Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

Covering cubic graphs with perfect matchings

Bachelor Thesis

2020
Ivan Agarský

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

Covering cubic graphs with perfect matchings

Bachelor Thesis

Study Programme:	Computer Science
Field of Study:	Computer Science
Department:	Department of Computer Science
Supervisor:	RNDr. Ján Mazák, PhD.

Bratislava, 2020
Ivan Agarský

76214648

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Ivan Agarský
Študijný program: informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Covering cubic graphs with perfect matchings
Pokrývanie kubických grafov perfektnými páreniami

Anotácia: Budeme skúmať nasledovný problém: aká časť hrán kubického grafu sa dá pokryť dvomi, tromi, štyrmi perfektnými páreniami? Zaujíma nás, pre ktoré dané racionálne číslo existujú grafy, kde podiel maximálneho počtu pokrytých hrán ku všetkým hranám je rovný tomuto číslu.

Vedúci: doc. RNDr. Ján Mazák, PhD.
Katedra: FMFI.KI - Katedra informatiky
Vedúci katedry: prof. RNDr. Martin Škoviera, PhD.

Dátum zadania: 29.10.2019

Dátum schválenia: 30.10.2019

doc. RNDr. Daniel Olejár, PhD.
garant študijného programu

....

študent vedúci práce

76214648

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Ivan Agarský
Study programme:	Computer Science (Single degree study, bachelor I. deg., full time form)
Field of Study:	Computer Science
Type of Thesis:	Bachelor´s thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: Covering cubic graphs with perfect matchings

Annotation: We will focus on the following problem: what proportion of the edges of a cubic graph can be covered by two, three, four perfect matchings? We are interested in determining the set of rational numbers such that there exists a cubic graph with the ratio of the maximum number of covered edges to the size of the graph equal to the given number.

Supervisor:	doc. RNDr. Ján Mazák, PhD.
Department:	FMFI.KI - Department of Computer Science
Head of department:	prof. RNDr. Martin Škoviera, PhD.
Assigned:	29.10.2019
Approved:	30.10.2019

doc. RNDr. Daniel Olejár, PhD.
Guarantor of Study Programme

....
Student

....
Supervisor

I wish to express my deepest gratitude to my advisor, Ján Mazák, for his time allotted to help and guide me through the writing of this thesis. Without his guidance, results of this work would be of much lower quality. Thanks should also go to my friends who helped me in various ways.

Abstrakt

Skúmali sme pokrytie kubických grafov perfektnými páreniami - podmnožinami hrán, z ktorých žiadne dve nie sú navzájom incidentné a zároveň každý vrchol grafu je incidentný s presne jednou z týchto hrán. Cyklická súvislosť predstavuje najmenší počet hrán ktorý musíme z grafu odstrániť aby sme dostali oddelené cykli vo výsledných komponentoch. Pozerali sme sa na pokrytia s tromi perfektnými páreniami. Zaujímali nás grafy s cyklickou súvislosťou 3 a viac. Za pomoci výpočtovej sily sme ukázali niektoré vlastnosti určitých multipólov a našli spôsob ako pre každé racionálne číslo z intervalu $\left(\frac{23}{27}, 1\right)$ nájsť graf s cyklickou súvislosťou 3, čije najlepšie pokrytie tromi perfektnými páreniami je presne toto číslo. Podarilo sa nám dostať podobný výsledok aj pre cyklickú súvislosť 4 a interval $\left(\frac{28}{31}, 1\right)$.

Kľúčové slová: kubický graf, perfektné párenie, multipól, cyklická súvislosť

Abstract

We studied covering of cubic graphs with perfect matchings (1-factors). Cyclic connectivity is the smallest number of edges which one need to remove from the graph to make its cycles separated in different components of the resulting graph. We looked at covering with three perfect matchings. Our interest was directed at graphs with cyclic connectivity 3 and more. With the help of a computer, we showed some attributes of certain multipoles and found a way to generate a graph with cyclic connectivity 3, for any rational number from the interval $\left(\frac{23}{27}, 1\right)$ such that its best covering with three perfect matchings covers precisely this percent of edges. We managed to achieve a similar result for cyclic connectivity 4 and interval $\left(\frac{28}{31}, 1\right)$.

Keywords: cubic graph, perfect matching, multipole, cyclic connectivity

Contents

Introduction

Perfect matchings are an important topic of study in graph theory. Many unsolved conjectures revolve around perfect matchings and covering graphs with them. We decided to study covering of cubic graphs with three perfect matchings. We modified an existing library for graph manipulation called *ba-graph*. Several functions that enabled us to search for perfect matchings in multipoles were added and tweaked. Our goal was to generate graphs that have a 3PM coverage equal to a given fraction we picked from a certain interval. We tried to keep the lower bound of the interval lowest possible. We built upon a known result about generating graphs with cyclic connectivity 2 and 3PM coverage equal to a given fraction from the interval $\left(\frac{4}{5}, 1\right)$ [11].

This work is divided into four parts. First part is about basic definitions and properties of graphs and graph-like structures. Second part is about previous work that was done in this field. Important unsolved conjectures are mentioned here. Next part explains how we approached this problem - what software we used and how we programmed it. Last part contains results that we achieved. Two theorems with elaborate proofs are presented there.

Chapter 1

Preliminaries

1.1 Elementary Definitions

What follows is a standard graph theory terminology as used in [3]. Section 1.1 gives brief and self-sufficient introduction to the most basic graph theory terms.

A *graph* is a pair (E, V) of sets such that the elements of E are 2-element subsets of set V. These 2-elements subsets are called *edges*. Elements of set V are called *vertices*. For notation ambiguity, V and E have empty intersection. Graphs are most commonly denoted by letter G. For a particular graph we denote $E(G)$ its edge set and $V(G)$ its vertex set.

Two vertices v and w in the graph G are said to be *adjacent* or *neighbouring* when there is an edge (v, w) in the graph G. Two edges are *adjacent* when they have a vertex in common. When edges connect same two vertices, they are called *multi-edges*. When an edge connects vertex with itself, it is called a *loop*. A *simple graph* is a graph containing no loops and no multi-edges. A *complete graph* is a graph in which addition of another edge between existing vertices would create a graph that is not simple. Complete graph containing n vertices is denoted as K^n.

Graph is called *finite* if its vertex and edge set is finite. In this thesis, we will be concentrating on finite graphs without loops and multi-edges. *Order* of a graph is cardinality of its vertex set.

Edge is *incident* with a vertex if that vertex is one of the two vertices the edge connects. *Degree* of a vertex is number of incident edges. *Cubic graphs* are graphs whose vertices are all of degree three. Most of the graphs studied in this thesis will be cubic.

