UNIVERSITÉ LIBRE DE BRUXELLES

MEMO-F524 - Master Thesis

Inherited vertex colouring of graphs - research on Krenn's conjecture

Merlin Hannon

Supervisors : Gwenaël JORET, Yelena YUDITSKY

Accademic year : 2023–2024

Contents

Introduction

Quantum computing has hugely developed in the last few years, with the rise of new technologies and the development of new algorithms. Advancements made in quantum information and computation promise theoretical ways to solve problems that are currently intractable with classical computers. It is already used a lot in the field of cryptography [KG21], and its potential uses extend to many other fields such as artificial intelligence [Ott+17], financial planning and medicine [Has+20]. However, quantum computers are still in their infancy and are not yet able to solve a lot of practical problems. One of the reasons for this is the difficulty to physically create some specific quantum states.

In its research, the physicist Mario Krenn has been working on the generation of high-dimensional multipartite quantum states [KGZ17]. In this same paper, he discovered a new surprising link between the generation of these quantum states and a specific problem in graph theory. The problem was summarized by Mario Krenn himself on his website [Kre]. Furthermore, to promote research on the subject, the physicist promised a reward of 3000€ to the first person who will solve his conjecture, and an additional reward of 1000€ was offered during the last 2 years to the best paper on the subject [Kre]. This problem, that we will refer to as the *Krenn's conjecture*, is about proving the existence of a bound on some quantity called the *weighted matching index* of a graph. It will be defined in details in the chapter 1 of this thesis.

The main points of this master thesis are the following: first, the Krenn's conjecture will be rigorously defined. This will be done through a progressive approach, starting from reminders of basic graph theory concepts (section 1.1), then looking at a simplified version of the conjecture (section 1.2), and at last introducing the Krenn's conjecture (section 1.3). This definition will be directly followed by a detailed explanation of its physical interpretations (section 1.4).

Then, a state of the art in the domain will be established in Chapter 2. In dressing the state of the art, we will focus on the special cases of the problem that were already solved by other researchers. Also, we will rewrite some proofs in our own terms, to get a better understanding of the reasoning.

Lastly, I will present my own new contributions to the problem in chapter 3, which can be summarized in 2 main points. Firstly, I proved algorithmically a new special case of the Krenn's conjecture. This proof will be presented by taking a simple known case of the problem, and then relaxing some of its constraints to generalize it. And secondly, I will present a new tool I developed, called EGPI, to experimentally test the conjecture on a large number of graphs. The use cases and technical implementation of this tool will be discussed in section ... At last, and to finish this master thesis, I present some experimental results I got using EGPI, and my interpretation of them in section 3.2.5.

1

Description of the problem

1.1 Prerequisites and used notations

In order to understand the following chapters, it is needed to (re)introduce some useful notations that are widely used in the context of graph theory [Die17].

1.1.1 The different definitions of graphs

A reader who is not familiar with the field of graph theory needs first to understand the basic concepts of graphs. The following definitions constitute the foundation of the domain, and I personally learned them in a course based on the book [Die17].

In the context of discrete mathematics, a *graph* $G = (V, E)$ is a set V of objects, called *vertices*, and a set E of tuples (or pairs) between these vertices, called *edges*. The notations $V(G)$ and $E(G)$ denote the set of vertices and the set of edges of G respectively. Graph theorists distinguish different types of graphs. The following definitions cover the graphs that will be used in the context of this master thesis.

Definition 1.1.1 (Simple graph)**.** A graph G is said to be *simple* if it has at most one edge between each pair of its vertices. Furthermore, a simple graph has no edge that begins and ends at the same vertex.

Definition 1.1.2 (Undirected graph)**.** A graph G is said to be *undirected* if $E(G)$ is a set of unordered pairs, i.e. if its edges are bidirectional. It is opposed to the concept of *directed graphs*, in which the edges are ordered tuples.

Definition 1.1.3 (Edge-coloured graph)**.** An *edge-colouring* k of a graph G is a function that maps each edge $e \in E(G)$ to a colour $k(e)$ (often represented as an unsigned integer). An *edge-coloured graph* G_k is a graph G equipped with an edge-colouring k.

Definition 1.1.4 (Mixed edge-coloured graph)**.** A *mixed edge-colouring* k of a graph G is a function that maps each edge $e = (v_1, v_2) \in E(G)$ to two colours $k(e, v_1)$ and $k(e, v_2)$ (often represented as unsigned integers). A *mixed edge-coloured graph* G_k is a graph G equipped with a mixed edge-colouring k.

Definition 1.1.5 (Weighted graph)**.** A *weighting* w of a graph G is a function that maps each edge $e \in E(G)$ to a weight $w(e)$. Most often, this weight is assumed to be a real number. However, in the context of this master thesis, this weight is a complex number. A *weighted graph* G^w is a graph G equipped with a weighting w.

1.1.2 Paths and connectivity concepts

The concepts of paths and connectivity are widely used in the mathematical field of graph theory. The definitions of these concepts are recalled here to ensure a better understanding of the following chapters [BM+76].

Definition 1.1.6 (Path)**.** Given a graph G, a *path* $P = (e_1, e_2, \dots, e_l)$ is an ordered sequence of different edges from $E(G)$ respecting the following property.

- $\forall i \in \{1, 2, \cdots, l-1\}$, the vertices e_i and e_{i+1} share a common vertex.
- $\forall$ vertex $v \in V(G)$, there is at most two edges $e \in P$ that contain v. In other words, P does not pass through the same vertex twice.

Definition 1.1.7 (Cycle)**.** Given a graph G, a *cycle* $C = (e_1, e_2, \dots, e_l)$ is an ordered sequence of different edges from $E(G)$ respecting the following property.

- C is a path.
- e_1 and e_l share a vertex. In other words, C begins and ends at the same vertex.

Definition 1.1.8 (Hamiltonian cycle)**.** A cycle H of a graph G is said to be *Hamiltonian* if and only if all the vertices in $V(G)$ appear in H.

Definition 1.1.9 (Arc of Hamiltonian cycle)**.** Let G be a graph that admits a Hamiltonian cycle $H = (\{v_1, v_2\}, \{v_2, v_3\}, \cdots, \{v_{n-1}, v_n\})$. In the context of this master thesis, what is called the *arc* from v_i to v_j on H, denoted by $H_{i,j}$, is defined as follows.

$$H_{i,j} = \begin{cases} \big\{\{v_i, v_{i+1}\}, \{v_{i+1}, v_{i+2}\}, \dots, \{v_{j-1}, v_j\}\big\} & \text{if } i < j \\ \big\{\{v_i, v_{i+1}\}, \{v_{i+1}, v_{i+2}\}, \dots, \{v_n - 1, v_n\}, \{v_n, v_1\}, \{v_1, v_2\}, \dots, \{v_{j-1}, v_j\}\big\} & \text{if } j < i \\ \{\} & \text{if } i = j \end{cases}$$

In other words, $H_{i,j}$ is the set of edges that builds the path from v_i to v_j along H when being allowed to go only in the positive direction. An example is shown in figure 1.1.

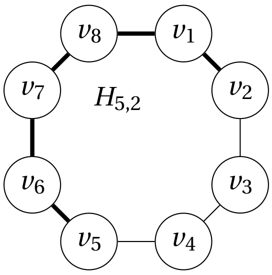

Figure 1.1: Visualization of the arc $H_{5,2}$ on a Hamiltonian path H of some graph G from v_5 to v_2. The arc is marked using thicker edges.

