Bachelor Thesis

Rainbow Matchings in Color-Spanned Graphs

Aaron Neugebauer

Date of Submission: July 4, 2022
Advisor: Prof. Dr. Alexander Wolff

Julius-Maximilians-Universität Würzburg
Lehrstuhl für Informatik I
Algorithmen und Komplexität

Abstract

Recent progressions in quantum physics provide new questions about perfect matchings and graph coloring. One such problem is about monochromatic weighted graphs and for which number of vertices in a graph how many monochromatic vertex colorings exist. In this work, we adapt this problem to provide an upper bound of the number of monochromatic vertex colorings. The adaptation is concerned with for which graph, every edge coloring provides a perfect rainbow matching. After introducing the problem, we propose multiple procedures including a brute-force approach, an integer linear program and SAT formulations to investigate the amount of monochromatic vertex coloring. Also, we present some statements to simplify the search for graphs containing rainbow matchings through isomorphic classes and case analysis's.

Zusammenfassung

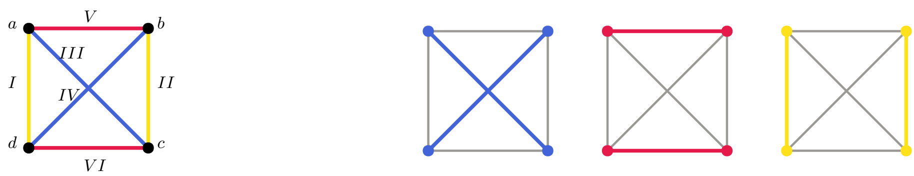
Durch neue Fortschritte in der Quantenphysik wurden neue Fragen aufgeworfen bezüglich perfekten rainbow matchings und Graphenfärbungen. Eines dieser Probleme befasst sich mit monochromatisch gewichteten Graphen und für welche Anzahl an Knoten wie viele monochromatischen Knotenfärbungen es gibt. In dieser Arbeit formen wir dieses Problem um, um eine obere Grenze für die Anzahl der monochromatischen Knotenfärbungen anzugeben. Die Umformung beschäftigt sich damit, für welche Graphen jede spannende Färbung ein perfektes rainbow matching enthält. Nachdem wir das Problem eingeführt haben, präsentieren wir verschiedene Vorgehensweisen, um dieses Problem zu berechnen für verschiedene Graphen. Zu diesen Vorgehensweisen gehören ein Brut-Force Algorithmus, ein Integer lineares Programm und SAT Formulierungen. Auch präsentieren wir Aussagen zum Vereinfachen der Suche nach Graphen, welche perfekte rainbow matchings besitzen.

Contents

1 Introduction

Due to the progress in quantum physics, new questions concerning graph theory arose [KGZ17, GEZK19, GCZK19]. Trying to predict photons in a quantum experiment, Krenn et al. [KGS19] formulated a problem about perfect matchings and graph coloring. This conjecture allows not only to anticipate an outcome of a photonic quantum experiment through graph theory, but such experiments could also predict properties of certain graphs.

In particular, the problem is about *monochromatic weighted* graphs and how many *vertex colorings* they admit. Consider a edge colored, weighted graph $G = (V, E)$. A perfect matching m of G consists of edges of G such that every vertex is incident to exactly one edge. A perfect matching m admits a vertex coloring τ, where we color each vertex depending on the color of its only incident edge in the perfect matching m. Such a vertex coloring is *monochromatic* if all vertices have the same color. The *weight* $\omega(\tau)$ of a vertex coloring τ is defined as the sum over the weight of all perfect matchings with the vertex coloring τ. Here, the weight of a single matching with the vertex coloring τ is the product of the edge weights. A colored, weighted graph is monochromatic weighted if the weight of a monochromatic vertex coloring is one and the weight of non-monochromatic vertex coloring is zero. For example, the graph G shown in Figure 1.1a has three different monochromatic vertex colorings as seen in Figure 1.1b. Since each vertex coloring is induced by exactly one matching, whose edges have all weight unit weight, the weight of each vertex coloring is one. Therefore, the graph G is monochromatic weighted. Currently, physicist are interested in graphs with an even number of vertices and complex weights.



(a) This graph G with unit edge weight is monochromatic weighted.

(b) Matchings of the graph G in Figure 1.1a with vertex coloring.

Fig. 1.1: Example for a monochromatic weighted graph G in Figure a with its vertex colorings admitted from the perfect matchings as seen in Figure b.

Until now, the graph in Figure 1.1a is the only known graph with three different vertex colorings. Bogdanov [Bog17] showed that for real edge weights greater than zero, there are no more than three different vertex colorings in a graph with four vertices and, for graphs with an even number of vertices greater than four, there are no more than two different vertex colorings. This result not only provides a lower bound for the number of vertex colorings, but also inspired Krenn to postulate the following conjecture.

Conjecture 1 ([KGZ17]). *Let G be a graph, and let n be the number of vertices of G. If $n = 4$ then there are at most three different monochromatic vertex colorings. If $n > 4$ and n is even, then the maximum number of different monochromatic vertex colorings is two.*

Theoretical approaches to solve the problem were made by Bogdanov [Bog17] and recently by Chandran and Gajjala [CG22], who proved for graphs without parallel edges, with an even number n of vertices greater equal than six and uniquely colored edges that the graph can't have $n - 3$ or more monochromatic vertex colorings. Even with advanced SAT formulations supported by artificial intelligence, Cervera-Lierta et al. [CKA21] could only show that monochromatic weighted graphs for six vertices colored in more than two colors do not exist. Similar, monochromatic weighted graphs with eight vertices colored in more than three colors do not exist.

1.1 Connection to Quantum Physics

The implications for the world of quantum physics are connected to the following experimental setup in Figure 1.2. The boxes I and II either produce two or no photons, which will travel on the black paths through crystals indicated by the boxes III, ..., VI. These crystals manipulate the photons in certain ways. At the end of each path, the photons can collide with the detectors a to d, which detect whether a photon hits it and read some intrinsic properties of the photon. Due to the effects of quantum interference, photons can cancel out each other. Physicist are interested in the case where each detector recognizes a photon. It is also important which proton which detector hits. Since it is not possible to read the mode number of a photon, which reveals information about the source of the photon, we only obtain a superposition of various possibilities. This possibility distribution contains information from which source the photons probably originate by assigning each superposition of possible photon sources a probability. Phase shifters and modified crystals can be used to change the mode number of a photon and therefore the probability distribution of the superposition.

An experimental setup can be expressed in terms of graph theory, using the analogies from the Table 1.1. Applying these analogies to the setup in Figure 1.2 results in the graph G in Figure 1.1a. When the resulting graph is monochromatic weighted, the photons are in a special state named Greenberg–Horne–Zeilinger (GHZ) state, which Krenn et al. are interested in. At foremost, the interest lies in how to generate such GHZ states [DMZ89] and their high-dimensional generalizations [MMMA18]. The dimension of entangled photons describes how much information can be stored. For example, 2-dimensional photons can either be in state 0 or in state 1 and therefore can hold one bit of information.

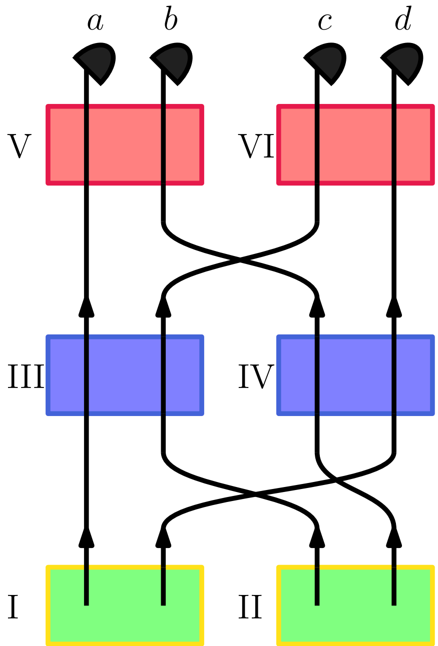


Fig. 1.2: Optical setup to generate 3-dimensional 4-photon GHZ states with the method of entanglement by path identity [KHLZ17, KGZ17]. Each box represents a crystal and the black lines depicts the path of photons. Box I and II will produce two entangled photons. All four photons are in a special entangled state when they hit the detectors a to d.

By answering this problem, Krenn et al. hope to gain new insights into the potential of the quantum interference since the examined phenomena has applications in spectroscopy [KPKK16], quantum imaging [LBC+14], the investigating of complementarity [HMM15], in superconducting cavities [LPHH16] and examining quantum correlation [HLL+17].

This problem leads to a new question in the field of graph theory. Krenn et al. [KGS19] published many additional questions regarding this field with implications for quantum physics.

1.2 Own Contribution

First, we formalize the problem about finding monochromatic vertex colorings; see Chapter 2. Then we show how to rephrase the problem into one concerning *rainbow matchings* in *color spanning graphs* to give an upper bound on k such that Conjecture 1 holds; see Chapter 2. In addition, we will recall results concerning rainbow matchings in color spanning graphs.

To calculate for which k rainbow matching always exists for a specific given graph with a spanning $(n - k)$-coloring, we will implement a simple brute-force algorithm and analyze its runtime and effectiveness; see Chapter 3.

Tab. 1.1: Analogies between the experimental setup and graph theory.

Quantum experiment	Graph theory
optical setup with crystals	undirected graph $G = (V, E)$
crystals	edges set E
optical path	vertices set V
mode number of photon	edge color
n photons entangled	perfect matching
number of possible n entangled photons	number of perfect matchings
largest dimension of photons	vertex degree
n-photon d-dimensional GHZ state	n-vertex graph with d disjoint perfect matchings

Next, we examine how to encode the problem as a *mathematical program* (MP). As we will show in Chapter 4.2, this MP formulation can be *linearized*, which allows us to use an ILP solver to test if a graph has a spanning $(n - k)$-coloring without a rainbow matching for a given graph. In particular, we examined with IBM's ILOG CPLEX Optimizer the viability of this approach to solve the ILP formulation.

Finally, we introduce a SAT formulation to search for spanning colorings for graphs without rainbow matchings; see Chapter 5.1. By allowing edges to remain uncolored, it is possible to generalize the SAT formulation to examine all subgraphs of a complete graph of a given size; see Chapter 5.2. Sometimes, it is possible to recursively simplify the examined graph. In this case, it remains to check multiple smaller problem instances; see Chapter 5.3.

We present some experimental results. For graphs with eight vertices, we compare Cervera-Lierta et al. [CKA21] results to ours and try to give a new upper bound for graphs with ten vertices. In addition, we compare the runtimes of different variants of our SAT formulation.

