark* 4-fluxo-crítico como menor. Essa observação abre uma nova abordagem na tentativa de resolução da Conjectura dos 5-fluxos de Tutte. Este trabalho é um início de pesquisa segundo essa nova abordagem buscando identificar entre os *snarks* de até 36 vértices quais são os *snarks* 4-fluxo-críticos.

Palavras-chaves: *Snark*. Conjectura dos 5-Fluxos. k -fluxo-crítico.

Abstract

The main theme of this dissertation are the k -flow-critical graphs, which are graphs that do not have a k -flow but once any two vertices (either adjacent or not) are identified the smaller graph thus obtained has a k -flow. Amongst those, we focused our study on *snarks*, which are cubic graphs that do not have a 3-edge-coloring, nor a 4-flow, as Tutte showed that a cubic graph has a 3-edge-coloring if and only if it has a 4-flow. Several famous conjectures can be reduced to *snarks*, and such fact motivates the study of the structure of such graphs. The 5-Flow Conjecture of Tutte, which states that every 2-edge-connected graph has a 5-flow is one of them. In 2013, Brinkmann, Goedgebeur, Hägglund and Markström generated all *snarks* of order at most 36. Silva, Pesci and Lucchesi observed that every 4-flow-critical *snark* has a 5-flow and that every non-4-flow-critical *snark* has a 4-flow-critical *snark* as a minor. This observation allows a new approach to try to resolve Tutte's 5-Flow Conjecture. This work is an attempt to start following this new approach by identifying which *snarks* of order at most 36 are 4-flow-critical.

Key-words: *Snark*. 5-Flow Conjecture. k -flow-critical.

Lista de ilustrações

Figura 1 – Grafo 3-aresta-colorido	22
Figura 2 – Grafo de Petersen	23
Figura 3 – Uma 4-coloração de faces do K_4	27
Figura 4 – Uma 3-coloração de arestas do K_4	28
Figura 5 – Uma 4-coloração de faces do K_4 com pares ordenados binários	29
Figura 6 – Uma 3-coloração de arestas do K_4 com pares ordenados binários	29
Figura 7 – Subgrafos H_1 (a) e H_2 (b) obtidos a partir do K_4	30
Figura 8 – A 4-coloração de faces obtida a partir da 3-coloração de arestas do K_4	31
Figura 9 – Ciclo hamiltoniano	31
Figura 10 – Coloração do ciclo hamiltoniano	32
Figura 11 – Grafo cúbico e planar não hamiltoniano	32
Figura 12 – Grafo de Petersen e a transformação de um vértice em um triângulo	34
Figura 13 – Produto interno entre dois grafos de Petersen	35
Figura 14 – Grafo estrela S_3	36
Figura 15 – <i>Flower-snark</i> J_5	37
Figura 16 – <i>Flower-snark</i> J_3	37
Figura 17 – Um 4-fluxo do K_4	38
Figura 18 – k -fluxo modular	40
Figura 19 – Vértices equilibrado (a), positivo (b) e negativo (c)	41
Figura 20 – Corte de T_x	42
Figura 21 – Inversão e alteração dos pesos das arestas de π_{xy}	43
Figura 22 – Multiplicação de um 4-fluxo do K_4 por 2	44
Figura 23 – Soma de dois 4-fluxos do K_4	45
Figura 24 – Possibilidades de equilíbrio de um vértice de grau 3	46
Figura 25 – O ciclo C e o corte $\partial(V(C))$ de um grafo com 4-fluxo	47
Figura 26 – Operações de identificação de vértices e contração de arestas	48
Figura 27 – Grafos de Petersen, quase-cúbico e cúbico subjacente	53
Figura 28 – Ordem e fixação dos pesos das arestas incidentes em um vértice w	57
Figura 29 – Combinações de pesos que equilibram módulo 4 um vértice de grau 3	60
Figura 30 – <i>Snark</i> de Blanche Descartes	69
Figura 31 – Grafo 4-aresta-crítico porém não 4-vértice-crítico	71
Figura 32 – Exemplo de produto interno que gera um grafo não 4-fluxo-crítico	73
Figura 33 – Grafo de Petersen - 1898	81
Figura 34 – Blanuša <i>snark</i> 1 - 1946	82
Figura 35 – Blanuša <i>snark</i> 2 - 1946	82
Figura 36 – Blanche Descartes <i>snark</i> - 1948	83

Figura 37 – Szekeres <i>snark</i> - 1973	84
Figura 38 – <i>Flower-snark</i> J_5 - 1975	85
Figura 39 – <i>Double-star snark</i> - 1975	86
Figura 40 – Celmins-Swart <i>snark</i> 1 - 1979	87
Figura 41 – Celmins-Swart <i>snark</i> 2 - 1979	87
Figura 42 – Goldberg <i>snark</i> 5 - 1981	88
Figura 43 – Watkins <i>snark</i> - 1989	89

Lista de tabelas

Tabela 1	–	Tempos da <i>Versão Força Bruta</i>	59
Tabela 2	–	Tempos da <i>Versão Força Bruta</i> e da <i>Versão de Fluxo - Variante 1</i>	61
Tabela 3	–	Tempos da <i>Versão de Coloração</i>	63
Tabela 4	–	Tempos da <i>Versão de Fluxo - Variante 2</i>	65
Tabela 5	–	Tempos das versões <i>VC</i> e <i>VF2</i> com todos <i>snarks</i> de ordem até 36	66
Tabela 6	–	Tempos das versões <i>VF2</i> e <i>VF3</i>	67
Tabela 7	–	Tempos das versões <i>VC</i> , <i>VF2</i> e <i>VF3</i> com todos <i>snarks</i> de ordem até 36	68
Tabela 8	–	Tempos das versões <i>VF3</i> original e <i>VF3</i> alterada	70
Tabela 9	–	Quantidades de <i>snarks</i> 4-fluxo-críticos para cada ordem	72

Conteúdo

	Introdução	21
1	CONCEITOS BÁSICOS	27
1.1	Coloração de Faces e Arestas	27
1.2	A Caça aos <i>Snarks</i>	32
1.2.1	Produto Interno	34
1.2.2	<i>Flower-snarks</i> - construção	36
1.3	k-Fluxos e suas propriedades	38
1.3.1	k -Fluxos modulares	40
1.3.2	Operações em k -Fluxos	43
1.4	k-Fluxos e colorações	45
1.5	Grafos Fluxo-Críticos	48
1.5.1	<i>Snarks</i> 4-Fluxo-Críticos	51
2	ALGORITMOS E RESULTADOS	55
2.1	<i>Versão Força Bruta</i> (ou <i>VFB</i>)	56
2.2	<i>Versão de Fluxo - Variante 1</i> (ou <i>VF1</i>)	59
2.3	<i>Versão de Coloração</i> (ou <i>VC</i>)	61
2.4	<i>Versão de Fluxo - Variante 2</i> (ou <i>VF2</i>)	63
2.5	<i>Versão de Fluxo - Variante 3</i> (ou <i>VF3</i>)	65
2.6	<i>Snarks</i> Grandes	67
2.7	Resultados	70
	Conclusão	75
	Referências	77
	APÊNDICE A – <i>SNARKS</i>	81

Introdução

No livro (BONDY; MURTY, 2008) encontramos um trecho de uma carta que Augustus de Morgan escreveu a William Rowan Hamilton em 1852 propondo o seguinte problema que, posteriormente, ganhou grande fama e ficou conhecido como o Problema das Quatro Cores (tradução livre do autor):

“Um dos meus estudantes [Frederick Guthrie, irmão de Francis] me pediu para explicar um fato que eu não sabia que era um fato — e ainda não sei. Ele diz que se uma figura for dividida de qualquer maneira e os compartimentos forem coloridos de tal forma que figuras com qualquer porção de fronteira comum sejam coloridas de maneiras diferentes — quatro cores podem ser necessárias, mas não mais...”

Segundo (MCKAY, 2013), a primeira publicação de tal problema não é devida a De Morgan em 1860 mas sim a um autor somente identificado como “F.G.” em 1854. Acredita-se que esse autor seja Francis ou Frederick Guthrie. Neste artigo McKay favorece Francis pois seu irmão, Frederick, teve o cuidado de atribuir a descoberta deste problema a ele em uma outra publicação.

Quais seriam essas figuras a que Augustus de Morgan se refere na carta? Podemos interpretar que seriam grafos desenhados no plano, onde os pontos de cruzamentos de linhas seriam os vértices, e os trechos das linhas entre dois pontos de cruzamento as arestas. As porções da figura a serem coloridas seriam as regiões do plano delimitadas por uma sequência de linhas, ou seja, as faces. A interpretação do termo fronteira pode ser ambígua. Para que a coloração seja possível, uma fronteira, ou aresta no caso de grafos, não pode ter dos seus dois lados uma mesma região, ou face, caso contrário uma aresta teria uma mesma cor dos seus dois lados. Tais arestas são chamadas de arestas de corte e, quando um grafo não apresenta arestas desse tipo, ele é dito ser 2-aresta-conexo. Formalmente, o Problema das Quatro Cores, também conhecido como Teorema das Quatro Cores, enuncia:

Teorema 1 (Teorema das Quatro Cores) *Todo grafo planar 2-aresta-conexo possui uma 4-coloração de faces.*

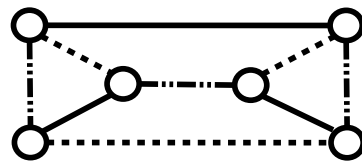
Uma k -coloração de faces é uma atribuição de no máximo k cores para as faces de um grafo planar G de forma que duas faces que compartilhem uma mesma aresta não tenham a mesma cor. Logo, o Teorema das Quatro Cores estabelece que $k = 4$ é o número máximo de cores necessárias para k -colorir um grafo planar.

Esse problema desafiou a comunidade científica por mais de um século, tempo durante o qual várias tentativas de provas foram apresentadas. Entre as inúmeras tentativas

de provas apresentadas está a de Tait ([TAIT, 1880](#)) que, em 1880, mostrou que encontrar uma 4-coloração de faces de um grafo planar cúbico é equivalente a encontrar uma 3-coloração de suas arestas.

Uma *3-coloração de arestas* de um grafo cúbico G é uma atribuição de no máximo três cores para as arestas de G de forma que em cada vértice incida exatamente uma aresta de cada cor, conforme ilustrado na Figura 1; cada tipo de tracejado representa uma cor.

Figura 1 – Grafo 3-aresta-colorido



Fonte: Produzido pelos autores

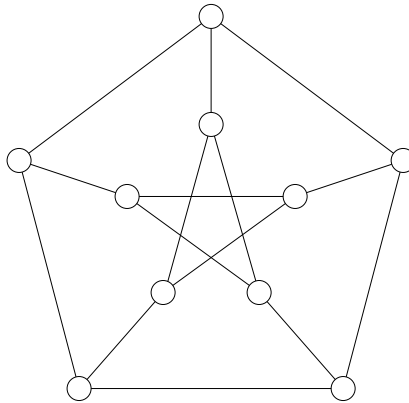
O Teorema das Quatro Cores é redutível a grafos cúbicos (veja ([SAATY, 1972](#), Conjecture C_2)), isto é, demonstrá-lo para grafos cúbicos implica em demonstrar para qualquer grafo planar. Todo grafo cúbico e hamiltoniano admite 3-coloração de arestas e, como Tait acreditava que todo grafo cúbico planar era hamiltoniano, ele pensou ter, com a demonstração dessa equivalência, demonstrado o Teorema das Quatro Cores. Apenas muitos anos depois, em 1946, Tutte ([TUTTE, 1946](#)) encontrou um grafo cúbico planar e não hamiltoniano, mostrando assim que a equivalência entre os dois problemas, apesar de muito interessante, não era suficiente para demonstrar o Teorema das Quatro Cores.

A equivalência de Tait suscitou um interesse em encontrar grafos cúbicos, mesmo que não planares, que não admitissem 3-coloração de arestas. Tais grafos são conhecidos como *snarks*¹. O primeiro *snark* foi descoberto por Petersen ([PETERSEN, 1898](#)) e acabou ficando conhecido como grafo de Petersen, ilustrado na Figura 2. Quase 50 anos depois, Blanuša ([BLANUŠA, 1946](#)) apresentou mais dois *snarks*, conhecidos como Blanuša *snarks*.

Em 1948, Blanche Descartes ([DESCARTES, 1948](#)) apresentou um outro *snark* bem maior, com 210 vértices. Nas décadas seguintes, poucos outros *snarks* foram descobertos. A aparente raridade de tais grafos motivou Martin Gardner ([GARDNER, 1976](#)) a chamar tais grafos de *snarks* se referindo ao poema “*The Hunting of the Snark*” de Lewis Carroll, o qual narra a tensa busca de uma criatura mística, temível e muito difícil de encontrar chamada *snark*. No entanto, na mesma época Isaacs ([ISAACS, 1975](#)) apresentou uma operação, conhecida como produto interno, que permite construir um novo *snark* a partir

¹ Formalmente, a definição de *snark* inclui mais restrições além de não admitir 3-coloração de arestas, as quais foram omitidas neste ponto para manter a leveza do texto. A definição precisa, mais técnica, é apresentada posteriormente na Seção 1.2.

Figura 2 – Grafo de Petersen



Fonte: Produzido pelos autores

de dois *snarks* conhecidos. No mesmo artigo (ISAACS, 1975), Isaacs mostrou também que existe uma família infinita de *snarks* conhecida como *flower-snarks* em que nenhum dos grafos da família pode ser construído a partir de outros *snarks* menores através da operação de produto interno. A contribuição de Isaacs deixou claro que existem infinitos *snarks* e ainda que há mais de uma forma de obter famílias infinitas de tais grafos. No entanto, até hoje, ainda não há um entendimento pleno de qual é a estrutura subjacente a tais grafos que inviabiliza uma 3-coloração de suas arestas.

Diversas conjecturas famosas sabidamente reduzem-se a *snarks*, e por esse motivo um bom entendimento da estrutura dessa classe de grafos é de interesse da comunidade científica. Como exemplo de tais conjecturas temos a Conjectura dos 5-Fluxos de Tutte. Para enunciar tal conjectura é necessário apresentar algumas definições. Dados um grafo não direcionado G , uma função D que define orientações para as arestas de G e uma função f que define pesos inteiros para as arestas de G , dizemos que o fluxo líquido $f(v)$ de um vértice v é a soma dos pesos das arestas que saem de v menos a soma dos pesos das arestas que entram em v . Um k -fluxo é um par de funções (D, f) tal que o peso $f(e)$ atribuído a cada aresta é não nulo e menor que k (isto é, pertence ao conjunto $\{1, 2, \dots, k-1\}$), com fluxo líquido nulo em todos os vértices. Com base nessas definições enunciamos a seguir esta conjectura:

Conjectura 1 (Conjectura dos 5-Fluxos de Tutte (TUTTE, 1954)) *Todo grafo 2-aresta-conexo admite um 5-fluxo.*

Essa conjectura é de grande interesse no contexto deste projeto. Tutte demonstrou que, para grafos planares, é equivalente admitir uma k -coloração de faces e um k -fluxo (veja (BONDY; MURTY, 2008, Theorem 21.2) para uma demonstração). Essa equivalência é enunciada a seguir.

Teorema 2 (Teorema de Tutte) *Um grafo planar admite k -coloração de faces se, e somente se, possui k -fluxo.*

Na verdade, Tutte ([TUTTE, 1954](#)) definiu o conceito de k -fluxo como forma de generalizar o conceito de coloração de faces sem depender de planaridade. Se nos lembrarmos do famoso Teorema das Cinco Cores de Heawood ([HEAWOOD, 1890](#)), o qual afirma que todo grafo planar sem arestas de corte possui uma 5-coloração de faces, percebemos que a Conjectura dos 5-fluxos de Tutte é uma generalização deste Teorema para grafos não necessariamente planares. Na mesma linha de raciocínio, Tutte propôs mais duas famosas Conjecturas enunciadas a seguir. Para enunciá-las é necessário primeiro definir o que é um *minor*. Um *minor* H de um grafo G é um grafo que pode ser obtido a partir de G através das operações de remoção de vértices e arestas ou contração de arestas.

Conjectura 2 (Conjectura dos 4-Fluxos ([TUTTE, 1966](#))) *Todo grafo 2-aresta-conexo e sem um minor do grafo de Petersen tem um 4-fluxo.*

Conjectura 3 (Conjectura dos 3-Fluxos) *Todo grafo 2-aresta-conexo e sem 3-cortes tem um 3-fluxo.*

A Conjectura dos 4-Fluxos é uma generalização do Teorema das Quatro Cores, e a Conjectura dos 3-Fluxos uma generalização do Teorema de Grötzsch ([GRÖTZSCH, 1958](#)). Diversas propriedades interessantes de k -fluxos e resultados parciais sobre as Conjecturas de Tutte podem ser encontrados nos livros de Zhang ([ZHANG, 1997](#)), Diestel ([DIESTEL, 1996](#)) e Bondy e Murty ([BONDY; MURTY, 2008](#)), bem como no artigo de Seymour ([SEYMOUR, 1995](#)).

Esta dissertação

Com foco nas três Conjecturas de Tutte, Silva e Lucchesi ([SILVA; LUCCHESI, 2008](#)) iniciaram o estudo de grafos *k -fluxo-críticos*; grafos que não admitem k -fluxo para um inteiro k especificado, mas passam a admitir quando uma operação de redução é aplicada ao grafo. Silva e Lucchesi consideram em seu estudo dois tipos de grafos fluxo-críticos: os *k -aresta-críticos* e os *k -vértice-críticos*. Grafos *k -aresta-críticos* são aqueles que não admitem k -fluxo, mas passam a admitir após a contração de uma aresta qualquer. Grafos *k -vértice-críticos* são aqueles que não admitem k -fluxo, mas passam a admitir após a identificação de dois vértices quaisquer.

É sabido que todo grafo k -aresta-crítico possui um $(k + 1)$ -fluxo. Portanto, todo *snark* 4-aresta-crítico (ou 4-vértice-crítico) possui 5-fluxo e não é contra-exemplo para a Conjectura dos 5-Fluxos de Tutte. Essa observação levou Silva e Lucchesi a investigarem

quais *snarks* conhecidos são 4-aresta-críticos, concluindo que diversos *snarks* famosos como o grafo de Petersen, os *snarks* de Blanuša, Loupekine, Celmins-Swart, *double star snark* e *snark de Szekeres* (veja o Apêndice A para figuras de alguns deles), bem como todos os grafos da família infinita dos *flower-snarks* (SILVA; LUCCHESI, 2012) são 4-aresta-críticos. Posteriormente, Silva, Pesci e Lucchesi (SILVA; PESCI; LUCCHESI, 2013) determinaram quantos e quais dos *snarks* com até 28 vértices são 4-aresta-críticos, concluindo que menos de 5% de tais *snarks* são 4-aresta-críticos, por outro lado observaram que todo *snark* G que não é 4-aresta-crítico possui um *snark* H que é 4-aresta-crítico como minor. Assim, um bom entendimento da estrutura dos *snarks* 4-aresta-críticos pode auxiliar na resolução da Conjectura dos 5-Fluxos. O conjunto de *snarks* de até 28 vértices analisado por Silva, Pesci e Lucchesi foi gerado por Brinkmann e Steffen (BRINKMANN; STEFFEN, 1998) e disponibilizado em (BRINKMANN et al., 2013a).

Posteriormente, em 2013, Brinkmann et al. (BRINKMANN et al., 2013b) conseguiram gerar um conjunto de todos os *snarks* de ordem até 36. O objetivo principal deste trabalho foi determinar quais *snarks* deste conjunto maior são 4-aresta-críticos e quais são 4-vértice-críticos. Como contribuições do desenvolvimento do trabalho temos:

- as quantidades e as listas de *snarks* 4-aresta-críticos e 4-vértice-críticos;
- um algoritmo mais rápido para verificar se um *snark* é 4-aresta(ou vértice)-crítico;
- a verificação de que nem todo *snark* 4-aresta-crítico é 4-vértice-crítico;
- a descoberta que a operação de produto interno não preserva fluxo criticalidade de *snarks*.

Parte dessas contribuições foram publicadas em um artigo na conferência LAGOS 2015 (CARNEIRO; SILVA; MCKAY, 2015). As listas de *snarks* 4-aresta-críticos e *snarks* 4-vértice-críticos foram disponibilizadas no *House of Graphs* (BRINKMANN et al., 2013a).

No Capítulo 1, fazemos toda a introdução teórica do que foi necessário para a realização deste trabalho. Nele são definidas colorações de arestas e faces de um grafo assim como algumas de suas relações. Na sequência apresentamos um breve histórico sobre *snarks*. Também definimos k -fluxo e apresentamos todas as propriedades que se fizeram necessárias para a certeza da correção do que foi desenvolvido.

No Capítulo 2, descrevemos as principais versões dos algoritmos desenvolvidos para a determinação dos *snarks* 4-aresta(ou vértice)-críticos assim como apresentamos os tempos necessários para as execuções dos mesmos. Ao final apresentamos as quantidades de *snarks* 4-aresta-críticos e de *snarks* 4-vértice-críticos de ordem até 36 identificados.

Finalmente, no último capítulo, apresentamos a conclusão do que foi realizado neste trabalho e elencamos algumas possíveis atividades como continuação deste trabalho.

1 Conceitos Básicos

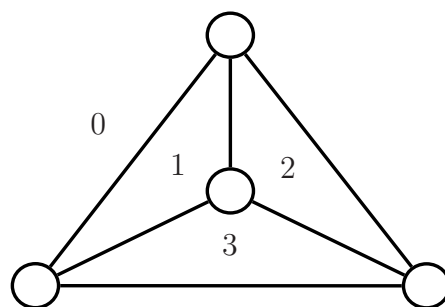
Neste capítulo utilizamos alguns termos e notações que não são definidos nesta dissertação por serem clássicos e seguirem o padrão utilizado no livro (BONDY; MURTY, 2008), tido como referência fundamental em Teoria dos Grafos. Alguns exemplos de definições omitidas são as definições dos conceitos de grau, caminho, ciclo, ciclo par, ciclo ímpar e decomposição. Recomendamos ao leitor que sinta falta de tais definições a procurá-las no livro citado.

1.1 Coloração de Faces e Arestas

Tait mostrou uma equivalência entre coloração de faces e arestas de vital importância no contexto deste trabalho, importância tal que será detalhada a seguir. Antes de reproduzir uma prova para esta equivalência, definimos aqui o que são uma k -coloração de faces e uma k -coloração de arestas.

Uma k -coloração de faces é uma atribuição de no máximo k cores para as faces de um grafo planar G de forma que faces adjacentes não compartilhem uma mesma cor. A Figura 3 ilustra uma 4-coloração de faces, na qual as cores são representadas pelos números de 0 a 3.

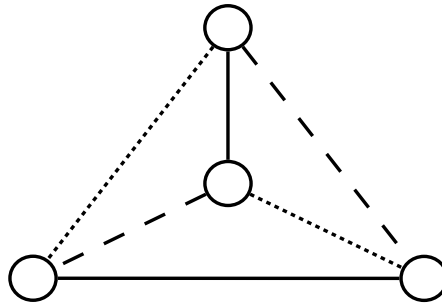
Figura 3 – Uma 4-coloração de faces do K_4



Fonte: Produzido pelos autores

Se um grafo possuir arestas de corte, não será possível encontrar uma k -coloração para o mesmo pois tais arestas possuiriam a mesma face (e a mesma cor) dos dois lados.

Analogamente, uma k -coloração de arestas de um grafo G é uma atribuição de no máximo k cores para as arestas de G de forma que arestas adjacentes possuam cores distintas. A Figura 4 ilustra uma 3-coloração de arestas do K_4 , sendo que cada cor de aresta é representada por um tracejado diferente.

Figura 4 – Uma 3-coloração de arestas do K_4 

Fonte: Produzido pelos autores

Um k -fator de um grafo G é um subgrafo gerador (isto é, com os mesmos vértices que G) k -regular, isto é, todo vértice do subgrafo possui grau k . O caso particular de $k = 1$ é tipicamente chamado de *emparelhamento perfeito*. Em uma 3-coloração de arestas de um grafo cúbico, o conjunto de arestas de uma mesma cor forma um emparelhamento perfeito, pois a cada vértice incide uma aresta de cada uma das três cores. A Proposição 1 segue diretamente desta observação.