Definition 1.1.10 (Connected graph)**.** A graph G is said to be *connected* if and only if, $\forall v_1, v_2 \in V(G), \exists$ a path P between v_1 and v_2 in G. If this property is not satisfied, then G is *disconnected.*

Definition 1.1.11 (Edge connectivity)**.** The *edge connectivity* of a graph G is the minimum number of edges we have to remove from $E(G)$ to make it disconnected.

Definition 1.1.12 (Maximum and minimum degree)**.** The *degree* of a vertex v in a graph G, denoted $d(v)$, is the number of edges touching v in $E(G)$. The *minimum degree* of G, denoted $\delta(G)$, and the *maximum degree* of G, denoted $\Delta(G)$, are defined as follows.

$$\begin{cases} \delta(G) = \min\limits_{v \in V(G)} d(v) \\ \Delta(G) = \max\limits_{v \in V(G)} d(v) \end{cases}$$

1.1.3 Matching definitions and notations

The studied problem in this master thesis is centered about the (non-)existence of matchings in specific graphs. For this reason, the matching-related concepts are (re-)defined here [Die17].

Definition 1.1.13 (Matching)**.** Given a graph G, two edges in $E(G)$ are said to be *independent* if they do not contain any vertex in common. A *matching* M of G is a set of edges $M \in E(G)$ that are independent 2 by 2.

Definition 1.1.14 (Perfect matching)**.** A matching M of a graph G is said to be *perfect* if it covers the whole set of G's vertices, i.e. if every vertex $v \in V(G)$ appears in one of the edges of M.

1.2 Simplified version of the conjecture (solved)

The following statements and definitions directly come from Mario Krenn's formulation of the problem in [Kre]. He first defines the notion of monochromatic graphs, followed by the notion of its matching index.

Definition 1.2.1 (Monochromatic graph)**.** An edge-coloured graph G_k is said to be *monochromatic* if all its perfect matchings are monochromatic, i.e., for all perfect matchings M, the edges of M are all the same colour.

An example of monochromatic graph is shown in figure 1.2.

Definition 1.2.2 (Matching index)**.** Let G be a graph. For any edge-colouring k such that G_k is monochromatic, let $c(G,k)$ be defined as the number of different colour classes containing at least one perfect matching of G_k. The *matching index* of G, denoted $c(G)$, is the maximum value that $c(G,k)$ can take.

$$c(G) = \max_{k \in \mathscr{K}(G)} (c(G,k))$$

Where $\mathscr{K}(G)$ describes here the set of all possible edges-colourings k of G such that G_k is monochromatic.

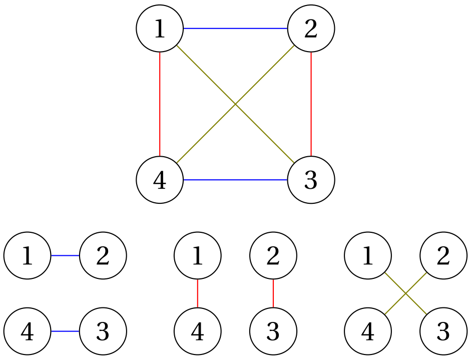

Figure 1.2: Example of monochromatic graph: an edge-coloured version of K_4. It has at most three monochromatic perfect matchings of different colours. Therefore, $c(K_4) = 3$.

Equipped with these concepts, the reader has now all the knowledge to understand the simplified version of the conjecture.

Theorem 1.2.3 (Simplified version of Krenn's conjecture)**.** For all graphs G, if G is isomorphic to K_4, then $c(G) = 3$. Otherwise, $c(G) \leq 2$.

The proof of theorem 1.2.3 was first proposed by Ilya Bogdanov in a post on a forum. [Bog] I rewrote it in my own terms to make it clearer, according to our own formulation of the problem.

Proof. Let G be a monochromatic graph with a matching index $c(G) \geq 2$, and let k be an edge-colouring of G such that $c(G,k) = c(G)$. Let M_1, M_2 be two monochromatic perfect matchings of G_k of different colours. Clearly, they are disjoint.

Claim. The union of M_1 and M_2 form a disjointed union of cycles of even length.

Proof of the claim. For each vertex $v \in V(G_k)$, v is touched by exactly one edge from M_1 and one edge of M_2 (by definition 1.1.14 of a perfect matching). So all the vertices are of degree 2 in $M_1 \bigcup M_2$. This already proves the first statement. Then, on each cycle in $M_1 \bigcup M_2$, the edges must be alternating between edges from M_1 and edges from M_2 (otherwise we would have two edges from the same perfect matching that touch the same vertex, which is forbidden by definition 1.1.13). This condition is satisfied only if the cycles are of even length. □

Let say there are $\mathscr{C}$ different cycles formed by $M_1 \bigcup M_2$. If $\mathscr{C} \geq 2$, then a new non-monochromatic perfect matching N can be found as follows.

$$N = \begin{cases} M_1 & \text{on the } k-1 \text{ first cycles} \\ M_2 & \text{on the last cycle} \end{cases}$$

As N contains edges from M_1 and from M_2, it is not monochromatic, and can't exist. Therefore, $\mathscr{C} = 1$ and the union of any 2 perfect matchings of different colours in a monochromatic graph forms a Hamiltonian cycle. We will denote $H = (v_1, v_2, \ldots, v_n)$ the Hamiltonian cycle formed by M_1 and M_2. We notice that n is an even number, since $|H| = 2 \cdot |M_1| = 2 \cdot |M_2|$.

Now let's consider the case where $c(G) = c(G,k) \geq 3$. Let M_3 be a third monochromatic perfect matching such that the colour of M_3 differs from the colours of M_1 and M_2. Let $e = \{v_i, v_j\} \in M_3$, where the indices i and j are denoting positions in H. From now, it will be considered without loss of generality that the colour of $\{v_i, v_{i+1}\}$ is 1. Indeed, if it is not the case, the colour 1 and 2 can be exchanged in the following reasoning.

- **If $j - i$ is odd :** using the notation introduced in definition 1.1.9, e splits H in two arcs $H_{i+1,j-1}$ and $H_{j+1,i-1}$ of odd lengths. Then we build a new non-monochromatic perfect matching N as follows :

$$\begin{aligned} N &= e \\ &\cup M_1 \cap H_{i+1,j-1} \\ &\cup M_2 \cap H_{j+1,i-1} \end{aligned}$$

The construction of N is shown in figure 1.3. Because N is not monochromatic, it can't exist in G. Therefore, this case is impossible.

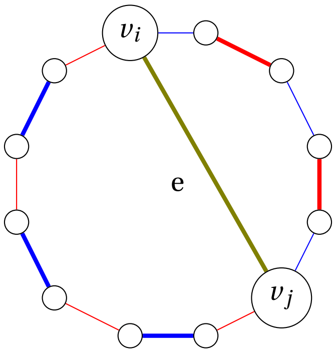

Figure 1.3: Construction of N (in bold) when $j-i$ is odd.

- **If $j-i$ is even:** then let's assume without loss of generality that e cuts the smallest arc possible $H_{i,j}$ in H. Indeed, if it is not the case, it is always possible to choose another edge from M_3. Let $e' = \{v_{i+1}, v_k\} \in M_3$. The existence of e' is certain by definition 1.1.14 of a perfect matching. By the previous case, we know that $k-(i+1)$ can not be odd, so it is even. v_k does not appear in $H_{i,j}$ since $H_{i,j}$ was the smallest arc possible delimited by e. Let C be defined as the cycle formed by the following edges.

$$C = \big\{\{v_i, v_{i+1}\}, \{v_{i+1}, v_k\}, \{v_k, v_{k-1}\}, \ldots, \{v_{j+1}, v_j\}, \{v_j, v_i\}\big\}$$

C has an even length, $H_{i+1,j}$ and $H_{k,i}$ have also an even length. This observation can be easily understood when looking at figure 1.4. Knowing all these parities, it is possible to find a new non-monochromatic perfect matching N.

$$\begin{aligned} N &= e \cup e' \\ &\cup M_1 \cap H_{i+1,j} \\ &\cup M_1 \cap H_{k,i} \\ &\cup M_2 \cap H_{j,k} \end{aligned}$$

This process is shown in figure 1.4. If G has more than 4 vertices, N is a non-monochromatic perfect matching and can not exist.