A sequence of edges which joins a sequence of vertices which are all distinct in graph G is called a *path* in G. Graph G is *connected* when each pair of vertices in G is connected by a path in G. A *subgraph* of graph G is a graph whose vertex set and edge set are subsets of $V(G)$ and $E(G)$, respectively. A *component* of a graph G is a subgraph in which all vertices are pairwise connected. An edge is called a *bridge* when its removal increases number of graph's components. A *bridgeless graph* is a graph which does not contain a bridge.

A *cycle* of a graph G is a subset of $E(G)$ that forms a path such that first and last vertex of the path is equal. A cycle that uses every vertex of a graph exactly once is called a *Hamiltonian cycle*. A graph that contains a Hamiltonian cycle is called a *Hamiltonian graph*.

1.2 Colouring

A vertex colouring of a graph $G = (V, E)$ is a map $c : V \to S$ such that $c(v) \neq c(w)$ whenever v and w are adjacent. The elements of set S are called *colours*. We will be interested in the smallest integer k such that G has a k-colouring that is a vertex colouring c with k colours. This k is the *(vertex-) chromatic number* of G. It is denoted by $\chi(G)$. A k-chromatic graph G is a graph with $\chi(G)$ equal to k. If $\chi(G)$ is less or equal to k, we call G k-colourable.

An *edge colouring* of $G = (V, E)$ is map $c : E \to S$ with $c(e) \neq c(f)$ for any adjacent edges e and f. The smallest integer k such that graph is *k-edge colourable*, i.e. has an edge colouring with k colours, is the *edge-chromatic number*, or *chromatic index* of G. It is denoted by $\chi'(G)$.

1.3 Matchings

All graphs in this chapter are simple, if not stated otherwise.

A set M of pairwise non-adjacent edges in a graph $G = (V, E)$ is called a matching. The vertices incident with edges in the matching are called *matched*. Vertices not incident with any edge of M are *unmatched*. A *maximal matching* is a matching M of a graph G such that the addition of another edge to M would no longer be a matching in G. A *maximum matching* is a matching that contains the largest possible number of edges. The *matching number* $\nu(G)$ of a graph G is the size of a maximum matching in G.

A *perfect matching* (or 1-*factor*) is a matching in which every vertex of a graph is incident with exactly one edge of the matching. This means that a perfect matching can only exist in a graph of an even order. A *near-perfect matching* is a matching in which exactly one vertex is unmatched. This occurs when the graph is of odd order. If, for every vertex in a graph, there is a near-perfect matching that omits only that vertex, the graph is called *factor-critical.*

1.4 Snarks

Snarks are a family of simple, bridgeless cubic graphs which cannot be properly coloured with three colours. Reasons why some cubic graphs are three colourable and others are not, are not fully understood [7]. Snarks are hard to be found computationally, because most cubic graphs are Hamiltonian and therefore 3-edge colourable [20]. Deciding whether a cubic graph is 3-edge colourable or not is NP-complete [8]. Smallest snark is the well-known *Petersen graph* [Figure 1.1]. Let us consider a snark G which contains an induced subgraph H, where $\chi'(H) > 3$ and $\chi'(G-H) \leq 3$. We can see that $G - H$ does not make the graph G a snark, so it can be cut off. The remaining subgraph H can be converted into a snark H' by introducing at most one vertex. Thus G is made out of smaller snark H' by introducing a certain number of unimportant vertices [1]. Removing unimportant vertices from a snark G is called a *reduction* of G. Reductions in which we cut k edges are called k*-reduction.* Examples for $k = 2, 3, 4$ can be seen in [1] (digon, triangle and quadrilateral). Another example of a snark is the Tietze's graph [Figure 1.2].

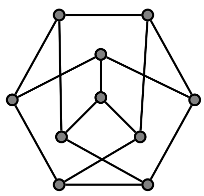

Figure 1.1: The Petersen graph

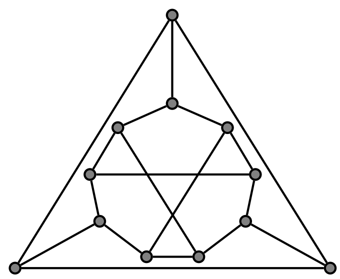

Figure 1.2: The Tietze's graph

1.5 Multipoles

An edge which is at both ends incident with a vertex is called a *proper edge* (also an *inner edge*). If one end of an edge is incident with a vertex and the other is not, that edge is called a *dangling edge.* If none of the ends are incident with a vertex, edge is called an *isolated edge.* An end of an edge that is not incident with a vertex is called a *semi-edge.*

Multipole is a graph that admits dangling and isolated edges. Multipoles are a great tool for building large graph structures from smaller building blocks. They can be crated in several ways. One of them is cutting an existing edge in a graph. Multipoles can be joined via dangling edges to create proper edges in the new larger graph. Figures

1.3 and 1.4 show two multipoles (created by cutting one edge in a K_4 graph) being joined.

Figure 1.3: Multipoles before joining

Figure 1.4: Two multipoles joined

1.6 Cyclic Connectivity

Let G be a connected graph. An *edge-cut* of G is any set S of edges of G such that $G - S$ is disconnected. Edge-cut is also called simply a *cut*. A cut is called *trivial* if it is made out of all edges incident with one vertex, otherwise it is called *non-trivial*. *Cocycle* is a type of an edge-cut which is a set $\delta_G(H)$ of all edges with exactly one end in H, where H is an induced subgraph of G. An edge-cut is said to be *cycle-separating* whenever at least two graph components of $G - S$ have cycles. Connected graph G is said to be *cyclically k-edge-connected* if no set of less than k edges is cycle-separating in G. Number $\beta(G)$ which is defined to be $|E(G)| - |V(G)| + 1$ is the cycle rank of G. The *cyclic connectivity* of G is the largest number $k \leq \beta(G)$ for which G is cyclically k-edge-connected. Figure 1.5 shows an example for k equals 4.

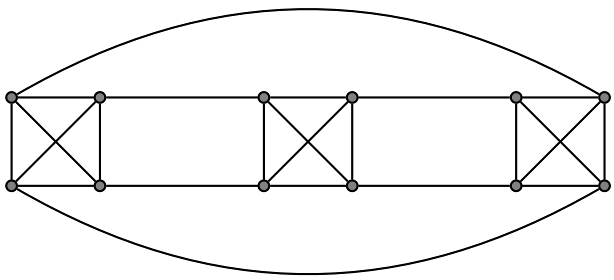

Figure 1.5: A cyclically 4-edge connected graph

Chapter 2

Previous work

2.1 Berge and Fulkerson

D. R. Fulkerson[1] stated the following conjecture in [6] (some authors attribute it to Berge). It was stated in the context of blocking and anti-blocking pairs of polyhedra. Here we present it in a more modern version, using context of perfect matchings.