Proposição 1 *Seja G um grafo cúbico que admite uma 3-coloração de arestas, então existe uma decomposição de G em 3 emparelhamentos perfeitos disjuntos.*

Note que, como em uma 3-coloração de um grafo cúbico cada cor é um emparelhamento perfeito, o conjunto de arestas de duas cores quaisquer forma um 2-fator, o qual só contém ciclos pares. É possível mostrar usando o famoso Teorema de Tutte (veja (BONDY; MURTY, 2008, Theorem 16.13)) que caracteriza a existência de emparelhamento perfeito em grafos, que todo grafo cúbico sem aresta de corte possui um emparelhamento perfeito. Portanto, os conjuntos de arestas de tais grafos podem ser decompostos em um 1-fator e um 2-fator; o que diferencia grafos que não admitem uma 3-coloração de arestas dos que admitem é a existência de ciclos ímpares no 2-fator para qualquer 1-fator escolhido.

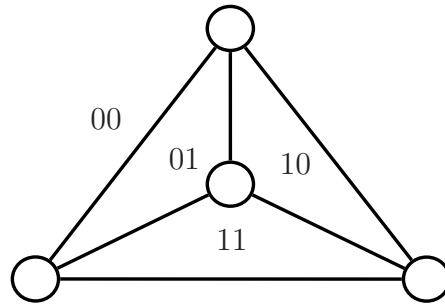
A equivalência de Tait diz que, se um grafo cúbico planar possuir uma 4-coloração de faces, então ele também admitirá uma 3-coloração de suas arestas; sendo a recíproca também verdadeira.

Teorema 3 (Equivalência de Tait) *Um grafo cúbico planar sem aresta de corte possui 4-coloração de faces se, e somente se, admite 3-coloração de arestas.*

Demonstração:

(\Rightarrow) Seja G um grafo cúbico planar sem aresta de corte. Suponha que G possua 4-coloração de faces. Mostraremos que é possível obter uma 3-coloração de arestas. Sejam as quatro cores da coloração de faces denotadas por pares ordenados binários: $C_0 = 00$, $C_1 = 01$, $C_2 = 10$ e $C_3 = 11$. A Figura 5 ilustra uma coloração do K_4 utilizando tais cores.

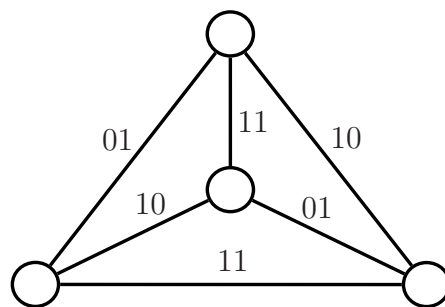
Figura 5 – Uma 4-coloração de faces do K_4 com pares ordenados binários



Fonte: Produzido pelos autores

Para cada aresta e de G atribui-se uma cor obtida através da soma das cores C_i e C_j das faces separadas por e . A soma dessas cores deve ser feita de maneira que cada coordenada seja somada módulo 2, por exemplo, se as faces possuem as cores C_2 e C_3 então a cor atribuída a e é C_1 , ou seja, $C_2 + C_3 = C_1$. A Figura 6 ilustra o resultado desse processo em relação à Figura 5.

Figura 6 – Uma 3-coloração de arestas do K_4 com pares ordenados binários



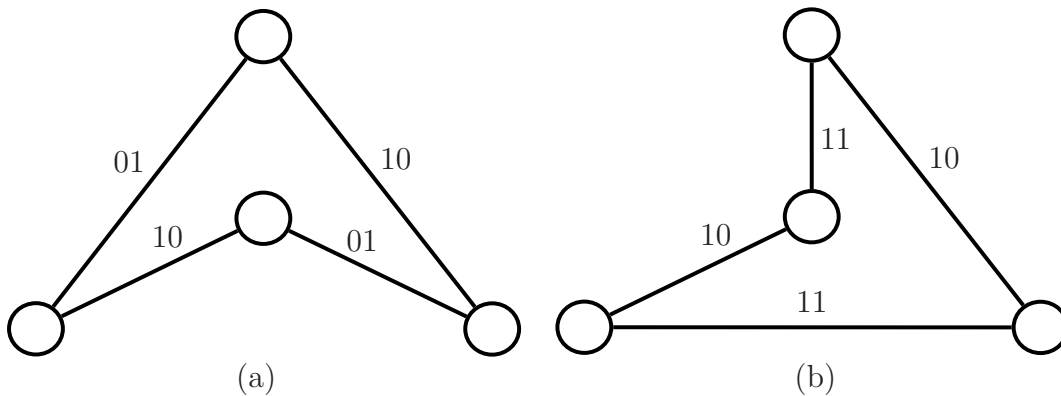
Fonte: Produzido pelos autores

Como G não possui aresta de corte, cada aresta é uma fronteira entre duas faces distintas, logo a cor 00 nunca ocorre como resultado da soma, ou seja, apenas as cores C_1 , C_2 e C_3 podem ocorrer como resultado da soma. Para que esta seja uma 3-coloração de arestas, é necessário mostrar que as arestas incidentes a um mesmo vértice possuem cores distintas. Como G é cúbico, três faces que compartilham um vértice são adjacentes duas a duas e portanto possuem cores distintas no conjunto $\{C_i, C_j, C_k\}$. Se 00 não pertence a

esse conjunto de cores então a soma de qualquer par delas resulta na terceira, no entanto, se $C_k = 00$, duas arestas possuirão as cores C_i e C_j enquanto que a terceira terá a cor resultante de $C_i + C_j$, que é a cor que não estava no conjunto original de cores das faces.

(\Leftarrow) Para a recíproca, suponha que G possui uma 3-coloração de arestas com cores definidas como pares ordenados: $a = 01$, $b = 10$ e $c = 11$. Sejam E_a , E_b e E_c os conjuntos de arestas com cada uma dessas cores. Como G é cúbico, cada cor a , b e c aparece em cada vértice e a união de qualquer par dos conjuntos E_a , E_b e E_c é um 2-fator H_i . Cada face de H_i é uma união de certas faces de G . Sejam $H_1 = E_a \cup E_b$ e $H_2 = E_b \cup E_c$. A Figura 7 ilustra os subgrafos H_1 e H_2 da Figura 6.

Figura 7 – Subgrafos H_1 (a) e H_2 (b) obtidos a partir do K_4

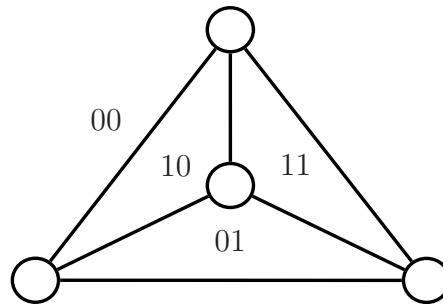


Fonte: Produzido pelos autores

Para cada face de G , atribua a cor de forma que a coordenada i ($i \in \{1, 2\}$) seja a paridade da quantidade de ciclos em H_i que a contenham. A Figura 8 ilustra as cores obtidas através desse processo em relação a Figura 7. Note que a face superior esquerda, interna ao triângulo da fronteira da face externa, está contida em um ciclo de H_1 e nenhum ciclo de H_2 , por isso sua cor é 10.

Sejam F e F' , duas faces de G que compartilham uma aresta e . Como G não possui aresta de corte então F e F' são distintas. A aresta e pertence a um ciclo C em pelo menos um dos subgrafos H_1 e H_2 , pertencendo a ciclos em ambos se for da cor b . Se a cor de e for a ou c , então a paridade da quantidade de ciclos contendo F e F' diferem em H_1 ou em H_2 , respectivamente. Se a cor de e for b , ambas paridades diferem. Como a paridade da quantidade de ciclos difere em pelo menos um de H_1 ou H_2 para qualquer par de faces que compartilham uma aresta, então as cores obtidas para as faces adjacentes são sempre distintas. Logo, o resultado é uma 4-coloração de faces, pois no máximo 4 pares de paridades distintas poderá existir.

□

Figura 8 – A 4-coloração de faces obtida a partir da 3-coloração de arestas do K_4 

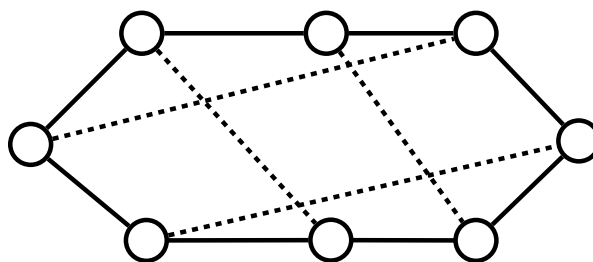
Fonte: Produzido pelos autores

Conforme mostrado na Proposição 2, todo grafo cúbico e hamiltoniano admite 3-coloração de arestas e como Tait acreditava que todo grafo cúbico planar era hamiltoniano, ele acreditava ter demonstrado o Teorema das Quatro Cores.

Proposição 2 *Todo grafo cúbico e hamiltoniano admite 3-coloração de arestas.*

Demonstração: Seja G um grafo cúbico e hamiltoniano e n o número de vértices de G . Todo grafo cúbico possui uma quantidade par de vértices, logo $n \equiv 0 \pmod{2}$. Como G é hamiltoniano, então existe um ciclo hamiltoniano, ou seja, um ciclo de tamanho n . O ciclo hamiltoniano está destacado na Figura 9.

Figura 9 – Ciclo hamiltoniano



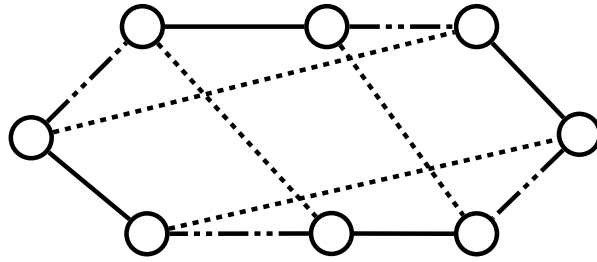
Fonte: Produzido pelos autores

Como o tamanho do ciclo é par, é possível colorir suas arestas com duas cores, alternando-as. A coloração do ciclo hamiltoniano da Figura 9 está ilustrado na Figura 10.

Com essa coloração do ciclo, todo vértice possui duas arestas coloridas com o mesmo par de cores, logo é possível colorir as arestas remanescentes com uma terceira cor.

□

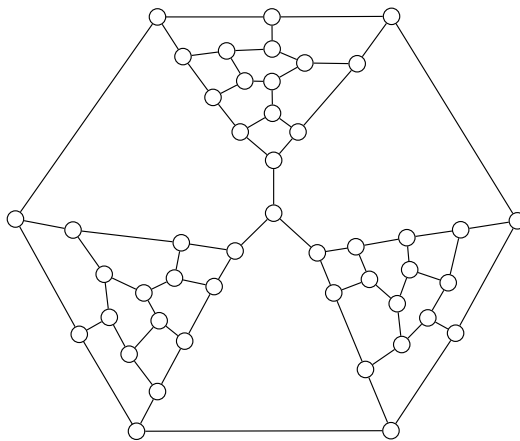
Figura 10 – Coloração do ciclo hamiltoniano



Fonte: Produzido pelos autores

Em 1946, Tutte encontrou um grafo cúbico e planar porém não hamiltoniano. Esse grafo mostra que, apesar de muito interessante, a equivalência de Tait não é uma prova para o Teorema das Quatro Cores. A Figura 11 que ilustra o grafo encontrado por Tutte é uma reprodução da figura apresentada em (BONDY; MURTY, 2008, Fig. 18.7).

Figura 11 – Grafo cúbico e planar não hamiltoniano



Fonte: Produzido pelos autores

1.2 A Caça aos *Snarks*

Conforme mencionado anteriormente, por mais de meio século desde a descoberta do grafo de Petersen, pouquíssimos exemplos de grafos cúbicos sem 3-coloração de arestas eram conhecidos. Foi esse fato que motivou batizar tais grafos de *snarks*. Porém, em 1975, Isaacs (ISAACS, 1975) apresentou uma operação conhecida como produto interno, que permite construir um novo *snark* a partir de dois *snarks* conhecidos. No mesmo artigo (ISAACS, 1975), Isaacs mostrou também que existe uma família infinita de *snarks* conhecida como *flower-snarks* em que nenhum dos grafos da família pode ser construído

a partir de outros *snarks* menores através da operação de produto interno. Nas próximas subseções, descrevemos as construções destas famílias infinitas apresentadas por Isaacs. Uma outra contribuição de Isaacs foi a descoberta de um *snark* de 30 vértices que não pertence a nenhuma dessas famílias, o *double-star snark*.

Alguns outros *snarks* foram descobertos na décadas seguintes. Os dois Celmins-Swart *snarks* de 26 vértices foram descobertos em 1979 (CELMINS; SWART, 1979). Goldberg apresentou, em 1981 (GOLDBERG, 1981), uma outra família infinita de *snarks* chamada Goldberg *snarks*. Em 1989, John J. Watkins (WATKINS, 1989), encontrou um *snark* de 50 vértices, conhecido como Watkins *snark*. O Apêndice A apresenta figuras que ilustram alguns dos *snarks* mencionados aqui.

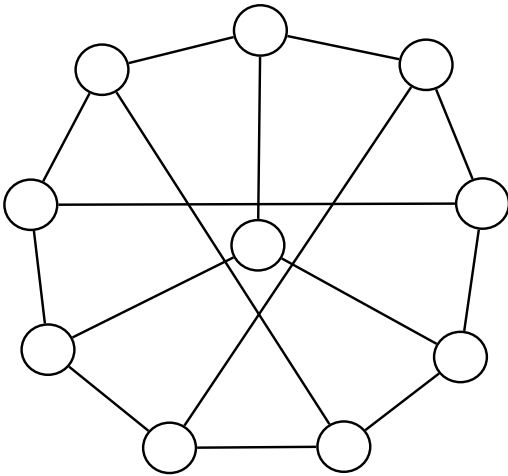
Existem algumas operações, mais simples que o produto interno, que permitem criar *snarks* maiores a partir de menores, por exemplo transformar um vértice em um triângulo. A Figura 12 ilustra a transformação de um vértice do grafo de Petersen em um triângulo. No entanto, estas não são consideradas de muito interesse pois não mudam essencialmente a estrutura do grafo. Na verdade, os *snarks* considerados mais interessantes são exatamente os minimais, isto é, aqueles que não são obtidos por simples modificações de *snarks* menores. Por isso os pesquisadores acabaram adicionando à definição de *snark* mais restrições relativas a duas características de grafos, a cintura e a conexidade cíclica. A *cintura* de um grafo é o comprimento do seu menor ciclo. Um grafo é *k-cíclicamente-aresta-conexo* se não existe um corte com menos de k arestas cuja remoção gere pelo menos duas componentes conexas com ciclos. Formalmente, um *snark* é definido como um grafo cúbico de cintura pelo menos cinco, 4-cíclicamente-aresta-conexo e que não admite 3-coloração de arestas. A partir deste ponto do texto, quando nos referirmos a *snarks*, estaremos nos referindo a esta definição formal.

A aparente raridade de *snarks* foi desmistificada por Isaacs, no entanto, a busca pelo entendimento da estrutura que não permite que tais grafos possuam 3-coloração de arestas continua. Em 1964, Vizing (VIZING, 1964) demonstrou que a quantidade mínima de cores necessárias para colorir as arestas de um grafo G , denotada $\chi'(G)$, depende do maior grau de seus vértices, denotado por $\Delta(G)$. Ele mostrou que todo e qualquer grafo pertence a uma de duas classes, *Classe 1* ou *Classe 2*. A *Classe 1* contém todos os grafos em que $\chi'(G) = \Delta(G)$. Já a *Classe 2* contém todos os outros grafos, eles têm $\chi'(G) = \Delta(G) + 1$. Claramente, todos os *snarks* pertencem à *Classe 2*. Algum tempo depois, em 1981, Holyer (HOLYER, 1981) mostrou que determinar se um grafo pertence à *Classe 1* ou à *Classe 2* é um problema NP-Completo.

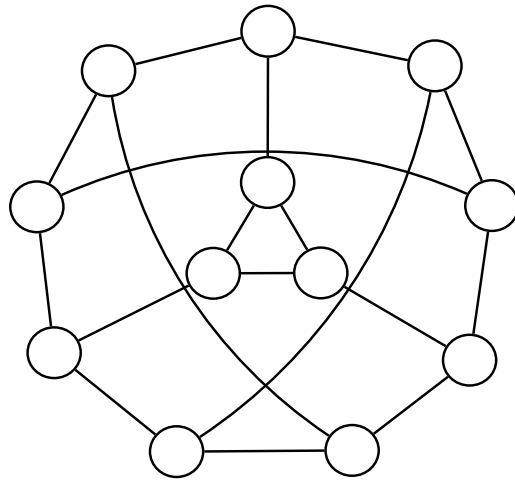
Em seu livro (WEST, 1996, 7.3.10. Lemma), West mostra operações de reduções de grafos cúbicos que não são *snarks* e que não possuem 3-coloração de arestas a *snarks*. Estas reduções eliminam 3-cortes não triviais e ciclos de tamanhos 3 e 4 mantendo a propriedade de não possuir 3-coloração de arestas. Estas reduções são operações de com-

Figura 12 – Grafo de Petersen e a transformação de um vértice em um triângulo

(a) Grafo de Petersen



(b) Transformação do vértice em um triângulo



Fonte: Produzido pelos autores

plexidade polinomial. Além disso, cada uma delas diminui o tamanho do grafo, portanto a quantidade de vezes que estas operações precisam ser executadas para obter um *snark* a partir de um grafo cúbico sem 3-coloração de arestas também é polinomial. Concluímos então que existe redução polinomial do problema de determinar se um grafo cúbico tem 3-coloração de arestas ao problema de determinar se um grafo cúbico com cintura pelo menos cinco e 4-cíclicamente-aresta-conexo é *snark*. Logo, determinar se um grafo cúbico é *snark* é um problema *NP-Completo*.

Tutte ([TUTTE, 1966](#)) conjecturou que todo *snark* possui o grafo de Petersen como minor. Esta conjectura foi confirmada em 1997 por Robertson, Sanders, Seymour e Thomas ([ROBERTSON; SEYMOUR; THOMAS, 1997](#)).

A seguir mostramos as duas construções apresentadas por Isaacs. A operação de produto interno e a construção de *Flower Snarks*.

1.2.1 Produto Interno

Sejam G e H dois grafos cúbicos, $e_1 = uv$ e $e_2 = wx$ arestas de G , e y e z um par de vértices adjacentes de H . Sejam y_1 e y_2 os vértices adjacentes a y distintos de z , e z_1 e z_2 os vértices adjacentes a z distintos de y . Considere as seguintes operações para a obtenção de um novo grafo cúbico S_{GH} a partir de G e H :

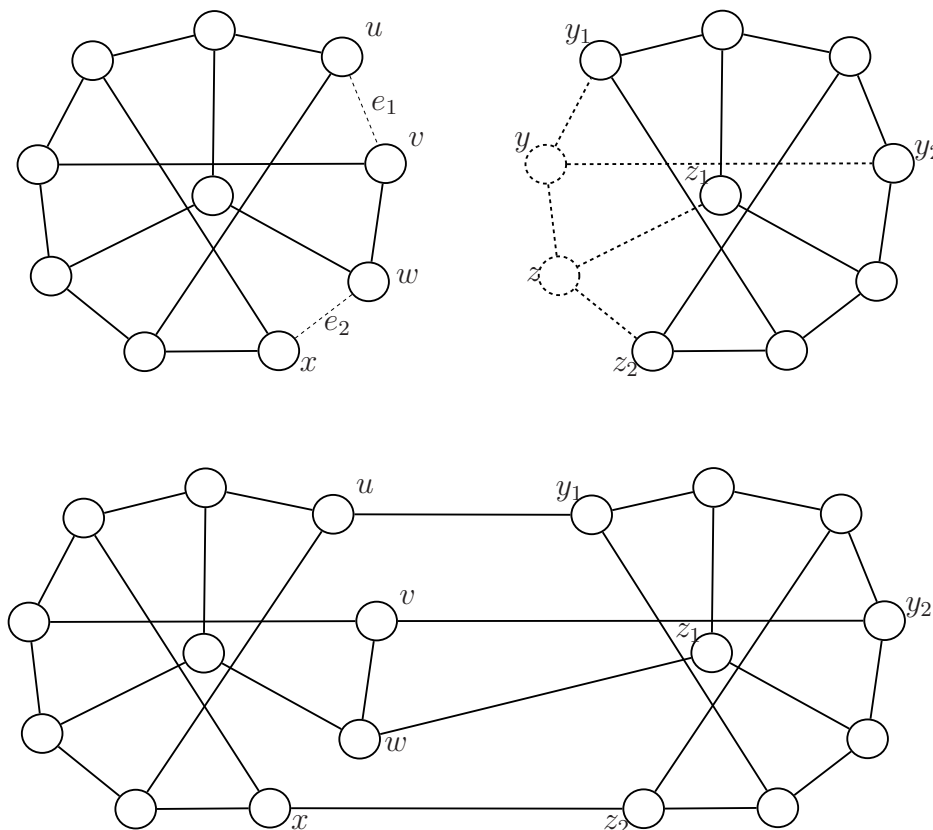
- remover as arestas e_1 e e_2 de G ;
- remover os vértices y e z de H ;
- adicionar as arestas: uy_1, vy_2, wz_1, xz_2

Todo grafo S_{GH} assim obtido é chamado de *produto interno* de G e H . E se ambos G e H forem livres de 3-coloração de arestas, o produto interno de G e H também o será, conforme mostrado na Proposição 3 a seguir.

Proposição 3 *Sejam G e H dois grafos cúbicos sem 3-coloração de arestas. Então o grafo S_{GH} obtido a partir do produto interno de G e H também é um grafo cúbico sem 3-coloração de arestas.*

Demonstração: Sejam $e_1 = uv$ e $e_2 = wx$ as arestas removidas de G , e y e z o par de vértices adjacentes removidos de H . Sejam y_1 e y_2 os vértices adjacentes a y distintos de z , e z_1 e z_2 os vértices adjacentes a z distintos de y . Sejam uy_1, vy_2, wz_1 e xz_2 as arestas adicionadas para a obtenção de S_{GH} . A Figura 13 ilustra a construção de um dos *snarks* de Blanuša a partir do produto interno de dois grafos de Petersen. Os grafos G e H são cúbicos, logo, por construção, S_{GH} também é.

Figura 13 – Produto interno entre dois grafos de Petersen



Fonte: Produzido pelos autores

Suponha por absurdo que S_{GH} tem uma 3-coloração de arestas f , vamos então mostrar que é possível obter uma 3-coloração de arestas de G ou de H . Como o conjunto

de arestas de cada cor c de f é um emparelhamento perfeito de S_{GH} , então c aparece um número par de vezes nas arestas em $\{uy_1, vy_2, wz_1, xz_2\}$. Se a cor de uy_1 for a mesma que a cor de vy_2 , então a cor de wz_1 é igual à cor de xz_2 . A partir da restrição de f a G e das atribuições das cores de uy_1 a uv e de wz_1 a wx , obtém-se uma 3-coloração de G , uma contradição. Se a cor de uy_1 for diferente da cor de vy_2 , então, sem perda de generalidade, podemos assumir que a cor de uy_1 é igual à cor de wz_1 , e que a cor de vy_2 é igual à cor de xz_2 . A partir da restrição de f a H , da atribuição da cor de uy_1 à yy_1 e à zz_1 , da atribuição da cor de vy_2 à yy_2 e à zz_2 , e da atribuição da terceira cor à yz , obtém-se uma 3-coloração de arestas de H , novamente uma contradição. Concluimos então que S_{GH} não pode ter 3-coloração.