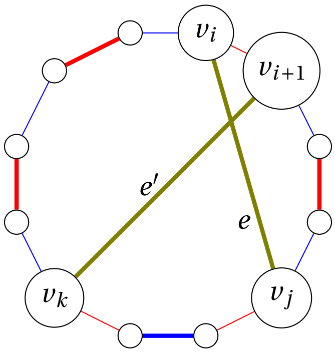

Figure 1.4: Construction of N (in bold) when $j - i$ is even.

In conclusion, K_4 is the only graph that has a matching index of 3. For all graphs G that have a size different from 4, $c(G) \leq 2$. □

1.3 Krenn's conjecture (unsolved)

All the graphs that were studied in the previous section were simple, undirected edge-coloured graphs. (Referring to the definitions 1.1.1, 1.1.2 and 1.1.3). However, in the context of Mario Krenn's research, these graphs constitute a particular case of all the graphs that need to be studied [KGZ17; Kre]. The exact motivations to extend the problem to a whole new level are directly motivated by quantum optics experiment, and will be described into details in section 1.4.

Definition 1.3.1 (Experiment graph)**.** In the context of this master thesis, what is called an *experiment graph* is a mixed-edge coloured weighted graph, refering to the definitions 1.1.4 and 1.1.5. Furthermore, experiment graphs allow arbitrary number of edges between the same pair of vertices; they are therefore not simple in the sense of definition 1.1.1. Nevertheless, they do not allow loops, i.e. no edge begins and ends at the same vertex.

The denomination of experiment graph was used by Gajjala and Chandran in [CG23]. The reasons behind this name will become clearer in section 1.4. The newly added weights in these experiment graphs introduce much more freedom compared to the previous case, and allow the definitions of new generalized concepts about perfect matchings. Again, these definitions come directly from [Kre].

Definition 1.3.2 (Weight of a matching)**.** The *weight* of a matching M, denoted $w(M)$, is the product of all the weights of its edges.

$$w(M) = \prod_{e \in M} w(e)$$

Definition 1.3.3 (Induced vertex colouring)**.** Given an experiment graph G_k^w, it is said that a vertex colouring κ is *induced* by a perfect matching M of G if and only if, for each vertex $v \in V(G)$, $\exists e \in M$ such that $\kappa(v) = k(e, v)$. (Using the notation introduced in definition 1.1.4). The induced vertex colouring by a perfect matching M is denoted $\kappa(M)$ in the context of this master thesis. The other way around, $\mathscr{M}_\kappa$ denotes the set of all perfect matchings that induce a vertex colouring κ.

Definition 1.3.4 (Feasible vertex colouring)**.** Let G_k^w be an experiment graph. It is said of a vertex colouring κ that it is *feasible* for G_k^w if and only if there is at least one perfect matching of G_k^w that induces κ.

Definition 1.3.5 (Weight of a vertex colouring)**.** The *weight* of a feasible vertex colouring κ of an experiment graph G_k^w, denoted $w(\kappa)$, is the sum of all the weights of the perfect matchings that induce this vertex colouring.

$$w(\kappa) = \sum_{M \in \mathscr{M}_\kappa} w(M)$$

This definition uses a notation introduced in definition 1.3.3.

Definition 1.3.6 (Perfectly monochromatic graph)**.** An experiment graph G_k^w is said to be *perfectly monochromatic* if the weights of all its feasible monochromatic vertex colourings are equal to 1, and the weights of all its feasible non-monochromatic vertex colourings are equal to 0.

An example of a perfectly monochromatic graph is presented in the figure 1.5. Notice that in a perfectly monochromatic graph, non-monochromatic perfect matchings are allowed as long as the weight of their induced vertex colouring is 0. This was not the case in simple monochromatic graphs. Just like the matching index of a monochromatic graph was defined in the previous section (definition 1.2.2), we generalize this notion by introducing a weighted matching index in perfectly monochromatic graphs.

Definition 1.3.7 (Weighted matching index)**.** Given a perfectly monochromatic graph G_k^w, let $\tilde{c}(G, k, w)$ be the number of different feasible monochromatic vertex colourings in G_k^w. We define the *weighted matching index* of G, denoted $\tilde{c}(G)$, as the maximum number that $\tilde{c}(G, k, w)$ can take for a mixed-edge colouring k and a weighting w.

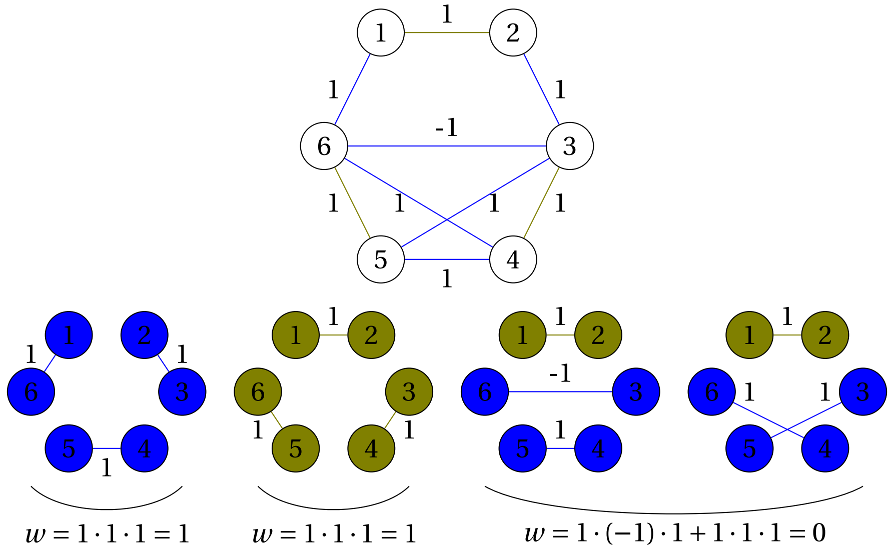
$w = 1 \cdot 1 \cdot 1 = 1$
$w = 1 \cdot 1 \cdot 1 = 1$
$w = 1 \cdot (-1) \cdot 1 + 1 \cdot 1 \cdot 1 = 0$

Figure 1.5: Example of a perfectly monochromatic graph G_k^w. With this mixed-edge colouring k and weighting w, we have $\tilde{c}(G,k,w) = 2$ because there are 2 feasible monochromatic vertex colourings in G_k^w. It can be shown that it is not possible to find a mixed-edge colouring k' and a weight attribution w' such that $\tilde{c}(G,k',w') > \tilde{c}(G,k,w)$. Therefore, $\tilde{c}(G) = 2$.

Observation 1.3.8 (Link between c and $\tilde{c}$)**.** For all graphs G, $\tilde{c}(G) \geq c(G)$.