Conjecture 1 (Berge-Fulkerson) *For every bridgeless cubic graph* $\mathbf{G}$*, there exist 6 perfect matchings* $M_1, M_2, \ldots, M_6$ *such that every edge* $e \in \mathbf{G}$ *is in exactly two of the matchings.*

If $\chi'(G) = 3$ (see 1.2) then we can easily find three distinct perfect matchings M_1, M_2 and M_3 - one for each color. These matchings cover all the edges in a graph. Taking each one of them twice, we get six perfect matchings having the property from the Berge-Fulkerson conjecture. This implies that the conjecture holds for every graph having $\chi'(G) = 3$. We will be interested in those cubic graphs where $\chi'(G) > 3$ (see 1.4). Berge[2] also states a weaker version of Conjecture 1.

Conjecture 2 (Berge) *For every bridgeless cubic graph* $\mathbf{G}$*, there exist 5 perfect matchings* $M_1, M_2, \ldots, M_5$ *such that every edge* $e \in \mathbf{G}$ *is in at least one of the matchings.*

Note: Even if the number 5 in Conjecture 2 is replaced by any larger number, the statement in general is not known to be true. [9]

G. Mazzuoccolo[3] studied previous two conjectures and proved their equivalence in

[1]Delbert Ray Fulkerson (1924 - 1976), an American mathematician, who co-authored the Ford-Fulkerson algorithm

[2]Claude Jacques Berge (1926 - 2002), a French mathematician, one of the founders of graph theory

[3]Giuseppe Mazzuoccolo, researcher at the Department of Computer Science, University of Verona

[14]. Mazzuoccolo called the 6 perfect matchings in Fulkerson's conjecture *a Fulkerson cover* and used the notation $\chi'_e(G)$ to denote the minimum number of 1-factors needed to cover the edge-set of G.

Note: Conjecture 2 can be restated as: *Let G be a bridgeless cubic graph. Then $\chi'_e(G) \leq 5$.*

He proved the following, which directly implies the equivalence of Conjectures 1 and 2: **(Mazzuoccolo)** *If for each bridgeless cubic graph G the relation $\chi'_e(G) \leq 5$ holds, then each bridgeless cubic graph admits a Fulkerson cover.*

P. Seymour[1] came up with a more generalized version of Berge-Fulkerson conjecture.

Definition (r-graph). *An r-graph is an r-regular graph G with even order with the property that every edge-cut which separates $V(G)$ into two sets of odd cardinality has size at least r.*

Conjecture 3 (the generalized Berge-Fulkerson conjecture) *Let G be an r-graph. Then there exist $2r$ perfect matchings $M_1, \ldots, M_{2r}$ of G with the property that every edge $e \in G$ is contained in exactly two of the matchings.* [21]

As a remark, we mention that for $r = 3$, this becomes the original Berge-Fulkerson conjecture. The following conjecture is directly implied by Berge-Fulkerson conjecture ($k = 2$) and for that reason is considered a weakening.

Conjecture 4 (the weak Berge-Fulkerson conjecture) *There exists an integer $k > 0$ such that every 3-graph G has $3k$ perfect matchings such that every edge $e \in G$ is contained in exactly k of the matchings.*

Berge also stated the following weakening, which is still open. It is trivially implied by Conjecture 2.

Conjecture 5 (Berge) *There exists a fixed integer k such that the edge set of every 3-graph can be written as a union of k perfect matchings.*

The Berge-Fulkerson conjecture has another consequence about perfect matchings known as Fan-Raspaud[2,3] conjecture [18]. First we will define an *FR-triple*.

[1]Paul Seymour (1950), a British mathematician, currently a professor at Princeton University

[2]Genghua Fan, a Chinese researcher at Fuzhou University

[3]André Raspaud, a researcher at Laboratoire Bordelais de Recherche en Informatique

Definition (FR-triple). *Let G be a bridgeless cubic graph. Then a list of perfect matchings (M_1, M_2, M_3) of graph G is called an FR-triple, if $M_1 \cap M_2 \cap M_3 = \emptyset$.*

Conjecture 7 (Fan-Raspaud) Every bridgeless cubic graph has an FR-triple.

This conjecture directly follows from the Berge-Fulkerson conjecture - it is enough to take any 3 perfect matchings from the original 6.[5] An existential oriented conjecture was stated by Berge.

Conjecture 6 (Berge) *There exists a fixed integer k such that every 3-graph has a list of k perfect matchings with empty intersection.*

An equivalent conjecture to Conjecture 7 was stated by Mačajova and Škoviera[1]. [12] They proved the equivalence in the same paper. Conjecture revolves around Fano colorings (see 1.2) and is as follows.

Conjecture 7 (Four-Line Conjecture) Every bridgeless cubic graph has a Fano colouring which uses at most four lines.

Since Fan-Raspaud conjecture is implied by Berge-Fulkerson and since Fan-Raspaud is equivalent to Four-Line Conjecture, this means that Four-Line Conjecture is implied by Berge-Fulkerson. Another consequence of Berge-Fulkerson is stated by Mazzuoccolo [16].

Conjecture 8 ($\mathbf{S_4}$-Conjecture) *For any bridgeless cubic graph G, there exist* 2 *perfect matchings M_1 and M_2 such that $G - (M_1 \cup M_2)$ is a bipartite subgraph of G.*[17]

Such pair of perfect matchings is called an S_4-pair. [17]

We can say that the Berge-Fulkerson conjecture is very important in graph theory. Its implications are far-reaching. Now we will study the tools that led to some successful proofs surrounding our problem. We will focus on union of perfect matchings in a graph. These unions do not necessarily need to have empty intersections, we will study those that do not.

[1] Edita Mačajová and Martin Škoviera, Slovak researchers at Comenius University

2.2 Numbers m_k

To better study unions of perfect matchings we will use notation introduced in [9] particularly regarding numbers m_k.

$$m_k = \inf_G \max_{M_1,\ldots,M_k} \frac{|\bigcup_i M_i|}{|E(G)|}$$

where the infimum is taken over all bridgeless cubic graphs G, and $M_1, \ldots M_k$ range over all perfect matchings of G. We can interpret this as a fraction of edges that can be covered using k perfect matchings (chosen to cover the most edges) so that the fraction holds for every bridgeless cubic graph. We can now again restate the Conjecture 2 as $m_5 = 1$.