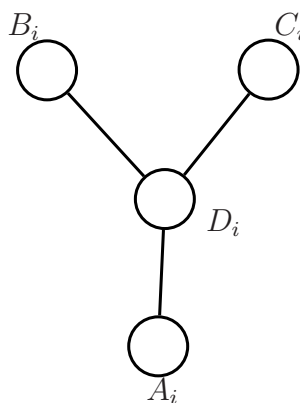
□

1.2.2 Flower-snarks - construção

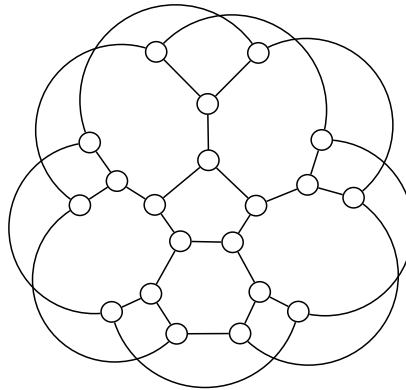
Para qualquer valor natural ímpar de $n \geq 5$, o *flower-snark* J_n pode ser construído da seguinte maneira:

- criar n cópias do grafo estrela S_3 ilustrado na Figura 14, onde i é o índice da cópia;
- adicionar as arestas $\{A_1, A_2\}, \{A_2, A_3\}, \dots, \{A_n, A_1\}$;
- adicionar as arestas $\{B_1, B_2\}, \{B_2, B_3\}, \dots, \{B_{n-1}, B_n\}$;
- adicionar as arestas $\{C_1, C_2\}, \{C_2, C_3\}, \dots, \{C_{n-1}, C_n\}$;
- adicionar as arestas $\{B_n, C_1\}$ e $\{C_n, B_1\}$.

Figura 14 – Grafo estrela S_3



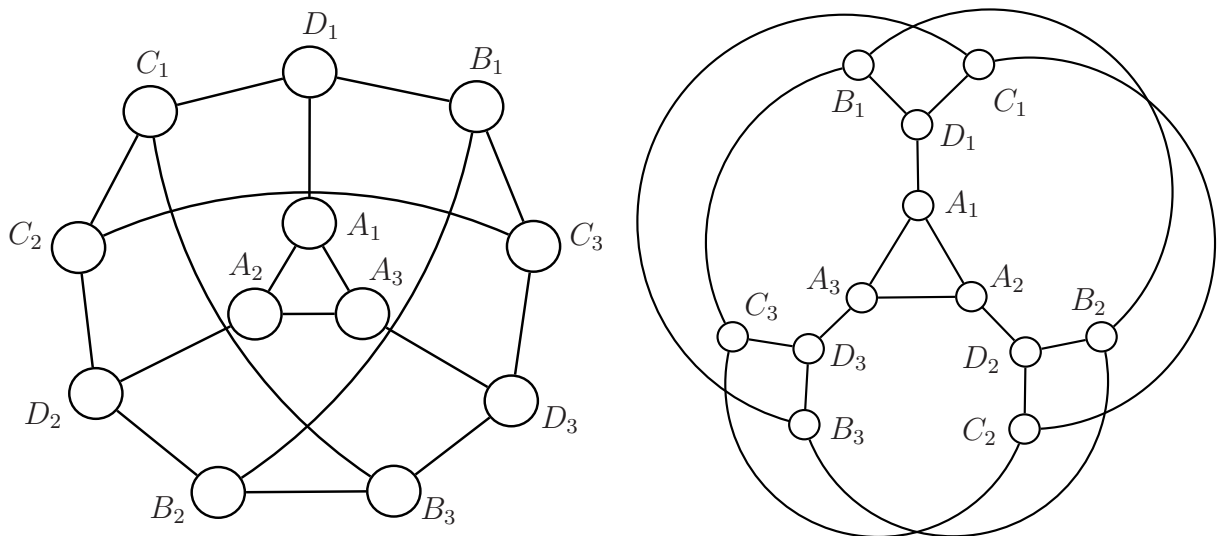
Fonte: Produzido pelos autores

Figura 15 – *Flower-snark* J_5 

Fonte: Produzido pelos autores

A Figura 15 ilustra o *flower-snark* J_5 . O J_3 , ilustrado na Figura 16 pode ser obtido a partir grafo de Petersen pela substituição de um dos vértices por um triângulo. Como a definição de *snarks* inclui a restrição de cintura pelo menos cinco então o J_3 não é um *snark*.

A explicação de porque tais grafos são snarks será omitida, mas está relacionada ao fato de que as arestas $\{C_n, D_1\}$ e $\{D_n, C_1\}$ produzem um tipo de cruzamento que inviabiliza a 3-coloração das arestas. As arestas de cada estrela S_3 precisam ser coloridas de uma forma padronizada para usar apenas três cores, e esse padrão é quebrado quando os ciclos são fechados com o cruzamento destas duas arestas; sem o cruzamento é perfeitamente possível 3-colorir os grafos.

Figura 16 – *Flower-snark* J_3 

Fonte: Produzido pelos autores

1.3 k -Fluxos e suas propriedades

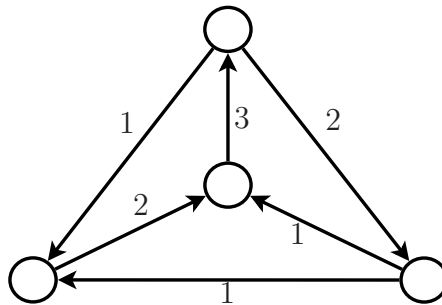
Nessa seção definimos o que é um k -fluxo e apresentamos algumas propriedades para que, posteriormente, relacionemos com a coloração de arestas vista na Seção 1.1.

Seja G um grafo não orientado, D uma função que define orientações para as arestas de G . Denotamos por $\partial^+(v)$ o conjunto das arestas incidentes em v orientadas em D saindo de v ; analogamente, $\partial^-(v)$ é o conjunto das arestas incidentes em v orientadas em D entrando em v . Seja f uma função que define pesos inteiros para as arestas de G . Dizemos que o fluxo líquido $f(v)$ ¹ de um vértice v é a soma dos pesos das arestas que saem de v menos a soma dos pesos das arestas que entram em v , ou seja,

$$f(v) = \sum_{e \in \partial^+(v)} f(e) - \sum_{e \in \partial^-(v)} f(e).$$

Um k -fluxo parcial é um par de funções (D, f) tal que todo vértice $v \in V$ tem fluxo líquido nulo e o peso $f(e)$ atribuído a cada aresta $e \in E$ é menor que k , isto é, pertence ao conjunto $\{0, 1, 2, \dots, k-1\}$. O suporte de um k -fluxo parcial, denotado $\text{sup}(D, f)$, é o conjunto de arestas de peso não nulo. Quando o $\text{sup}(D, f) = E$, dizemos que (D, f) é um k -fluxo. A Figura 17 ilustra um 4-fluxo do K_4 . A partir dessa definição enunciamos a Proposição 4.

Figura 17 – Um 4-fluxo do K_4



Fonte: Produzido pelos autores

Proposição 4 *Todo k -fluxo é um $(k+1)$ -fluxo.*

Demonstração: Pela definição, se a toda aresta $e \in E$ é atribuído um peso inteiro que pertence ao conjunto $\{1, 2, \dots, k-1\}$ então esse peso também pertence ao conjunto $\{1, 2, \dots, k\}$. \square

¹ Faremos um abuso de notação denotando por f tanto a função peso definida nas arestas quanto o fluxo líquido definido nos vértices. Quando há parâmetro e o mesmo é um vértice, a notação se refere a fluxo líquido do vértice; caso contrário a notação refere-se a uma função peso nas arestas.

Seja X um conjunto de vértices de um grafo G e $\partial(X)$, denominado *corte* de X , o conjunto das arestas com um extremo em X e outro no complemento \bar{X} de X . Denotamos por $\partial^+(X)$ o conjunto das arestas orientadas em D saindo de X e entrando em \bar{X} ; analogamente, denotamos por $\partial^-(X)$ o conjunto das arestas orientadas em D saindo de \bar{X} e entrando em X . Dado um par de funções (D, f) de G que definem orientações e pesos, o *fluxo líquido no corte* $\partial(X)$, denotado $f(X)$, é dado pela soma dos pesos das arestas que saem de X menos a soma dos pesos das arestas que entram em X , ou seja,

$$f(X) = \sum_{e \in \partial^+(X)} f(e) - \sum_{e \in \partial^-(X)} f(e).$$

Note que, se X é vazio, então $f(X) = 0$. O fluxo líquido em qualquer corte $\partial(X)$ é dado pela soma dos fluxos líquidos dos vértices em X ; além disso, os fluxos líquidos em X e seu complemento \bar{X} devem ser complementares. Ambas afirmações são demonstradas na Proposição 5 a seguir.

Proposição 5 *Dado um par de funções (D, f) que definem orientações e pesos para as arestas de um grafo G , e um subconjunto X de vértices de G , vale que:*

$$\begin{aligned} f(X) &= \sum_{v \in X} f(v), \text{ e} \\ f(X) &= -f(\bar{X}). \end{aligned}$$

Demonstração: Pela definição de fluxo líquido em um vértice temos que

$$f(v) = \sum_{e \in \partial^+(v)} f(e) - \sum_{e \in \partial^-(v)} f(e).$$

Portanto,

$$\sum_{v \in X} f(v) = \sum_{v \in X} \left[\sum_{e \in \partial^+(v)} f(e) - \sum_{e \in \partial^-(v)} f(e) \right] \quad (1.1)$$

$$= \sum_{e \in \partial^+(X)} f(e) - \sum_{e \in \partial^-(X)} f(e) \quad (1.2)$$

$$= f(X). \quad (1.3)$$

A igualdade (1.2) segue do fato que toda aresta com ambos os extremos em X aparecem duas vezes na soma, uma vez somando e outra subtraindo, não afetando portanto o resultado final. A igualdade $f(X) = -f(\bar{X})$ segue diretamente da observação que as arestas em $\partial(\bar{X})$ são as mesmas de $\partial(X)$, apenas com as orientações invertidas. \square

Pela Proposição 5, todos os cortes de um k -fluxo têm fluxo líquido igual a zero e nenhum grafo com aresta de corte pode ter k -fluxo, independentemente do valor de k .

Uma aresta de corte precisaria ter peso nulo, contradizendo a definição. Além disso, se um par de funções (D, f) deixa algum vértice com fluxo líquido diferente de zero, então deve existir pelo menos mais um vértice com fluxo líquido não nulo.

Existe uma caracterização bem conhecida para os grafos que possuem 2-fluxo, esses são exatamente os grafos pares, ou seja, grafos em que todo vértice tem grau par. Demonstramos tal caracterização na Proposição 6.

Proposição 6 *Um grafo G possui 2-fluxo se, e somente se, é par.*

Demonstração: (\Rightarrow) Seja (D, f) um 2-fluxo de um grafo G . Como o peso de todas as arestas de G é um e o fluxo líquido de todos os vértices de G é nulo então temos que o número de arestas entrando em qualquer vértice é igual à quantidade de arestas saindo; logo a quantidade de arestas incidentes a qualquer vértice é par, ou seja, G é par.

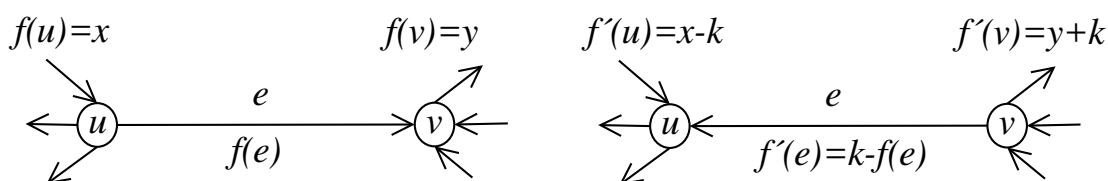
(\Leftarrow) Seja G um grafo par, vamos mostrar que é possível obter uma orientação D tal que a quantidade de arestas entrando é igual à quantidade de arestas saindo de qualquer vértice. Supomos que G é conexo, caso contrário, é possível obter a orientação da mesma maneira para cada componente conexa. Se G é conexo e par então G é Euleriano; logo, ele possui um circuito Euleriano. Se orientarmos as arestas no sentido do percurso do circuito Euleriano, certamente conseguiremos a orientação desejada. \square

1.3.1 k -Fluxos modulares

Um k -fluxo modular parcial (D, f) é definido de forma muito similar a um k -fluxo parcial, com a única diferença que é permitido que o fluxo líquido em cada vértice seja um múltiplo de k , isto é, congruente a $0 \pmod k$. Quando $\text{sup}(D, f) = E$, dizemos que (D, f) é um k -fluxo modular. Claramente, todo k -fluxo é também um k -fluxo modular.

Sabemos que a fixação de uma orientação arbitrária D para as arestas de G não afeta o fato de um grafo admitir um k -fluxo modular ou não, conforme demonstrado na Proposição 7 a seguir.

Figura 18 – k -fluxo modular

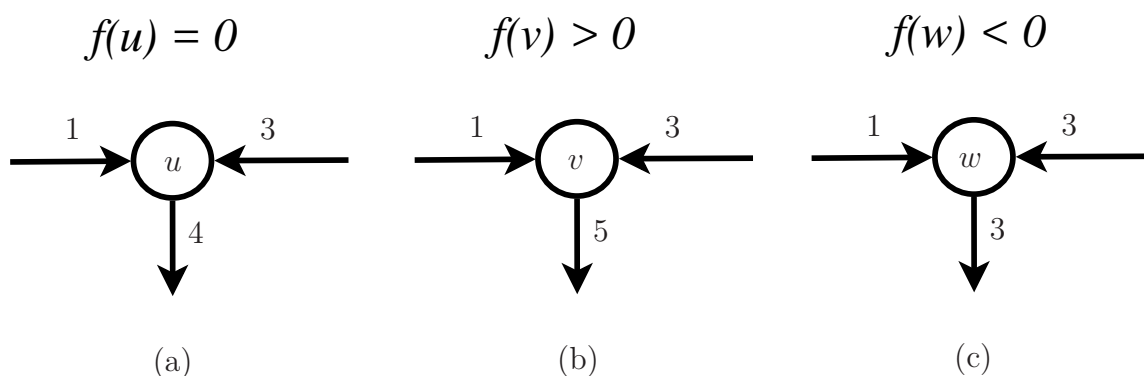


Proposição 7 Se G tem um k -fluxo modular parcial (D, f) , tem também um k -fluxo modular parcial (D', f') de mesmo suporte para qualquer orientação D' distinta de D .

Demonstração: Em um k -fluxo modular, seja $e = uv$ uma aresta do suporte. Podemos inverter a orientação de e e complementar seu peso módulo k , conforme ilustrado na Figura 18. O resultado é um novo k -fluxo modular parcial de mesmo suporte, já que o novo peso da aresta continua dentro do conjunto válido e apenas os fluxos líquidos dos extremos u e v de e são alterados, e por um fator k . Portanto, se um grafo G tem um k -fluxo modular parcial para alguma orientação D das arestas, tem também um k -fluxo modular parcial para qualquer outra orientação D' , com mesmo suporte. \square

Dada uma orientação D e uma função f que atribui pesos às arestas, um vértice é dito *equilibrado* quando $f(v) = 0$, *positivo* quando $f(v) > 0$ e *negativo* quando $f(v) < 0$. A Figura 19 ilustra vértices equilibrados, positivos e negativos. O *desvio do equilíbrio* de (D, f) é a soma dos fluxos líquidos dos vértices positivos; ou zero, caso não haja vértices positivos. Note que o desvio do equilíbrio será também o valor absoluto da soma dos fluxos líquidos dos vértices negativos, pela Proposição 5 e pelo fato que $f(X) = 0$ quando X é o conjunto de vértices do grafo.

Figura 19 – Vértices equilibrado (a), positivo (b) e negativo (c)



Fonte: Produzido pelos autores

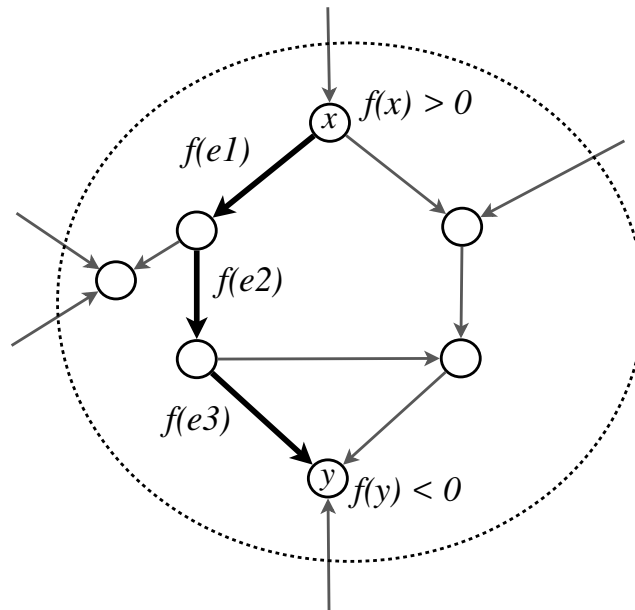
Em um k -fluxo modular, dado um vértice positivo x , sempre existe um caminho orientado de x a um vértice negativo y , conforme demonstrado no Lema 1.

Lema 1 Dado um vértice positivo x de um k -fluxo modular (D, f) , existe um caminho orientado em D de x a um vértice negativo y .

Demonstração: Seja $R(x)$ o conjunto de todos os vértices alcançáveis a partir de x por caminhos orientados segundo D e $T_x = R(x) \cup \{x\}$. Por construção, se o corte $\partial(T_x)$

contém arestas então ele é composto apenas por arestas que entram em T_x conforme ilustrado na Figura 20. Portanto, $f(T_x) \leq 0$ valendo a igualdade somente se o corte $\partial(T_x)$ é vazio. Pela Proposição 5 e pelo fato de que x é positivo, concluímos que existe vértice negativo y em T_x . Certamente, $x \neq y$, visto que x é positivo. Assim, $y \in R(x)$, e existe caminho direcionado de x a y .

Figura 20 – Corte de T_x



Fonte: Produzido pelos autores

□

Tutte se baseou nessa observação para mostrar que todo k -fluxo modular pode ser transformado em um k -fluxo, conforme demonstrado na Proposição 8.

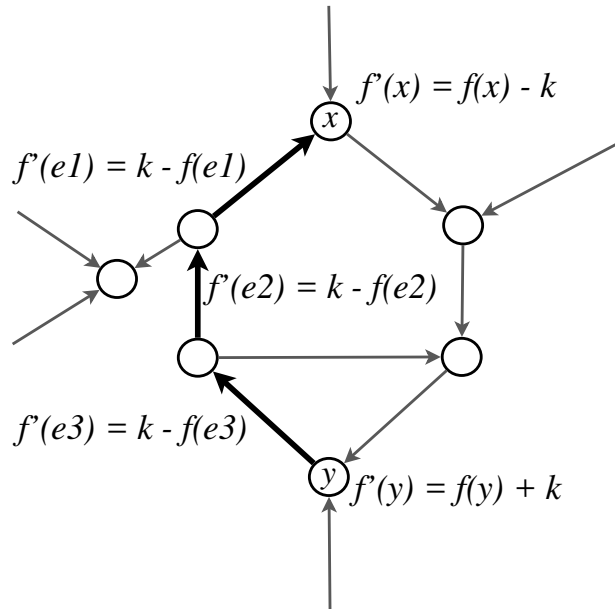
Proposição 8 *Se um grafo possui um k -fluxo modular, então também possui um k -fluxo.*

Demonstração: A demonstração segue por indução no desvio do equilíbrio d do k -fluxo modular (D, f) . Se $d = 0$, então (D, f) já é um k -fluxo. Portanto, podemos supor que $d > 0$. Vamos mostrar que (D, f) pode ser transformado em um k -fluxo modular (D', f') com desvio do equilíbrio menor.

Seja x um vértice positivo e y um vértice negativo alcançável a partir de x cuja existência, de acordo com o Lema 1, é garantida. Denotamos por π_{xy} o caminho de x a y orientado em D . Se a orientação das arestas de π_{xy} forem invertidas e seus pesos complementados módulo k , é possível provar que o k -fluxo modular (D', f') obtido possui

desvio do equilíbrio menor do que o de (D, f) . A Figura 21 mostra o resultado desse processo em relação a Figura 20.

Figura 21 – Inversão e alteração dos pesos das arestas de π_{xy}



Fonte: Produzido pelos autores

Como a cada vértice interno v do caminho π_{xy} incidem duas arestas de direções opostas, ao inverter suas orientações e complementar seus pesos módulo k , somamos e subtraímos k do fluxo líquido de v , ou seja, ele não é alterado. Quanto ao vértice x , a única aresta de π_{xy} incidente era orientada saindo de x . Então, ao obter (D', f') , subtraímos k do fluxo líquido de x . Analogamente, para y , somamos k ao fluxo líquido. Concluímos então que (D', f') é um k -fluxo modular com desvio do equilíbrio menor, precisamente $d - k$. Segue por hipótese de indução que G tem um k -fluxo. \square

Como todo k -fluxo é também um k -fluxo modular, segue da Proposição 8 que determinar se um grafo admite um k -fluxo é equivalente a determinar se admite um k -fluxo modular.

Corolário 1 *Um grafo possui k -fluxo se, e somente se, possui k -fluxo modular.*

1.3.2 Operações em k -Fluxos

Definimos aqui duas operações que posteriormente aplicaremos a k -fluxos. Ambas operações são realizadas em k -fluxos de um grafo G sem aresta de corte e resultam em k -fluxos (parciais) também de G .

Seja (D, f) um k -fluxo, a multiplicação escalar de (D, f) por n , sendo $n \in \mathbb{N}^*$, denotada por

$$(D, f') = n \cdot (D, f),$$

resulta em um k' -fluxo (D, f') onde:

$$\forall e \in E, f'(e) = n \cdot f(e), \text{ e}$$

$$k' = n \cdot k - n + 1$$

O par de funções (D, f') é um k' -fluxo pois, $f'(e)$ estará no intervalo $\{1, \dots, n \cdot k - n\}$ para toda aresta e e, para qualquer vértice v , as arestas nele incidentes tiveram seus valores multiplicados por n , um fator comum na somatória do fluxo líquido de v , ou seja,

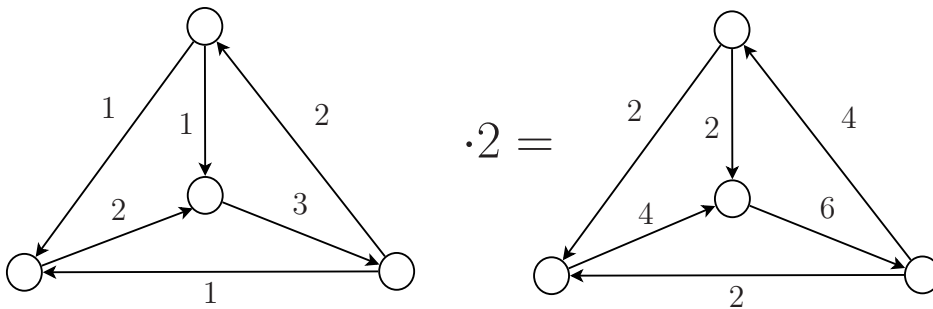
$$\forall v \in V, f'(v) = \sum_{e \in \partial^+(v)} n \cdot f(e) - \sum_{e \in \partial^-(v)} n \cdot f(e) \quad (1.4)$$

$$= n \cdot \left(\sum_{e \in \partial^+(v)} f(e) - \sum_{e \in \partial^-(v)} f(e) \right) \quad (1.5)$$

$$= n \cdot f(v). \quad (1.6)$$

Claramente, se $f(v) = 0$ então $f'(v) = 0$ também. A Figura 22 ilustra a multiplicação de um 4-fluxo do K_4 por 2, que nos permite obter um 7-fluxo para o K_4 .