Proof. The structure of this proof follows the one proposed by Chandran and Gajjala in [CG22]. Let $G_k = (V,E)$ be a monochromatic graph. Let $n_M(b)$ be the number of different monochromatic perfect matchings in G of the colour b. Now we assign the following weights to each edge e of G :

$$w(e) = \begin{cases} \left(\frac{1}{n_M(k(e))}\right)^{\frac{2}{|V|}} & \text{if there are PMs of the colour } k(e) \\ 1 & \text{if there are none} \end{cases}$$

Because G_k is monochromatic, we know that all of its perfect matchings are monochromatic. For each of these monochromatic perfect matching M, we have

$$w(M) = \left(\left(\frac{1}{n_M(k(M))}\right)^{\frac{2}{|V|}} \right)^{\frac{|V|}{2}} = \frac{1}{n_M(k(M))}$$

It follows that for each monochromatic vertex colouring κ of colour b,

$$w(\kappa) = n_M(b)\frac{1}{n_M(b)} = 1$$

Hence, G_k^w is perfectly monochromatic and $c(G) = \tilde{c}(G)$. Of course, this means that $\tilde{c}(G) \geq c(G)$. □

The reader now possesses all the needed tools to understand the Krenn's conjecture, which is the heart of this master thesis research.

Conjecture 1.3.9 (Krenn's conjecture)**.** Let G be a graph. If G is isomorphic to K_4, then $\tilde{c}(G) = 3$. Otherwise, $\tilde{c}(G) \leq 2$.

This last statement is, as its name says, a conjecture: it was not yet proven and no constant upper bound is currently known for the weighted matching index of an experiment graph. However, some special cases were already studied. These are analyse in the next chapter. But before getting lost in maths, the reader is encouraged to read the next section, which justifies the study of this conjecture and explains why any progress on it is important.

1.4 Motivations

While the Krenn's conjecture as defined in conjecture 1.3.9 may seem very theoretical at first sight, it is actually directly motivated by important questions in the field of quantum physics. To understand why the answer to this problem has a real impact, the reader must acquire some very basic knowledge of quantum physics. Please note, however, that a reader only interested in the graph theory question can pass this section, as it is not a prerequisite to understand the rest of the master thesis.

In 1935, A. Einstein, B. Podolsky and N. Rosen discovered that quantum physics theory implied a strange phenomenon considered as impossible. They observed for the first time that the theory predicts the ability of two particles' quantum states to depend on each other even if the particles are separated by an arbitrary distance [EPR35]. Such two particles are said to be in an *entangled* state. Furthermore, in 1964, J.S. Bell stated three inequalities that must be respected by any quantum theory in the hypothesis of a deterministic local theory using hidden variables [Bel64]. The phenomenon of quantum entanglement was breaking these inequalities, and many scientists were therefore very sceptical about it. It was specifically redefining the ancient vision of the locality principle, defended by Albert Einstein. However, quantum entanglement still doesn't allow information exchange at a speed greater than the speed of light [Sie20].

Since then, quantum entanglement has been verified experimentally [J12], proving the Bell's inequalities to be wrong and opening new exciting usages of the phenomenon, mainly in the field of quantum computation. In 1989, Greenberger, Horne and Zeilinger theorized for the first time special entangled states involving three particles [GHZ89]. These states are known today as GHZ-states, and were experimentally observed in 1999 [Bou+99]. Using the Dirac's notation [Dir39], a GHZ-state is formally defined as follows.

$$GHZ = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

A reader who's not familiar with the Dirac's formalism can interpret such a state as a system composed of 3 qubits that can't be described separately from each other. The whole system has eight classical states $\{\,|\,0\rangle, |\,1\rangle\}^3$, and is in a perfect mixed state between the classical states $|000\rangle$ and $|111\rangle$. In this base case, the involved particles have only two classical states, $|0\rangle$ and $|1\rangle$. Therefore, the system is said to be of dimension $d = 2$. Generalized GHZ-states accept other dimensions, and can involve more than $n = 3$ qubits.

$$GHZ_{n,d} = \frac{1}{\sqrt{d}}(|0\rangle^n + |1\rangle^n + \cdots + |d-1\rangle^n)$$

The use of photonic technologies allowed physicists to experimentally create GHZ-states [Wan+16]. However, it is still unknown if every GHZ-state can be created this way, it seems especially difficult to create higher dimensional states. In 2017, Mario Krenn studied this question and discovered an unexpected link between the ability to experimentally create high-dimensional GHZ-states and graph theory. Indeed, he could show that a large class of experiments could be represented as perfectly monochromatic graphs [KGZ17]. Furthermore, given an experiment that allows to create GHZ-states of dimension d involving n qubits and its corresponding perfectly monochromatic graph G:

$$\begin{cases} n & = & |V(G)| \\ d & = & \tilde{c}(G) \end{cases}$$

This lead to the formulation of the **Krenn's conjecture** 1.3.9 in graph theory, which remains unsolved [Kre]. Finding any constant bound on the perfect matching index of a perfectly monochromatic graph would allow physicists to know the limits of this class of experiments. On the other hand, finding counter-examples to the conjecture would lead to the ability to create higher dimensional GHZ-states.

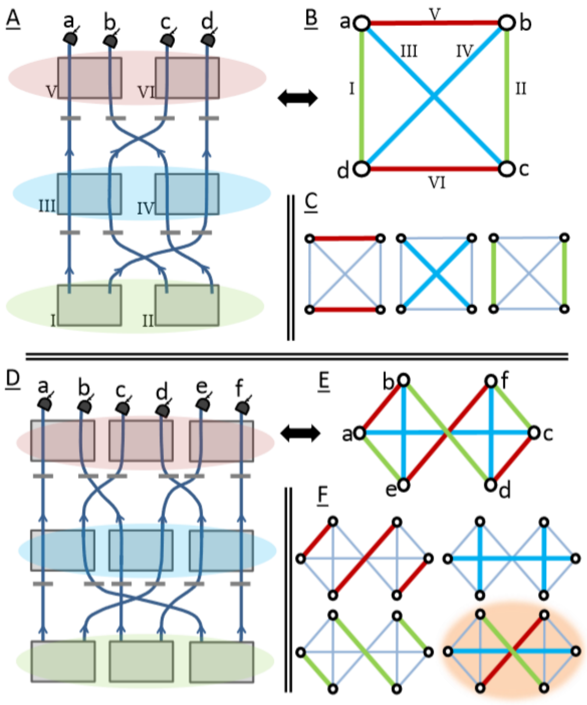

Figure 1.6: This figure was taken from Mario Krenn's paper of 2017 [KGZ17]. It represents two examples of experiments to create entangled states using photonic technologies. On one hand, the experiment described in A corresponds to the graph B that is monochromatic as shown in C. Therefore, it allows to create a GHZ-state of dimension 3 involving four particles $|\psi\rangle = GHZ_{4,3} = \frac{1}{\sqrt{3}}(|0000\rangle + |1111\rangle + |2222\rangle)$. On the other hand, the experiment described in D is represented by the graph in E, which is non-monochromatic as shown in F. Therefore, the state it creates is not a GHZ-state. $|\psi'\rangle = \frac{1}{2}(|000000\rangle + |111111\rangle + |222222\rangle + |121200\rangle)$.

At last, please note that the study of GHZ-states is considered as a fundamental question in quantum physics. For instance, Alain Aspect, John F. Clauser and Anton Zeilinger received in 2022 a Nobel Prize for their experiments on quantum entanglement, and their work was closely related to the topic [Sci22]. Finding new ways to experimentally create such states would open new intriguing gates in the fields of quantum computing [GK20] and cryptography [PHM18].