Number $\mathbf{m_1}$

Lemma 1. *$m_1 = 1/3$.*

Proof. Every perfect matching covers the same number of edges in a given graph, $|V(G)|/2$ precisely (graphs with odd order can not have a perfect matching). In every cubic graph a perfect matching covers one in three edges at a given vertex. This implies that $m_1 = 1/3$. □

Number $\mathbf{m_2}$

Upper bound for m_2 can be easily determined by the Petersen graph. This graph has 15 edges and is a snark. Two perfect matchings can cover at most 9 edges *i.e.* $m_2 \leq 3/5$. It has already been shown that $m_2 = 3/5$ [9]. Powerful tool used in the proof of mentioned equality is the Perfect Matching Polytope theorem of Edmonds [4].

Number $\mathbf{m_3}$

The Petersen graph can be again used to show that three perfect matchings can cover at most 4/5 of the edges (see 2.1). This implies that $m_3 \leq 4/5$. It has been proven that $27/35 \leq m_3 \leq 4/5$ [9]. Many authors think that $m_3 = 4/5$.

2.3 Recent progress

There are some important results in graph coverings that hold for certain types of graphs or for graphs with a certain amount of edges. It was known that for a graph to have k perfect matchings, that cover all the edges, it needed to satisfy following inequality: $k > \log_{3/2}(|E(G)|)$. [19] This was later improved by Mazzuoccolo [15].

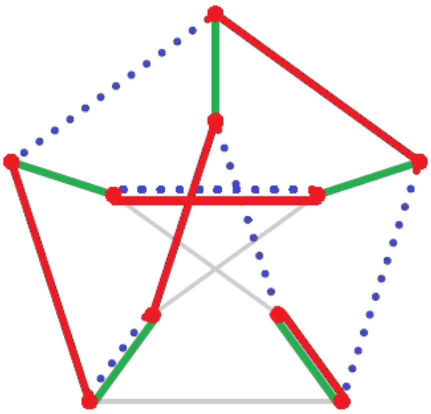

Figure 2.1: One possible triple of perfect matchings in the Petersen graph

Theorem 1 (**Mazzuoccolo**). *If G is a bridgeless cubic graph with fewer than $\frac{2^t}{\sqrt{t}}$ edges, then there is a covering of G by t perfect matchings.*

In the same paper he made a new lower bound for m_5. He studied the sequence defined by the recurrence $a_k = \frac{t}{2t+1}(1 - a_{k-1}) + a_{k-1}$ and $a_0 = 0$. It was announced (without a proof) that $a_k \leq m_k, \forall k$ when the sequence was introduced in [9]. Mazzuoccolo provides a proof of the relation between a_k and m_k, and shows that $a_5 = \frac{215}{231}$, directly implying that $m_5 \geq \frac{215}{231}$ [15].

Theorem 2 (**Mazzuoccolo**). *Let G be a bridgeless cubic graph. There exist five perfect matchings of G that cover at least $\lceil \frac{215}{231}|E(G)| \rceil$ edges of G.*

The following definition and theorem are important, because they show that Berge-Fulkerson theorem holds for some classes of snarks, namely Goldberg graphs. [5]

Definition (**Goldberg graphs**). *Let H be the graph depicted in 2.2. Let G_k (k odd) be a cubic graph obtained from k copies of H ($H_0 \dots H_{k-1}$ where the name of vertices are indexed by i) by adding edges $a_i a_{i+1}$, $c_i c_{i+1}$, $e_i e_{i+1}$, $f_i f_{i+1}$ and $h_i h_{i+1}$ (subscripts are taken modulo k). Graphs G_k are called Goldberg graphs. G_5 is also known as the Goldberg snark 2.3.*

Theorem 3. *For any odd $k \geq 5$, G_k can be provided with a Fulkerson covering.*

Let us now consider the following conjectures introduced in [18]. $\nu_F(e)$ is the number of perfect matchings of the FR-triple F that contain the edge e.

Conjecture 9 *For any bridgeless cubic graph G, any edge $e \in G$ and $i \in \{0, 1, 2\}$, there is an FR-triple F of G, such that $\nu_F(e) = i$.*

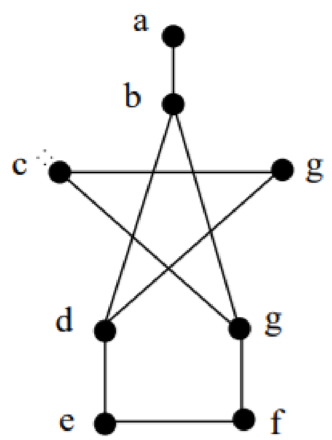

Figure 2.2: Graph H [5]

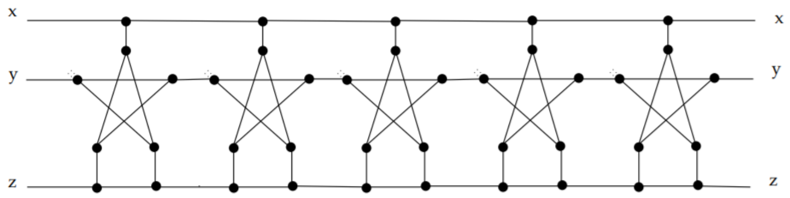

Figure 2.3: Goldberg snark [5]

Conjecture 10 *Let G be a bridgeless cubic graph, e and f be adjacent edges, $0 \leq i$, $j \leq 2$ be two numbers with $1 \leq i + j \leq 3$. Then G contains an FR-triple F, such that $\nu_F(e) = i$ and $\nu_F(f) = j$.*

Berge-Fulkerson conjecture implies Conjecture 10, Conjecture 10 implies Conjecture 9, and Conjecture 9 implies Fan-Raspaud conjecture. It was proved that Fan-Raspaud conjecture and Conjecture 9 are equivalent. It was also proved that if Conjecture 10 has a counter-example then it must be a cyclically 4-edge-connected graph.[18]

Mkrtchyan[1] notes that it is unknown whether the smallest counter-example to Fan-Raspaud conjecture is a cyclically 4-edge-connected graph. It was shown that the smallest counter-example to Conjecture 9 must be a 3-edge-connected. It is unknown whether Conjecture 10 is equivalent to Fan-Raspaud conjecture.[18]

[1]Mkrtchyan Vahan, researcher at Gran Sasso Science Institute, Italy

Chapter 3

Computational methods

To study perfect matchings, we relied on extensive computational power. Many cubic graphs and multipoles were searched through.