2 Improving the Upper Bound

Recall that Cervera-Lierta et al. [CKA21] have tried to find possible configurations for monochromatic weighted graphs. This shows that even for graphs with few vertices, the problem is computationally difficult. Due to this reason, we introduce an approach to compare an upper bound for the number of different vertex colorings in a monochromatic weighted graph. First, however, we formalize the definitions for vertex coloring, monochromatic weighted graphs and spanning coloring given in Chapter 1.

Definition 2 (Vertex Coloring). *Consider a graph $G = (V, E)$ with a perfect matching m and an edge coloring $\alpha\colon E \to \mathcal{F}$, where $\mathcal{F} = \{1, 2, \ldots, f\}$. Then $\tau\colon V \to \mathcal{F}$ is a* vertex coloring *with respect to the perfect matching m if $\tau(v) = \alpha(e)$, where $e \in m$ is the unique edge incident to v. If all vertices are colored the same, then τ is* monochromatic.

Definition 3 (Induced Vertex Coloring). *Any coloring α of a perfect matching m of a graph induces a (unique) coloring $\hat{\alpha}$ of the vertices of the graph such that $\hat{\alpha}(v) = \hat{\alpha}(u) = \alpha(vu)$ for each edge vu of m.*

Definition 4 (Monochromatic Weighted Graph). *Consider a graph $G = (V, E)$ that admits a perfect matching, with complex edge weights $\omega\colon E \to \mathbb{C}$ and edge coloring $\alpha\colon E \to \mathcal{F}$. Let $\mathcal{M}$ be the set of all perfect matchings of G, and, for a given vertex coloring $\tau\colon V \to \mathcal{F}$, let $\mathcal{M}(\tau) = \{m \in \mathcal{M}\colon\ \tau \text{ is a vertex coloring with respect to } m\}$ be the set of all perfect matchings that induce τ. Let*

$$\omega(\tau) = \sum_{m \in \mathcal{M}(\tau)} \prod_{e \in m} \omega_e$$

be the weight of τ. If, for every monochromatic vertex coloring τ, $\omega(\tau) = 1$ and, for every non-monochromatic vertex coloring $\bar{\tau}$, $\omega(\bar{\tau}) = 0$, then G is a monochromatic weighted graph.

Note, that the definition of monochromatic weighted graphs given by Krenn [KGS19] does not prohibit edges, which have the same end vertices. Such edges are called parallel. Since we are only interested in graphs without parallel edges, we assume, that all monochromatic weighted graphs do not contain parallel edges.

Definition 5 (Spanning Coloring). *Let $G = (V, E)$ be a graph and $\alpha\colon E \to \mathcal{F}$ be a coloring of the graph. Here $\mathcal{F}$ is the set of colors. A coloring α is spanning for a graph G if, for every vertex $v \in V$ and for every color $f \in \mathcal{F}$, there exists an edge incident to v with color f.*

In Conjecture 1 Krenn [KGZ17] made a statement about how many monochromatic vertex colorings a monochromatic weighted graph has. Let $\mathcal{G}_n$ be the family of graphs with n vertices that are monochromatic weighted. For $G \in \mathcal{G}_n$, let $C(G)$ be the number of monochromatic vertex colorings of G. For any even positive integer n, let

$$C(n) = \max_{G \in \mathcal{G}_n} C(G)$$

be the maximum of $C(G)$ over all graphs G in $\mathcal{G}_n$. As previously mentioned, Bogdanov [Bog17] showed that $C(4) = 3$ and that, for every even number $n \geq 6$ and graphs with *positive real edge weights*, $C(n) = 2$. It follows that $C(n) \geq 2$ for arbitrary edge weights.

This leaves the question of an upper bound for $C(n)$ for arbitrary edge weights, which we investigate with the help of two claims proposed by Ravsky [Rav21a, Rav21b]. First, we introduce each claim and show how it bounds $C(n)$ for a given n. Secondly, we show which upper bounds can be verified through theoretical approaches.

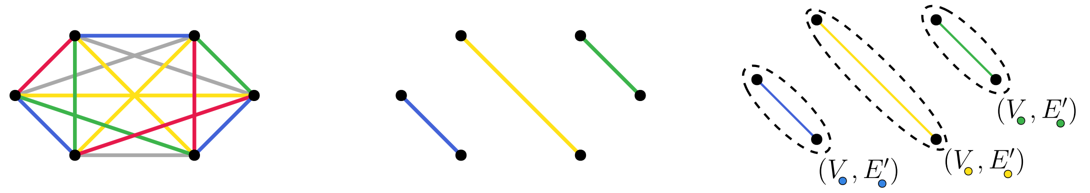
2.1 First Bound of C(n)

Ravksy's [Rav21b] first claim asks for a colored perfect matching in partially colored complete graph with an even number n of vertices and $n - k$ colors, whose coloring is spanning. Furthermore, the perfect matching must not only have one color and, for each color $f \in \{1, 2, \dots, n-k\}$, the graph, containing all vertices, which are incident to the matching and the color class of f, and all edges, which are uncolored or have color f, between the vertices, needs to contain a unique perfect matching. We want to know which number of vertices n and number of colors $n - k$ satisfy this claim. Mainly, we want to know the largest k for which Claim 6 holds. We define $k'(n)$ as the largest k for which Claim 6 holds.

Claim 6 ([Rav21b])**.** *Consider the graph $G = (V, E) = K_n$. Let $\alpha\colon E \setminus R \to \mathcal{F} = \{1, 2, \dots, n-k\}$ be a $(n-k)$-spanning coloring of $(V, E \setminus R)$, where $R \subset E$ is a set of uncolored edges of G. For any $f \in \mathcal{F}$, let $E_f = \{e \in E\colon \alpha(e) = f\}$ be the edges of G with color f. Then there exists a colored perfect matching $m \subseteq E \setminus R$ of G and only colored edges such that for each $f \in \mathcal{F}$ with $V_f \neq \varnothing$, the graph (V_f, E'_f) has a unique perfect matching. Here V_f is the set of vertices of G incident to edges in $m \cap E_f$ and E'_f is the set of edges of $E_f \cup R$ with both end vertices in V_f.*

For example, consider the graph G in Figure 2.1a with $n = 6$ and $k = 2$. Here, the four color classes are the sets $E_1, \dots E_4$. Let m be the perfect matching in Figure 2.1b. Then, for every color $f \in \mathcal{F}$, $m \nsubseteq E_f$ and therefore all edges do not have the same color. Claim 6 holds for $k = 2$, because every graph (V_f, E'_f) admits a unique perfect matching as in Figure 2.1c. Note that $V_{\bullet} = \emptyset$, because $m \cap E_{\bullet} = \emptyset$, and therefore $(V_{\bullet}, E'_{\bullet})$ is an empty graph, whose perfect matching is also unique. So, in this case, $k'(6) \geq 2$.

To get a better understanding of the graphs (V_f, E'_f), consider the graph G in Figure 2.2a. For the perfect matching m in Figure 2.2b the graph $(V_{\circ}, E'_{\circ})$ in Figure 2.2c has

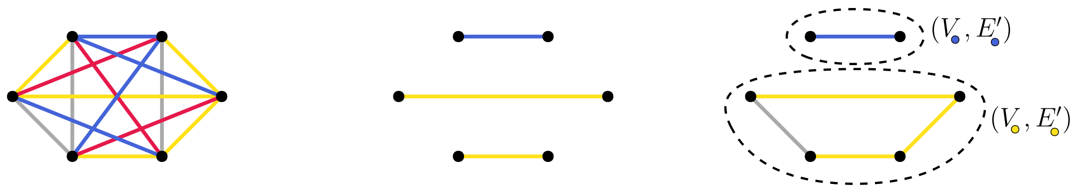


(a) Graph G with a spanning 4-coloring in colors ●, ●, ●, ● and uncolored edges ●.

(b) Perfect matching m of graph G in Figure a satisfying Claim 6.

(c) Resulting graphs (V_f, E'_f) of perfect matching m in Figure b.

Fig. 2.1: Example of a graph with unique perfect matchings in graphs (V_f, E'_f), for $f \in \{1, \ldots, n-k\}$.



(a) Graph G with a spanning 3-coloring in colors ●, ●, ● and uncolored edges in ●.

(b) Example perfect matching m of graph G in Figure a.

(c) Resulting graphs (V_f, E'_f) of perfect matching m in Figure b.

Fig. 2.2: Example for a graph with multiple perfect matchings in (V_f, E'_f).

two perfect matchings. This means that m is not a candidate for the perfect matching, whose existence Claim 6 guarantees.

We now want to show the connection between Claim 6 and Conjecture 1 from Capter 1. Recall that Conjecture 1 stats a graph with four vertices has at most three different monochromatic vertex collorings and every graph with a even number of vertices greater than four has at most two different monochromatic vertex collorings. To connect Claim 6 and Conjecture 1, we need to introduce the definition of *contributing* edges for a color of a graph. Ravsky [Rav22] shows that an edge can only be contributing to one color, which helps to prove the connection.

Definition 7 (Contributing Edge)**.** *Let $n \geq 4$ be an even number, and let G be a monochromatic weighted graph with n vertices and colors $\mathcal{F}$. For an edge $e = (u, v)$ of G and color $c \in \mathcal{F}$, let $\mathcal{M}_e^c$ be the set of all c-monochromatic perfect matchings of the induced graph $G \setminus \{u, v\}$. Let*

$$W_e^c = \sum_{m \in \mathcal{M}_e^c} \prod_{e \in m} \omega_e^c$$

be the weight of $\mathcal{M}_e^c$, where ω_e^c is the weight of an edge e of G, when considering that e has color c. An edge e of G is contributing *for a color c if both ω_e^c and W_e^c are non-zero.*

Lemma 8 ([Rav22])**.** *Any edge e of a monochromatic graph is contributing for at most one color.*

Proof. Suppose for a contradiction that an edge e of a monochromatic weighted graph is contributing for distinct colors c and d. Let τ be a vertex coloring of the graph such that the end vertices of e are colored in c and the remaining vertices are colored in d. The definition of the contributing edge implies that both ω_e^c and W_e^d are non-zero, so $\omega(\tau) = \omega_e^c W_e^d \neq 0$, a contradiction. $\square$

Lemma 9. *If $G = (V, E)$ is a monochromatic graph with at least 4 vertices then for each vertex v and each color c of G there exists an edge of G incident to v and contributing for c.*

Proof. Let $\tau : V \to \{c\}$ be a constant vertex coloring. It is easy to see that $0 \neq \omega(\tau) = \sum_{u \in V \setminus \{v\}} \omega_{(v,u)}^c W_{(v,u)}^c$. Then one of the summands $\omega_{(v,u)}^c W_{(v,u)}^c$ is non-zero, that is the edge (v, u) is contributing for c. $\square$

Via the following Proposition 10, proposed by Ravsky [Rav22], we will connect the graphs in Claim 6 and monochromatic weighted graphs. The connection is made with the values $C(n)$ and $k'(n)$.