2 State of the art

During the last few years, the problem was studied by a few researchers around the world [KGZ17; Bog; CG22; CG23]. While none of them could find any constant upper bound on the weighted matching index of perfectly monochromatic graphs (defined in definitions 1.3.7 and 1.3.6), some special cases of the conjecture were proven to be true. Furthermore, non-constant bounds could also be found in more general cases of experiment graphs. This section aims to present these results. But before getting started, it could be interesting to introduce some useful definitions and tools that will be used a lot during the different seen proofs.

Definition 2.0.1 ((Non-)redundant experiment graph)**.** In the context of this master thesis, an experiment graph G_k^w as defined in definition 1.3.1 is said to be *non-redundant* if it satisfies all the following properties.

- $\forall e \in E(G), w(e) \neq 0$
- $\forall e \in E(G), \exists$ a perfect matching M such that $e \in M$
- $\forall (v_1, v_2) \in (V(G))^2, v_1 \neq v_2$, and $\forall$ colour pair (r, g), $\exists$ at most one unique edge e between v_1 and v_2 such that $k(e, v_1) = r$ and $k(e, v_2) = g$, using notations introduced in definition 1.1.4.

Otherwise, it is said to be *redundant*.

Definition 2.0.2 (Non-redundant induced graph)**.** Let G_k^w be a redundant experiment graph. Then its *non-redundant-induced graph* is the graph built by doing successively the following operations.

- $\forall e \in E(G)$ such that $w(e) = 0$, delete e.
- $\forall e \in E(G)$ that does not belong to any perfect matching M of G_k^w, delete e.

- $\forall (v_1, v_2) \in (V(G))^2, v_1 \neq v_2$ And $\forall$ colour pair (r, g), if there are multiple edges $e_1, e_2, \ldots, e_m$ such that

$$\forall i \in \{1, \ldots, m\},\ k(e_i, v_1) = r \text{ and } k(e_i, v_2) = g$$

  Then, replace them by one single edge $e = (v_1, v_2)$ that has the following properties.

$$k(e, v_1) = r \text{ And } k(e, v_2) = g$$
$$w(e) = \sum_{i=1}^{m} w(e_i)$$

This definition and denomination is entirely motivated by the following observation.

Observation 2.0.3 (Non-redundancy is enough)**.** Let G_k^w be a redundant experiment graph, and let $G'^{w'}_{k'}$ be its non-redundant induced graph. Then $\tilde{c}(G, k, w) = \tilde{c}(G', k', w')$.

Proof. To prove this last observation, the procedure will be to show that each of the transformations that was applied on G_k^w did not change its weighted matching index.

1. **Transformation 1:** $\forall e \in E(G)$ such that $w(e) = 0$, delete e.
   Let $e = \{v_1, v_2\}$ be a zero-weighted edge of G_k^w. At most, e can contribute to the weights of all the feasible vertex colourings κ such that $\kappa(v_1) = k(e, v_1)$ and $\kappa(v_2) = k(e, v_2)$. But since $w(e) = 0$, then all the perfect matchings M such that $e \in M$ have a weight $w(M) = 0$. Therefore, e's contribution to κ is null, and removing it doesn't change anything to the feasibility of κ.

2. **Transformation 2:** $\forall e \in E(G)$ that do not belong to any perfect matching M of G_k^w, delete e.
   Since e does not belong to any perfect matching M, its weight can not contribute to any feasible vertex colouring κ. Removing it has therefore no impact.

3. **Transformation 3:** $\forall (v_1, v_2) \in (V(G))^2, v_1 \neq v_2$ and $\forall$ colour pair (r, g), if there are multiple edges $e_1, e_2, \ldots, e_m$ such that

$$\forall i \in \{1, \ldots, m\},\ k(e_i, v_1) = r \text{ and } k(e_i, v_2) = g$$

   Then, replace them by one single edge $e = (v_1, v_2)$ that has the following properties.

$$k(e, v_1) = r \text{ and } k(e, v_2) = g$$
$$w(e) = \sum_{i=1}^{m} w(e_i)$$

Indeed, let κ be a feasible vertex colouring such that $\kappa(v_1) = r$ and $\kappa(v_2) = g$. We need to introduce some new notations for this specific part of the proof, which are inspired by the notations we previously defined in definition 1.3.3 of an induced vertex colouring.

- $\mathcal{M}_\kappa$ Is the set of all perfect matching M of G that induce a vertex colouring κ on G.
- $\mathcal{M}_\kappa^{e_i}$ Is the set of perfect matchings M of G that induce a vertex colouring κ on G such that $e_i \in M$.
- $\mathcal{M}_\kappa^*$ Is the set of perfect matchings M of G that have no edges from $\{e_1, \ldots, e_m\}$.
- $\mathcal{M}'_\kappa$ Is the set of all perfect matching of G' that induce a vertex colouring κ on G'.
- $\mathcal{M}'^{e}_{\kappa}$ Is the set of perfect matchings of G' that induce a vertex colouring κ on G' such that $e_i \in M$.
- $\mathcal{M}'^{*}_{\kappa} = \mathcal{M}_\kappa^*$ Is the set of perfect matchings M of G' that have do not contain e.

Using these new notations, and by the definition 1.3.5 of the weight of a vertex colouring,

$$\begin{aligned}
w(\kappa) &= \sum_{M\in\mathcal{M}_\kappa} w(M) \\
&= \sum_{i=1}^{m} \sum_{M\in\mathcal{M}_\kappa^{e_i}} w(M) + \sum_{M\in\mathcal{M}_\kappa^*} w(M) \\
&= \sum_{i=1}^{m} w(e^i) \sum_{M\in\mathcal{M}_\kappa^{e_i}} w(M\backslash e^i) + \sum_{M\in\mathcal{M}_\kappa^*} w(M) \\
&= w(e) \sum_{M\in\mathcal{M}'^{e}_{\kappa}} w(M\backslash e) + \sum_{M\in\mathcal{M}'^{*}_{\kappa}} w(M) \\
&= \sum_{M\in\mathcal{M}'_\kappa} w(M)
\end{aligned}$$

This result proves the transformation 3 did not have any influence on the weight of all induced vertex colourings.

□

This last observation is convenient since it allows researchers to focus on non-redundant experiment graphs to prove bounds on their matching index, and these bounds are still valid in redundant experiment graphs without any loss of generality. An example of such transformation is shown in figure 2.1.

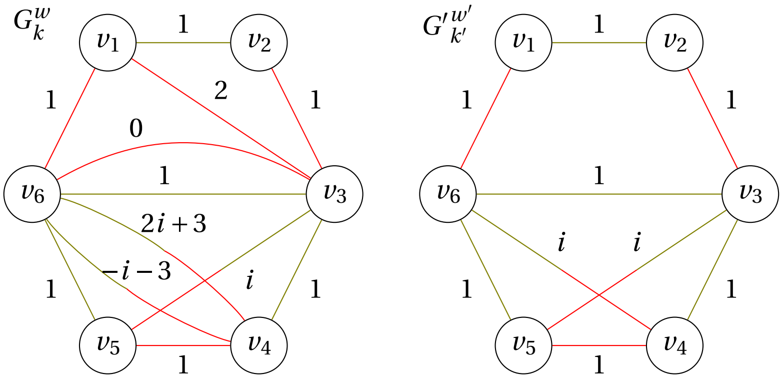

Figure 2.1: In this figure, G_k^w is an experiment graph and $G'^{w'}_{k'}$ is its non-redundant induced graph. In this particular case, it consisted in doing the following operations. First of all, the 0-weighted edge between v_3 and v_6 is removed. Secondly, the edge $\{v_1, v_3\}$ is also deleted because it does not belong to any perfect matching (indeed, including it in a perfect matching M would prevent M to cover v_2). Lastly, the two edges between v_4 and v_6 are combined in one single edge, since they have the same colour at each endpoint. The reader can verify that the resulting graph $G'^{w'}_{k'}$ has a weighted matching index $\tilde{c}(G', k', w') = 2$. Therefore, $\tilde{c}(G, k, w) = 2$ by observation 2.0.3.