3.1 Introduction to graph library *ba-graph*

Graph library *ba-graph* contains useful tools for working with graphs and other graph-like structures. In this library a graph is represented as a collection of graph neighbourhoods called *rotations*. Rotation represents a neighbourhood of a certain vertex. This gives us possibility to refer to the underlying vertex, meaning we can easily iterate over all vertices in the graph. Each rotation contains a collection of *incidences*. Incidence has references to the underlying edge, both vertices, rotations and half-edges. Graph also contains its order (vertex count) and its size (edge count) among other things. Adding vertices and edges to a graph is very simple and straightforward.

```
Graph G(createG()); //creates an empty graph
Vertex v1 = createV();
Vertex v2 = createV();
addV(G, v1);
addV(G, v2);
addE(G, v1, v2); //adds en edge between vertex v1 and vertex v2
```

Each vertex contains a *number*, used to refer to the vertex itself. For example, we can refer to a vertex with number 0 as follows

```
Vertex v = G[0].v();
```

Notice that without *v()* we get the rotation for that vertex. We can refer to edges in a similar way. Here we refer to an edge between vertices with numbers 0 and 1. We can also check whether this edge is a loop. Similarly as in rotations, without the $e()$ we get an incidence.

```
Edge e = G[0][1].e();
bool is_loop = G[0][1].is_loop();
```

This library implements some very good graph searching functions. We could want to find what number does a certain vertex have or whether does a graph contain a given number.

```
Number n = G.find(v)->n();
bool contains = G.contains(n);
```

We can also list all incidences that satisfy some predicates. We can also pass rotation predicates to the search function. predicate $all()$ does not filter out any rotations or incidences. The $primary()$ predicate filters out all non-primary incidences.

```
G.list(RP::all(),IP::all());
G.list(RP::all(),IP::primary());
```

This can be used in relatively complicated, but very useful ways. Another example follows.

```
G[1].find(v2)->r2().v()==v2 //true
```

Here we search for a rotation with number 1, then we find an incidence between number 1 and vertex $v2$. Then the incidence gives us the second rotation $r2$, which we use to access its vertex which is actually the vertex $v2$.

3.2 CNF formulas

A boolean formula is in a conjunctive normal form (CNF) if it represents a conjunction of one or more clauses, where every clause is a disjunction of literals. Boolean formula in conjunctive normal form is simply called CNF formula. Problem of finding perfect matchings in a graph can be encoded in a CNF formula. A sat solver can then check whether this formula is satisfiable and can list all possible configurations in which it is satisfiable. Figure 3.1 is an example of how we can find a perfect matching in a graph with the help of a CNF formula. Every edge is labelled with a literal, meaning it can either be in a perfect matching or not. Now we can construct a CNF formula by making constraints for every vertex neighbourhood. We can start by looking at the bottom left vertex. A perfect matching must include precisely one edge incident with this vertex. It can be either a, b or c. A formula for this vertex would be:

$$(a \vee b \vee c) \wedge (\neg a \vee \neg b) \wedge (\neg b \vee \neg c) \wedge (\neg a \vee \neg c)$$

First clause makes sure that at least one edge is included in the matching. Other clauses make sure that no two edges are included in the matching at the same time. A CNF formula for the whole graph can now be simply a conjunction between CNF formulas for each vertex.

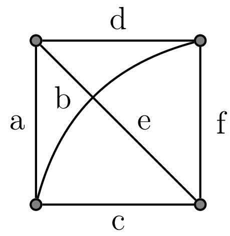

Figure 3.1: Finding a perfect matching in a graph

3.3 Adding possibly not covered vertices parameter

First thing we needed to do to be able to work with multipoles and their perfect matchings was enabling some vertices not to be covered in a 3PM covering. This was done because we wanted to represent a dangling edge using two vertices with one vertex being uncovered i.e. ignored from a perfect matching. One of the functions which was changed was *cnf_kfactor*. This function builds a CNF for a k-factor given graph and a number k. We added a optional parameter which makes sure that chosen vertices can be incident with at most k edges of the k-factor, but do not have to be incident with one (we removed the first clause from the example in 3.2).

```
inline CNF cnf_kfactor(const Graph &G, int k,
   const std::vector<Number>& possiblyNotCoveredVertices)
```

Similarly we changed the function *cnf_perfect_matching*. Chosen vertices doesn't have to be covered with a perfect matching. This makes possible for us to find a perfect matchings inside a multipole by specifying all the dangling vertices. Number of other functions were also changed in a similar way. We also made them accept *Number* and *Vertex* vectors for easier and more flexible usage.

3.4 Stored graphs

Library *ba-graph* can work with *graph6* and *sparse6* graph storing formats. We worked with pre-generated graphs with up to 36 vertices, mainly concentrating on snarks with cyclic connectivity 4 and girth 5. Some of the files used were

cubic_cc04_g05.10.snark
cubic_cc04_g05.18.snark
cubic_cc04_g05.20.snark
cubic_cc04_g05.22.snark

where *cc* means cyclic connectivity, *g* means girth and the last number indicates order of graphs in that file. Accessing these graphs in code is simple, we can just make a graph configuration, then read the graphs we need and access a certain graph by simple indexing. Following code sample can be used to access all graphs from the first file in the list above. Files are located in the folder *resource/graphs/snarks* relative to the library's root directory.

```
Configuration cfg;
cfg.load_from_string("{\"storage\": {\"dir\": \"../../resources/graphs\"}}");
auto sg = StoredGraphs::create<SnarkStorageDataC4G5>(cfg);
auto order = 10;

auto graphCount = sg->get_graphs_count(order);
for(int i = 0; i < graphCount; i++)
{
    Graph G(sg->get_graph(order, i));
}
```

3.5 Sat solvers in *ba-graph*

Library *ba-graph* uses couple of different sat solvers for certain tasks. In our work we needed to ensure that we get all possible solutions for coverings, so we used the all sat solver. It was implemented either by CryptoMiniSat all solver or BDD all solver. Sat solvers are relatively fast and therefore very useful in computations that are easily transformable to a CNF satisfiability problem. As we showed in 3.2, finding a perfect matching is a good example of a CNF satisfiability problem. We can ask the solver to return all the solutions that satisfy the CNF formula, thus getting all possible perfect matchings. Then we can easily iterate through every three-combinations.

3.6 Creating multipoles

There are a multipole ways to create multipoles :). We created multipoles by cutting edges or vertices in cubic graphs. By splitting an edge we get a 2-pole i.e. the resulting graph has 2 dangling edges [Figure 3.2]. By splitting a vertex we get a 3-pole with 3 dangling edges [Figure 3.3].