Figura 22 – Multiplicação de um 4-fluxo do K_4 por 2



Fonte: Produzido pelos autores

Sejam (D_1, f_1) e (D_2, f_2) um k_1 -fluxo e um k_2 -fluxo de G , respectivamente. A segunda operação que definimos é a soma de fluxos, denotada

$$(D_3, f_3) = (D_1, f_1) + (D_2, f_2).$$

A soma de fluxos resultará em um k_3 -fluxo parcial onde:

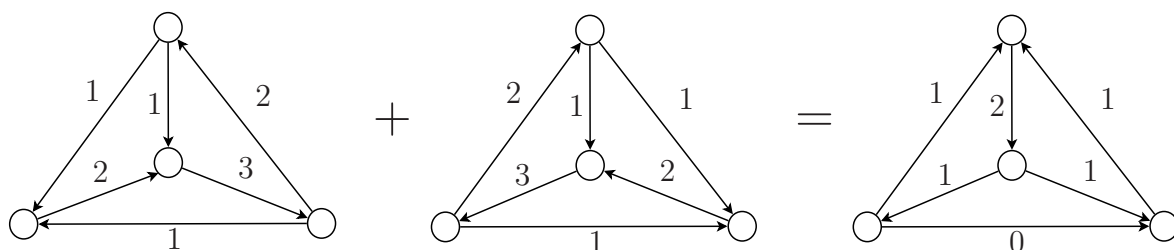
$$k_3 = k_2 + k_1 - 1 \quad (1.7)$$

$$\forall e \in E, D_3(e) = \begin{cases} D_1(e), & \text{se } f_1(e) > f_2(e) \\ D_2(e), & \text{se } f_1(e) \leq f_2(e) \end{cases} \quad (1.8)$$

$$\forall e \in E, f_3(e) = \begin{cases} f_1(e) + f_2(e), & \text{se } D_1(e) = D_2(e) \\ f_1(e) - f_2(e), & \text{se } D_1(e) \neq D_2(e) \text{ e } f_1(e) > f_2(e) \\ f_2(e) - f_1(e), & \text{se } D_1(e) \neq D_2(e) \text{ e } f_2(e) \geq f_1(e). \end{cases} \quad (1.9)$$

Note que em (D_3, f_3) , para todo vértice v o fluxo líquido $f_3(v) = f_1(v) + f_2(v) = 0$ segundo a definição de soma de fluxos. Além disso, $f_3(e) \in \{0, \dots, k_1 + k_2 - 2\}$ para toda aresta e . Portanto, (D_3, f_3) é um k_3 -fluxo parcial. É possível que o suporte de (D_3, f_3) não seja o conjunto de arestas E , pois caso uma aresta e tenha orientações opostas em D_1 e D_2 porém o mesmo peso em f_1 e f_2 então $f_3(e) = 0$. A Figura 23 ilustra a soma de dois 4-fluxos do K_4 .

Figura 23 – Soma de dois 4-fluxos do K_4



Fonte: Produzido pelos autores

1.4 k -Fluxos e colorações

Tutte encontrou uma equivalência análoga à de Tait que relaciona a existência de uma 3-coloração de arestas em um grafo cúbico qualquer à existência de 4-fluxo. A equivalência de Tutte é enunciada a seguir e de crucial importância no contexto deste projeto.

Teorema 4 (Equivalência de Tutte) *Um grafo cúbico admite 3-coloração de arestas se, e somente se, possui 4-fluxo.*

Demonstração: (\Rightarrow) Seja G um grafo cúbico 3-aresta colorido; sejam $\{1,2,3\}$ as cores da coloração. Sem perda de generalidade, a partir da remoção das arestas de cor 3 de G obtém-se um 2-fator C_{12} . Um 2-fator C_{23} pode ser obtido da mesma maneira removendo

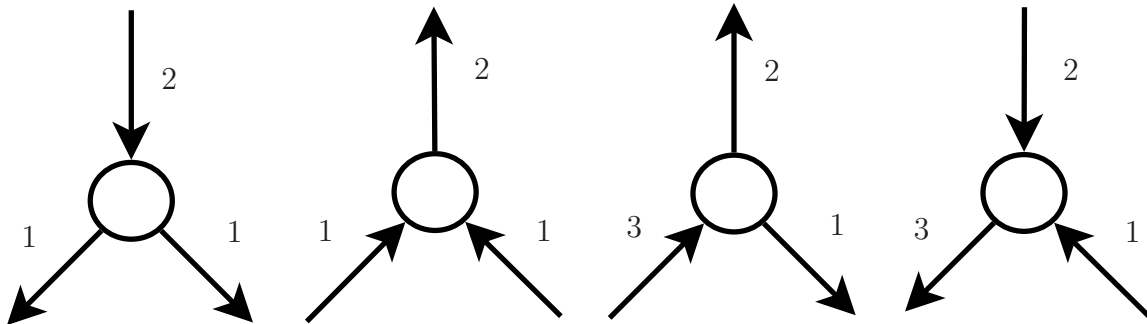
as arestas de cor 1. Como C_{12} e C_{23} são 2-fatores, ambos possuem 2-fluxo, digamos (D_1, f_1) e (D_2, f_2) , respectivamente. Então, tomando

$$(D, f) = (D_1, f_1) + 2 \cdot (D_2, f_2),$$

temos um 4-fluxo parcial de G . O balanceamento em cada vértice segue do fato que somamos dois fluxos. É fácil ver que em (D, f) o maior valor de peso de qualquer aresta é 3, sendo que apenas as arestas de cor 2 podem ter tal peso, quando suas orientações coincidem em (D_1, f_1) e (D_2, f_2) ; quando não coincidem, também não têm peso nulo pois (D_2, f_2) foi multiplicado por 2. As demais arestas, de cor 1 ou 3, terão peso 1 ou 2, respectivamente. Concluímos então que (D, f) é um 4-fluxo de G .

(\Leftarrow) Vamos agora supor que G tem 4-fluxo e mostrar como conseguimos obter uma 3-coloração. Como G é cúbico, cada vértice equilibrado tem, necessariamente, uma aresta de peso 2 e duas de peso 1 em direção oposta, ou uma de peso 1 e outra de peso 2 na mesma direção e outra de peso 3 em direção oposta; a Figura 24 ilustra as possíveis atribuições de pesos e orientações às arestas incidentes em um vértice equilibrado de um grafo cúbico. Portanto, as arestas de peso 2 formam um emparelhamento perfeito de G , digamos M .

Figura 24 – Possibilidades de equilíbrio de um vértice de grau 3

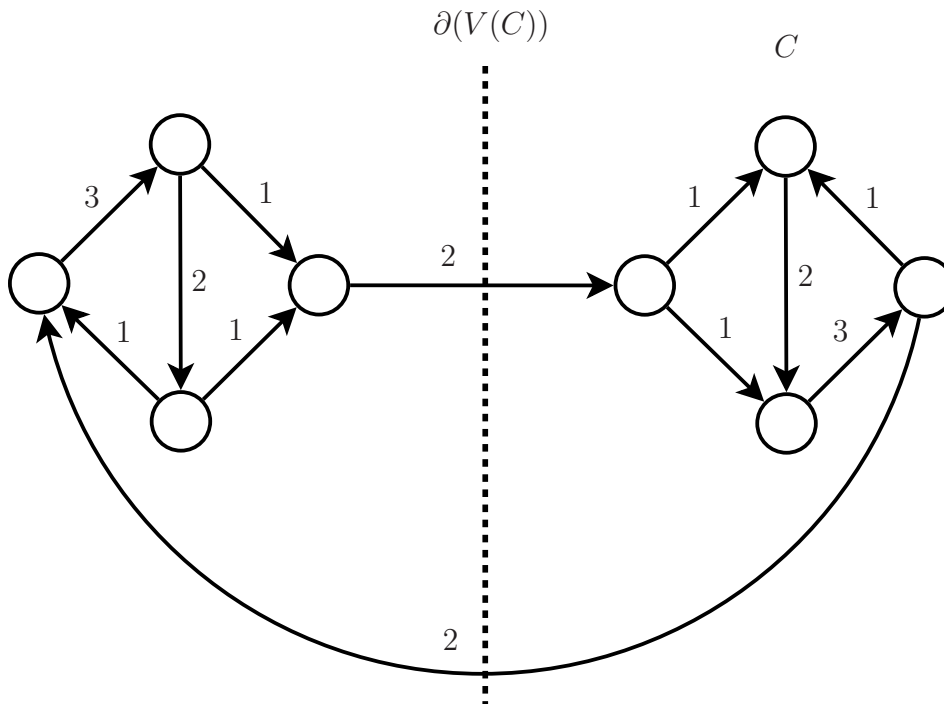


Fonte: Produzido pelos autores

As arestas de M devem ser coloridas com a cor 1. Resta mostrar que as arestas restantes, $E(G) - M$, podem ser coloridas com as duas cores restantes. Mas o subgrafo $G - M$ é um 2-fator, portanto é necessário que contenha apenas ciclos pares para que possa ser colorido com duas cores. Considere um ciclo C de $G - M$. Por construção, o corte $\partial(V(C))$ em G associado a C possui somente arestas de peso 2 e, a partir da Proposição 5, segue que $f(V(C)) = 0$. Logo temos o mesmo número de arestas de peso 2 entrando e saindo em $\partial(V(C))$. Toda aresta de peso 2 que incide em $V(C)$ mas não pertence a $\partial(V(C))$, tem exatamente 2 extremos em C . Como G é cúbico, o número total de vértices em C é dado pelo número de arestas em $\partial(V(C))$, que é par, mais duas vezes o número de arestas de peso 2 com ambos os extremos em C . Logo, C é um ciclo par. Esse argumento é válido

para todo ciclo de $G - M$, então, $G - M$ pode ser colorido com duas cores e G com três. A Figura 25 ilustra um 4-fluxo, um ciclo C e o corte $\partial(V(C))$ de um grafo cúbico.

Figura 25 – O ciclo C e o corte $\partial(V(C))$ de um grafo com 4-fluxo



Fonte: Produzido pelos autores

□

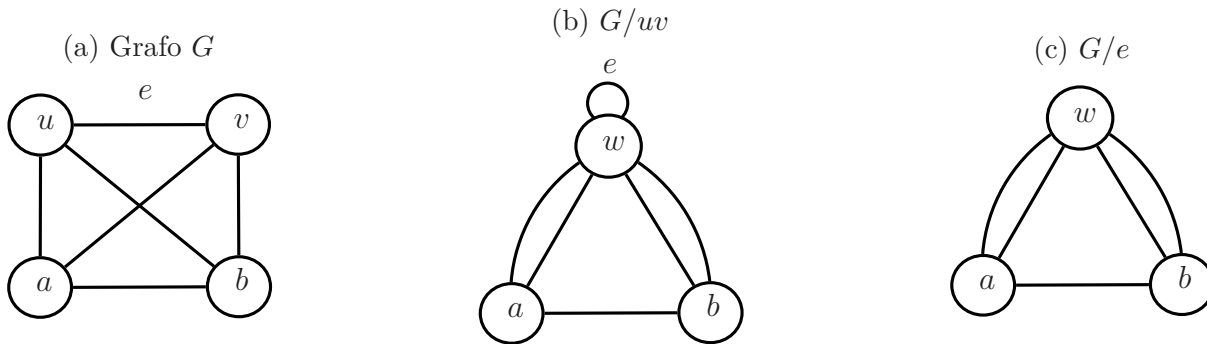
Vale lembrar que, conforme mostrado por Holyer (HOLYER, 1981), determinar se um grafo cúbico admite 3-coloração de arestas é um problema NP -Completo. Então, como consequência da equivalência de Tutte 4 e da observação de Holyer, determinar se um grafo cúbico admite 4-fluxo é um problema NP -Completo também.

É sabido que um eventual contra-exemplo mínimo para a Conjectura dos 5-Fluxos deve ser cúbico. Essa redução pode ser encontrada na dissertação de Mestrado de Leila Silva (SILVA, 1991, Redução 3.13). Pela Proposição 4, todo grafo que tem 4-fluxo também tem 5-fluxo, logo, pelo Teorema 4, um eventual contra-exemplo para a Conjectura dos 5-Fluxos não pode admitir uma 3-coloração de arestas. Desde 1988, é sabido que um contra-exemplo mínimo para a Conjectura dos 5-Fluxos deve ser 5-ciclicamente-aresta-conexo e ter cintura pelo menos sete (veja (JAEGER, 1988, Teorema 9.3)); portanto, necessariamente é um *snark*. Mais recentemente, Kochol mostrou que um contra-exemplo mínimo para a Conjectura dos 5-Fluxos deve ser 6-ciclicamente-aresta-conexo (KOCHOL, 2004) e ter cintura pelo menos onze (KOCHOL, 2010).

1.5 Grafos Fluxo-Críticos

Para uma melhor compreensão dos detalhes apresentados a seguir, definimos aqui duas operações de redução aplicadas a grafos, a *identificação* de dois vértices e a *contração* de uma aresta. Para *identificar* dois vértices u e v de um grafo G , adjacentes ou não, de um grafo G é necessário substituí-los por um único vértice w e, para toda aresta incidente em u ou v , torná-la incidente em w . Denotamos o grafo obtido a partir da operação de *identificação* dos vértices u e v de G por G/uv . Para *contrair* uma aresta $e = uv$ de G é necessário remover e e *identificar* os vértices u e v . Denotamos o grafo obtido a partir da operação de *contração* da aresta e de G por G/e . A Figura 26 ilustra tais operações.

Figura 26 – Operações de identificação de vértices e contração de arestas



Fonte: Produzido pelos autores

No artigo (SILVA; LUCCHESI, 2008), Silva e Lucchesi iniciaram o estudo de grafos k -fluxo-críticos; grafos que não admitem k -fluxo para um inteiro k especificado, mas passam a admitir quando uma operação de redução é aplicada ao grafo. Silva e Lucchesi consideram em seu estudo dois tipos de grafos fluxo-críticos: os k -aresta-fluxo-críticos (ou simplesmente k -aresta-críticos) e os k -vértice-fluxo-críticos (ou simplesmente k -vértice-críticos). Grafos k -aresta-críticos são aqueles que não admitem k -fluxo, mas que o grafo resultante após a contração de uma aresta qualquer admite um k -fluxo. Grafos k -vértice-críticos são aqueles que não admitem k -fluxo, mas o grafo resultante após a identificação de dois vértices quaisquer admite um k -fluxo. Todo grafo k -vértice-crítico sem laços (arestas cujos extremos coincidem) é também k -aresta-crítico, mas a recíproca não é verdadeira; um contra-exemplo pode ser encontrado na tese de doutorado de Silva (SILVA, 2009, Figura 2.1).

Como a contração de arestas não gera cortes novos e nem novos *minors*, contra-exemplos mínimos para qualquer uma das Conjecturas de Tutte é, necessariamente, um grafo k -aresta-crítico. Já a contração de conjuntos de vértices pode gerar novos *minors*, mas não novos cortes, então contra-exemplos mínimos para as Conjecturas dos 5- e 3-Fluxos são também k -vértice-críticos. Portanto, um conhecimento mais profundo da estrutura de tais grafos pode auxiliar na resolução destas conjecturas.

Existe mais de uma maneira de determinar se um grafo é k -vértice-crítico e/ou k -aresta-crítico. No mesmo artigo (SILVA; LUCCHESI, 2008, Theorem 3.2), Silva e Lucchesi apresentaram formas equivalentes de definir grafos k -fluxo-críticos. Antes de apresentarmos tais equivalências, primeiro precisamos definir o que é um (u, v) - k -fluxo. Seja G um grafo sem aresta de corte. Um (u, v) - k -fluxo modular, ou simplesmente (u, v) - k -fluxo, é um par de funções de orientação e peso (D, f) , com pesos das arestas de G no conjunto $\{1, 2, \dots, k-1\}$ e tal que todos os vértices de G , exceto u e v , têm fluxo líquido congruente a $0 \pmod k$.

As Proposições 9 e 10, encontrados em (SILVA; LUCCHESI, 2008), mostram definições equivalentes de grafos k -aresta-críticos e k -vértice-críticos, respectivamente.

Proposição 9 *Seja G um grafo conexo sem aresta de corte e sem k -fluxo, com pelo menos duas arestas. Então, as propriedades a seguir são equivalentes:*

- a) *Para toda aresta e de G , G/e admite k -fluxo modular;*
- b) *Para toda aresta e de G , $G - e$ admite k -fluxo modular;*
- c) *Para toda aresta $e = uv$ de G , G admite um (u, v) - k -fluxo;*

Demonstração:

(a) \Rightarrow (b): Seja $e = uv$ uma aresta de G e (D, f) um k -fluxo modular de G/e . Vamos mostrar que o par de funções (D', f') obtido a partir da extensão de (D, f) a G é um k -fluxo modular parcial com $\text{sup}(D', f') = E - e$. Note que como (D', f') é uma extensão de (D, f) , então os únicos vértices que podem não estar equilibrados módulo k são u e v . Assim, vamos mostrar que os vértices u e v ficam equilibrados módulo k em (D', f') . Suponha por absurdo que (D', f') não equilibre u e v . Então, pela Proposição 5, $f'(u) = -f'(v)$ em G . Sem perda de generalidade, suponha que u é o vértice negativo e v é o vértice positivo. Ao orientar a aresta e de u para v e atribuir peso $f(v) \pmod k$ a e , u e v tornam-se equilibrados módulo k , ou seja, um k -fluxo modular (D'', f'') de G é obtido, uma contradição à hipótese de que G não tem k -fluxo. Concluimos assim que a extensão (D', f') a $G - e$ é um k -fluxo modular. Logo, se para toda aresta e de G , G/e admitir um k -fluxo modular então $G - e$ também admitirá um k -fluxo modular para toda aresta e .

(b) \Rightarrow (c): Seja $e = uv$ uma aresta de G e (D, f) um k -fluxo modular de $G - e$. Da mesma maneira que argumentado anteriormente, a extensão de (D, f) a G é um k -fluxo modular parcial (D', f') com $\text{sup}(D', f') = E - e$.

Vamos mostrar que o k -fluxo modular parcial (D', f') de G com $\text{sup}(D', f') = E - e$ pode ser transformado em um (u, v) - k -fluxo de G . Ao atribuir uma orientação e um peso arbitrário a e (dentro do intervalo $\{1, 2, \dots, k-1\}$), obtemos um par de funções (D'', f'') que tornam u e v não equilibrados módulo k . Como todos os outros vértices de

G continuam equilibrados módulo k em (D'', f'') e $\text{sup}(D'', f'') = E$ então (D'', f'') é um (u, v) - k -fluxo de G . Logo, se para toda aresta e de G , $G - e$ admitir um k -fluxo modular então G admitirá um (u, v) - k -fluxo para toda aresta e .

(c) \Rightarrow (a): Seja $e = uv$ uma aresta de G e (D, f) um (u, v) - k -fluxo de G . Vamos mostrar que a restrição de (D, f) a G/e é um k -fluxo modular (D', f') de G/e . Pela Proposição 5, $f(u) = -f(v)$ em G . O fluxo líquido do vértice w obtido após a contração de e é igual ao fluxo líquido no corte $\partial(X)$ para $X = \{u, v\}$. Pela Proposição 5, $f'(w) = f(u) + f(v) \equiv 0 \pmod{k}$. Como todos os vértices de G/e são equilibrados módulo k por (D', f') então este é um k -fluxo modular de G/e . Logo, se para toda aresta e , o grafo G admitir um (u, v) - k -fluxo, G/e admitirá um k -fluxo modular para toda aresta e .

□

Proposição 10 *Seja G um grafo conexo sem aresta de corte e sem k -fluxo. Então, as propriedades a seguir são equivalentes:*

- a) *Para todo par de vértices (u, v) de G , G/uv admite k -fluxo modular;*
- b) *Para todo par de vértices (u, v) de G , $G + uv$ admite k -fluxo modular;*
- c) *Para todo par de vértices (u, v) de G , G admite um (u, v) - k -fluxo;*

Demonstração:

(a) \Rightarrow (b): Sejam u e v dois vértices de G e (D, f) um k -fluxo modular de G/uv . Vamos mostrar que o par de funções (D', f') obtido a partir da extensão de (D, f) a G é um (u, v) - k -fluxo. Note que como (D', f') é uma extensão de (D, f) , então os únicos vértices que podem ficar não equilibrados módulo k são u e v . Assim, vamos mostrar que os vértices u e v ficam não equilibrados módulo k em (D', f') . Suponha por absurdo que (D', f') equilibre u e v módulo k . Então (D', f') é um k -fluxo modular de G , uma contradição à hipótese de que G não tem k -fluxo. Concluimos assim que (D', f') é um (u, v) - k -fluxo de G .

Sem perda de generalidade, suponha que u é o vértice negativo e v é o vértice positivo. Ao adicionar uma nova aresta $e = uv$ a G , orientar e de u para v e atribuir peso $f(v) \pmod{k}$ a e , u e v tornam-se equilibrados módulo k , ou seja, um k -fluxo modular (D'', f'') de $G + uv$ é obtido.

Logo, se para todo par de vértices (u, v) de G , G/uv admitir um k -fluxo modular então $G + uv$ também admitirá um k -fluxo modular para todo par de vértices (u, v) .

(b) \Rightarrow (c): Sejam u e v dois vértices de G , (D, f) um k -fluxo modular de $G + uv$, e $e = uv$ a aresta adicionada a G para obter $G + uv$. Vale observar que ainda que já exista uma aresta ligando os vértices u e v em G , a nova aresta e é sempre adicionada.

Vamos provar que o par de funções (D', f') obtidos a partir da restrição de (D, f) a G é um (u, v) - k -fluxo. Note que como (D', f') é uma restrição de (D, f) , então os únicos vértices que podem ficar não equilibrados módulo k são u e v . Assim, vamos mostrar que os vértices u e v ficam não equilibrados módulo k em (D', f') . Sem perda de generalidade, suponha que e é orientada em D saindo de u e entrando em v . Claramente, a remoção de e faz com que $f'(u) = f(e)$ e $f'(v) = -f(e)$. Concluímos assim que (D', f') é um (u, v) - k -fluxo de G . Logo, se para todo par de vértices (u, v) de G , $G + uv$ admitir um k -fluxo modular então G admitirá um (u, v) - k -fluxo para todo par de vértices (u, v) .

(c) \Rightarrow (a): Seja (D, f) um (u, v) - k -fluxo de G . Pela Proposição 5, $f(u) = -f(v)$; logo, ao identificar tais vértices, o fluxo líquido do vértice resultante será congruente a $0 \pmod k$, ou seja, um k -fluxo modular é obtido. Se, para todo par de vértices (u, v) , G admitir um (u, v) - k -fluxo então G/uv admitirá um k -fluxo modular para todo par de vértices (u, v) .

□

1.5.1 Snarks 4-Fluxo-Críticos

Silva e Lucchesi (SILVA; LUCCHESI, 2007, Proposition 3.10) mostraram que todo grafo k -aresta-crítico possui um $(k + 1)$ -fluxo. A Proposição 11 apresenta uma demonstração desse fato. Segue desta proposição que todo *snark* 4-aresta-crítico (ou 4-vértice-crítico) possui 5-fluxo e não é contra-exemplo para a Conjectura dos 5-Fluxos de Tutte. Motivados por esse fato, Silva e Lucchesi investigaram e concluíram que muitos *snarks* famosos como o grafo de Petersen, os de Blanuša, de Loupekine e muitos outros são 4-aresta-críticos. Também descobriram que os grafos da família infinita dos *flower-snarks* (SILVA; LUCCHESI, 2012) também são 4-aresta-críticos.

Em 2013, Silva, Pesci e Lucchesi (SILVA; PESCI; LUCCHESI, 2013) avaliaram todos os *snarks* de ordem até 28 e observaram que nem todos *snarks* são 4-aresta-críticos. No entanto, também observaram que todo *snark* que não é 4-aresta-crítico possui um *snark* 4-aresta-crítico como menor. Logo, como consequência da Proposição 11, uma possível abordagem para tentar resolver a Conjectura dos 5-Fluxos de Tutte seria a extensão de um 5-fluxo de um *snark* 4-aresta-crítico G para um *snark* H que não é 4-aresta-crítico mas que possui G como menor.

Proposição 11 *Seja G um grafo conexo sem aresta de corte e $e = uv$ uma aresta de G tal que $G - e$ tem k -fluxo. Então G tem $(k + 1)$ -fluxo.*

Demonstração: Seja (D, f) um k -fluxo de $G - e$ e $R(u)$ o conjunto de todos os vértices alcançáveis a partir de u por caminhos orientados segundo D , e $T_u = R(u) \cup \{u\}$. Se houver

arestas saindo de T_u então haveria vértices alcançáveis a partir de u que não estariam em $R(u)$, logo não existem arestas assim. Se houver arestas entrando em T_u então $f(T_u) < 0$ algo que, como consequência da Proposição 5, não pode ocorrer em um k -fluxo. Como não existem arestas entrando ou saindo de T_u e $G - e$ é conexo então $T_u = V(G)$.