2.1 Special cases that were already proven

2.1.1 Restrictions on the weights

Lemma 2.1.1 (Real, positive weights)**.** Let G_k^w be a perfectly monochromatic graph which has only positive, real weights. If G is isomorphic to K_4, then $\tilde{c}(G, k, w) \leq 3$. Otherwise, $\tilde{c}(G, k, w) \leq 2$.

Proof. This is a direct result from Bogdanov's proof, described in theorem 1.2.3 [Bog]. Because all the weights of G_k^w are real, positive numbers, it means that the weight of any perfect matching is positive. Therefore, the weight of any feasible vertex colouring is positive by definition 1.3.5. But all the non-monochromatic feasible vertex colourings in a perfectly monochromatic graph should have a weight of 0 by definition 1.2.1. It follows that G_k^w has no non-monochromatic perfect matching. Then $\tilde{c}(G, k, w) = c(G, k)$. □

2.1.2 Restrictions on the matching index

In 2022, Gajjala and Chandran analysed the Krenn's conjecture by separating the graphs in different subclasses according to their matching index (see definition 1.2.2). Here are their results.

Lemma 2.1.2 (Graphs with a matching index of 0)**.** If G is a graph with a matching index of 0, then the Krenn's conjecture is true for G and $\tilde{c}(G) = c(G) = 0$.

Proof. Let G be a graph with a matching index of 0. Then, G has no perfect matchings (otherwise it would be feasible to colour all the edges of G in the same colour and find a monochromatic perfect matching, which contradicts the fact that $c(G) = 0$ by definition 1.2.2 of a matching index). Because it has no perfect matchings, there is no mixed-edge colouring k and weight attribution w such that G_k^w has a feasible monochromatic vertex colouring. It follows that $\tilde{c}(G) = 0$. □

Lemma 2.1.3 (Graphs with a matching index of 2)**.** If G is a graph with a matching index of 2, then the Krenn's conjecture is true for G and $\tilde{c}(G) = c(G) = 2$.

This last lemma is proved by Chandran and Gajjala in [CG22]. The proof is not trivial and uses some very interesting observations about the structure of a perfectly monochromatic graph with $\tilde{c}(G) = 2$. The reader is highly encouraged to have a look at it in the original paper in order to believe this claim. Nevertheless, the proof will not be discussed here since it would be a bit redundant. Thanks to their proof, Chandran and Gajjala could pose the following theorem, which is a summary of the two last lemmas 2.1.2 and 2.1.3.

Theorem 2.1.4 (Chandran - Gajjala's theorem)**.** Let G be a perfectly monochromatic graph. If $c(G) \neq 1$, then the Krenn's conjecture is true and $\tilde{c}(G) = c(G)$.

Having that last theorem, it would be a natural question to ask ourselves if, for any graph G, $c(G) = \tilde{c}(G)$. Unfortunately, it is not the case, since a counter-example was found by Chandran and Gajjala. [CG22]

Observation 2.1.5. There exists a graph G that satisfies $c(G) = 1$ and $\tilde{c}(G) = 2$. This counter example is shown in figure 2.2.

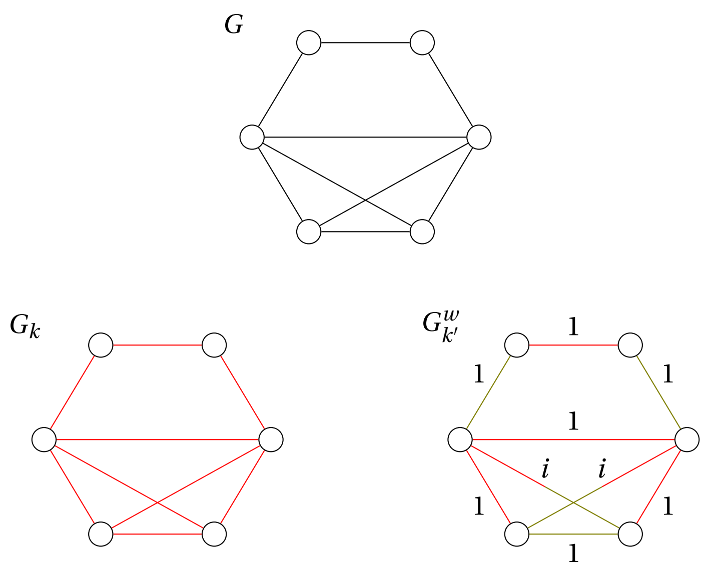

Figure 2.2: Example of graph G that has a matching index of 1 and a weighted matching index of 2. On this figure, the edge-colouring k is an example of edge colouring that gives $c(G,k) = 1$. Furthermore, the mixed-edge colouring k' and the weight attribution w are examples of mixed-edge colourings and weights attributions that result in $\tilde{c}(G,k',w) = 2$. Another example of mixed-edge colouring and weight attribution that gives the same matching index on G is available in figure 1.5. It has still to be shown that it is impossible to find other edge-colourings / weight attributions that would lead to bigger (weighted) matching indexes. This proof is not presented here, but is available in Chandran and Gajjala's original paper [CG22].

2.2 Known non-constant bounds on the weighted matching index

Despite the failure to find any constant bound on the weighted matching index of experiment graphs up to now, some interesting non-constant bounds were nevertheless found. Chandran and Gajjala discovered in [CG22] two interesting bounds in terms of minimum degree and edge connectivity that I considered to be relevant to rewrite here. The two following theorems are due to them.

Lemma 2.2.1 (Upper bound in terms of minimum degree)**.** Let G be a graph, and $\delta(G)$ be its minimum degree as defined in definition 1.1.12. Then $\tilde{c}(G) \leq \delta(G)$.

Proof. Let G_k^w be a perfectly monochromatic graph such that $\tilde{c}(G,k,w) = \tilde{c}(G)$. It is known by definition 1.3.7 of the weighted matching index that G_k^w has at least one monochromatic perfect matching for $\tilde{c}(G)$ different colour classes. In the context of this proof, these perfect matchings are denoted $M_1, M_2, \ldots, M_{\tilde{c}(G)}$. Because these perfect matchings are of different colours, they can not share any edge.

Let $v \in V(G)$ be a vertex of minimum degree $d(v) = \delta(G)$. This vertex must be covered by $M_1, M_2, \ldots, M_{\tilde{c}(G)}$. M_1 Covers it through an edge e_1, M_2 through an edge e_2, …, $M_{\tilde{c}(G)}$ through an edge $e_{\tilde{c}(G)}$ (as shown in figure 2.3). Then v has at least $\tilde{c}(G)$ outgoing edges, and $d(v) = \delta(G) \geq \tilde{c}(G)$. □

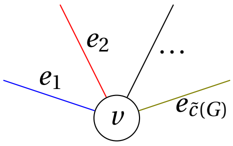

Figure 2.3: Visualization of the proof that $\tilde{c}(G) \leq \delta(G)$ for every perfectly monochromatic graph G.

Theorem 2.2.2 (Upper bound in terms of edge connectivity)**.** Let G be a graph, and $\lambda(G)$ be its edge connectivity as defined in definition 1.1.11. Then $\tilde{c}(G) \leq \lambda(G)$.