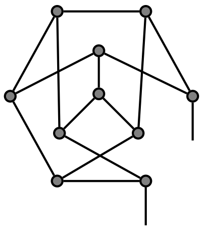

Figure 3.2: A 2-pole

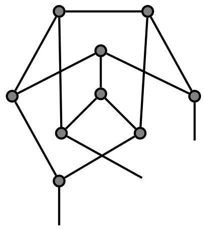

Figure 3.3: A 3-pole

For cyclic connectivity 2 we made multipoles by cutting one of the edges. For cyclic connectivity 3 we made multipoles by cutting one of the vertices. Cyclic connectivity 4 can be achieved by splitting two edges or by removing two neighbouring vertices. Finally, cyclic connectivity 5 can be achieved by removing a vertex and cutting an edge that was not incident with the removed vertex.

3.7 Calculating uncovered edges

Snarks usually have many perfect matchings. We want to know how many edges can be covered with 3 perfect matchings. The function *minimum_uncovered_edges_3PM* calculates how much edges can't be covered with 3 perfect matchings. The function *minimum_uncovered_edges_3PM_for_multipoles* is similar and does the same thing but works with multipoles, one just needs to specify number of dangling edges. This function works by simply iterating through every possible three-combination of all perfect matchings in the graph. For every combination we calculate amount of uncovered inner and uncovered dangling edges in the graph using the function *uncovered_edges_for_multipoles_count*. Then we find the combination with minimum uncovered edges. Finally we need to add those combinations that have the same amount of uncovered edges but in a different arrangement i.e. number of uncovered dangling edges is not the same. This is done, because different dangling edges covering configurations can be useful in different scenarios. An example of this situation can be as follows:

A 2-pole has 3 inner and 2 dangling edges uncovered. Additionally there is a combination in which 5 inner and 0 dangling edges are uncovered. In the end both will be included in the result returned by the function.

3.8 Source code

The source code for *ba-graph* library as it appeared when this thesis was being written is supplied as an attachment to this work. The source code for our program used to search and generate graphs and multipoles is also provided as an attachment. Additionally a textual file is provided that summarizes all the changes made to the *ba-graph* library. Finally, we also added some stored graph files that were used to generate multipoles.

Chapter 4

Results

Problem 1. *Let $r \in (0,1) \cap Q$ be a fraction and $k > 2$ an integer. Is it true that there exists a cyclically k-connected cubic graph G such that $m_3(G) = r$?*

Proofs that follow rely on a so-called *dilution* technique, appearing in an unpublished paper *On covering cubic graphs with three perfect matchings* by J. Mazák and E. Mačajová [11].

4.1 Cyclic connectivity 3

We will show that for any $r \in (\frac{23}{27}, 1)$ we can find a cyclically 3-connected cubic graph G for which $m_3(G) = r$. Let us start with a cyclically 3-connected ladder graph L_n^c (symbol ^{c} stands for cyclically connected) [Figure 4.1].

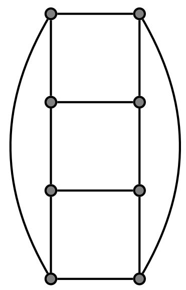

Figure 4.1: Cyclically 3-connected ladder graph L_4^c

We can easily see that for any natural number n, $m_3(L_n^c)$ is equal to 1. Next, we will start exchanging some vertices of this graph with a multipole made from splitting one vertex in the Petersen graph. This will push the m_3 coverage down as the Petersen graph cannot be fully covered with 3 perfect matchings. Finally if we exchange all the vertices, we show that the m_3 of such graph is $\frac{23}{27}$.

Let A denote the Petersen multipole and let B be a vertex with 3 dangling edges. Graph $G_{a,b}$ will be a graph made from exchanging a vertices of the graph $L^c_{(a+b)/2}$ with vertex split Petersen multipoles. We can easily see that $(a+b)$ must always be even. Additionally, $\frac{a+b}{2}$ must also be even, otherwise graph $L^c_{(a+b)/2}$ does not have a 3PM configuration covering all outer edges in the B graphs.

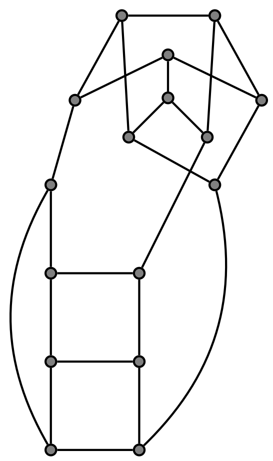

Figure 4.2: Graph $G_{7,1}$

Figure 4.2 shows an example of a graph $G_{a,b}$ for $a = 7$ and $b = 1$.

Number of edges in the graph $G_{a,b}$ can be calculated by counting inner and outer edges of the graph, where outer are the edges that connect graphs A and B. There are 12 inner and 3 outer edges in every copy of the graph A. Graph B contains only 3 outer edges. The number of edges in the whole graph can now be calculated easily as

$$12A + \frac{3}{2}(A + B)$$

(we needed to divide by 2 because we counted every outer edge twice). Computing all possible 3PM coverages of graph A shows that 2 edges can be uncovered only when we have a configuration in which 2 PMs cover a unique dangling edge and the third PM covers all 3 dangling edges. To be able to connect multipole A with the rest of the graph, we must ensure that the 3 dangling edges are covered with 3 different perfect matchings. This can easily be done by taking only the third dangling edge in the perfect matching that covers all dangling edges. Our computation also showed that there are no configurations in which there are more than 2 uncovered inner edges in the graph A. Thus the maximum number of covered edges in a 3PM coverage in the graph $G_{a,b}$ is

$$10A + \frac{3}{2}(A + B)$$

This gives us the m_3 for our graph.

$$m_3(G_{a,b}) = \frac{10A + \frac{3}{2}(A+B)}{12A + \frac{3}{2}(A+B)} = \frac{23A + 3B}{27A + 3B}$$

Theorem 4. *For each fraction $\frac{p}{q} \in (\frac{23}{27}, 1)$, there exists infinitely many bridgeless cubic graphs G such that $m_3(G) = \frac{p}{q}$.*

Proof. Let us consider the graph $G_{a,b}$ for $a = 3(q-p)$, $b = 27p - 23q$ (since $1 > \frac{p}{q} > \frac{23}{27}$, both a and b are positive).

$$m_3(G_{a,b}) = \frac{23(3(q-p)) + 3(27p - 23q)}{27(3(q-p)) + 3(27p - 23q)} = \frac{12p}{12q} = \frac{p}{q}$$

so $G_{a,b}$ satisfies the required property. In fact $G_{na,nb}$ for any integer n satisfies this property too. □

Now we can formulate a more general theorem for cyclic connectivity 3. Let us consider that we found a graph C that has lower 3PM coverage than graph A. Let us say that it has y uncovered inner edges out of x edges (all outer edges are covered). There are $x - 3$ inner and 3 outer edges in this graph. This gives us the m_3 for graph $G_{c,b}$.