Proposition 10 ([Rav21b])**.** *For any even $n \geq 6$, holds $C(n) < n - k'(n)$.*

Proof. Suppose for a contradiction that there exists a monochromatic graph (V, E^*) with n vertices whose edges are colored in colors from a set F such that $|F| = n - k'(n)$. Construct a set $\hat{E} \subseteq E^*$ and an $(n - k')$-spanning coloring of $(V, \hat{E})$ of colors of F as follows. Let $c \in F$ be any color. For each vertex v of V, applying Lemma 9, we pick an edge $e_{v,c} \in E^*$ which is incident to v and contributing for c. Put $\hat{E}_c = \{e_{v,c} : v \in V, c \in C\}$. Note that the set $\hat{E}_c$ of edges is spanning. Finally put $\hat{E} = \bigcup_{c \in F} \hat{E}_c$. By Lemma 2.1, the sets $\hat{E}_c$ and $\hat{E}_d$ are disjoint for any distinct $c, d \in F$, so the edge coloring of $\hat{E}$ which assigns a color c to each edge $e \in \hat{E}_c$, is correctly defined.

Now let $G = (V, E)$ be a complete graph. Put $R = E \setminus \hat{E}$. Now, using Claim 6 and its notation, pick a perfect matching $m \subseteq \hat{E}$ such that for each $c \in F$ with $V_c \neq \varnothing$, the graph (V_c, E'_c) has a unique perfect matching. Let τ be the coloring of V induced by m (and so non-monochromatic). Then $\omega(m) = \prod_{c \in F} W_c^{V_c}$ (if $V_f = \varnothing$ then we set $W_c^{V_c} = 1$). Since for each $c \in F$ with $V_c \neq \varnothing$, the graph (V_c, E'_c) has a unique perfect matching (which is necessarily $m \cap E'_c$), the graph on the vertex set V_c whose edges are colored in c, has (the same) unique perfect matching. Thus $W_c^{V_c} = \prod_{e \in m \cap E'_f} \omega_e^c \neq 0$, since each edge of $m \cap E'_c$ is contributing for the color c and so $\omega_e^c \neq 0$. Then $\omega(\tau) \neq 0$, a contradiction. $\square$

2.2 Second Limitation of the Upper Bound

Since calculating the value $k'(n)$ is still complicated, we will introduce Claim 11 to give $k'(n)$ a lower bound, as we will show in Lemma 12. For that purpose, we introduce $k(n)$ as the largest $k < n$ for which Claim 11 holds.

An edge coloring of a graph G is *spanning*, if each color class is spanning. A *rainbow matching* is a matching where every edge has a different color.

Claim 11. *Let $k < n$ and G be a graph on n vertices with a spanning $(n-k)$-edge coloring. Then G has a perfect rainbow matching.*

Recall the graph F in Figure 2.1a. As a matter of fact, the graph has a prefect rainbow matching as seen in Figure 2.1b. The edges, which are all colored differently, are a perfect matching. Therefore, this graph satisfies Claim 11 for this coloring. For an example where Claim 11 does not hold, consider the graph in Figure 2.3. There does not exist a perfect matching where all edges are colored differently. Hence, $k(6) < 6$.

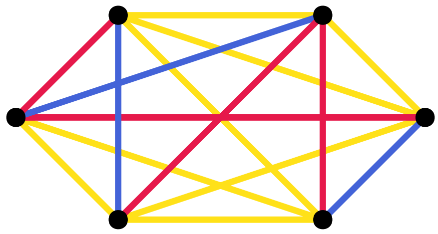


Fig. 2.3: Graph with six vertices with a spanning 3 edge coloring in colors ●, ●, ●, which does not contain a perfect rainbow matching.

With the help of Lemma 12 we will show that $k(n)$ of Claim 11 is a lower bound of $k'(n)$. Afterwards, we can show the connection between Claim 11 and the Conjecture 1 with Corollary 13.

Lemma 12 ([Rav21a])**.** *For even $n \geq 6$, $k(n) \leq k'(n)$.*

Proof. We will show that for any given even $n \geq 6$, Claim 6 holds if Claim 11 holds. Thus, $k(n) \leq k'(n)$.

Consider a graph $\tilde{G} = (\tilde{V}, \tilde{E})$ with n vertices and a $(n-k)-$coloring α, where Claim 11 holds. Let $\tilde{m}$ be a perfect rainbow matching of G. Due to the spanning coloring of $\tilde{G}$, there exist spanning pairwise edge disjoint subgraphs $(V, E_1), \ldots, (V, E_{n-k})$ of $G = (V, E) = K_n$ such that for every $f \in \{1, \ldots, n-k\}$, E_f is the set of all edges $e \in E$ with color f. Therefore, $R = E \setminus \tilde{E}$ is the set of uncolored edges in G. Since there exists a rainbow matching $\tilde{m}$ in $\tilde{G}$, there exists the perfect matching m in G where, for every $f \in \{1, \ldots, n-k\}$, there is at most one edge of m in E_f. Thus, $m \nsubseteq E_f$, for all $f \in \{1, \ldots, n-k\}$, because $(V, E_1), \ldots, (V, E_{n-k})$ are edge disjoint. For every $f \in \{1, \ldots, n-k\}$, $|V_f| \in \{0, 2\}$, since $m \cap E_f$ contains a maximum of one edge. So, there is a maximum of one edge in $E_f \cup R$ with both end vertices in V_f. It follows that every subgraph (V_f, E'_f) consists of two or none vertices and one or none edges like in Figure 2.1c, which has a unique perfect matching. □

Corollary 13. *For even $n \geq 6$, $C(n) < n - k(n)$.*

Proof. Follows from Propositions 10 and 12 since $C(n) < n - k'(n) \leq n - k(n)$. □

2.3 Theoretical Achievements

Before we start using a computer to calculate the value of $k(n)$ in Chapter 3 and forwards, theoretical results can limit the range of $k(n)$. At first, we present a simple observation regarding Claim 11, which give us an upper bound for $k(n)$.

Lemma 14. *Let $n \geq 6$ be even. Then $k(n) \leq n/2$.*

Proof. Consider any graph G with n vertices, where there exists a spanning ℓ-coloring with $\ell < n - n/2 = n/2$. Then a perfect rainbow matching consists of $n/2$ edges colored in different colors. With the pigeonhole principle, it follows that there is no perfect rainbow matching, since we have fewer colors than edges in a perfect matching. □

A better bound can be achieved with the following Lemma 15 of Ravsky [Rav21a].

Lemma 15 ([Rav21a])**.** *Let $n \geq 6$ be even and $n' = n/2$. Then $k(n) \leq n' - 2$, if n' is even, and $k(n) \leq n' - 1$, if n' is odd.*

Proof. Let $G = (V, E)$ be a complete graph with vertices $v_1, \ldots, v_{n'}, u_1, \ldots, u_{n'}$. Consider the following edge coloring of G. For each $1 \leq i, j \leq n'$, the edge $v_i u_j$ has the color $i + j \mod n'$ and all other edges of G are colored in n', see Table 2.1. It is easy to see that the constructed coloring is spanning. For example, see Figure 2.4. For $n' = 3$, this graph's coloring is spanning.

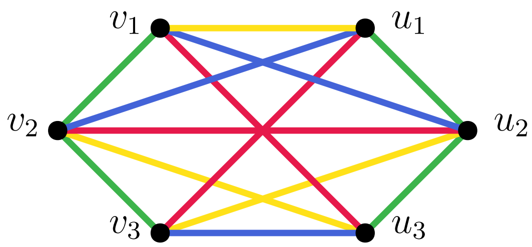


Fig. 2.4: Graph with vertices $v_1, v_2, v_3, u_1, u_2, u_3$, where, for $1 \leq i, j \leq 3$ edge v_i, u_j has color $i + j \mod 3$ and every other edge has color 4. Here the color ● is 0, ● is 1, ● is 3 and ● is 4.

To obtain a perfect rainbow matching, we need to choose n' entries such that each column and row is only chosen once.

Let G' be the bipartite subgraph of G with the parts $\{v_1, \ldots, v_{n'}\}$ and $\{u_1, \ldots, u_{n'}\}$. If n' is even, Aharoni et al. [ABKZ17] showed that G' as in Figure 2.2 with the induced edge-coloring has no rainbow perfect matching. If n' is odd then to assure absence of such matching we remove from G all edges of color 0.

Suppose for a contradiction that G has a rainbow perfect matching m. By the above, m has at most $n' - 1$ edges of G'. But this bound is never tight, because if m has exactly $n' - 1$ edges of G' then the remaining edge of m also belongs to G'. Thus m has at most $n' - 2$ edges of G'. But then the remaining two edges of m has to be colored in n', a contradiction. □

Tab. 2.1: For $r, s \in \{v_1, \ldots, v_{n'}, u_1, \ldots, u_{n'}\}$, the entry (r, s) in the table contains the color of the edge rs in the graph G.

	v_1	v_2	$\cdots$	$v_{n'-1}$	$v_{n'}$	u_1	u_2	$\cdots$	$u_{n'-1}$	$u_{n'}$
v_1	n'		$\cdots$		n'	2	3	$\cdots$	0	1
v_2	n'		$\cdots$		n'	3	4	$\cdots$	1	2
$\vdots$	$\vdots$				$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
$v_{n'}$	n'		$\cdots$		n'	1	2	$\cdots$	$n'-1$	0
u_1	2	3	$\cdots$	0	1	n'		$\cdots$		n'
u_2	3	4	$\cdots$	1	2	n'		$\cdots$		n'
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$				$\vdots$
$u_{n'}$	1	2	$\cdots$	$n'-1$	0	n'		$\cdots$		n'

Tab. 2.2: The coloring of the sub-graph G' with only edges between $v_1, \ldots, v_{n'}$ and $u_1, \ldots, u_{n'}$, which does not contain a rainbow matching.

	u_1	u_2	$\cdots$	$u_{n'-1}$	$u_{n'}$
v_1	2	3	$\cdots$	0	1
v_2	3	4	$\cdots$	1	2
$\vdots$	$\vdots$		$\vdots$	$\vdots$	
$v_{n'}$	1	2	$\cdots$	$n'-1$	0

Furthermore, Kostochka and Yancy [KY12] proved that an edge-colored graph G, where every vertex is incident to at least ℓ distinct colors, then G has a rainbow matching of size $\lceil \ell/2 \rceil$. If $\ell = n-1$, where an even $n \geq 6$ is the number of vertices of G, then there exists a perfect rainbow matching, since $\lceil \ell/2 \rceil = n/2$. Therefore, the next Corollary follows.