O vértice v pertence a T_u , logo existe um caminho orientado $\pi_{uv} = \{u, x_1, \dots, x_n, v\}$ em $G - e$. O par de funções (D', f') obtido a partir da extensão de (D, f) a G é um k -fluxo parcial de G tal que $f'(e) = 0$ e a aresta e é orientada de v a u em D' . Então $\text{sup}(D', f') = E - e$. O caminho π_{uv} juntamente com a aresta $e = vu$ formam um ciclo direcionado C em D' e um $(k + 1)$ -fluxo (D', f'') pode ser obtido a partir de (D', f') somando-se o valor um ao peso de toda aresta do ciclo C , isto é, através dos seguintes passos:

- para toda aresta $e' \in \pi_{uv}$, atribuir $f''(e') = f'(e') + 1$;
- atribuir $f''(e) = 1$.

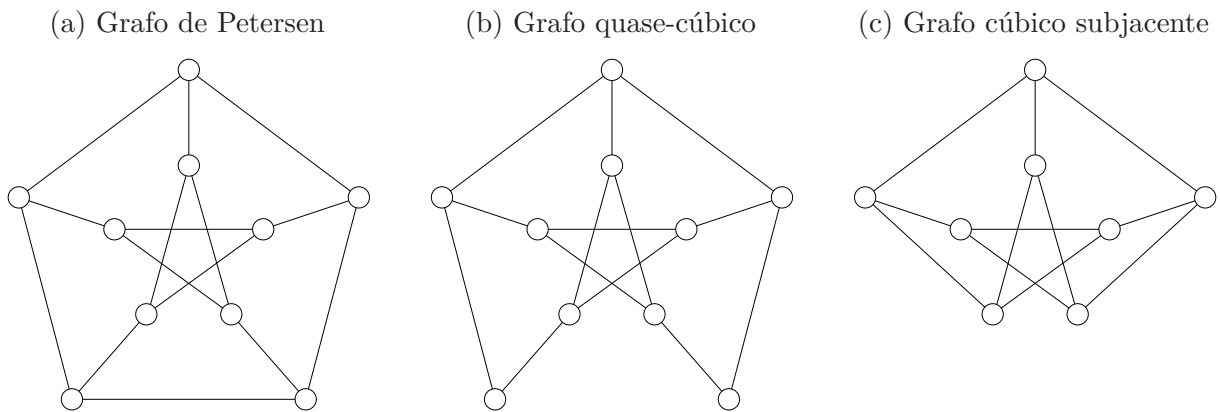
Claramente, o intervalo de pesos atribuídos às arestas por f'' será $\{1, 2, \dots, k\}$ uma vez que, da definição de k -fluxo, o intervalo de pesos atribuídos por f é $\{1, 2, \dots, k - 1\}$. Além disso, todos os vértices se mantêm equilibrados pois somamos o mesmo valor 1 ao peso de todas as arestas do ciclo direcionado C . Então, (D'', f'') é um $(k + 1)$ -fluxo. \square

Ainda em 2013, Brinkmann et al. (BRINKMANN et al., 2013b) conseguiram gerar todos os *snarks* de ordem até 36. Com o intuito de continuar o trabalho de Silva, Pesci e Lucchesi, identificamos quais *snarks* desta nova base são 4-aresta-críticos e quais são 4-vértice-críticos. Descrevemos no próximo capítulo o processo que nos permitiu realizar tal identificação. No entanto, primeiro é necessário apresentar definições e demonstrar proposições que são de vital importância para a explicação da parte prática deste projeto.

Um grafo é dito *quase-cúbico* quando possui dois vértices u e v de grau dois e o restante de grau três. Claramente, a remoção de uma aresta $e = uv$ de um grafo cúbico G gera um grafo quase-cúbico. O grafo cúbico obtido a partir da contração de duas arestas de $G - e$, uma incidente em u e outra incidente em v , é cúbico; denotamos tal grafo por $G * e$. Dizemos que $G * e$ é o *grafo cúbico subjacente* de $G - e$. A Figura 27 ilustra o grafo de Petersen, um grafo quase-cúbico obtido a partir do grafo de Petersen e o grafo cúbico subjacente obtido a partir do quase-cúbico.

Se o grafo quase-cúbico $G - e$ admite um 4-fluxo (D, f) então o vértice u possui uma aresta entrando e outra saindo, ambas com o mesmo peso; o mesmo vale para o vértice v . Claramente, o par de funções (D', f') obtidos a partir da restrição de (D, f) a $G * e$ também é um 4-fluxo. Dessas definições, observações e da Proposição 9 segue a Proposição 12.

Figura 27 – Grafos de Petersen, quase-cúbico e cúbico subjacente



Fonte: Produzido pelos autores

Proposição 12 *Seja G um snark e $e = uv$ uma aresta de G . Então G admite um (u, v) -4-fluxo se e somente se $G * e$ admite um 4-fluxo.*

Uma (u, v) -3-coloração de arestas de um grafo quase-cúbico tal que u e v têm grau 2 é similar a uma 3-coloração de arestas de um grafo cúbico com a diferença que cada par de arestas incidentes em cada um dos vértices de grau dois possuem cores iguais. A partir desta definição, demonstramos as Proposições 13 e 14.

Proposição 13 *Seja G um snark e $e = uv$ uma aresta de G . Então $G - e$ admite uma (u, v) -3-coloração de arestas se e somente se $G * e$ admite uma 3-coloração de arestas.*

Demonstração: (\Rightarrow) Seja C uma (u, v) -3-coloração de arestas de $G - e$. Pela definição, as arestas incidentes em u possuem a mesma cor; o mesmo é válido para v . Claramente, as arestas incidentes no vértice obtido a partir da contração de uma das arestas de u têm cores distintas; o mesmo vale para o vértice obtido a partir da contração de uma das arestas de v . Logo, $G * e$ obtido a partir de tais contrações admite 3-coloração de arestas.

(\Leftarrow) Seja C uma 3-coloração de arestas de $G * e$, x_1 e x_2 os vértices adjacentes a u , y_1 e y_2 os vértices adjacentes a v , e $e_1 = ux_1$ e $e_2 = vy_1$ as arestas contraídas de $G - e$ para obter $G * e$. A extensão das cores de C a $G - e$ atribui cores a todas arestas de $G - e$ exceto e_1 e e_2 . Se as cores das arestas $e_3 = ux_2$ e $e_4 = vy_2$ de $G - e$ forem atribuídas a e_1 e e_2 , respectivamente, uma (u, v) -3-coloração de $G - e$ é obtida. Concluímos assim que se $G * e$ admite uma 3-coloração de arestas então $G - e$ admite uma (u, v) -3-coloração. \square

Proposição 14 *Seja G um snark e $e = uv$ uma aresta de G . Então G admite um (u, v) -4-fluxo se e somente se $G - e$ admite uma (u, v) -3-coloração.*

Demonstração: (\Rightarrow) Se G admite um (u, v) -4-fluxo então, pela Proposição 12, o $G * e$ admite 4-fluxo. Logo, pela equivalência de Tutte (Teorema 4), $G * e$ admite uma 3-coloração de arestas. Então, pela Proposição 13, $G - e$ admite uma (u, v) -3-coloração de arestas. Concluímos assim que, se um snark G que possui uma aresta $e = uv$, admite um (u, v) -4-fluxo, então $G - e$ admite uma (u, v) -3-coloração.

(\Leftarrow) Como todas as proposições utilizadas acima são equivalências, o mesmo raciocínio pode ser feito no sentido contrário. \square

O Corolário 2 é consequência das Proposições 14 e 9.

Corolário 2 *Seja G um snark. Então G é 4-aresta-crítico se e somente se $G - e$ admite uma (u, v) -3-coloração para toda aresta e de G .*

2 Algoritmos e Resultados

Como um primeiro passo na direção de tentar resolver a Conjectura dos 5-Fluxos (Conjectura 1), Silva, Pesci e Lucchesi (SILVA; PESCI; LUCCHESI, 2013) identificaram os *snarks* 4-vértice-críticos de ordem até 28, que era, até 2012, o maior conjunto de *snarks* disponível publicamente.

Em 2013, Brinkmann et al. (BRINKMANN et al., 2013b) conseguiram gerar um conjunto de todos os *snarks* de ordem até 36, uma tarefa computacional massiva, dado o fato que o número de *snarks* cresce exponencialmente com a ordem. Esse conjunto de *snarks* pode ser encontrado em (BRINKMANN et al., 2013a). Nesta dissertação, apresentamos as quantidades de *snarks* 4-aresta-críticos e 4-vértice-críticos nesse conjunto maior, assim como uma descrição do processo de desenvolvimento do algoritmo que nos permitiu processar esse conjunto de *snarks* apesar do seu tamanho consideravelmente maior que o anterior.

Conforme mencionado no capítulo anterior, determinar se um grafo cúbico possui 4-fluxo é um problema NP-Completo. As Proposições 9 e 10, que mostram maneiras equivalentes de determinar se um grafo é 4-fluxo-crítico, sugerem um algoritmo natural que envolve a resolução de um número polinomial de problemas NP-Completo, ou seja, um algoritmo de complexidade exponencial. Ainda não é claro se há ou não uma caracterização dos *snarks* 4-fluxo-críticos que permita seu reconhecimento em tempo polinomial, logo os todos algoritmos descritos a seguir são de complexidade exponencial.

Primeiramente, descrevemos o programa criado por Silva (SILVA, 2008) que testa se um grafo que não admite 4-fluxo é 4-vértice-crítico. O algoritmo implementado por Silva está baseado na caracterização apresentada na Proposição 10. Pela Proposição 7, é possível fixar uma orientação para as arestas do grafo e buscar, dentre todas as permutações de funções peso válidas, uma que deixe precisamente dois vértices u e v desbalanceados, ou seja, com fluxo líquido não nulo módulo 4. Se para todo par de vértices u e v do grafo for encontrado um (u, v) -4-fluxo, então tal grafo é 4-vértice-crítico (pela Proposição 10(a)). É precisamente essa computação que é feita no programa de Silva, com a diferença de que, para simplificar a implementação, ao invés de buscar uma função para cada par de vértices, o programa gera recursivamente todas as permutações de pesos e, para aquelas que deixam precisamente dois vértices u e v desbalanceados, registra ter achado um (u, v) -4-fluxo para esse par de vértices. Algumas podas simples são feitas para conter o crescimento desnecessário da árvore de busca, como por exemplo parar a recursão de uma dada permutação caso mais de dois vértices já estejam desbalanceados considerando apenas os pesos já fixados; ou se existem precisamente dois vértices desbalanceados até o

momento, mas esse é um par para o qual já foi encontrado um (u, v) -4-fluxo.

As podas utilizadas no algoritmo de Silva foram boas o suficiente para lidar com o conjunto de *snarks* de ordem até 28. No entanto, uma estimativa inicial do tempo necessário para processar o novo conjunto de *snarks* de até 36 vértices utilizando o mesmo algoritmo, mesmo se 72 instâncias diferentes fossem processadas em paralelo em um *cluster*, seria de aproximadamente 6 meses.

O programa implementado por Silva foi originalmente utilizado para testar se certas construções de grafos sem 4-fluxo, quando aplicadas a grafos 4-vértice-críticos, geram um novo grafo 4-vértice-crítico. Como essas construções não garantem preservação de regularidade do grafo, o programa não assume que o grafo inicial seja cúbico. Para o caso particular de grafos cúbicos, é possível fazer mais podas da árvore de busca e, a partir desta observação, decidimos investigar se uma nova implementação, específica para *snarks* (que são cúbicos) levaria a uma melhora de desempenho na verificação. Felizmente, obtivemos um resultado muito positivo. A seguir apresentamos uma descrição mais detalhada das propriedades observadas que permitiram a melhora no desempenho, bem como das variantes de versões experimentadas.

O nosso algoritmo é dividido em duas versões, ambas são algoritmos de *backtracking*. A primeira, chamada de *Versão de Fluxo*, procura por um (u, v) -4-fluxo para todos pares (u, v) e a outra, chamada de *Versão de Coloração*, procura por uma (u, v) -3-coloração das arestas do grafo para todo par (u, v) de vértices adjacentes. Pela Proposição 14 determinar se um grafo cúbico possui uma (u, v) -3-coloração de arestas é equivalente a determinar se ele possui um (u, v) -4-fluxo.

2.1 Versão Força Bruta (ou VFB)

Antes de discutir as versões desenvolvidas e utilizadas neste trabalho, explicaremos um simples algoritmo, que chamamos de *Versão Força Bruta* (ou *VFB*), sem qualquer estratégia de poda que gera todas as funções peso possíveis para um grafo. Fazemos isso para melhor ilustrar a base das outras versões que é a recursividade, assim como salientar a importância das podas comparando o tempo de execução deste algoritmo com os tempos dos outros.

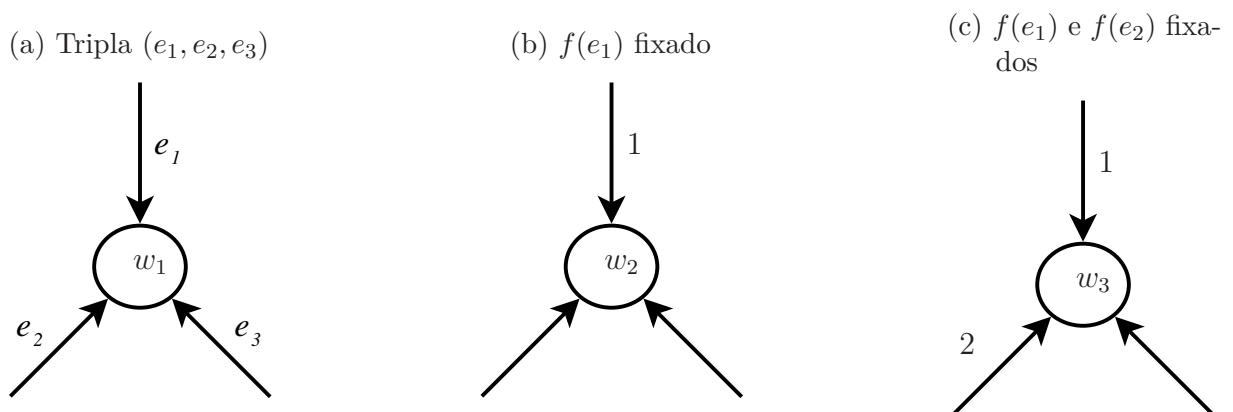
O objetivo desta versão é gerar todas as funções peso $f : E \rightarrow \{1, 2, 3\}$ possíveis buscando identificar durante a geração aquelas que representam (u, v) -4-fluxos. São 3^m funções a serem geradas no total. Existe mais de uma maneira para gerar esse conjunto de funções. Uma maneira natural é usar um algoritmo recursivo (evidentemente com *backtracking*) para fixar o peso de uma única aresta a cada nível da recursão. É fácil ver que dessa forma a altura da árvore de busca será m e o grau de cada nó será 3. É exatamente isso que é feito no algoritmo de Silva (SILVA, 2008). Já na versão apresentada

nesta dissertação, os pesos das arestas são fixados em blocos de forma a ter todos os pesos de arestas incidentes em um determinado vértice fixados antes da próxima chamada recursiva, ou seja, a recursão ocorre nos vértices e não nas arestas. Desta forma, a altura será menor, n . Por outro lado, os graus dos nós da árvore de busca serão potencialmente maiores. O grau exato do nó da árvore de busca depende do número de arestas incidentes no vértice que ainda não tiveram seus pesos fixados, o qual pode ser 0, 1, 2 ou 3. Como existem 3 possíveis valores de peso, então os possíveis graus dos nós da árvore de busca podem ser 27, 9, 3 ou 1, respectivamente.

No que tange a geração de funções peso, ambas abordagens são equivalentes. Optamos pela segunda maneira para facilitar a implementação e viabilizar o uso de certas observações teóricas como mecanismos de poda da árvore de busca. Utilizamos essa abordagem em todas as versões descritas neste capítulo. Uma vez feita a escolha dessa abordagem, sempre teremos ao longo da execução do algoritmo dois tipos de vértices, os quais chamaremos de vértices *rígidos* e vértices *não rígid*os. Um vértice w é rígido quando já houve uma chamada recursiva em um nível anterior tendo w como parâmetro. Evidentemente, antes do início da execução do algoritmo, todos os vértices são não rígid

os. A cada nível da recursão, escolhemos um vértice w não rígido para se tornar rígido. Seja (e_1, e_2, e_3) a tripla ordenada que representa a lista de adjacência de w . Seja (f_1, f_2, f_3) a tripla ordenada que representa uma combinação de pesos que podem ser fixados à (e_1, e_2, e_3) . Uma combinação (f_1, f_2, f_3) é *compatível* com w se, para cada aresta e_i incidente em w cujo peso foi fixado em um nível anterior da recursão, o peso f_i é precisamente o valor já fixado. Para facilitar a implementação, criamos uma rotina $C_p(w)$ que retorna a lista de combinações compatíveis com w . A Figura 28a ilustra a lista de

Figura 28 – Ordem e fixação dos pesos das arestas incidentes em um vértice w



Fonte: Produzido pelos autores

adjacências de um vértice w_1 . A Figura 28b ilustra quando a aresta e_1 possui um peso

fixado e, para esse caso,

$$C_p(w_2) = \{(1, 1, 1), (1, 1, 2), (1, 1, 3), (1, 2, 1), (1, 2, 2), (1, 2, 3), (1, 3, 1), (1, 3, 2), (1, 3, 3)\}.$$

A Figura 28c exemplifica uma situação quando e_1 e e_2 possuem pesos fixados, nesse caso

$$C_p(w_3) = \{(1, 2, 1), (1, 2, 2), (1, 2, 3)\}.$$

Note que não necessariamente uma combinação compatível deixa um vértice equilibrado módulo 4.

O Algoritmo 1 é um pseudocódigo da nossa implementação. Os parâmetros de entrada são um grafo $G = (V, E)$ com orientação arbitrária D , um vértice não rígido w_i , uma função peso f que tem como domínio apenas as arestas incidentes nos vértices rígidos e uma lista l de pares de vértices (u, v) para os quais um (u, v) -4-fluxo foi encontrado. A primeira chamada para esta função é feita com w_1 como parâmetro, escolhido arbitrariamente, nenhum peso definido em f e $l = \emptyset$.

Dados: (D, w_i, f, l)

para cada combinação compatível c de $C_p(w_i)$ **faça**

 // Após a execução da instrução abaixo, w_i passa a ser rígido
 fixar em f os pesos das arestas incidentes em w_i determinados pela combinação c ;

 // Essa condição é verdadeira no último nível da recursão, ou seja, quando um nó folha da árvore de busca é atingido

se todos os vértices de V são rígidos **então**

se f possui apenas dois vértices quaisquer u e v desbalanceados **então**

 | adicionar o par (u, v) a l ;

fim

senão

 | chamar recursivamente esse algoritmo para $(D, w_{(i+1)}, f, l)$;

fim

 // *backtrack*; w_i deixa de ser rígido

 remover de f os pesos das arestas incidentes em w_i que foram fixadas pela combinação c ;

fim

Algoritmo 1: Versão Força-Bruta

Após a execução deste algoritmo para um *snark* G , basta verificar se um (u, v) -4-fluxo foi encontrado para todo par de vértices (u, v) para determinar se G é 4-vértice-crítico, e apenas os pares adjacentes para determinar se é 4-aresta-crítico.

A Tabela 1 mostra o tempo de execução desta versão para cada conjunto de *snarks* agrupados por suas ordens. Claramente, esta versão é muito ineficiente. Os tempos listados em todas as tabelas apresentadas neste capítulo foram obtidos a partir de execuções realizadas em um *notebook* Dell com 8GB de memória RAM, processador Intel i7 e sistema operacional Windows 8.1.

Tabela 1 – Tempos da Versão Força Bruta

Ordem	Snarks	VFB [s]
10	1	2,55
18	2	*

Fonte: Produzido pelos autores.

Nota: * interrompido por ser superior a 24 horas (86.400 segundos).

2.2 Versão de Fluxo - Variante 1 (ou VF1)

Para mostrar a melhoria no desempenho que cada uma das estratégias utilizadas na Versão de Fluxo causou, nós dividimos esta versão em 3 variantes. Todas as variantes têm como base a Versão Força-Bruta, ou seja, são algoritmos de *backtracking* que procuram por um (u, v) -4-fluxo para todo par de vértices (u, v) .

A primeira variante possui uma abordagem bastante similar à de Silva. As mesmas podas são utilizadas, isto é, a recursão é interrompida caso:

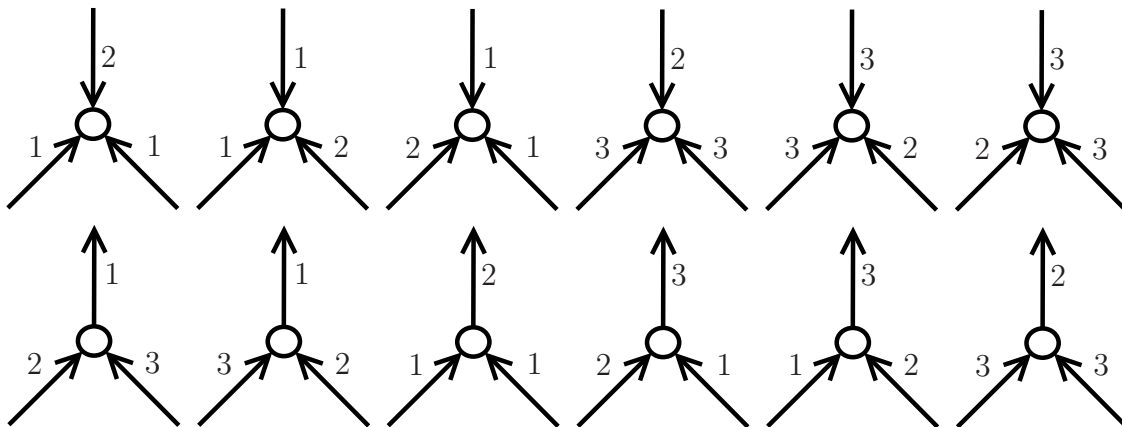
- mais de dois vértices já estejam desbalanceados,
- dois vértices u e v estejam desbalanceados, mas o par (u, v) já pertence à lista l , ou
- a lista l já permita determinar se o grafo é 4-vértice-crítico ou 4-aresta-crítico, dependendo do que se pretende determinar.

A primeira diferença entre esta variante e o algoritmo de Silva é que há uma etapa de pré-processamento que determina uma ordem $S = (w_1, w_2, \dots, w_n)$ que define a sequência dos vértices que se tornarão rígidos ao longo da recursão. O ideal é que a ordem S seja tal que o vértice $w_{(i+1)}$ seja um dos vértices w_j , $j > i$, com maior número de vizinhos no trecho (w_1, w_2, \dots, w_i) . Se essa propriedade é satisfeita para todo i então temos que a garantia que $C_p(w_i)$ tem o menor tamanho possível a cada instante. Assim garantimos um menor grau da árvore de busca a cada nível. Se existe mais de uma ordem S que satisfaça essa propriedade, a escolha entre elas é arbitrária. Na implementação, para determinar a ordem S simplesmente construímos um *heap* de máximo H com os vértices ainda não adicionados a S ordenados pela quantidade de vizinhos já em S . Claramente um algoritmo de complexidade polinomial.

A segunda e principal diferença dessa variante é o fato de assumirmos que o grafo é *snark*, portanto cúbico. Para um grafo cúbico não existem muitas maneiras que podemos atribuir pesos às arestas incidentes em um vértice w de modo que ele fique equilibrado módulo 4. Primeiro, observe que uma orientação D deve direcionar todas as arestas incidentes em w na mesma direção ou duas na mesma direção e a terceira na direção oposta.

Para cada um desses casos, existem seis combinações de pesos que deixam w equilibrado módulo 4. Esses casos são ilustrados na Figura 29. Note que, se as orientações das arestas fossem opostas às apresentadas, o equilíbrio módulo 4 seria mantido nos vértices com as mesmas combinações de pesos nas arestas.