The proof of this last theorem is not trivial and uses some concepts that need to be properly defined here. It was first described by Chandran and Gajjala in [CG22], and we are using the same concepts than them here.

Definition 2.2.3 (Redundant edge)**.** An edge e from a graph G is said to be *redundant* if it does not belong to any perfect matching of G.

Definition 2.2.4 (Redundant colour class)**.** A colour class r from an edge colouring k of a graph G is said to be *redundant* if there is no monochromatic perfect matching of colour r in G_k.

Definition 2.2.5 (Redundant mixed-colour class)**.** A mixed-colour class (r, b) from an edge-colouring k of a graph G is said to be *redundant* if at least one of the two following conditions is true.

$$\begin{cases} r \text{ is a redundant colour class} \\ b \text{ is a redundant colour class} \end{cases}$$

Proof of Theorem 2.2.2. TODO : this proof can be simplified by using the concepts introduced at the beginning of the chapter. I'm currently working on it.

This theorem is proven by contradiction. Using the notations introduced in definitions 1.3.7 and 1.1.11, let's assume there exists a graph G'' such that $\tilde{c}(G'') \geq \lambda(G'') + 1$. By removing all the redundant edges (defined in definition 2.2.3) of G'', it results in a graph

G' with $\tilde{c}(G'') = \tilde{c}(G')$. This manipulation preserves the weighted matching index of the graph.

$$\tilde{c}(G') \geq \lambda(G'') + 1$$

Therefore, there exists a mixed-edge colouring k' and a weight attribution w' such that

$$\tilde{c}(G') = \tilde{c}(G', k', w') \geq \lambda(G'') + 1$$

From ${G'}_{k'}^{w'}$, let apply a new transformation that removes all edges that belong to a redundant colour class and all edges that belong to a redundant mixed-colour class (defined in definitions 2.2.4 and 2.2.5), obtaining G_k^w. It is straightforward to see that, if ${G'}_{k'}^{w'}$ was perfectly monochromatic, G_k^w is still perfectly monochromatic. Indeed, by removing all the edges from a redundant colour class, no monochromatic perfect matching is removed by definition 2.2.4. If there were (non-monochromatic) perfect matchings containing edges of that colour class, they are all destroyed, so there is no induced vertex colouring of G anymore that has vertices of the removed colour class. Then, it's not needed any more to worry about the weight of such vertex colourings.

$\tilde{c}(G)$ did not decrease, because no non-redundant colour class was removed from ${G'}_{k'}^{w'}$, and did not increase, since removing edges can not create new perfect matchings. So again:

$$\tilde{c}(G) = \tilde{c}(G') = \tilde{c}(G'')$$

We also notice that the edge-connectivity of a graph can not increase when we remove edges. Therefore

$$\begin{aligned} \tilde{c}(G) &\geq \lambda(G'') + 1 \\ &\geq \lambda(G') + 1 \\ &\geq \lambda(G) + 1 \end{aligned}$$

By definition 1.1.11 of the edge-connectivity of a graph, G can be cut in two disconnected parts S and S' by removing $\lambda(G)$ edges. Note that, because G has perfect matchings, $|V(G)|$ is an even number and therefore $|S|$ and $|S'|$ are of the same parity.

1. **If $|S|$ and $|S'|$ are odd** then, for all monochromatic perfect matching M, M contains at least one crossing edge, i.e. an edge with one endpoint in S and the other endpoint in S' (see figure 2.4).

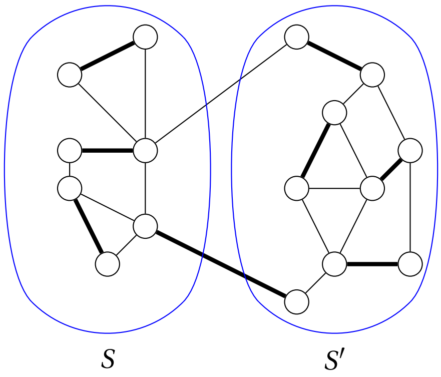

Figure 2.4: Existence of a crossing edge between S and S' in a perfect matching M of G if $|S|$ and $|S'|$ are odd.

Let $E(S,S')$ be the set of crossing edges from S to S'. The maximum number of different colour classes that can be found on $E(S,S')$ is $\lambda(G)$ (otherwise more edges wold be needed). The conclusion is that the monochromatic perfect matchings of G_k^w can be of a most $\lambda(G)$ different colours. Hence,

$$\tilde{c}(G'') = \tilde{c}(G) \leq \lambda(G) \leq \lambda(G'')$$

This forms a contradiction with the statement that $\tilde{c}(G'') \geq \lambda(G'') + 1$.

2. **If $|S|$ and $|S'|$ are even** then let's separate all the colour classes of G_k^w in

$$\begin{cases} [r] & = & \{1,2,\ldots,r\} & = & \text{all the colours such that none of the monochromatic} \\ & & & & \text{perfect matchings of these colours intersects } E(S,S') \\ [r+1,r+r'] & = & \{r+1,r+2,\ldots,r+r'\} & = & \text{all the colours such that there exists a monochromatic} \\ & & & & \text{perfect matching of this colour intersecting } E(S,S') \end{cases}$$

Clearly, $\tilde{c}(G'') = \tilde{c}(G) = r + r'$. For all colours $i \in [r+1, r+r']$, there exists a monochromatic perfect matching M which intersects $E(S,S')$. And since $|S|$ and $|S'|$ are even, M contains at least two edges from $E(S,S')$, as shown in figure 2.5.

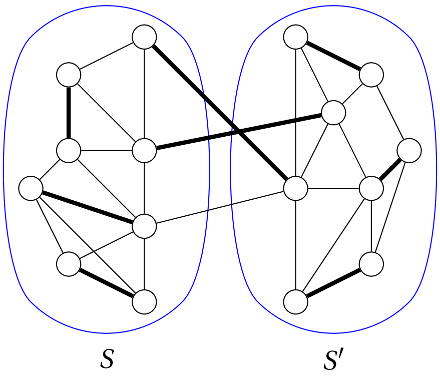

Figure 2.5: Existence of 2 crossing edges in a crossing perfect matching M of G if $|S|$ and $|S'|$ are even numbers.

Then there exist at least 2 edges of colour i in $E(S,S')$ for all $i \in [r+1, r+r']$. It follows that $\lambda(G'') \geq \lambda(G) \geq 2 \cdot r'$.

2.1 **If** $r \leq 1$ **and** $r' \geq 1$, then

$$\begin{aligned}\tilde{c}(G'') &= r + r' \\ &\leq 2 \cdot r' \\ &\leq \lambda(G'')\end{aligned}$$

2.2 **If** $r \leq 1$ **and** $r' = 0$, then it is trivial because G is connected.

$$\begin{aligned}\lambda(G'') &\geq 1 \\ &\geq r + r' \\ &= \tilde{c}(G'')\end{aligned}$$

2.3 **Else,** $r \geq 2$. Then one can pick two colours from $[r]$ (let say 1 and 2).

Claim. There should be at least two mixed edges of colour $(1,2)$ in $E(S,S')$.