Corollary 16. *Let $n \geq 6$ be even. Then $k(n) \geq 1$. Hence, $k'(n) \geq 1$.*

Proof. The first part follows from the reasoning above. The second part follows directly from Lemma 12. □

3 A Brute-Force Approach

Since further theoretical progress on Conjecture 1 turns out to be rather difficult even for small numbers n, as seen by the advanced by Chandran and Gajjala [CG22], one can try to verify the Conjecture 1 via computers for small n. Recall that Cervera-Lierta et al. [CKA21] already made some limitations for $n = 6$ and $n = 8$. Here they tackled Conjecture 1 itself.

We, on the other hand, will try to improve the upper bound of $C(n)$ with Claim 11 and Corollary 13. By searching for counterexamples for Claim 11 for a given n and k, we try to reduce the upper bound of $C(n)$. A counterexample is a spanning $(n-k)$-coloring for a given graph $G = (V, E)$ for which no perfect rainbow matching exists.

For now, we will introduce a brute-force algorithm to search for such counterexamples. Let $G = (V, E)$ be a graph with n vertices, where $n \geq 6$ is even. Here $\mathcal{F} = \{1, \ldots, n-k\}$ is a set of colors and $\alpha\colon E \to \mathcal{F}$ is a coloring of the edges in G. As previously introduced, $\mathcal{M}$ contains all perfect matchings of G and, for every perfect matching $m \in \mathcal{M}$, m contains $n/2$ elements. The algorithm is as follows.

Algorithm 1: Brute-Force Algorithm(G, n, k)

Input: Graph $G = (V, E)$ with n vertices, where $n \geq 6$ is even, $k < n$.
Output: Coloring α if counterexample exist, else NO.

$\mathcal{F} = \{1, \ldots, n-k\}$
for all $\alpha\colon E \to \mathcal{F}$ **do**
    if is_spanning_coloring(G, α) **and** has_rainbow_matching(G, α) **then**
        return α;
return FALSE;

We will now show that Algorithm 1 is correct. For every coloring $\alpha\colon E \to \mathcal{F}$, we test if it is a spanning coloring with Algorithm 2 and does not contain a perfect rainbow matching with Algorithm 3. If the test is successful, the given graph with the coloring α is a desired counterexample, otherwise we keep searching. When all colorings fail the test, there does not exist a counterexample and thus Claim 11 holds for the given graph G, n and k. This depends on the correctness of Algorithm 2 and Algorithm 3. Algorithm 2 returns only FALSE if, for a given coloring α, there exists a vertex, where the set of colors from the incident edges does not contain all colors in $\mathcal{F}$. Therefore, the coloring α would not be spanning, and the Algorithm 2 is correct. If in Algorithm 3 exists two edges e, e' in the given perfect matching m with the same color, either e or e' would set $colors[\alpha(e)]$ to 1. The later edge would then break the inner loop, which hinders the algorithm to return TRUE. Thus, Algorithm 3 returns TRUE if and only if

Algorithm 2: is_spanning_coloring(G, α)

Input: Graph $G = (V, E)$, coloring $\alpha\colon E \to \mathcal{F}$.

Output: TRUE if α is spanning coloring of the graph G, else FALSE.

for $v \in V$ **do**
    $colors \leftarrow \emptyset$
    for $u \in N(v)$ **do**
        $colors \leftarrow colors \cup \{\alpha(uv)\}$
    if $colors$ is not $\mathcal{F}$ **then**
        return FALSE
return TRUE

Algorithm 3: has_rainbow_matching(G, α)

Input: Graph $G = (V, E)$, coloring $\alpha\colon E \to \mathcal{F}$.

Output: TRUE if graph G with coloring α has a perfect rainbow matching, else FALSE.

for $m = \{e_1, \ldots, e_{|E|/2}\} \in \mathcal{M}$ **do**
    $colors$ is Boolean array of size $|\mathcal{F}|$.
    for $e \in m$ **do**
        if $colors[\alpha(e)]$ **then**
            break
        else
            $colors[\alpha(e)] \leftarrow 1$
    return TRUE
return FALSE

there exists a perfect matching, where every edge has a different coloring inherited from α. In other words, a perfect rainbow matching. With the correctness of Algorithm 2 and Algorithm 3, the Algorithm 1 is now also correct.

To get a grasp on the runtime of Algorithm 1, we will calculate its worst-case runtime for $G = K_n$ and $k = 1$. In Algorithm 2 we iterate over V and $N(v)$ for a given $v \in V$. Therefore, this results in a runtime of $\mathcal{O}(|V|^2)$. Note that $\mathcal{M}$ contains $(n-1)!! = (n-1)(n-3)\ldots 3 \cdot 1$ elements. Assume that the inner for-loop of Algorithm 3 always breaks when processing the last edge and every matching has at least two edges with same colors, then the runtime results to $\mathcal{O}((n-1)!!\,|E|/2)$. When Algorithm 1 needs to check each condition for every coloring, the runtime is $\mathcal{O}(|\mathcal{F}|^{|E|}\,|V|^2\,(n-1)!!\,|E|/2) = \mathcal{O}(|V|^{|E|}\,|V|^2\,(n-1)!!\,|E|/2)$ in the worst case, since $\mathcal{O}(\mathcal{F}) = \mathcal{O}(n-k) = \mathcal{O}(|V|)$, because of Lemma 14. If we assume a processor runs with 10Ghz, then we need about $3.7 \cdot 10^{12}$ years for $n = 8$ and $6.9 \cdot 10^{33}$ years for $n = 10$ in the worst case. Such a runtime is not reasonable, and we need another approach.

4 An Integer Linear Program

As we have seen in Chapter 3, that a normal brute-force algorithm is not feasible. We need a better method to search for counterexamples. An alternative is an *integer linear program* (ILP). Even though, binary integer linear programming is one of 21 Karp's NP-complete problems, there are many good solvers available for such problems. Therefore, we will encode the problem in Claim 11 as an ILP and see in this chapter wether we can calculate the value of $k(n)$ for small n. This would give new upper bounds for $C(n)$.

At first, we will model an ILP for Claim 11, which searches for counterexamples like in Chapter 3; see Section 4.1 and Section 4.2. In Section 4.3 we will present the results gathered from implementing the ILP.

4.1 Mathematical Program Formulation

Before we model an ILP, we introduce a *mathematical programming* (MP) formulation. Such MP are not necessarily linear, but can be *linearized*, which is the process of replacing non-linear constraints by equivalent linear constraints. In Section 4.2 we will show how to linearize the problematic expressions to turn the below program into an ILP.

As an input, we get a graph $G = (V, E)$ with an even number $n \geq 6$ of vertices and an integer $k < n$. Let $\mathcal{F} = \{1, \ldots, n-k\}$ be the set of all colors. Let $\mathcal{M}$ contain all perfect matchings of G. Each matching $m \in \mathcal{M}$ consists of $|V|/2$ independent edges $\{e_1, \ldots, e_{|V|/2}\}$. For any edge e, let $\alpha(e) \in \mathcal{F}$ be the color of e. For any edge e of G and for any $f \in \mathcal{F}$, the boolean variable $c_{e,f}$ indicates whether edge e has color f. We define $\mathcal{I}$ as the set $\{(i,j) \mid i,j \in \{1, \ldots, |V|/2\}, i \neq j\}$.

The MP is defined by the following constraints. First, we express that each edge has a single color:

$$\sum_{f \in \mathcal{F}} c_{e,f} = 1 \qquad \text{for each } e \in E.$$

Next, we force each color class to be spanning.

$$\sum_{u \in N(v)} c_{uv,f} \geq 1 \qquad \text{for each } v \in V \text{ and for each } f \in \mathcal{F}.$$

Then we connect $\alpha(e)$ and c_e, f for edges $e \in E$ and color $f \in \mathcal{F}$. This constraint will be linearized in Section 4.2.

$$\alpha(e) = f \Leftrightarrow c_{e,f} = 1 \qquad \text{for each } e \in E \text{ and for each } f \in \mathcal{F}.$$

Now we define the difference of colors for each pair of edges in a perfect matching. This will be used to test whether two edges have the same color in a matching. The absolute value is not linear, but we provide a method to linearize this function in Section 4.2

$$d_{m,i,j} := |\alpha(e_i) - \alpha(e_j)| \qquad \text{for each } m \in \mathcal{M} \text{ and for each } (i,j) \in \mathcal{I}.$$

Finally, we ensure that each matching contains at least two edges of the same color. In other words, the following restriction ensures that the given matching is not a rainbow matching.

$$\min_{(i,j)\in\mathcal{I}} d_{m,i,j} = 0 \qquad \text{for each } m \in \mathcal{M}.$$

The last constraint ensures that the matching is not a rainbow matching. The non-linearity of the minimum function will be addressed in Section 4.2.

4.2 Linearization of the Mathematical Program

Although the functions minimum, equivalence, and absolute value are not linear, there exist formulations to use them in an ILP. We now linearize the above-mentioned problematic functions. Thereby, we show that the mathematical formulation in Section 4.1 can be converted into a valid ILP.

Absolute value We want to calculate the absolute value y of the difference $x_1 - x_2$, where $x_1, x_2 \geq 0$. So $y = |x_1 - x_2|$. To achieve this, we need three additional variables d_1, d_2, and U. We lowerbound U by both x_1 and x_2. The boolean values d_1 and d_2 indicate the following.

$$\begin{aligned} &d_1 : 1 \text{ when } x_1 - x_2 \text{ is positive, otherwise } 0. \\ &d_2 : 1 \text{ when } x_2 - x_1 \text{ is positive, otherwise } 0. \end{aligned}$$

With this formulation we can calculate the absolute value y in an ILP.

$$\begin{aligned} &0 \leq x_i \leq U && \text{for each } i \in \{1,2\}; \\ &0 \leq y - (x_1 - x_2) \leq 2 \cdot U \cdot d_2; \\ &0 \leq y - (x_2 - x_1) \leq 2 \cdot U \cdot d_1; \\ &d_1 + d_2 = 1. \end{aligned}$$

Minimum To calculate the minimum y of a set $\{x_1, \ldots, x_n\}$ we need to know the upper and lower bounds U and L. In our case we know that $d_{m,i,j} \in \{0, \ldots, |V|/2\}$, so $U = |V|/2$ and $L = 0$. Here we also need additional boolean variables $d_1, \ldots, d_n$. Those indicate the following.

$$d_i : 1 \text{ when } x_i \text{ is the minimum, otherwise } 0 \qquad \text{for each } i \in \{1, \ldots, n\}$$

Now we can calculate the minimum in the following way in an ILP.

$$
\begin{aligned}
L \le x_i &\le U && \text{for each } i \in \{1,\dots,n\};\\
y &\le x_i && \text{for each } i \in \{1,\dots,n\};\\
y &\ge x_i - (U-L)(1-d_i) && \text{for each } i \in \{1,\dots,n\};\\
\sum_{i\in\{1,\dots,n\}} d_i &= 1.
\end{aligned}
$$

Equivalence Now we need to solve the equation $(a = b) \Leftrightarrow (c = d)$ in an ILP. At first, we need a boolean variable x to describe if $a = b$. For that, we can use the absolute and minimum function. Let $x = 1 - \min\{|a - b|, 1\}$. The same way, we define y in terms of c and d. Now we can linearize the equivalency via $x = y$ in an ILP.