Figura 29 – Combinações de pesos que equilibram módulo 4 um vértice de grau 3



Fonte: Produzido pelos autores

Observando a Figura 29, concluímos que, se uma das arestas incidentes no vértice já possuir um peso atribuído pelo algoritmo, então existem exatamente duas combinações compatíveis que deixam este vértice equilibrado módulo 4. Caso duas arestas já possuam pesos atribuídos então existe no máximo uma combinação.

Se em um determinado nível da recursão, tivermos precisamente dois vértices u e v já rígidos e desbalanceados, então não faz mais sentido considerar combinações que permitam novos vértices rígidos e desbalanceados. Nesse caso, para todo vértice não rígido w definimos $C'_p(w)$ como sendo o subconjunto das combinações de $C_p(w)$ que deixam w equilibrado módulo 4. Conforme mencionado anteriormente, a quantidade de combinações de $C'_p(w)$ será no máximo dois. No caso da Figura 28b, temos que:

$$C'_p(w_2) = \{(1, 1, 2), (1, 2, 1)\}.$$

A ordem S , em conjunto com a observação do número mais restrito de combinações compatíveis que deixam um vértice equilibrado módulo 4, tendem a reduzir bastante o grau dos nós da árvore de busca. Estas podas só ocorrem a partir do momento que já existem dois vértices rígidos e desbalanceados. Antes disso, quase todos os vértices possuem até 9 combinações compatíveis; um valor relativamente alto quando comparado a uma ou duas combinações que deixam um vértice equilibrado módulo 4.

A Tabela 2 é uma extensão da Tabela 1 de tempo de execução da *Versão Força Bruta* e, como podemos observar, as podas descritas acima tiveram um grande impacto

no tempo de execução. Esta variante permitiu determinar quais *snarks* de ordem até 30 são 4-vértice-críticos, ou seja, reproduziu e estendeu os resultados obtidos por Silva, Pesci e Lucchesi (SILVA; PESCI; LUCCHESI, 2013).

Tabela 2 – Tempos da Versão Força Bruta e da Versão de Fluxo - Variante 1

Ordem	Snarks	VFB [s]	VF1 [s]
10	1	2,55	0,00
18	2	*	0,00
20	6		0,06
22	20		0,31
24	38		1,23
26	280		16,51
28	2.900		138,11
30	28.399		3033,61
32	293.059		*

Fonte: Produzido pelos autores.

Nota: * interrompido por ser superior a 24 horas (86.400 segundos).

2.3 Versão de Coloração (ou VC)

Pela Proposição 14, procurar por um (u, v) -4-fluxo para um par (u, v) de vértices adjacentes é equivalente a procurar por uma (u, v) -3-coloração para o mesmo par. Baseados neste fato, criamos uma nova versão de algoritmo que procura por uma (u, v) -3-coloração para todo par (u, v) de vértices adjacentes. Claramente, esta versão é específica para determinar se um dado *snark* G é 4-aresta-crítico mas não permite determinar se G é 4-vértice-crítico. Apesar desta limitação, note que esta versão tem uma grande vantagem natural, o número de pares de vértices adjacentes em grafo cúbico é linear na quantidade de vértices, ou seja, a quantidade de problemas exponenciais a serem resolvidos é menor.

Em analogia às combinações de pesos que definimos na Versão Força-Bruta, definimos nesta versão o conceito de *combinação de cores* como uma tripla ordenada de cores $\{c_1, c_2, c_3\}$ que podem ser atribuídas à tripla ordenada $\{e_1, e_2, e_3\}$, a lista de adjacências de um vértice w . Dizemos que uma combinação de cores é *compatível* com w se esta não altera as cores previamente determinadas das arestas incidentes em w . Uma combinação de cores compatível com w é *própria* se ela atribui cores distintas às arestas incidentes em w . Para esta versão, definimos como $C_c(w)$ a lista de combinações de cores próprias de w . Também, de maneira similar à descrita anteriormente, esta lista está limitada a 0,

1, 2 ou 6 combinações dependendo de quantas arestas incidentes em w já possuem cores atribuídas e de quais são essas cores. O número de combinações de cores próprias de um vértice w é 6 quando nenhuma das arestas incidentes em w foram coloridas. Caso uma aresta tenha sido colorida então a quantidade de combinações é sempre 2. Apenas quando w já possui duas arestas coloridas então a quantidade de combinações pode ser 0 ou 1, sendo 1 se as cores forem distintas e 0 caso contrário.

Em uma (u, v) -3-coloração, com exceção de u e v , todos os vértices têm uma aresta de cada cor. Devido a isso, nos pareceu infrutífera a ideia de atribuir cores às arestas de um grafo sem o cuidado de manter as arestas incidentes na vasta maioria dos vértices com cores distintas.

Isto nos fez mudar uma característica fundamental da rotina recursiva, a qual nesta versão, ao invés de procurar em uma mesma execução por uma (u, v) -3-coloração para todo par (u, v) de vértices adjacentes, ela procura por apenas uma (u, v) -3-coloração para um par (u, v) determinado previamente. Apesar desta alteração nos obrigar a executar a rotina recursiva uma vez para cada par de vértices adjacentes do grafo, ela apresenta a vantagem de, a cada nível da recursão, avaliar apenas combinações de cores próprias para cada vértice distinto de u e v . Isso reduz os graus de todos os nós da árvore de recursão diferentes da raiz a no máximo dois.

Uma (u, v) -3-coloração pode ser transformada em outra apenas permutando os nomes das cores. Uma consequência para esse fato é a redução do grau da raiz da árvore de busca a 1. Se uma (u, v) -3-coloração A foi obtida para uma combinação de cores própria do primeiro vértice w_1 de S então, para obter uma nova (u, v) -3-coloração para qualquer outra combinação própria de w_1 , basta permutar as cores de A .

A alteração da rotina recursiva também permitiu mudar a maneira como S é construída, antes de uma chamada à rotina recursiva para um dado par (u, v) de vértices adjacentes, uma nova ordem de vértices S é criada com todos os vértices menos u e v . Uma vez que a recursão ocorre nos vértices, a existência de uma ordem S menor, claramente reduz a altura da árvore de busca quando comparada à *Versão de Fluxo - Variante 1*. Por outro lado, ao atingir um nó folha da árvore de busca, não temos garantia que a coloração C obtida é uma (u, v) -3-coloração. Então ao final é necessário adicionar uma verificação que avalia se a coloração obtida é uma (u, v) -3-coloração ou não. Sabemos que, ao executar tal verificação, todos os vértices menos u e v são rígidos, logo possuem combinações de cores próprias. Para que C seja uma (u, v) -3-coloração, basta verificar se o par de arestas incidentes em u que foram coloridas compartilham a mesma cor; o mesmo vale para v . Se, para cada um desses pares, essa restrição for atendida então, pela definição, C é uma (u, v) -3-coloração.

Finalmente, foi possível adicionar um critério de poda à rotina recursiva. Quando procuramos por uma (u, v) -3-coloração atribuímos uma combinação de cores c a um vértice

w_i de S a cada nível da rotina recursiva. Se algum vizinho não rígido x de w_i , com $x \neq u$ e $x \neq v$, tiver $C_c(x) = \emptyset$ então c é descartada e a próxima combinação é avaliada. Claramente, se uma combinação de cores c é descartada então menos chamadas recursivas são realizadas, ou seja, o grau do nó da árvore de busca é diminuído.

Esta versão foi rápida o suficiente para determinar quais *snarks* de ordem até 36 são 4-arestas-críticos em menos de um dia; um tempo razoalmente menor que o previsto inicialmente. O tempo de execução desta versão pode ser visto na Tabela 3. Note que a propriedade sendo verificada é diferente, isto é, esta versão verifica se os *snarks* são 4-arestas-críticos enquanto que as versões anteriores verificam se são 4-vértice-críticos, portanto não cabe comparação com os dados das versões anteriores.

Tabela 3 – Tempos da Versão de Coloração

Ordem	Snarks	VC [s]
10	1	0,00
18	2	0,00
20	6	0,00
22	20	0,00
24	38	0,03
26	280	0,28
28	2.900	0,59
30	28.399	11,11
32	293.059	147,56
34	3.833.587	2.850,23
36	60.167.732	62.827,38

Fonte: Produzido pelos autores.

2.4 Versão de Fluxo - Variante 2 (ou VF2)

Intrigados com a melhora apresentada pela versão anterior, decidimos criar uma nova variante da Versão de Fluxo com uma abordagem similar à Versão de Coloração. Para cada *snark* G , a quantidade de pares de vértices a serem testados para determinar se G é 4-vértice-crítico é quadrática na quantidade de vértices. Já a quantidade de arestas, ou pares de vértices adjacentes, a serem verificadas é linear na quantidade de vértices. Claramente, devido à distinção dos problemas, a Versão de Coloração resolve uma quantidade de problemas exponenciais menor, no entanto, algumas características da implementação da

Versão de Coloração puderam ser aproveitadas para essa variante; todas as modificações descritas a seguir são semelhantes, senão idênticas, a essas características.

A primeira mudança em relação à primeira variante desta versão é a fixação do par de vértices (u, v) para o qual um (u, v) -4-fluxo é procurado. Essa alteração permitiu também uma mudança na criação da ordem S em que os vértices são processados. Como o par de vértices que ficarão desbalanceados é conhecido, eles não são adicionados a S ; logo, quando a recursão torna rígido um vértice w de S , apenas as combinações de $C'_p(w)$ são consideradas. Essa alteração permitiu reduzir, à exceção da raiz, o grau de todos os nós da árvore de busca a no máximo dois.

Conforme descrito na primeira variante, o primeiro vértice w de S é escolhido arbitrariamente e será o único com as três arestas sem pesos atribuídos antes de se tornar rígido. Note que, ao multiplicar os pesos das arestas de um (u, v) -4-fluxo por 3 módulo 4, nós obtemos um outro (u, v) -4-fluxo no qual as arestas de peso 3 se tornam arestas de peso 1 e vice-versa, enquanto que as arestas de peso 2 não se alteram. Portanto, somente as três combinações mais à esquerda de cada linha ilustradas na Figura 29 devem ser consideradas para deixar w equilibrado módulo 4. Observe também que, as três combinações às quais nos referimos dependem das orientações das arestas incidentes em w .

A última modificação feita é a verificação dos vizinhos do vértice w sendo processado em qualquer nível da recursão. Ao procurar por um (u, v) -4-fluxo, se uma combinação c de $C'_p(w)$ impossibilita o equilíbrio módulo 4 de um vizinho x de w , com $x \neq u$ e $x \neq v$, então c é descartada e a próxima combinação é avaliada. Claramente, se uma combinação c é descartada então menos chamadas recursivas são realizadas, ou seja, o grau do nó da árvore de busca é diminuído.

Quando todos os vértices de S se tornam rígidos, é necessário verificar se o par de funções (D, f) obtido é um (u, v) -4-fluxo. O algoritmo garante que todos os vértices menos u e v estejam equilibrados módulo 4 logo é necessário apenas verificar se os vértices u e v estão desbalanceados módulo 4. Se estiverem, (D, f) é um (u, v) -4-fluxo do grafo. Caso contrário, um 4-fluxo modular foi encontrado para o grafo, uma contradição para as execuções realizadas uma vez que apenas *snarks* são avaliados.

Essas modificações permitiram avaliar rapidamente quais *snarks*, dentre aqueles 4-aresta-críticos identificados usando a *Versão de Coloração*, são 4-vértice-críticos. Os tempos de execução desta variante para cada ordem estão descritos na Tabela 4.

Somente para efeito de comparação com as outras versões, uma outra execução desta variante foi feita com todos os *snarks* de ordem até 36. Os tempos dessa segunda execução estão detalhados na Tabela 5; uma extensão da Tabela 3. Como é possível avaliar, essas modificações surtiram efeito na velocidade de execução, o que permitiu determinar quais *snarks* de ordem até 34 são 4-vértice-críticos sem necessariamente antes

Tabela 4 – Tempos da Versão de Fluxo - Variante 2

Ordem	Snarks 4-aresta-críticos	VF2 [s]
10	1	0,00
18	2	0,01
20	1	0,01
22	2	0,01
24	0	0,00
26	111	1,72
28	33	0,60
30	115	3,21
32	29	0,72
34	40.330	1.445,48
36	14.548	455,60

Fonte: Produzido pelos autores.

identificar quais são 4-aresta-críticos. Apesar de sabermos a diferença entre os problemas, comparamos esta versão com a *Versão de Coloração* para mostrar a proximidade dos tempos e salientar o quão importante foi reestruturar esta versão baseada nas observações realizadas na *Versão de Coloração*.

2.5 Versão de Fluxo - Variante 3 (ou VF3)

Para determinar se um grafo é k -vértice-crítico basta encontrar um (u, v) - k -fluxo para todo par de vértices (u, v) , logo se for possível encontrar um (u, w) - k -fluxo a partir de um (u, v) - k -fluxo, então a quantidade de problemas exponenciais a serem resolvidos é diminuída. Chamamos de (v, w) - k -transformação a obtenção de um (u, w) - k -fluxo a partir de um (u, v) - k -fluxo, com $w \neq v$. Quando não nomeamos os vértices para essa operação, a chamamos apenas de *transformação*. Mostramos a seguir uma condição que, quando satisfeita, permite realizar uma (v, w) - k -transformação.

Seja (D, f) um (u, v) - k -fluxo de um grafo G , onde u e v não são necessariamente adjacentes. Seja w um vértice de G distinto de u e v e adjacente a v . Sejam $e = vw$ e $G' = G - e$. Sejam D' e f' as restrições de D e f a G' . Quando $f'(v) \not\equiv 0 \pmod{k}$ então é possível obter um (u, w) - k -fluxo em G . Considere a expansão h de f' a G , onde $h(e)$ é

Tabela 5 – Tempos das versões *VC* e *VF2* com todos *snarks* de ordem até 36

Ordem	<i>Snarks</i>	<i>VC</i> [s]	<i>VF2</i> [s]
10	1	0,00	0,00
18	2	0,00	0,00
20	6	0,00	0,01
22	20	0,00	0,03
24	38	0,03	0,03
26	280	0,28	1,53
28	2.900	0,59	1,94
30	28.399	11,11	22,57
32	293.059	147,56	346,93
34	3.833.587	2.850,23	8.375,22
36	60.167.732	62.827,38	*

Fonte: Produzido pelos autores.

Nota: * interrompido por ser superior a 24 horas (86.400 segundos).

definido da seguinte forma:

$$h(e) = \begin{cases} -f'(v) \pmod{k}, & \text{se } e \text{ estiver saindo de } v; \\ f'(v) \pmod{k}, & \text{se } e \text{ estiver entrando em } v. \end{cases} \quad (2.1)$$

Então o par (D, h) é um (u, w) - k -fluxo pois a definição de h força o balanceamento módulo k de v e, conseqüentemente, o desbalanceamento módulo k de w .

Ao realizar uma (v, w) - k -transformação, talvez existam outras transformações que possam ser realizadas no (u, w) - k -fluxo obtido.

Se uma transformação for realizada, quando possível, para cada um dos vizinhos de u e v assim como para todos os novos pares encontrados, então vários (x, y) - k -fluxos para pares distintos são encontrados, o que diminui a quantidade de problemas exponenciais a serem resolvidos.

Nesta variante, ao realizar essas diversas transformações, o algoritmo acaba encontrando rapidamente os pares de vértices com as maiores chances de não possuírem (u, v) -4-fluxo.

A aplicação desse processo de transformação mostrou um ganho na eficiência do algoritmo principalmente quando o conjunto de *snarks* testados eram, em sua maioria, 4-fluxo-críticos, pois, uma vez encontrado um (u, v) -4-fluxo para um par de vértices de um grafo crítico, eram encontrados (u, v) -4-fluxos para quase todos os outros pares. Como

pode ser visto na Tabela 6, essa abordagem apresentou uma melhora significativa no tempo de identificação de *snarks* 4-vértice-críticos de ordem até 36, já que o conjunto testado eram os *snarks* 4-aresta-críticos que, em sua maioria, também são 4-vértice-críticos.

Tabela 6 – Tempos das versões VF2 e VF3

Ordem	<i>Snarks</i> 4-aresta-críticos	VF2 [s]	VF3 [s]
10	1	0,00	0,00
18	2	0,01	0,00
20	1	0,01	0,00
22	2	0,01	0,00
24	0	0,00	0,00
26	111	1,72	0,39
28	33	0,60	0,16
30	115	3,21	0,50
32	29	0,72	0,25
34	40.330	1.445,48	72,71
36	14.548	455,60	35,41

Fonte: Produzido pelos autores.

Da mesma maneira que na seção anterior, essa variante foi executada utilizando todos os *snarks* de ordem até 36. Os tempos dessa execução estão descritos na Tabela 7 que é uma extensão da Tabela 5.

Apesar de ajudar, essa abordagem não mostrou ganhos quando aplicada a grafos maiores como o de Blanche Descartes, uma vez que ainda é necessário encontrar um (u, v) -4-fluxo para pelo menos um par de vértices; processo de complexidade exponencial.

2.6 Snarks Grandes

Conforme mencionado na terceira variante da *Versão de Fluxo*, apesar de acelerarem o processo de determinação de quais *snarks* de ordem até 36 são 4-fluxo-críticos, as podas descritas tiveram pouco ou quase nenhum efeito quando aplicadas a *snarks* maiores como o de Blanche Descartes, um *snark* de 210 vértices, ilustrado na Figura 30.

Com o intuito de tentar melhorar a execução do algoritmo para *snarks* maiores, algumas tentativas de mudanças no algoritmo foram implementadas. Descrevemos aqui a que apresentou o melhor desempenho.

Conforme descrito anteriormente, na Variante 1 da *Versão de Fluxo*, antes de

Tabela 7 – Tempos das versões VC, VF2 e VF3 com todos *snarks* de ordem até 36

Ordem	<i>Snarks</i>	VC [s]	VF2 [s]	VF3 [s]
10	1	0,00	0,00	0,00
18	2	0,00	0,00	0,00
20	6	0,00	0,01	0,01
22	20	0,00	0,03	0,02
24	38	0,03	0,03	0,06
26	280	0,28	1,53	0,47
28	2.900	0,59	1,94	2,24
30	28.399	11,11	22,57	26,55
32	293.059	147,56	346,93	363,6
34	3.833.587	2.850,23	8.375,22	7.419,34
36	60.167.732	62.827,38	*	*

Fonte: Produzido pelos autores.

Nota: * interrompido por ser superior a 24 horas (86.400 segundos).

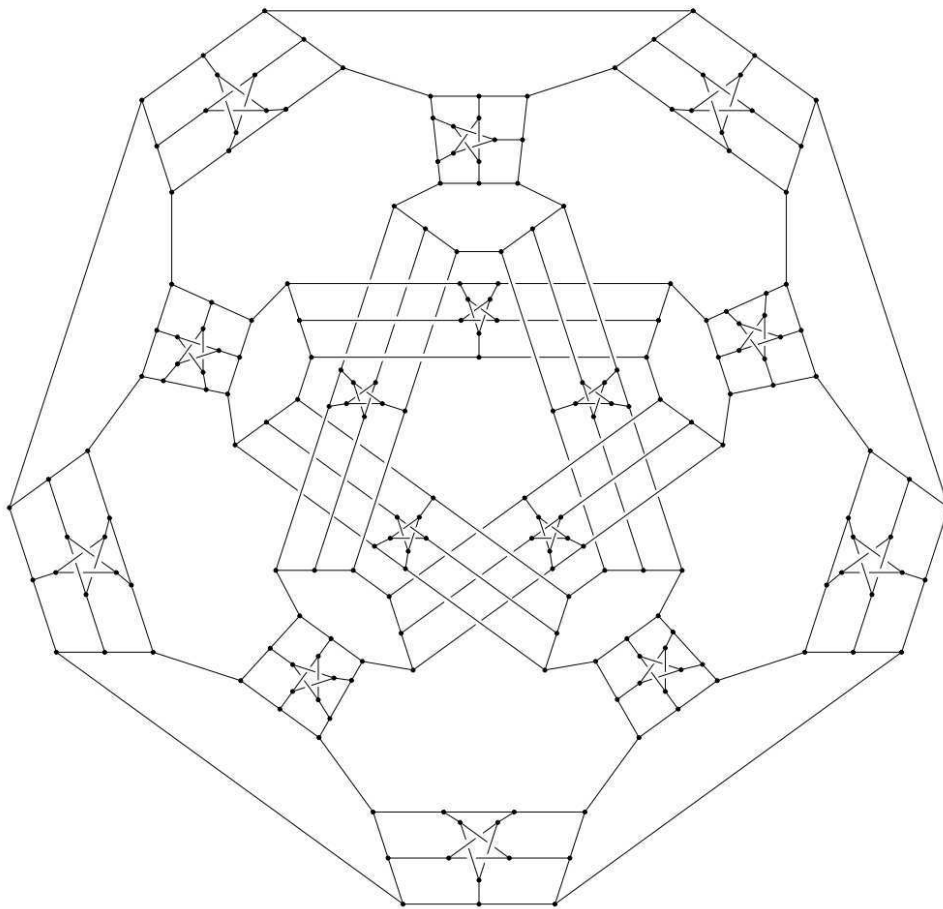
procurar por um (u, v) -4-fluxo, uma ordem S é estabelecida. Caso haja mais de uma maneira de criar a ordem S na qual os vértices do *snark* G se tornarão rígidos, uma é escolhida arbitrariamente. Para lidar com *snarks* maiores, a construção de S foi alterada. Seja G um *snark*. Dentre as possíveis maneiras de criar S de forma a obedecer a restrição descrita anteriormente, é escolhida uma que obedeça as seguintes regras de construção:

- adicionar a S um ciclo $C = \{w_1, w_2, \dots, w_a\}$ onde a é a cintura de G ;
- adicionar a S um caminho $P = \{w_{a+1}, w_{a+2}, \dots, w_{a+b}\}$ de $G - S$ onde:
 - o tamanho de P seja o menor possível;
 - w_{a+1} e w_{a+b} possuam pelo menos um vizinho, cada, em S .
- repetir o passo anterior até que todos os vértices de $G - \{u, v\}$ pertençam a S .

Essa alteração na criação de S permitiu descobrir (u, v) -4-fluxo para alguns pares de vértices do *snark* de Blanche Descartes. Em conjunto com a operação de transformação descrita na *Versão de Fluxo - Variante 3*, essa alteração nos permitiu encontrar (u, v) -4-fluxos para 21.585 pares de vértices dos 21.945 existentes do *snark* de Blanche Descartes. Infelizmente, para os pares restantes, o algoritmo não é rápido o suficiente. Anterior a essa alteração, a *Versão de Fluxo - Variante 3* não conseguia encontrar um (u, v) -4-fluxo para nenhum par de vértices (u, v) de tal *snark* em um dia ou menos. Nela, é possível observar

10 eneágonos ligados por estruturas que parecem estrelas. Em cada eneágono existem 9 vértices que podem ser pareados de 36 maneiras distintas. Como existem 10 eneágonos, é possível obter 360 pares distintos de vértices. São exatamente para esses 360 pares de vértices que o algoritmo não conseguiu um (u, v) -4-fluxo. Infelizmente, como alguns destes pares são adjacentes, não foi possível nem determinar se tal grafo é 4-aresta-crítico.

Figura 30 – *Snark* de Blanche Descartes



Fonte: Wikipedia

Para ilustrar a melhora que tal alteração surtiu quando aplicada a *snarks* maiores, a Tabela 8 compara as execuções da *Versão de Fluxo - Variante 3* com e sem tal alteração da construção de S . Para essas execuções, os *snarks* utilizados foram os da família de *flower-snarks* que, conforme descrito no Capítulo 1 são 4-vértice-críticos.

Vale ressaltar que tal alteração, apesar de bastante eficiente quando aplicada a *snarks* grandes, não mostrou um desempenho satisfatório quando aplicada ao conjunto de *snarks* de ordem até 36. A construção de S , por ser mais complexa que a descrita na *Versão de Fluxo - Variante 3*, acaba demorando mais do que a determinação se um *snark*

Tabela 8 – Tempos das versões *VF3* original e *VF3* alterada

<i>Flower-snark</i>	<i>VF3</i> original [s]	<i>VF3</i> alterado [s]
J_5	0	0
J_7	0	0,01
J_9	0,01	0,05
J_{11}	0,19	0,06
J_{13}	2,42	0,08
J_{15}	37,99	0,11
J_{17}	617,31	0,22
J_{19}	10.559,74	0,25
J_{21}	*	0,84
J_{23}	*	2,35
J_{25}	*	6,14
J_{27}	*	26,40

Fonte: Produzido pelos autores.