$$m_3(G_{c,b}) = \frac{(x-y-3)C + \frac{3}{2}(C+B)}{(x-3)C + \frac{3}{2}(C+B)} = \frac{(2x-2y-3)C + 3B}{(2x-3)C + 3B}$$

Theorem 5. *For each fraction $\frac{p}{q} \in (\frac{2x-2y-3}{2x-3}, 1)$, where x is the number of edges and y the number of uncovered edges in the lower bound graph, there exists infinitely many bridgeless cubic graphs G such that $m_3(G) = \frac{p}{q}$.*

Proof. Let us consider the graph $G_{c,b}$ for $c = 3(q-p)$, $b = (2x-3)p - (2x-2y-3)q$ (since $1 > \frac{p}{q} > \frac{2x-2y-3}{2x-3}$, both c and b are positive).

$$m_3(G_{c,b}) = \frac{(2x-2y-3)C + 3B}{(2x-3)C + 3B}$$

$$= \frac{(2x-2y-3)3(q-p) + 3((2x-3)p - (2(x-y)+3)q)}{(2x-3)3(q-p) + 3((2x-3)p - (2(x-y)+3)q)} = \frac{2yp}{2yq} = \frac{p}{q}$$

so $G_{c,b}$ satisfies the required property. In fact $G_{nc,nb}$ for any integer n satisfies this property too. □

4.2 Cyclic connectivity 4

We will show that for any $r \in (\frac{28}{31}, 1)$ we can find a cyclically 4-connected cubic graph G for which $m_3(G) = r$.

Consider a multipole made from splitting two neighbouring vertices in the Petersen graph. Let us call it graph B. This multipole has 4 dangling edges [Figure 4.3]. Graph B can be fully covered with 3 perfect matchings [Figure 4.3].

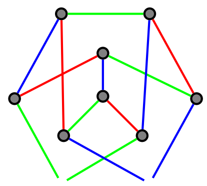

Figure 4.3: The graph B with 3PM coverage

Let us start with a cyclically 4-connected graph B^c_n made from connecting n B graphs in a circular way, by connecting 2 of the dangling edges to the right graph and remaining 2 dangling edges to the graph on the left.

We can easily see that for any natural number n, $m_3(B^c_n)$ is equal to 1. Next, we will start exchanging some B subgraphs of this graph with a multipole made from splitting 2 neighbouring vertices in a graph with order 22 and size 33. Let us denote this graph A [Figure 4.4]. It has 3 inner edges uncovered in a 3PM coverage. Notice that 3 of the covered edges are coloured green. Green edge represents and edge that was covered with 2 perfect matchings (it's easy to see which two by looking at colours of neighbouring edges). Gray edge on the other hand represents an uncovered edge.

Note: Dangling edges have vertices of degree one on them. This is only due to how *Gephi* works, these vertices don't actually exist.

Computing power showed that in this configuration dangling edges are covered with two perfect matchings in the graph A. This means that it connects nicely with the graph B. This way can create a graph $G_{a,b}$ consisting of a copies of graph A and b copies of graph B. The number of edges in the whole graph can now be calculated easily as

$$10B + 29A + \frac{4A + 4B}{2}$$

(again we divided by two to negate duplicate edge counting from joined dangling edges). Extensive computation showed that there are no better 3PM coverages of the graph A that this one. The maximum number of covered edges is thus

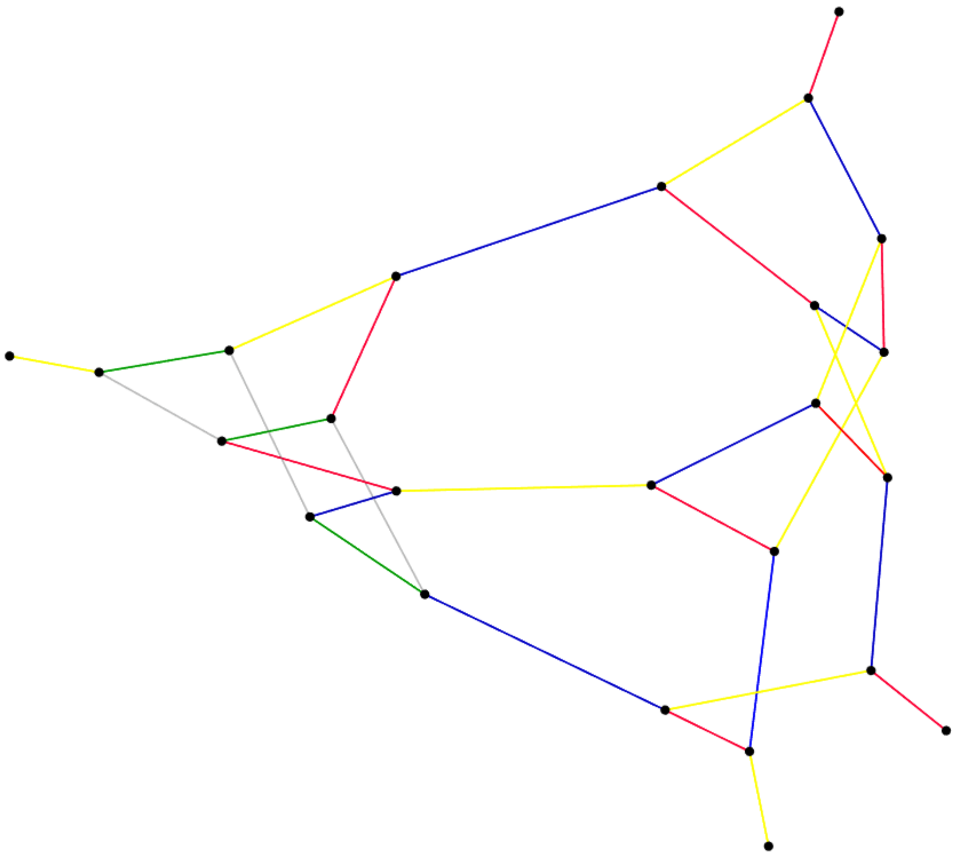

Figure 4.4: Graph A, visualized with *Gephi* [2]

$$10B + 26A + \frac{4A + 4B}{2}$$

This gives us the m_3 for our graph.

$$m_3(G_{a,b}) = \frac{10B + 26A + 2(A + B)}{10B + 29A + 2(A + B)} = \frac{12B + 28A}{12B + 31A}$$

Theorem 6. *For each fraction* $\frac{p}{q} \in (\frac{28}{31}, 1)$*, there exists infinitely many bridgeless cubic graphs* G *such that* $m_3(G) = \frac{p}{q}$.