4.3 ILP Results

With a valid ILP formulation, we can now investigate for which n and k Claim 11 holds. First, we will analyze Claim 11 for $n = 6$ and afterwards for $n = 8$. Throughout this section, we will be using the ILP solver IBM ILOG CPLEX Optimizer [IBM] version 22.1.0.0 on an AMD Ryzen 2500U CPU to gather the results. Also, IBM ILOG CPLEX Optimizer function to convert the MP into an ILP was used with the formulation in Section 4.1.

For $n = 6$ the following results were gathered. The graph in Figure 4.1 is the output for $k = 3$ after a runtime of 0.05 seconds. This graph does not contain any perfect rainbow matching and therefore is a counterexample as described in Chapter 3. Thus, Claim 11 does not hold for $n = 6$ and $k = 3$, as Lemma 15 states. With $k = 2$ and $k = 1$, after 4.81 and 0.95 seconds respectively, no counterexample was found and Claim 11 holds. It follows, that $k(6) = 2$ and $C(6) < 6 - 2 = 4$ with Corollary 13.

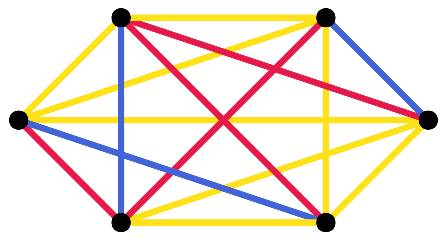
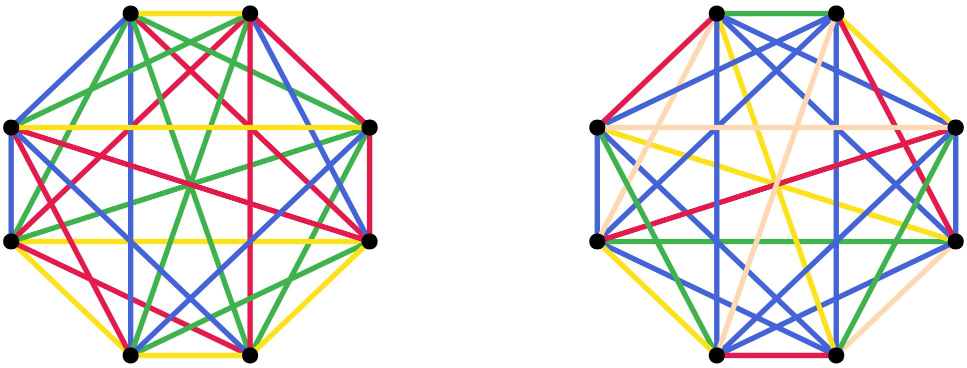


Fig. 4.1: Output graph of the ILP formulation for $n = 6$ and $k = 3$ with a spanning 3-coloring consisting of colors ●, ●, ●.

Consider $n = 8$. With $k = 4$, the ILP solver returned the counterexample depicted in Figure 4.2a after 2.59 seconds. For $k = 3$, the given counterexample can be seen in Figure 4.2b, which was produced after 30.99 seconds. So, Claim 11 does not hold for $n = 8$ and $k \in \{3, 4\}$. It follows that $k(8) \le 2$.

The ILP solver was not able to terminate when using $n = 8$ and $k \in \{1, 2\}$ in over a week of processing, even on stronger hardware, if a complete graph with n vertices has



(a) Output graph of the ILP formulation for $n = 8$ and $k = 4$ with a spanning 4-coloring consisting of colors ●, ●, ●, ●.

(b) Output graph of the ILP formulation for $n = 8$ and $k = 3$ with a spanning 5-coloring consisting of colors ●, ●, ●, ●, ●.

Fig. 4.2: Counterexamples to Claim 11 with eight vertices and $k = 3, 4$.

a spanning $(n - k)$-edge coloring. Namely, the hardware was an AMD Ryzen 3600. This leads to the conclusion, that our ILP implementation is not powerful enough to calculate $k(n)$. In the following Chapter 5 we will therefore try to find $k(n)$ with the help of SAT formulations.

5 A SAT Formulation

Since the ILP solver does not find counterexamples in a reasonable time, we encode the problem in Claim 11 into a *conjunctive normal form* (CNF) and solve it by checking its satisfiability. A CNF is a conjunction of one or more clauses consisting of disjunctions. Solving such SAT problems have made major advancements, as Fichte et al. [FHS20] showed. On the Conjecture 1 Cervera-Lierta et al. [CKA21] have gathered further knowledge by using SAT solvers. We hope to find counterexamples faster with a SAT solver compared to ILPs as some problems can be solved faster in this way as Brown et al. [BZG20] have investigated. In the following, we introduce two slightly different SAT formulations, which either has an advantage over the other, and a pattern to simplify certain cases.

5.1 SAT Formulation for Single Graphs

We want to search for a counterexample like in Chapter 3 for a given graph $G = (V, E)$ with n vertices and a given positive integer $k < n$. For this purpose we need to ensure that each edge is colored in a unique color, the coloring is spanning and no rainbow matching exists. We introduce a Boolean variable $c_{e,f}$ that is true if edge e has color f. Let $\mathcal{I} = \{\{i, j\} \mid i, j \in \{1, \dots, n/2\}, i \neq j\}$. Further variables will be introduced when required.

Unique coloring First, we need to ensure that each edge has a distinct color. By restricting all edges e to have at least one and at most one color $f \in \mathcal{F}$, we ensure that every edge is colored in exactly one color. It is rather simple to restrict every edge to have at least one color.

$$\bigwedge_{e \in E} \bigvee_{f \in \mathcal{F}} c_{e,f}$$

It is more difficult to restrict every edge to have at most one color. Using quantifiers, this can be written as follows.

$$\forall e \in E \; \exists f \in \mathcal{F} \; \forall f' \in \mathcal{F} \setminus \{f\} \colon \neg c_{e,f'}$$

We now replace the quantifiers by conjunctions and disjunctions.

$$\bigwedge_{e \in E} \bigvee_{f \in \mathcal{F}} \bigwedge_{f' \in \mathcal{F} \setminus \{f\}} \neg c_{e,f'}$$

Since the formulation is not in CNF, we need to isolate the last conjunction and solve it separately. To this end, we introduce a new variable $\beta_{e,f}$ that describes if the edge e

is not colored in any color of $\mathcal{F} \setminus \{f\}$. Therefore, the following must hold for all edges e and colors f.

$$\begin{aligned}
&\beta_{e,f} \leftrightarrow \bigwedge_{f' \in \mathcal{F}\setminus\{f\}} \neg c_{e,f'} \\
\equiv &\left(\beta_{e,f} \rightarrow \neg \bigvee_{f' \in \mathcal{F}\setminus\{f\}} c_{e,f'}\right) \wedge \left(\neg\beta_{e,f} \rightarrow \bigvee_{f' \in \mathcal{F}\setminus\{f\}} c_{e,f'}\right) \\
\equiv &\left(\beta_{e,f} \vee \bigvee_{f' \in \mathcal{F}\setminus\{f\}} c_{e,f'}\right) \wedge \left(\neg\beta_{e,f} \vee \neg \bigvee_{f' \in \mathcal{F}\setminus\{f\}} c_{e,f'}\right) \\
\equiv &\left(\beta_{e,f} \vee \bigvee_{f' \in \mathcal{F}\setminus\{f\}} c_{e,f'}\right) \wedge \left(\neg\beta_{e,f} \vee \bigwedge_{f' \in \mathcal{F}\setminus\{f\}} \neg c_{e,f'}\right) \\
\equiv &\left(\bigvee_{f' \in \mathcal{F}\setminus\{f\}} c_{e,f'} \vee \beta_{e,f}\right) \wedge \left(\bigwedge_{f' \in \mathcal{F}\setminus\{f\}} \neg c_{e,f'} \vee \neg\beta_{e,f}\right)
\end{aligned}$$

The complete expression for unique colors as a CNF is as follows.

$$\begin{aligned}
&\bigwedge_{e \in E} \bigvee_{f \in \mathcal{F}} c_{e,f} \wedge \bigwedge_{e \in E} \bigvee_{f \in \mathcal{F}} \beta_{e,f} \wedge \\
&\bigwedge_{e \in E} \bigwedge_{f \in \mathcal{F}} \left(\bigvee_{f' \in \mathcal{F}\setminus\{f\}} c_{e,f'} \vee \beta_{e,f}\right) \wedge \bigwedge_{e \in E} \bigwedge_{f \in \mathcal{F}} \bigwedge_{f' \in \mathcal{F}\setminus\{f\}} \neg c_{e,f'} \vee \neg\beta_{e,f}
\end{aligned}$$

Spanning A color $f \in \mathcal{F}$ is spanning if each vertex is incident to at least one edge, which has color f.

$$\forall v \in V\colon \sum_{u \in N(v)} c_{uv,f} \geq 1 \quad \Leftrightarrow \quad \forall v \in V\colon \bigvee_{u \in N(v)} c_{uv,f} \quad \Leftrightarrow \quad \bigwedge_{v \in V} \bigvee_{u \in N(v)} c_{uv,f}$$

Thus a coloring is spanning if

$$\bigwedge_{f \in \mathcal{F}} \bigwedge_{v \in V} \bigvee_{u \in N(v)} c_{uv,f}.$$

Rainbow Matching A coloring for the graph G has no rainbow matching if in each matching there are two edges with the same color.

$$\forall m = (e_1, \ldots, e_{|V|/2}) \in \mathcal{M}\ \exists\{i,j\} \in \mathcal{I}\ \exists f \in \mathcal{F}\colon c_{e_i,f} \wedge c_{e_j,f}$$

This can be translated into the following Boolean expression.

$$\bigwedge_{m=(e_1,\ldots,e_{|V|/2}) \in \mathcal{M}} \bigvee_{\{i,j\} \in \mathcal{I}} \bigvee_{f \in \mathcal{F}} c_{e_i,f} \wedge c_{e_j,f}$$