Nota: * interrompido por ser superior a 24 horas (86.400 segundos).

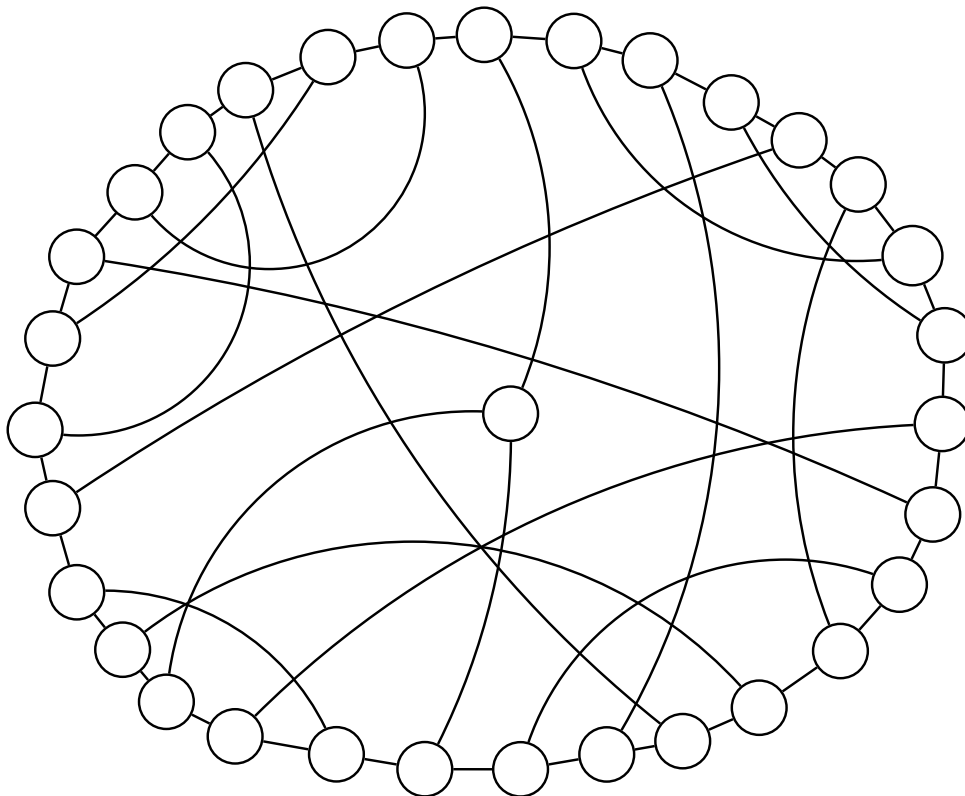
de ordem pequena é 4-vértice-crítico.

2.7 Resultados

Todas as variantes de todas as versões foram executados e seus resultados, as quantidades de *snarks* 4-vértice-críticos e 4-aresta-críticos identificados, são apresentados na Tabela 9. Inicialmente, não sabíamos se todo *snark* 4-aresta-crítico seria necessariamente também 4-vértice-crítico. Em 2013, Silva e Lucchesi (SILVA; PESCI; LUCCHESI, 2013) apresentaram resultados nos quais todos os *snarks* de ordem até 28 que são 4-aresta-críticos também são 4-vértice-críticos, o que indicava que isso poderia ser verdade. No entanto, Silva (SILVA, 2009) já havia apresentado um grafo 3-aresta-crítico porém não 3-vértice-crítico. Esses dois fatos que indicavam respostas contraditórias não nos ajudavam a responder essa questão. Nossa dúvida foi respondida negativamente após a execução dos algoritmos descritos anteriormente. Conforme é possível observar na Tabela 9, existem *snarks* 4-aresta-críticos porém não 4-vértice-críticos. Os menores *snarks* com tal característica têm ordem 32. A Figura 31 ilustra um desses grafos.

Este trabalho foi desenvolvido em conjunto com o professor Brendan McKay do Research School of Computer Science da Australian National University. O professor Brendan McKay tem vasta experiência na implementação de geradores de grafos com

Figura 31 – Grafo 4-aresta-crítico porém não 4-vértice-crítico



Fonte: Produzido pelos autores

propriedades específicas e é autor de um famoso programa chamado *nauty* utilizado para computar grupos de automorfismos de grafos orientados e não orientados. Na fase inicial deste trabalho, tivemos a oportunidade de discutir com ele maneiras de otimizar o nosso algoritmo. Ele, de forma independente, implementou sua própria versão de um algoritmo similar aos descritos anteriormente e chegou aos mesmos resultados que os apresentados na Tabela 9.

As listas de *snarks* 4-aresta-críticos e de *snarks* 4-vértice-críticos foram disponibilizadas no site *House of Graphs* (BRINKMANN et al., 2013a). O site *House of Graphs* tem por objetivo manter disponível uma base de grafos interessantes. Nele, é possível pesquisar grafos por nomes e/ou características. Para encontrar as listas de *snarks* 4-aresta-críticos e de *snarks* 4-vértice-críticos basta realizar uma busca utilizando as palavras-chave “*critical snark*”, “*edge-critical snark*” ou “*vertex-critical snark*”, dependendo do que se deseja encontrar.

Todos os grafos disponibilizados nesse site estão no formato *g6*, um formato utilizado para armazenar grafos de uma maneira compacta. O conjunto de *snarks* de ordem até 36 utilizado neste trabalho foi obtido neste site, logo todos os *snarks* estavam no formato *g6*. Para a execução das versões apresentadas anteriormente, o formato esperado não era

Tabela 9 – Quantidades de *snarks* 4-fluxo-críticos para cada ordem

Ordem	<i>Snarks</i>	<i>Snarks</i> 4-aresta-críticos	<i>Snarks</i> 4-vértice-críticos
10	1	1	1
18	2	2	2
20	6	1	1
22	20	2	2
24	38	0	0
26	280	111	111
28	2.900	33	33
30	28.399	115	115
32	293.059	29	13
34	3.833.587	40.330	40.328
36	60.167.732	14.548	13.720

Fonte: Produzido pelos autores.

o *g6* então uma etapa de conversão do formato *g6* para o formato esperado foi necessária. Existe um outro programa chamado *showg*, também criado pelo professor Brendan McKay, capaz de converter grafos representados no formato *g6* para diversos outros formatos. O formato esperado pelas versões do algoritmo descritas anteriormente não está na lista de formatos para o qual o programa consegue converter. O formato esperado pelo algoritmo descrito nas seções anteriores é o seguinte:

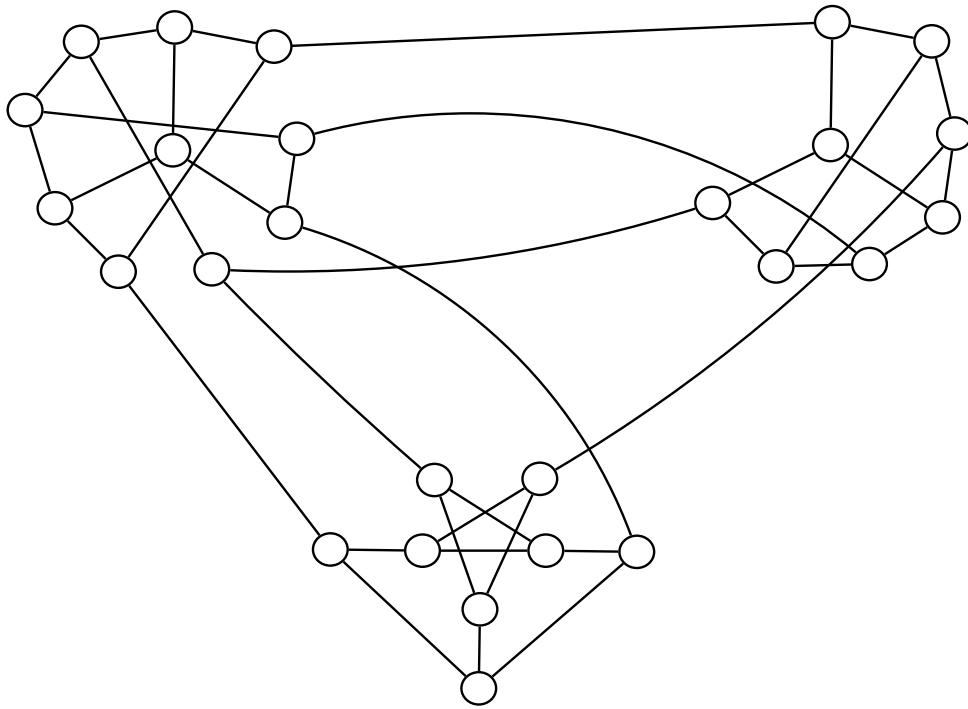
- a primeira linha deve conter dois valores inteiros N e M que representam a quantidade de vértices e arestas, respectivamente;
- a próximas M linhas devem conter dois valores inteiros a e b indicando que existe uma aresta $e = ab$.

Os vértices são representados por valores entre 0 e $N - 1$. Para atender a essa formatação, o programa *showg* foi alterado e a conversão do formato *g6* para o formato esperado foi realizada.

Uma outra contribuição deste trabalho foi a descoberta que a operação de produto interno não preserva a propriedade de 4-fluxo criticalidade. Conforme descrito no Capítulo 1, Silva e Lucchesi (SILVA; LUCCHESI, 2012) mostraram que todos os grafos da família *flower-snarks* são 4-aresta-críticos. No entanto, não se sabia se o produto interno de quaisquer dois *snarks* 4-aresta-críticos produziria um novo *snark* 4-aresta-crítico. A Figura 32 ilustra a construção, utilizando a operação de produto interno, de um *snark*

não 4-aresta-crítico a partir de dois 4-aresta-críticos (grafo de Petersen e um dos *snarks* de Blanuša).

Figura 32 – Exemplo de produto interno que gera um grafo não 4-fluxo-crítico



Fonte: Produzido pelos autores

Conclusão

Um eventual contra-exemplo para a Conjectura dos 5-Fluxos deve ser um *snark*. Silva e Lucchesi introduziram o conceito de grafos k -fluxo-críticos e mostraram que todo grafo que apresenta tal propriedade admite $(k + 1)$ -fluxo, logo todo *snark* 4-fluxo-crítico não é um contra-exemplo para a Conjectura dos 5-Fluxos. Silva e Lucchesi também mostraram que todo *snark* possui como minor um *snark* 4-fluxo-crítico. Essa observação permitiu uma nova maneira de abordar a Conjectura dos 5-Fluxos pois, se for possível mostrar como obter um 5-fluxo para *snarks* não 4-vértice-críticos a partir do 5-fluxo de seus respectivos minors então a conjectura estaria provada. Com esse intuito, Silva e Lucchesi identificaram quais *snarks* de até 28 vértices são 4-fluxo-críticos.

Em 2013, Brinkmann et al. (BRINKMANN et al., 2013b), conseguiram gerar todos os *snarks* de ordem até 36. Se o mesmo algoritmo, utilizado anteriormente por Silva e Lucchesi, fosse utilizado para determinar quais *snarks* dessa nova base são 4-fluxo-críticos, o tempo estimado seria de aproximadamente seis meses, um tempo considerado muito alto.

A proposta deste projeto era criar um algoritmo mais eficiente, mesmo que utilizando computação paralela, para conseguir processar a nova base de *snarks* em um tempo consideravelmente menor. O resultado, além de outras contribuições, foi positivo. Foram desenvolvidos dois algoritmos, um para determinar se um *snark* é 4-aresta-crítico e outro para determinar se um *snark* é 4-vértice crítico. Ambas versões apresentaram um desempenho razoavelmente melhor para realizar tais tarefas sem a necessidade de paralelismo; levaram pouco menos de um dia. No entanto, observamos que para *snarks* maiores, com ordem na casa das centenas, esses algoritmos ainda não apresentavam um desempenho satisfatório, não chegando a resposta alguma para *snarks* como o de Blanche Descartes. Algum estudo foi feito nesse sentido e conseguimos uma melhora perceptível.

Trabalhos Futuros

A partir dessa nova base de *snarks* 4-fluxo-críticos, talvez seja possível estudar suas estruturas e identificar quais são os minors 4-fluxo-críticos dos outros *snarks* não 4-fluxo-críticos. Com esse tipo de informação talvez seja possível determinar uma maneira de obter um 5-fluxo para um *snark* não 4-fluxo-crítico a partir de seu minor 4-fluxo-crítico.

Também, uma outra linha de pesquisa seria um aprofundamento no estudo das estruturas de *snarks* maiores para melhorar os algoritmos aqui apresentados. Uma possibilidade também seria estudar uma maneira de utilizar computação paralela na busca por um (u, v) -4-fluxo ou uma (u, v) -3-coloração para um determinado par de vértices (u, v) .

Vale lembrar que ainda não sabemos se há ou não uma caracterização dos *snarks* 4-fluxo-críticos que nos permita reconhecê-los em tempo polinomial. É inclusive uma interessante questão de pesquisa; se o problema de determinar se um *snark* é 4-fluxo-crítico é NP-Completo ou não.

Publicações

Conforme descrito no capítulo anterior, o professor Brendan McKay esteve envolvido durante o desenvolvimento deste trabalho. Esse trabalho conjunto, além de nos possibilitar a validação dos resultados, resultou na publicação do artigo intitulado *A Faster Test for 4-Flow-Criticality in Snarks* no VIII Latin-American Algorithms, Graphs and Optimization Symposium (LAGOS 2015) (CARNEIRO; SILVA; MCKAY, 2015).

Referências

- BLANUŠA, D. Problem cetiriju boja. *Glasnik Mat. Fiz. Astr. Ser. II*, v. 1, p. 31–42, 1946. Citado na página 22.
- BONDY, J. A.; MURTY, U. S. R. *Graph Theory*. [S.l.]: Springer, 2008. Citado 6 vezes nas páginas 21, 23, 24, 27, 28 e 32.
- BRINKMANN, G. et al. House of graphs: A database of interesting graphs. *Discrete Applied Mathematics*, Elsevier, v. 161, n. 1, p. 311–314, 2013. Available at <<http://hog.grinvin.org>>. Citado 3 vezes nas páginas 25, 55 e 71.
- BRINKMANN, G. et al. Generation and properties of snarks. *Journal of Combinatorial Theory, Series B*, Elsevier, v. 103, n. 4, p. 468–488, 2013. Citado 4 vezes nas páginas 25, 52, 55 e 75.
- BRINKMANN, G.; STEFFEN, E. Snarks and reducibility. *Ars Combinatoria*, CHARLES BABBAGE RES CTR PO BOX 272 ST NORBERT POSTAL STATION, WINNIPEG, MB R3T 2N2, CANADA, v. 50, p. 292–296, 1998. Citado na página 25.
- CARNEIRO, A. B.; SILVA, C. N. da; MCKAY, B. A faster test for 4-flow-criticality in snarks. *Electronic Notes in Discrete Mathematics*, v. 50, p. 193 – 198, 2015. ISSN 1571-0653. LAGOS'15 – {VIII} Latin-American Algorithms, Graphs and Optimization Symposium. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1571065315001882>>. Citado 2 vezes nas páginas 25 e 76.
- CELMINS, U. A.; SWART, E. R. *The Constructions of Snarks*. [S.l.], 1979. Citado na página 33.
- CHETWYND, A. G.; WILSON, R. J. Snarks and Supersnarks. In: ALAVI, Y. et al. (Ed.). *The Theory and Applications of Graphs*. [S.l.]: Wiley, 1981. p. 215–241. Citado na página 81.
- DESCARTES, B. Network-colourings. *The Mathematical Gazette*, v. 32, p. 67–69, 1948. Citado na página 22.
- DIESTEL, R. *Graph Theory*. [S.l.]: Springer-Verlag, 1996. (Graduate Texts in Mathematics, v. 173). Citado na página 24.
- GARDNER, M. Mathematical games: Snarks, boojums and other conjectures related to the four-color-map theorem. *Scientific American*, v. 234, p. 126–130, 1976. Citado na página 22.
- GOLDBERG, M. K. Construction of class 2 graphs with maximum vertex degree 3. *Combin. Theory Ser. B*, v. 31, p. 282–291, 1981. Citado na página 33.
- GRÖTZSCH, H. Ein Dreifarbensatz für dreikreisfreie Netze auf der Kugel. *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe*, v. 8, p. 109–120, 1958. Citado na página 24.

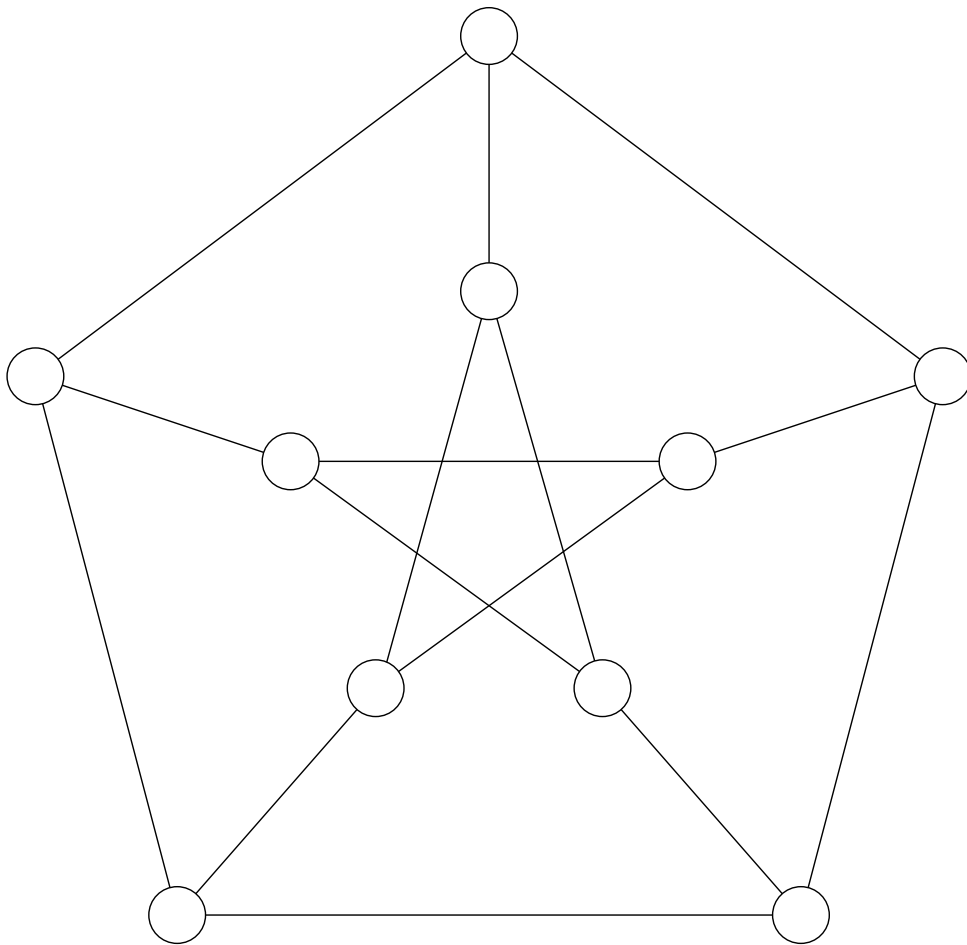
- HEAWOOD, P. J. Map-Color Theorem. *Quarterly Journal of Pure and Applied Mathematics*, v. 2, p. 193–200, 1890. Citado na página 24.
- HOLYER, I. The NP-completeness of the edge-coloring. *SIAM J. Comput.*, v. 10, p. 718–720, 1981. Citado 2 vezes nas páginas 33 e 47.
- ISAACS, R. Infinite families of non-trivial trivalent graphs which are not tait colorable. *Amer. Math. Monthly*, v. 82, p. 221–239, 1975. Citado 3 vezes nas páginas 22, 23 e 32.
- JAEGGER, F. Nowhere-zero flows problems. In: BEINEKE, L. W.; WILSON, R. J. (Ed.). *Selected Topics in Graph Theory*. [S.l.]: Academic Press, 1988. v. 3, p. 71–95. Citado na página 47.
- KOCHOL, M. Reduction of the 5-flow conjecture to cyclically 6-edge-connected snarks. *J. Combin. Theory Ser. B*, v. 90, p. 139–145, 2004. Citado na página 47.
- KOCHOL, M. Smallest counterexample to the 5-flow conjecture has girth at least eleven. *J. Combin. Theory Ser. B*, v. 100, p. 381–389, 2010. Citado na página 47.
- MCKAY, B. D. A note on the history of the four-colour conjecture. *Journal of Graph Theory*, v. 72, p. 361–363, 2013. Citado na página 21.
- PETERSEN, J. Sur le théoreme de tait. *L'intermédiaire des Mathématiciens*, v. 5, p. 225–227, 1898. Citado na página 22.
- ROBERTSON, N.; SEYMOUR, P. D.; THOMAS, R. Tutte's edge colouring conjecture. *J. Combin. Theory Ser. B*, v. 70, p. 166–183, 1997. Citado na página 34.
- SAATY, T. L. Thirteen Colorful Variations on Guthrie's Four-Color Conjecture. *American Math. Monthly*, v. 79, p. 2–43, 1972. Citado na página 22.
- SEYMOUR, P. D. Nowhere-zero flows. In: GRAHAM, R. L.; GRÖTSCHEL, L. L. M. (Ed.). *Handbook of Combinatorics*. [S.l.]: Elsevier, 1995. cap. 4, p. 289–299. Citado na página 24.
- SILVA, C. N. da. *A tester for 4-flow critical graphs*. 2008. Available at <<http://www.dcomp.sor.ufscar.br/candida/publications>>. Citado 2 vezes nas páginas 55 e 56.
- SILVA, C. N. da. *Fluxos Inteiros e Colorações*. Tese (Doutorado) — Instituto de Computação – UNICAMP, 2009. Em Português. Citado 2 vezes nas páginas 48 e 70.
- SILVA, C. N. da; LUCCHESI, C. L. *Flow-Critical Graphs*. [S.l.], 2007. Citado na página 51.
- SILVA, C. N. da; LUCCHESI, C. L. Flow-critical graphs. *Electronic Notes in Discrete Mathematics*, v. 30, p. 165–170, 2008. Citado 3 vezes nas páginas 24, 48 e 49.
- SILVA, C. N. da; LUCCHESI, C. L. Flower-snarks are flow-critical. Available at <<http://www.dcomp.sor.ufscar.br/candida/publications>>. 2012. Citado 3 vezes nas páginas 25, 51 e 72.
- SILVA, C. N. da; PESCI, L.; LUCCHESI, C. L. Snarks and flow-critical graphs. *Electronic Notes in Discrete Mathematics*, v. 44, p. 299–305, 2013. Citado 5 vezes nas páginas 25, 51, 55, 61 e 70.

- SILVA, L. M. de Almeida e. *Fluxos Inteiros em Grafos*. Dissertação (Mestrado) — Departamento de Ciência da Computação – UNICAMP, 1991. Em Português. Citado na página 47.
- TAIT, P. G. Remarks on the colouring of maps. In: *Proc. Roy. Soc. Edinburgh*. [S.l.: s.n.], 1880. v. 10, n. 729, p. 501–503. Citado na página 22.
- TUTTE, W. T. On hamiltonian circuits. *Journal of the London Mathematical Society*, Oxford University Press, v. 1, n. 2, p. 98–101, 1946. Citado na página 22.
- TUTTE, W. T. A contribution to the theory of chromatic polynomials. *Can. J. Math.*, v. 6, p. 80–91, 1954. Citado 2 vezes nas páginas 23 e 24.
- TUTTE, W. T. On the algebraic theory of graph colorings. *J. Combin. Theory*, v. 1, p. 15–50, 1966. Citado 2 vezes nas páginas 24 e 34.
- VIZING, V. G. On an estimate of the chromatic class of a p-graph. *Diskret. Analiz*, v. 3, n. 7, p. 25–30, 1964. Citado na página 33.
- WATKINS, J. J. Snarks. *Ann. New York Acad. Sci.*, v. 576, p. 606–622, 1989. Citado na página 33.
- WEST, D. B. *Introduction to Graph Theory*. [S.l.]: Prentice Hall, 1996. Citado na página 33.
- ZHANG, C.-Q. *Integer Flows and Cycle Covers of Graphs*. [S.l.]: Marcel Dekker, 1997. Citado na página 24.