Proof. Let us consider the graph $G_{a,b}$ for $a = 12(q-p)$, $b = 31p - 28q$ (since $1 > \frac{p}{q} > \frac{28}{31}$, both a and b are positive).

$$m_3(G_{a,b}) = \frac{12(31p - 28q) + 28(12(q - p))}{12(31p - 28q) + 31(12(q - p))} = \frac{3 \cdot 12p}{3 \cdot 12q} = \frac{p}{q}$$

so $G_{a,b}$ satisfies the required property. In fact $G_{na,nb}$ for any integer n satisfies this property too. $\square$

Now we can formulate a more general theorem for cyclic connectivity 4. Let us consider that we found a graph C that has lower 3PM coverage than graph A. Let us say that it has y uncovered inner edges out of x edges (all outer edges are covered). There are $x-4$ inner and 4 outer edges in this graph. This gives us the m_3 for graph $G_{c,b}$.

$$m_3(G_{c,b}) = \frac{10B + (x-y-4)C + 2(C+B)}{10B + (x-4)C + 2(C+B)} = \frac{(x-y-2)C + 12B}{(x-2)C + 12B}$$

Theorem 7. *For each fraction* $\frac{p}{q} \in (\frac{x-y-2}{x-2}, 1)$, *where x is the number of edges and y the number of uncovered edges in the lower bound graph, there exists infinitely many bridgeless cubic graphs G such that* $m_3(G) = \frac{p}{q}$.

Proof. Let us consider the graph $G_{c,b}$ for $c = 12(q-p)$, $b = (x-2)p - (x-y-2)q$ (since $1 > \frac{p}{q} > \frac{x-y-2}{x-2}$, both c and b are positive).

$$m_3(G_{c,b}) = \frac{(x-y-2)C + 2B}{(x-2)C + 2B}$$

$$= \frac{(x-y-2)12(q-p) + 12((x-2)p - (x-y-2)q)}{(x-2)12(q-p) + 12((x-2)p - (x-y-2)q)} = \frac{12yp}{12yq} = \frac{p}{q}$$

so $G_{c,b}$ satisfies the required property. In fact $G_{nc,nb}$ for any integer n satisfies this property too. $\square$

Conclusion

Importance

Results achieved can be deemed somewhat interesting, but what is more important is the *dilution* technique used in those proofs. This technique can be applied to any coverings and any interval $(x, 1)$ as long as the lower bound graph is found successfully (upper bound graph can stay the same, because it is fully covered).

Further study

Results presented in the Chapter 4 can be improved and extended in various ways. One way would be to increase the computing power or provide more time for the computation to run longer and find graphs that maybe have better (lower) 3PM coverage than the graphs used in these proofs. Another way would be to study cyclic connectivity of higher order. Finding graphs with cyclic connectivity 5 and 6 can be relatively easily added to the program and used to provide similar results to the ones presented in this work, albeit with higher cyclic connectivity.

Altogether different way to study coverings with perfect matchings would be to study coverings with 4 or 5 perfect matchings. We chose coverings with 3 perfect matchings, because they showed very promising. We did not need to search for a long time to find graphs that had good 3PM coverage. On the other hand, almost all small snarks are completely covered with 4 perfect matchings, so one would need to search bigger snarks, but unfortunately those are out of reach for today's computers.

Bibliography

[1] Miroslav Chladný and Martin Škoviera. Factorisation of snarks. *The Electronic Journal of Combinatorics*, 17(1):32, 2010.

[2] The Gephi Consortium. Gephi 0.9.2. `https://gephi.org/`, 2017.

[3] Reinhard Diestel. *Graph theory 3rd ed.* Graduate texts in mathematics. 2005.

[4] Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of research of the National Bureau of Standards B*, 69(125-130):55–56, 1965.

[5] Jean-Luc Fouquet and Jean-Marie Vanherpe. On Fulkerson conjecture. *arXiv preprint arXiv:0906.1086*, 2009.

[6] Delbert Ray Fulkerson. Blocking and anti-blocking pairs of polyhedra. *Mathematical programming*, 1(1):168–194, 1971.

[7] Jan Goedgebeur, Edita Máčajová, and Martin Škoviera. Smallest snarks with oddness 4 and cyclic connectivity 4 have order 44. *arXiv preprint arXiv:1712.07867*, 2017.

[8] Ian Holyer. The np-completeness of edge-coloring. *SIAM Journal on computing*, 10(4):718–720, 1981.

[9] Tomáš Kaiser, Serguei Norine, et al. Unions of perfect matchings in cubic graphs. In *Topics in discrete mathematics*, pages 225–230. Springer, 2006.

[10] Robert Lukoťka and Ján Mazák. ba-graph. `https://bitbucket.org/relatko/ba-graph`, 2018.

[11] Edita Máčajová and Ján Mazák. On covering cubic graphs with three perfect matchings. *unpublished*.

[12] Edita Máčajová and Martin Škoviera. Fano colourings of cubic graphs and the Fulkerson conjecture. *Theoretical computer science*, 349(1):112–120, 2005.

[13] Edita Máčajová and Martin Škoviera. On a conjecture of Fan and Raspaud. *Electronic Notes in Discrete Mathematics*, 34:237–241, 2009.

[14] Giuseppe Mazzuoccolo. The equivalence of two conjectures of Berge and Fulkerson. *Journal of Graph Theory*, 68(2):125–128, 2011.

[15] Giuseppe Mazzuoccolo. Covering a cubic graph with perfect matchings. *Discrete Mathematics*, 313(20):2292–2296, 2013.

[16] Giuseppe Mazzuoccolo. New conjectures on perfect matchings in cubic graphs. *Electronic Notes in Discrete Mathematics*, 40:235–238, 2013.

[17] Giuseppe Mazzuoccolo and Jean Paul Zerafa. An equivalent formulation of the Fan-Raspaud conjecture and related problems. *arXiv preprint arXiv:1811.08363*, 2018.

[18] Vahan V Mkrtchyan and Gagik N Vardanyan. On two consequences of Berge-Fulkerson conjecture. *AKCE International Journal of Graphs and Combinatorics*, 2019.

[19] Viresh Patel. Unions of perfect matchings in cubic graphs and implications of the Berge-Fulkerson conjecture. Technical report, Citeseer, 2006.

[20] Robert W. Robinson and Nicholas C. Wormald. Almost all cubic graphs are hamiltonian. *Random Structures & Algorithms*, 3(2):117–125, 1992.

[21] PD Seymour. Some unsolved problems on one-factorizations of graphs. *Graph Theory and Related Topics. Academic Press, New York*, 1979.