Since $c_{e_i,f} \wedge c_{e_j,f}$ is problematic within a clause of a CNF formulation, we introduce a new variable $\gamma_{e_i,e_j,f}$, which is equivalent to $c_{e_i,f} \wedge c_{e_j,f}$.

$$\bigwedge_{m=(e_1,\ldots,e_{|V|/2})\in\mathcal{M}} \bigvee_{\{i,j\}\in\mathcal{I}} \bigvee_{f\in\mathcal{F}} \gamma_{e_i,e_j,f}$$

To ensure the equivalence, we use the following formulation for all $e, e' \in E$ and $f \in \mathcal{F}$.

$$\begin{aligned}
&\gamma_{e,e',f} \leftrightarrow c_{e,f} \wedge c_{e',f} \\
\equiv\ &(\gamma_{e,e',f} \rightarrow (c_{e,f} \wedge c_{e',f})) \wedge ((c_{e,f} \wedge c_{e',f}) \rightarrow \gamma_{e,e',f}) \\
\equiv\ &(\neg\gamma_{e,e',f} \vee (c_{e,f} \wedge c_{e',f})) \wedge (\gamma_{e,e',f} \vee \neg(c_{e,f} \wedge c_{e',f})) \\
\equiv\ &(\neg\gamma_{e,e',f} \vee c_{e,f}) \wedge (\neg\gamma_{e,e',f} \vee c_{e',f}) \wedge (\gamma_{e,e',f} \vee \neg c_{e,f} \vee \neg c_{e',f})
\end{aligned}$$

Therefore, the complete expression for rainbow matchings in CNF is as follows.

$$\begin{aligned}
&\bigwedge_{m=(e_1,\ldots,e_{|V|/2})\in\mathcal{M}} \bigvee_{\{i,j\}\in\mathcal{I}} \bigvee_{f\in\mathcal{F}} \gamma_{e_i,e_j,f} \\
&\qquad \wedge \bigwedge_{e,e'\in E} \bigwedge_{f\in\mathcal{F}} (\neg\gamma_{e,e',f} \vee c_{e,f}) \wedge (\neg\gamma_{e,e',f} \vee c_{e',f}) \wedge (\gamma_{e,e',f} \vee \neg c_{e,f} \vee \neg c_{e',f})
\end{aligned}$$

The number of variables for this SAT formulation equals $2|E||\mathcal{F}| + |E|^2|\mathcal{F}|$ and the number of clauses is $2|E| + |E||\mathcal{F}| + |E||\mathcal{F}|^2 + |\mathcal{F}||V| + |\mathcal{M}| + 3|E|^2|\mathcal{F}|$.

To verify that there exists a spanning $(n-k)$-coloring of a graph with n vertices , that does not contain a rainbow matching, we need to examine every possible graph with n vertices. With the help of the following lemma, we can reduce the number of graphs to check. Then we only need to inspect representatives from each isomorphism class.

Lemma 17. *Let $G = (V_G, E_G)$, $H = (V_H, E_H)$ be two isomorphic graphs. There exists a coloring α for G, which contains a perfect rainbow matching, if and only if there exists a coloring α' for H, which contains a perfect rainbow matching.*

Proof. Let $G = (V_G, E_G)$, $H = (V_H, E_H)$ be isomorphic graphs with an isomorphic function $f : V_G \to V_H$. Let $\mathcal{F}$ be a set of colors and $\alpha\colon V_G \to \mathcal{F}$ be a coloring of G, such that there exists a perfect rainbow matching $m = \{u_1v_1, \ldots, u_nv_n\}$ with $n = |V|/2$. Due to the fact that m is a perfect matching in G and f is bijective, $m' = \{f(u_1)f(v_1), \ldots, f(u_n)f(v_n)\}$ is also a perfect matching for H. For an edge uv let $\alpha'(uv) := \alpha(f^{-1}(u)f^{-1}(v))$ be a coloring of H. Then m' is a perfect rainbow matching in H. The inverse function f^{-1} of f exists, since f is bijective, and therefore the other direction holds. □

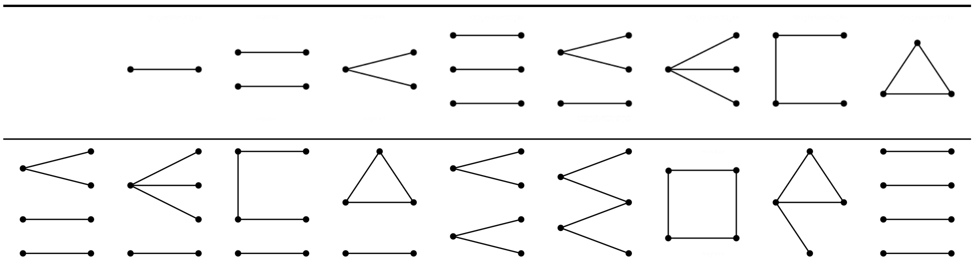
Since we are dealing with dens graphs, it is difficult to visualize such equivalence classes. For this reason, we will introduce the next lemma.

Lemma 18. *The graphs $G = (V_G, E_G)$, $H = (V_H, E_H)$ are isomorphic if and only if the* edge inverse *graph $\bar{G} = (V_G, \bar{E}_G) = (V_G, \binom{V_G}{2} \setminus E_G)$ and $\bar{H} = (V_H, \bar{E}_H) = (V_H, \binom{V_H}{2} \setminus E_H)$ are isomorphic.*

Proof. Let the graphs $G = (V_G, E_G)$, $H = (V_H, E_H)$ be isomorphic. Then there exists a bijective function $f : V_G \to V_H$, such that, for all edges $uv \in E_G$, the edge $f(u)f(v)$ is an element of E_H. Let the edge $uv \in \bar{E}_G$ be arbitrary. Therefore, $uv \notin E_G$. Due to the reason that G, H are isomorphic, the edge $f(u)f(v)$ is not an element of E_H. This leads to the conclusion that $f(u)f(v) \in \bar{E}_H$. The other direction can be shown in an analog fashion, since the edge inverse graph of an edge inverse graph is the original graph. □

For example, we want to test Claim 11 for $n = 8$ and $k = 2$. Then we can consider the graph K_8 and need to test all possible subgraphs. Since K_8 has 28 edges and at least $4 \cdot 6 = 24$ edges are needed for a coloring to be spanning with eight vertices, we need only to examine each subgraph with at least 24 edges. Therefore, each inverse-edge graph of K_8 has a maximum of four edges. Table 5.1 shows these edge-inverse graphs with none to four edges and up to eight edges. Here, isolated edges are neglected.

Tab. 5.1: Edge-inverse graphs with none to four edges and up to eight vertices. Here isolated edges are neglected.



To test Claim 11, for a given k, we choose one representative of each isomorphism class and see if the generated SAT formulation for each representative is unsatisfyable. If each representative is unsatisfyable, Claim 11 holds for the given k due to Lemma 17 and Lemma 18.

Searching for counterexamples in such a way leads to a great overhead. The SAT formulation in the following section tries to circumvent this problem by dynamically excluding edges.

5.2 General SAT Formulation

As the previous formulation only checks one specific graph, we need to check every representative of the isomorphic class with n vertices to test if there exists a spanning $(n-k)$-coloring, which also contains a perfect rainbow matching. To circumvent this problem, we can adapt the previous formulation.

Allowing edges to be excluded dynamically by the SAT formulation would fix the problem. This can be archived by allowing uncolored edges. If an edge is uncolored, all matchings, including this edge, would be ignored when checking for a rainbow matching. Therefore, not coloring an edge has the same effect as excluding it from the graph.

Unique coloring The only difference to the previous restrictions from Section 5.1 for unique coloring is to discard the rule, that an edge needs at least one color. Hence, we require only to guarantee every edge has at most one color. For this case, we already know the CNF.

$$\bigwedge_{e\in E}\bigvee_{f\in\mathcal{F}}\beta_{e,f}\wedge\bigwedge_{e\in E}\bigwedge_{f\in\mathcal{F}}\left(\bigvee_{f'\in\mathcal{F}\setminus\{f\}}c_{e,f'}\vee\beta_{e,f}\right)\wedge\bigwedge_{e\in E}\bigwedge_{f\in\mathcal{F}}\bigwedge_{f'\in\mathcal{F}\setminus\{f\}}\neg c_{e,f'}\vee\neg\beta_{e,f}$$

Spanning This formulation stays the same, because it behaves the same including uncolored edges.

$$\bigwedge_{f\in\mathcal{F}}\bigwedge_{v\in V}\bigvee_{u\in N(v)}c_{uv,f}.$$

Rainbow Matching In this part, a matching $m\in\mathcal{M}$ needs to be a rainbow matching or include an uncolored edge. With this familiar Boolean expression, we construct a CNF.

$$\bigwedge_{m=(e_1,\dots,e_{|V|/2})\in\mathcal{M}}\left(\bigvee_{\{i,j\}\in\mathcal{I}}\bigvee_{f\in\mathcal{F}}c_{e_i,f}\wedge c_{e_j,f}\vee\bigvee_{e\in m}\bigwedge_{f\in\mathcal{F}}\neg c_{e,f}\right)$$

From the previous part, we know how to handle $c_{e_i,f}\wedge c_{e_j,f}$ with $\gamma_{e_i,e_j,f}$. In a familiar fashion, we will make a conjunction over all atoms $\neg c_{e,f}$, which describes if an edge e is not colored. For that purpose, let δ_e be true if and only if the edge e is not colored. Therefore, the following restrictions need to hold in the SAT formulation for every edge.

$$\delta_e\leftrightarrow\bigwedge_{f\in\mathcal{F}}\neg c_{e,f}\equiv\delta_e\leftrightarrow\beta_{e,1}\wedge\neg c_{e,1}\equiv(\neg\delta_e\vee\neg\beta_{e,1})\wedge(\neg\delta_e\vee c_{e,1})\wedge(\delta_e\vee\beta_{e,1}\vee\neg c_{e,1})$$

Recall that $\beta_{e,f}$ describes if the edge e is not colored in any color of $\mathcal{F}\setminus\{f\}$. The resulting SAT formulation is as follows.

$$\begin{aligned}&\bigwedge_{m=(e_1,\dots,e_{|V|/2})\in\mathcal{M}}\left(\bigvee_{\{i,j\}\in\mathcal{I}}\bigvee_{f\in\mathcal{F}}\gamma_{e_i,e_j,f}\vee\bigvee_{e\in m}\delta_e\right)\\ \wedge&\bigwedge_{e\in E}(\neg\delta_e\vee\neg\beta_{e,0})\wedge(\neg\delta_e\vee c_{e,0})\wedge(\delta_e\vee\beta_{e,0}\vee\neg c_{e,0})\\ \wedge&\bigwedge_{e,e'\in E}\bigwedge_{f\in\mathcal{F}}(\neg\gamma_{e,e',f}\vee c_{e,f})\wedge(\neg\gamma_{e,e',f}\vee c_{e',f})\wedge(\gamma_{e,e',f}\vee\neg c_{e,f}\vee\neg c_{e',f})\end{aligned}$$

In total, this approach would result in $|E|$ more variables and $2\,|E|$ more clauses for the SAT formulation. As a trade-off, we can check if there exists a spanning $(n-k)$-coloring for any graphs with n vertices, which also contains a rainbow matching, instead of only for specific graphs. To see which method results in better runtimes, see Section 5.5.