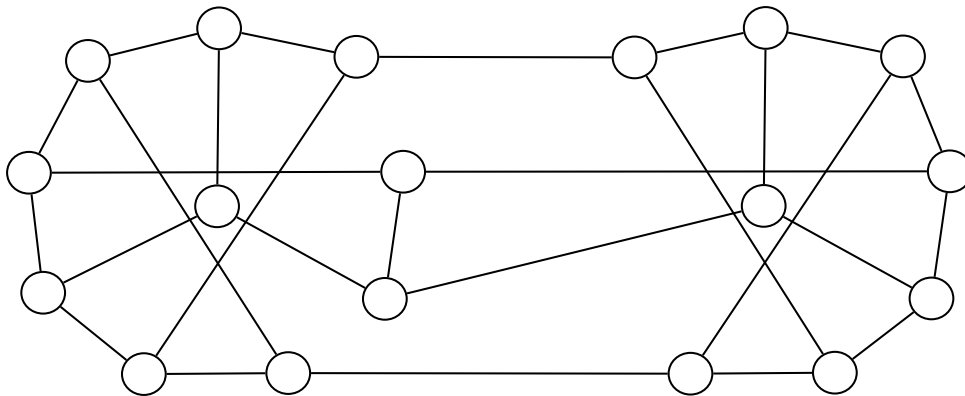
APÊNDICE A – *Snarks*

A seguir, ilustramos alguns *snarks* mencionados no Capítulo 1. Outras ilustrações destes grafos podem ser encontradas em ([CHETWYND; WILSON, 1981](#)).

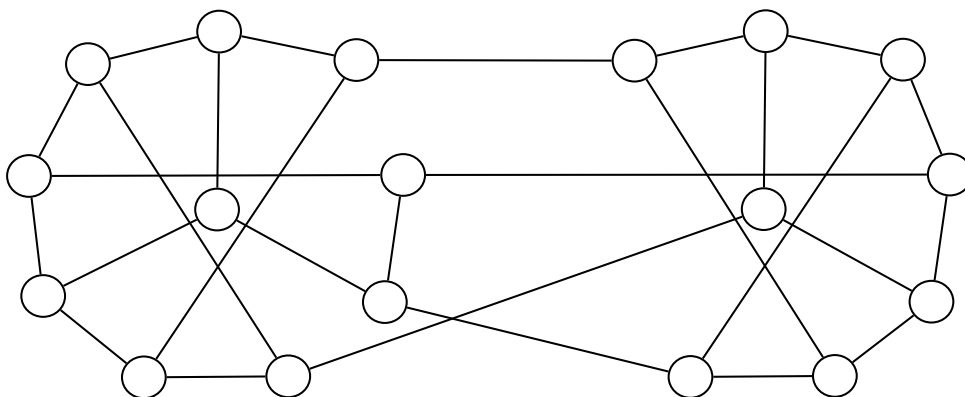
Figura 33 – Grafo de Petersen - 1898



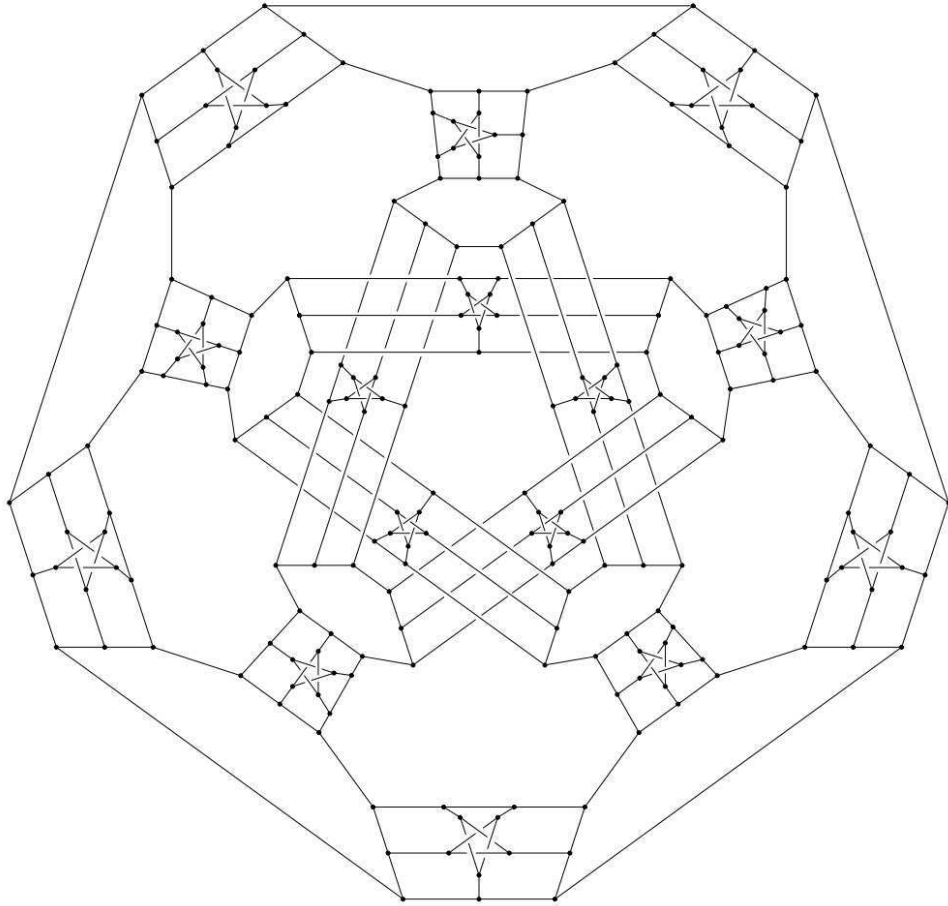
Fonte: Produzido pelos autores

Figura 34 – Blanuša *snark* 1 - 1946

Fonte: Produzido pelos autores

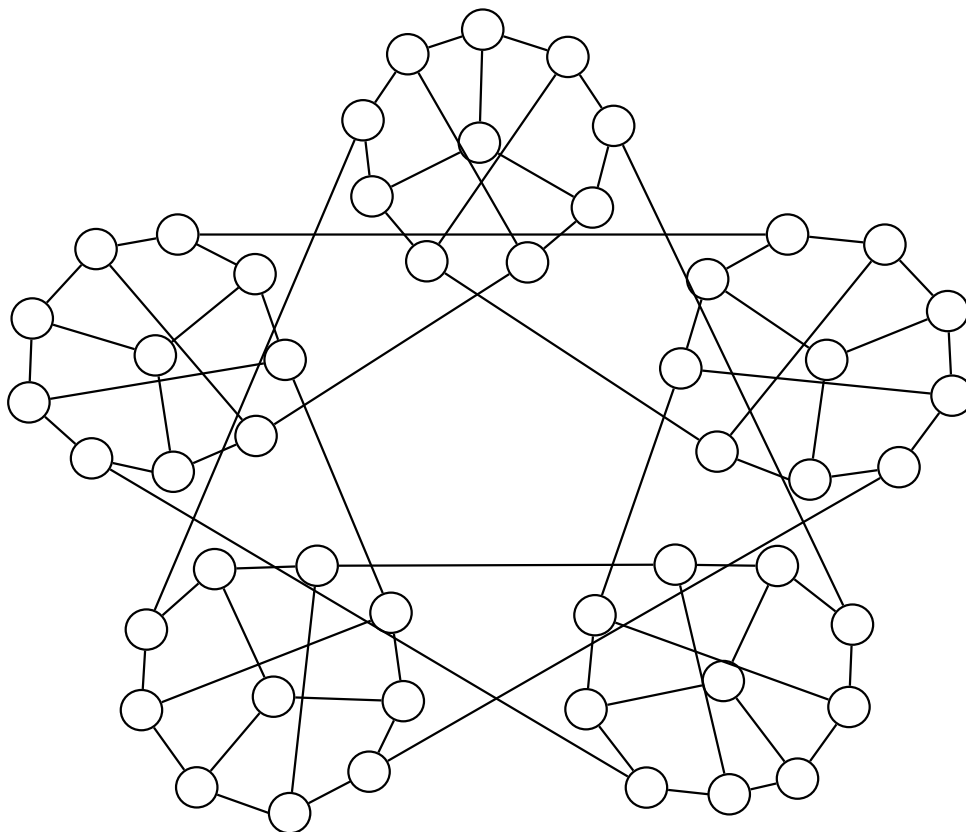
Figura 35 – Blanuša *snark* 2 - 1946

Fonte: Produzido pelos autores

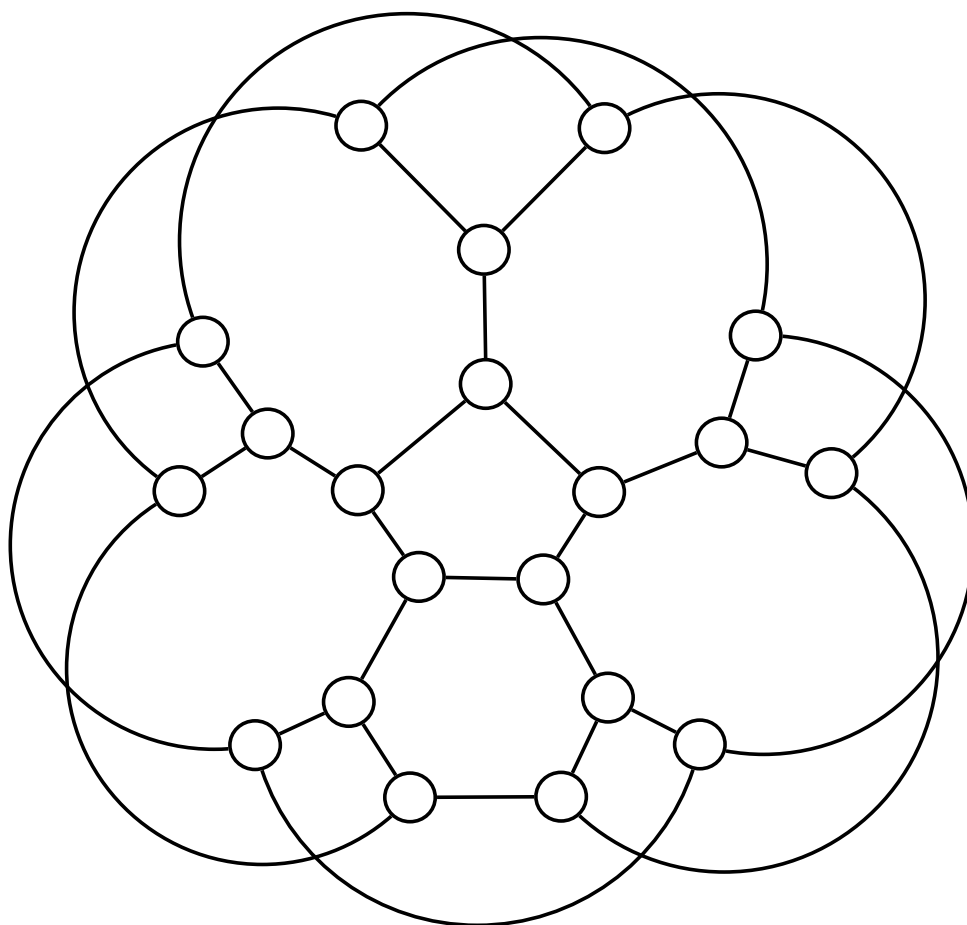
Figura 36 – Blanche Descartes *snark* - 1948

Fonte: Wikipedia

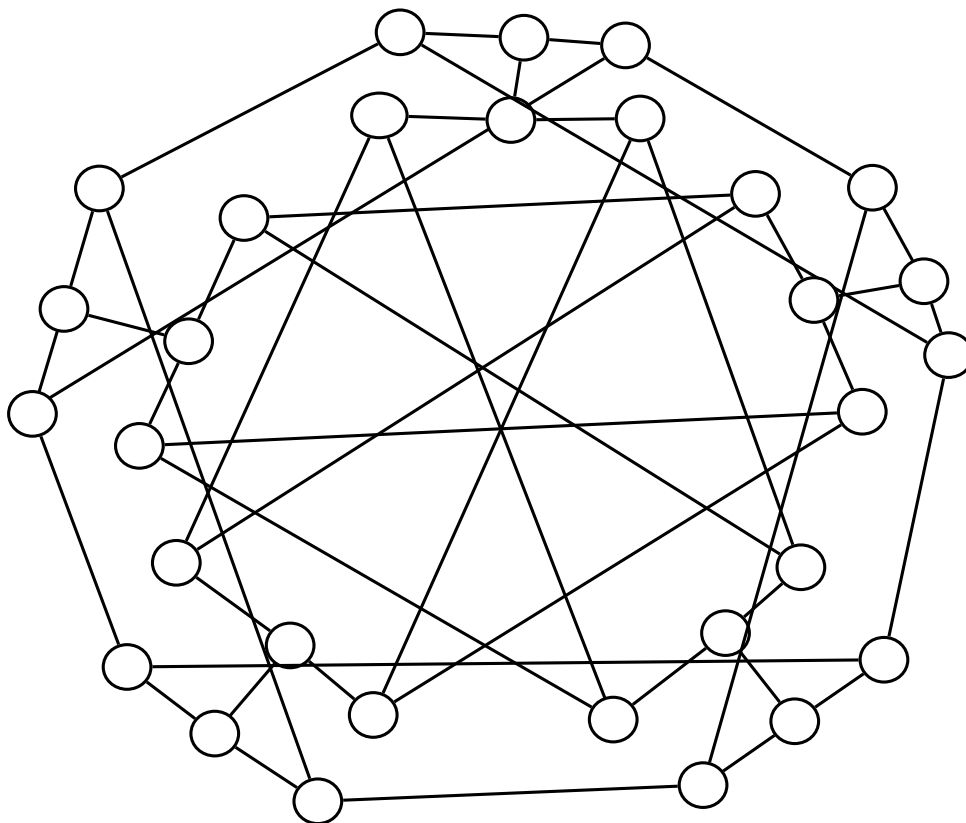
Figura 37 – Szekeres snark - 1973



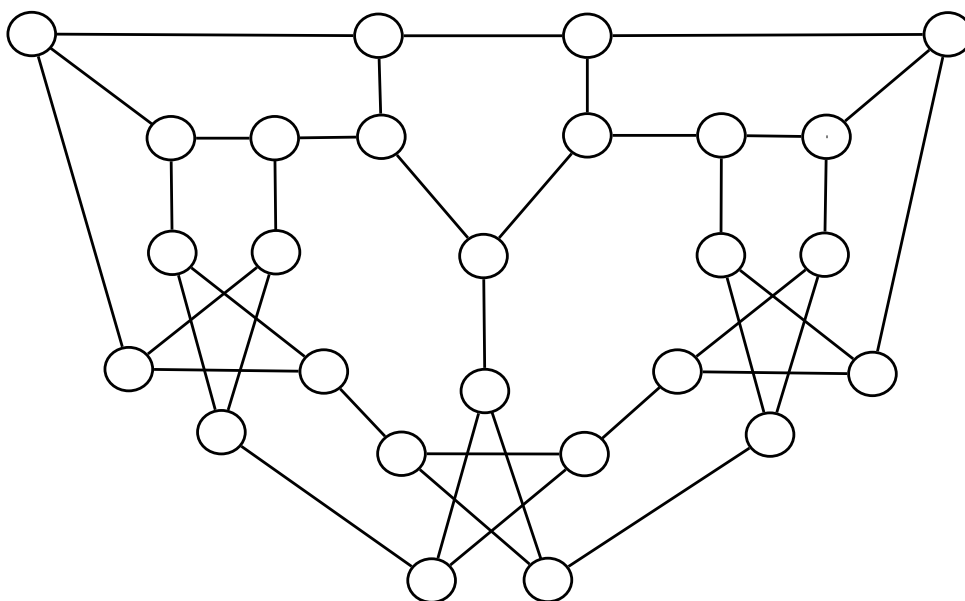
Fonte: Produzido pelos autores

Figura 38 – *Flower-snark* J_5 - 1975

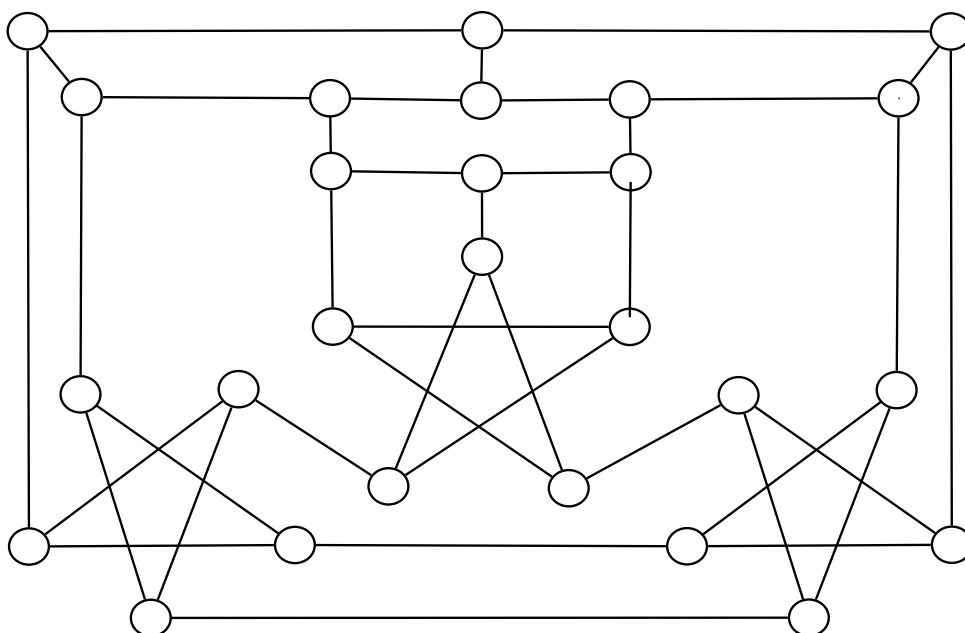
Fonte: Produzido pelos autores

Figura 39 – *Double-star snark* - 1975

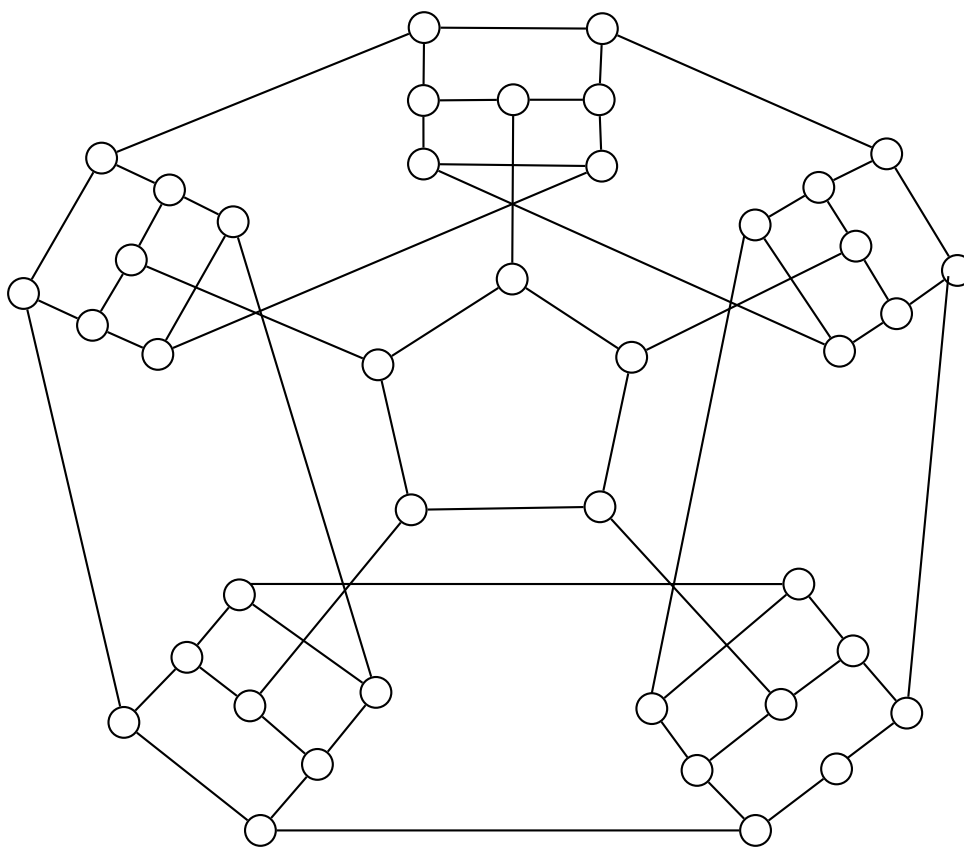
Fonte: Produzido pelos autores

Figura 40 – Celmins-Swart *snark* 1 - 1979

Fonte: Produzido pelos autores

Figura 41 – Celmins-Swart *snark* 2 - 1979

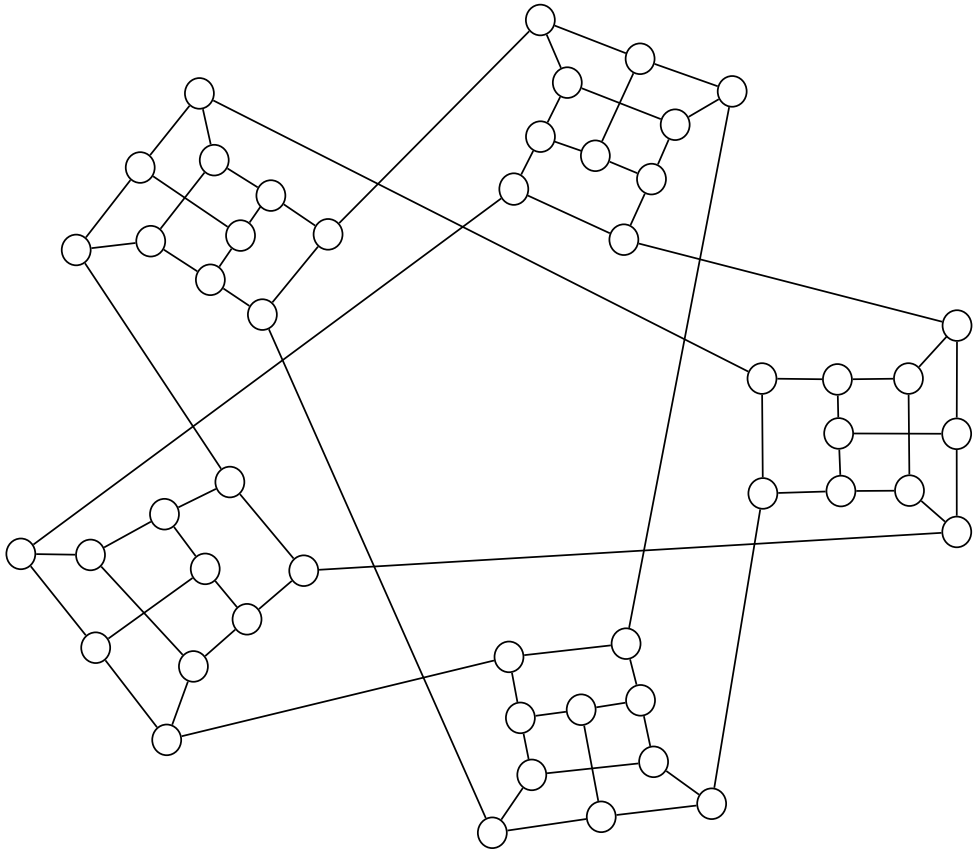
Fonte: Produzido pelos autores

Figura 42 – Goldberg *snark* 5 - 1981

Fonte: Produzido pelos autores

A Figura 42 ilustra um dos *snarks* da família infinita de Goldberg.

Figura 43 – Watkins snark - 1989



Fonte: Produzido pelos autores