5.3 Case Analysis for Ten Vertices

Consider a graph with ten vertices and a spanning 8-coloring, where each edge is colored. Each color class needs to contain at least five edges, otherwise it would not be spanning. Thus, we need at least $5 \cdot 8 = 40$ edges to ensure all colors are spanning. The complete graph with ten vertices, K_{10} has $9 \cdot 10/2 = 45$ edges. This leaves use with at most five edges, that we can to distribute to some color class. Hence, there are at least three color classes with exactly five edges. When inspecting an edge induced subgraph of two of these colors, the resulting graph is either C_{10} or $C_4 + C_6$, with alternating edge coloring, as Ravsky [Rav22] has noticed. Since one of these two case must occur, we can fix the color for these edges allowing us to focus on the remaining 35 edges which need to be colored in six colors, which simplifies the SAT formulation. If the resulting graph has at least two colors with only five edges, the same procedure can be applied again using two different colors, which further reduces the complexity.

Out of the seven distributions of 35 edges on six spanning colors, only one has exactly one color with five edges, while all other colors have six edges, which makes it impossible to use the previous method in this case, due to the fact, that at least six edges do not necessarily form a perfect matching.

(a) Edge distribution after applying the first method once.

Colors	1	2	3	4	5	6
Edges	5	5	5	5	6	9
	5	5	5	5	7	8
	5	5	5	6	6	8
	5	5	5	6	7	7
	5	5	6	6	6	7
	5	6	6	6	6	6

(b) Edge distribution after applying the first method exhaustively.

Colors	1	2	3	4	5	6
Edges	0	0	0	0	6	9
	0	0	0	0	7	8
	0	0	5	6	6	8
	0	0	5	6	7	7
	0	0	6	6	6	7
	5	6	6	6	6	6

Tab. 5.2: The tables show a example distribution of how many edges are colored in a certain color in multiple stages of the case analysis.

After the first reduction, the edges can be distributed onto the arbitrarily chosen colors $1, 2, 3, 4, 5, 6 \in \mathcal{F}$ shown in Table 5.2a. Here the first row specifies the colors while every other row describes a distribution of the edges. Since the first five distributions also have at least two colors with exactly five edges, we can continue to apply the method until no two colors have five edges, which gives use the remaining distribution in Table 5.2b. As a disadvantage, we need to check 2^m cases when applying m reductions, because we need to consider each case of removing C_{10} or $C_4 + C_6$ recursively.

This analysis holds also for graphs with more vertices or graphs with two color classes forming a perfect matching. Also, all edges do not necessarily need to be colored. Consider the graph K_{12} and its spanning 10-coloring. We have six edges which are needed to distribute to ten colors since we need $10 \cdot 6 = 60$ edges to ensure the spanning property of a coloring and K_{12} has 66 edges. Therefore, we have at least four color classes, which have six edges, after assigning each left out edge a color. Here, the induced subgraph either induces C_{12}, $C_8 + C_4$ or $C_4 + C_4 + C_4$, which can be handled similarly has above. With these tools, we can now try to calculate the values of $k(n)$.

5.4 SAT Results

Now that we have a few tools to investigate Claim 11 using SAT formulations, we will show our experimental results in this chapter. The SAT solver Kissat [Bie], which is a state-of-the-art SAT solver [HJS+], was used to solve the SAT formulations on an Intel Xeon Skylake processor with 3.2 GHz. Note that Kissat only runs on one CPU thread at a time.

At first, we want to check if Claim 11 holds for $n = 8$ and $k = 2$, since the ILP in Chapter 4 could not solve this instance. The SAT solver could show with each SAT formulation in Sections 5.1 and 5.2, that the instance is unsatisfiable and thus Claim 11 holds due to the absence of a counterexample. This leads to the conclusion that $k(8) = 2$ and $C(8) < 8-2 = 6$. Compared to Cervera-Lierta's et al. [CKA21], our result is worse, because they could show that $C(8) < 4$.

Where we made some progress, is on graphs with ten vertices. While applying the method in Section 5.3, we found a spanning 6-coloring for the graph $K_{10} - C_{10}$ in Figure 5.1, which is a counterexample to Claim 11. The method from Section 5.3 now cannot be used, since it only gives a conclusion for every spanning 8-coloring of K_{10} if $K_{10} - C_{10}$ and $K_{10} - C_6 - C_4$ do contain a perfect rainbow matching. This results in an upper bound for $k(10)$ with $k(10) < 4$.

With the method in Section 5.1, we found a representative with ten vertices and $k = 3$, which does not contain a rainbow matching. This representative is depicted with its coloring in Figure 5.2. With this graph it follows that $k(10) < 3$ and with Corollary 16 the value of $k(10)$ must be either one or two. Observe that the graph has three vertices of degree seven, where each incident edge has one of the seven colors, and seven vertices of degree nine, where three edges have the same color. Ravsky [Rav21a] recognized the coloring pattern in means of Steiner triple systems, which allowed him to provide the following lemma.

Lemma 19 ([Rav21a])**.** *If* $n/2 \equiv \pm 1 \pmod 6$*,* $k(n) \leq n/2 - 3$.

The method could also limit the graphs, which certainly contain a rainbow matching for every coloring. For graphs with ten vertices and $k = 2$, we could show, that 19 of 25 isomorphic classes do not contain a counterexample. We can describe these classes with edge-inverse graphs. Next to the empty graph, the other five graphs consist of one to five edges, which do not share any vertices.

The other method did not supply a result as well after a considerable amount of computational time. To be exact, each problem instance was running over 1500 hours without delivering an answer for ten vertices and $k = 2, 3, 4$. Here, all the different options were used in Kissat to get a solution to no avail. We will go into detail about those options and their time benefits in Section 5.5.

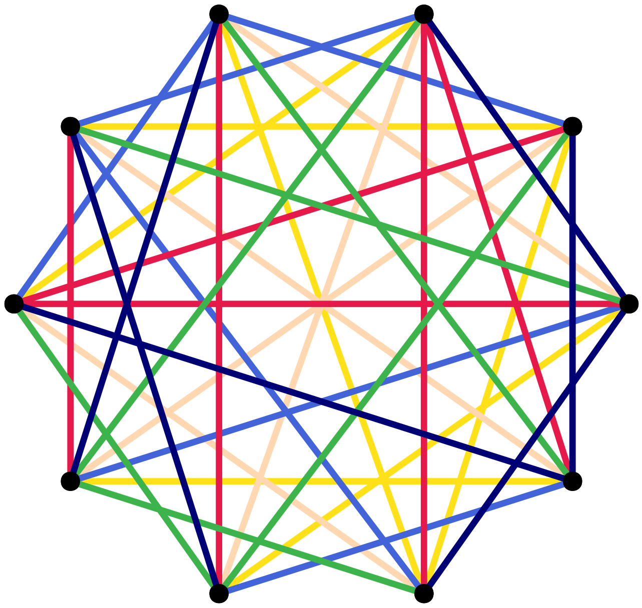


Fig. 5.1: Output graph of the SAT formulation for $n = 10$ and $k = 4$ with a spanning 6-coloring consisting of colors ●, ●, ○, ●, ○, ●.

These results show that none of our SAT formulation is efficient enough to calculate the value of $k(10)$. When considering the K_{10} and $k = 2$, there are a total of 16965 variables and 51605 clauses, where $45^2 \cdot 8 = 16200$ variables and 48600 clauses alone are claimed by defining γ, which indicates if the edges e, e' both have the color f. Especially when no edges are excluded and no simplifications can be made by the SAT solver, the runtime can be very high, since to solve a conjunctive normal form is an NP-hard problem. In the next section, we will see, that the runtime can differ greatly between seemingly identical hard SAT formulations.

5.5 SAT Program Performance

Kissat has different options for solving SAT formulations, whose runtimes we will analyze in this section. We run each of the three options on the single K_8 with the methods from Chapter 5. The runtime of each pass will be presented in this chapter. The three options are *unsat*, *sat* and without any argument, which is the default configuration of Kissat. Here, unsat should be used to get a faster output for instances, which are not satisfyable. Contrary to this, sat should speed up the computational time if an instance is satisfyable. The option with no further arguments should be the middle ground of the two options and should give a good runtime for each type of instances.

To test these options, we run each instance a minimum of four up to ten times on a single Intel Xeon Skylake core with 3.2 GHz with the graph K_8 and $k \in \{1, 2, 3, 4\}$. The accumulated runtime will be averaged for each instance. The Tables 5.3, 5.4 show the runtime. Here the option normal refers to not using other options such as sat or

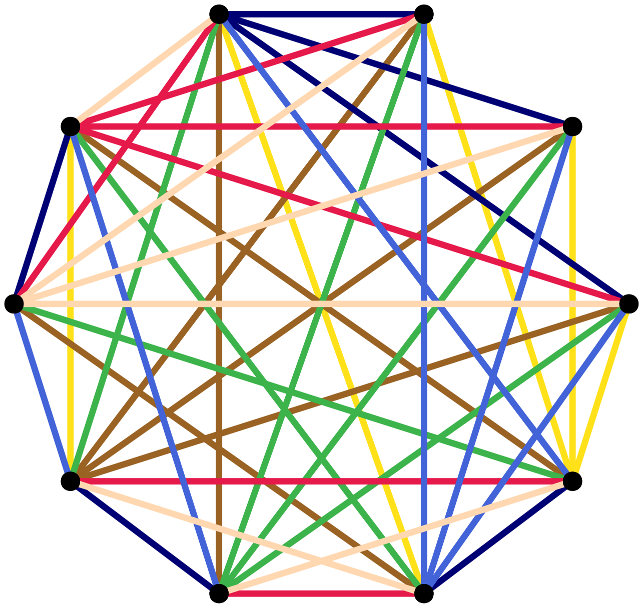


Fig. 5.2: Output graph of the SAT formulation for $n = 10$ and $k = 3$ with a spanning 7-coloring consisting of colors ●, ●, ●, ●, ●, ●, ●.

unsat. Each entry contains the runtime in seconds. Note that we did not compute one representative of each isomorphic class in Table 5.3.

For $k = 3, 4$ the runtime for each method and option are quite equal. The SAT formulation here is satisfyable, meaning that the option sat should perform the best. This trend can only be clearly seen in Table 5.3 for $k = 3$, while the other cases show no stark differences in the runtime. One notable outlier in the table is the runtime for normal and $k = 3$, which is higher than the other two options.

Contrary to the low runtimes for $k = 3, 4$, the completion time for $k = 2$ for each method exploded. Since in this case the SAT formulations are unsatisfyable, the option unsat should perform the best, which we cannot confirm with our formulations. Here the runtimes increase from sat throughout normal to unsat considerable, while also the formulation from Section 5.2 runs faster than the other, even though it contains $|E|$ more variables and $2|E|$ more clauses.

For $k = 1$ the runtime drop again into the one to two minute range. While the runtime increases in Table 5.3, the runtime decreases in Table 5.4. Since this instance is unsatisfyable, the runtime should theoretical increase.

In most cases, the second method performed better than the first one. Even though the sat option is not always the best option, in this analysis it performed the best on average. For these reasons, the second method and the option sat is on average the best configuration to inspect a single graph.

Tab. 5.3: Runtime of SAT formulation in Section 5.1 on K_8 in seconds.

k	sat	normal	unsat
4	0.00	0.00	0.00
3	0.15	1.26	0.84
2	13111.37	293219.64	314729.21
1	99.06	245.34	246.87

Tab. 5.4: Runtime of SAT formulation in Section 5.2 on K_8 in seconds.

k	sat	normal	unsat
4	0.01	0.02	0.02
3	0.46	0.43	0.68
2	10138.01	21929.19	568511.64
1	99.22	77.25	77.25

6 Conclusion

In this paper, we wanted to investigate Krenn's Conjecture 1. Especially the upper bound of the number $C(n)$ of distinct monochromatic vertex colorings in monochromatic weighted graphs. For this purpose, we introduced Ravky's Claim 6 and Claim 11 and showed, how they bound $C(n)$.

Followed by this, we looked into different methods to investigate, for which even number $n \geq 6$ and $k < n$, Claim 11 holds. While a brute-force algorithm, as we have discovered, has no chance of solving the problem in reasonable time, the integer linear program show, that Claim 11 holds for all possible values of k with $n = 6$ and $n = 8$ with the exception to $n = 8$ and $k = 2$. With the help of a SAT formulation, we could also show, that Claim 11 holds for $n = 8$ and $k = 2$. Also, the SAT formulation could find counterexamples for Claim 11 for $n = 10$ and $k \geq 3$. For $n = 10$ and $k = 2$, we could only confirm, that there do not exist counterexamples, for all graphs with n vertices and spanning coloring, except for six graphs.

This leaves the question open for the value of $k(10)$ and a bound for $C(10)$ through this method. The SAT formulation could not make any further progress for $n \geq 12$, due to the computational complexity of the problem.

Even though, we improved the upper bound on $k(n)$ for graphs with n vertices, where $n/2 \equiv \pm 1 \pmod 6$, regarding Claim 11 because of the counterexample in Figure 5.2.

We recognized the coloring pattern of this graph in means of Steiner triple systems, which allowed us to provide Lemma 19.

Lemma 19 ([Rav21a]). *If* $n/2 \equiv \pm 1 \pmod 6$*,* $k(n) \leq n/2 - 3$.

Since Claim 11 only gives a lower bound for Claim 6, with which we bound $C(n)$, one could try to directly investigate Claim 6 with a SAT formulation. This SAT formulation is more complex than the one we introduced in this thesis, because the definition of Claim 6 is more complex, especially with the uniqueness of the graphs (V_f, E'_f). Furthermore, Ravsky [Rav] proposed a design, connected to rainbow matchings in color-spanned graphs, to bound $C(n)$ from Krenn's Conjecture 1.

Although, we did not improve the bound for Krenn's conjecture, we gave further insights into the interesting field of rainbow matchings in color-spanned graphs.

Bibliography

[ABKZ17] Ron Aharoni, Eli Berger, Dani Kotlar, and Ran Ziv: *On a conjecture of Stein*, volume 87 of *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, pages 203–211. Springer International Publishing, 2017, 10.1007/s12188-016-0160-3.

[Bie] Armin Biere: Kissat SAT solver. `https://github.com/arminbiere/kissat`, accessed: June 24, 2022.

[Bog17] Ilya Bogdanov: Graphs with only disjoint perfect matchings. MathOverflow, 2017. `https://mathoverflow.net/q/267013`, accessed: April 25, 2022.

[BZG20] Hannah Brown, Lei Zuo, and Dan Gusfield: Comparing integer linear programming to SAT-solving for hard problems in computational and systems biology. In Carlos Martín-Vide, Miguel A. Vega-Rodríguez, and Travis Wheeler (editors): *Algorithms for Computational Biology*, volume 12099 of *Lecture Notes in Computer Science*, pages 63–76. Springer International Publishing, 2020, 10.1007/978-3-030-42266-0_6.

[CG22] L. Sunil Chandran and Rishikesh Gajjala: Perfect matchings and quantum physics: Progress on Krenn's conjecture. arXiv, 2022. 10.48550/ARXIV.2202.05562.

[CKA21] Alba Cervera-Lierta, Mario Krenn, and Alán Aspuru-Guzik: Design of quantum optical experiments with logic artificial intelligence. arXiv, 2021. 10.48550/ARXIV.2109.13273.

[DMZ89] Greenberger Daniel M., Horne Michael A., and Anton Zeilinger: Going beyond Bell's theorem. In Menas Kafatos (editor): *Bell's Theorem, Quantum Theory and Conceptions of the Universe*, volume 37 of *Fundamental Theories of Physics*, pages 69–72. Springer Netherlands, 1989, 10.1007/978-94-017-0849-4_10.

[FHS20] Johannes K. Fichte, Markus Hecher, and Stefan Szeider: A time leap challenge for SAT-Solving. In Helmut Simonis (editor): *Principles and Practice of Constraint Programming*, volume 12333 of *Lecture Notes in Computer Science*, pages 267–285. Springer International Publishing, 2020, 10.1007/978-3-030-58475-7_16.

[GCZK19] Xuemei Gu, Lijun Chen, Anton Zeilinger, and Mario Krenn: Quantum experiments and graphs. III. High-dimensional and multiparticle entanglement. *Physical Review A*, 99(3), 2019, 10.1103/physreva.99.032338.

[GEZK19] Xuemei Gu, Manuel Erhard, Anton Zeilinger, and Mario Krenn: Quantum experiments and graphs II: Quantum interference, computation, and state generation. *Proceedings of the National Academy of Sciences*, 116(10):4147–4155, 2019, 10.1073/pnas.1815884116.

[HJS+] Marijn Heule, Matti Jarvisalo, Martin Suda, Markus Iser, Tomáš Balyo, and Nils Froleyks: SAT competition 2021 results. `https://satcompetition.github.io/2021/results.html`, accessed: Mai 30, 2022.

[HLL+17] Armin Hochrainer, Mayukh Lahiri, Radek Lapkiewicz, Gabriela Barreto Lemos, and Anton Zeilinger: Quantifying the momentum correlation between two light beams by detecting one. *Proceedings of the National Academy of Sciences*, 114(7):1508–1511, 2017, 10.1073/pnas.1620979114.

[HMM15] Axel Heuer, Ralf Menzel, and Peter W. Milonni: Induced coherence, vacuum fields, and complementarity in biphoton generation. *Physical Review Letters*, 114(5), 2015, 10.1103/physrevlett.114.053601.

[IBM] IBM: IBM ILOG CPLEX Optimizer. `https://www.ibm.com/analytics/cplex-optimizer`, accessed: Jun 24, 2022.

[KGS19] Mario Krenn, Xuemei Gu, and Daniel Soltész: Questions on the structure of perfect matchings inspired by quantum physics, 2019. 10.48550/ARXIV.1902.06023.

[KGZ17] Mario Krenn, Xuemei Gu, and Anton Zeilinger: Quantum experiments and graphs: Multiparty states as coherent superpositions of perfect matchings. *Phys. Rev. Lett.*, 119:240403, 2017, 10.1103/PhysRevLett.119.240403.

[KHLZ17] Mario Krenn, Armin Hochrainer, Mayukh Lahiri, and Anton Zeilinger: Entanglement by path identity. *Phys. Rev. Lett.*, 118:080401, 2017, 10.1103/PhysRevLett.118.080401.

[KPKK16] Dmitry A. Kalashnikov, Anna V. Paterova, Sergei P. Kulik, and Leonid A. Krivitsky: Infrared spectroscopy with visible light. *Nature Photonics*, 10(2):98–101, 2016, 10.1038/nphoton.2015.252.

[KY12] Alexandr V. Kostochka and Matthew P. Yancey: Large rainbow matchings in edge-cloured graphs. *Combinatorics, Probability and Computing*, 21:255–263, 2012, 10.1017/S0963548311000605.

[LBC+14] Gabriela Barreto Lemos, Victoria Borish, Garrett Cole, Sven Ramelow, Radek Lapkiewicz, and Anton Zeilinger: Quantum imaging with undetected photons. *Nature*, 512:409–412, 2014, 10.1038/nature13586.

[LPHH16] Pasi Lähteenmäki, Gheorghe Sorin Paraoanu, Juha Hassel, and Pertti J. Hakonen: Coherence and multimode correlations from vacuum fluctuations in a microwave superconducting cavity. *Nature Communications*, 7(12548), 2016, 10.1038/ncomms12548.

[MMMA18] Erhard Manuel, Malik Mehul, Krenn Mario, and Zeilinger Anton: Experimental Greenberger–Horne–Zeilinger entanglement beyond qubits. *Nature Photonics*, 12:759–764, 2018, 10.1038/s41566-018-0257-6.

[Rav] Alex Ravsky: The existence of special designs. MathOverflow. `https://mathoverflow.net/questions/423223/the-existence-of-special-designs`.

[Rav21a] Alex Ravsky: A perfect rainbow matching in an edge-colored graph with spanning color classes. MathOverflow, 2021. `https://mathoverflow.net/q/396913`, accessed: June 24, 2022.

[Rav21b] Alex Ravsky: A special perfect matching in a complete edge-colored graph. MathOverflow, 2021. `https://mathoverflow.net/q/396653`, accessed: June 24, 2022.

[Rav22] Alex Ravsky: Private communication. Chatroom of University of Würzburg, 2022. written up until: April 29, 2022.