Proof of the claim. By contradiction, suppose not. Let's consider 2 monochromatic perfect matchings M_1 and M_2 of colour 1 and 2 that induce the monochromatic vertex colourings $1_{V(G)}$ and $2_{V(G)}$ respectively. M_1 And M_2 do not have any edge in $E(S,S')$. In general, in the context of this proof, i_A will denote the monochromatic vertex colouring of colour i of the vertices in A. Because G_k^w is perfectly monochromatic, the weights of $1_{V(G)}$ and $2_{V(G)}$ must be 1. Since $1,2 \in [r]$, there are no normal edges of colour 1 nor 2 in $E(S,S')$. Then we have the following relations, where $w(i_A)$ denotes the weight of the monochromatic vertex colouring of colour i on the subgraph A.

$$\begin{aligned}w(1_{V(G)}) &= w(1_S) \cdot w(1_{S'}) &= 1 \\ w(2_{V(G)}) &= w(2_S) \cdot w(2_{S'}) &= 1\end{aligned}$$

Therefore, $w(1_S)$, $w(1_{S'})$, $w(2_S)$, $w(2_{S'})$ must be non-zeros.

Now let's consider a non-monochromatic vertex colouring κ in which S is coloured 1 and S' is coloured 2. κ is feasible because we can build a perfect matching by taking all the edges from M_1 on S and all the edges from M_2 on S'. Because G_k^w is perfectly monochromatic, $w(\kappa)$ is 0 by definition 1.3.6. Also, since $|S|$ and $|S'|$ are even, every perfect matching contains an even number of edges from $E(S,S')$. But by assumption, there is at most one crossing edge of colour $\{1,2\}$ and no crossing edges of colour 1 nor 2. In conclusion, no perfect matching that induces κ can contain an edge from $E(S,S')$. Then,

$$w(\kappa) = w(1_S) \cdot w(2_{S'}) = 0$$

But $w(1_S)$ and $w(2_{S'})$ are non-zeros, so this is impossible and creates a contradiction. This proves the claim. □

We proved our claim. Therefore, for a pair of colours $i, j \in [r]$, there should be at least two mixed edges of colour $\{i, j\}$ in $\{S, S'\}$.

$$\text{Minimum number of edges in } E(S,S') = 2\binom{r}{2} + 2 \cdot r' \leq \lambda(G'')$$

Finally, it means that

$$\begin{aligned} \tilde{c}(G'') &= r + r' \\ &\leq 2\binom{r}{2} + 2 \cdot r' \\ &\leq \lambda(G'') \end{aligned}$$

And this ends the proof.

□

This proof was interesting to rewrite here since it uses some very interesting reasoning about the structure of perfectly monochromatic graphs and the definitions of their weights. Some similar reasoning is sometimes reused in this master thesis in order to prove my own new results.

Last but not least, Chandran and Gajjala showed more recently an upper bound on the weighted matching index of graphs in terms of their number of vertices [CG23].

Theorem 2.2.6 (Upper bound in terms of number of vertices)**.** Let G_k^w be an experiment graph that has n vertices. Then,

$$\tilde{c}(G, k, w) \leq \frac{n}{\sqrt{2}}$$

The proof of theorem 2.2.6 is detailed in Chandran and Gajjala's publication [CG23].

3 New contributions

There are 2 intuitive ways to tackle the problem of the Krenn's conjecture. The first way, done in section 3.1, is to use a theoretical approach to find interesting properties about perfectly monochromatic graphs. In this approach, I hope that the conjecture is true and try to prove it in some very restricted cases. Also, I present 2 different ways of thinking to the problems by showing an equivalency between 2 cases in the subsection 3.1.1.

The second approach is a computational approach, and is presented in section 3.2. The goal of this second approach will be to generate a big number of random perfectly monochromatic experiment graphs, and extract conclusions from these data. To do so, it will present a new tool I developed, called EGPI, that aims to perform such experiments.

3.1 Theoretical approach

3.1.1 Problem reduction

Lemma 3.1.1 (PM graphs with integer weights)**.** Let G_k^w be a perfectly monochromatic graph that has only edge weights included in $\mathbb{Z}$. For all upper bounds $\beta \in \mathbb{N}$, if the following conjecture is true:

- $\forall {G'_{k'}}^{w'}$ perfectly monochromatic graphs that have only weights included in $\{-1, 1\}$, $\tilde{c}(G', k', w') \leq \beta$.

then we have also $\tilde{c}(G, k, w) \leq \beta$.

Proof. Let G_k^w be a perfectly monochromatic graph that has only edge weights included in $\mathbb{Z}$. We will show that we can build a perfectly monochromatic graph that has only weights included in $\{-1, 1\}$ and that has the same weighted matching index than G_k^w.

First, we choose an edge in G_k^w that has a weight different from 1 nor -1 (if such an edge doe not exist, we are done). We replace this edge by $|w(e)|$ parallel edges of the same colour, and that have all a weight of $\frac{w(e)}{|w(e)|}$ (1 if $w(e) > 0$, -1 if $w(e) < 0$). Remember this can be done because experiment graphs allow multi edges by definition 1.3.1. This creates a new graph ${G'_{k'}}^{w'}$. That process is illustrated in figure 3.1.

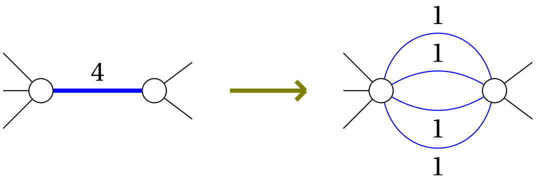

Figure 3.1: Illustration of the transformation from an experiment graph with integer weights to an experiment graph with weights included in $\{-1, 1\}$.

Let κ be a feasible vertex colouring of G_k^w. What is the weight of κ in G_k^w ? To express it, we will denote by

$$\left\{\begin{array}{ll} M_\kappa & \text{the set of perfect matchings of } G_k^w \text{ that induce the vertex colouring } \kappa \\ M_\kappa^e & \text{the set of perfect matchings of } G_k^w \text{ that induce the vertex colouring } \kappa \text{ and contain } e \\ M_\kappa^{\neg e} & \text{the set of perfect matchings of } G_k^w \text{ that induce the vertex colouring } \kappa \text{ and do not contain } e \end{array}\right.$$

The weight of κ in G_k^w is

$$\begin{aligned} w(\kappa \text{ in } G_c^w) &= \sum_{M \in M_\kappa} w(M) \\ &= \sum_{M \in M_\kappa^{\neg e}} w(M) + \sum_{M \in M_\kappa^e} w(M) \end{aligned}$$

The next step, which is the heart of the proof, consists of computing the weight of the vertex colouring κ in ${G'_{k'}}^{w'}$. We will denote by

$$\left\{\begin{array}{ll} M'_\kappa & \text{the set of perfect matchings of } {G'_{k'}}^{w'} \text{ that induce the vertex colouring } \kappa \\ {M'_\kappa}^{e} & \text{the set of perfect matchings of } {G'_{k'}}^{w'} \text{ that induce the vertex colouring } \kappa \\ & \text{and contain an edge that was derived from } e \\ {M'_\kappa}^{\neg e} & \text{the set of perfect matchings of } {G'_{k'}}^{w'} \text{ that induce the vertex colouring } \kappa \\ & \text{and does not contain an edge that was derived from } e \end{array}\right.$$

In addition to these concepts, for every perfect matching $M \in M_\kappa^e$, let $\mathscr{M}'(M)$ be the set of corresponding perfect matchings M' in ${M'_\kappa}^{e}$. In other words, $\mathscr{M}'(M)$ denotes the perfect matchings that are the same as M on every edge except e, and that contain one of the edges that were added when e was removed. It follows that, for every $M \in M_\kappa^e$, $|\mathscr{M}'(M)| = w(e)$. Also, given $M \in M_\kappa^e$, for each perfect matching $M' \in \mathscr{M}'(M)$, $w(M') = \frac{w(M)}{|w(e)|}$. Finally, we notice the following relations between the different sets we defined :

$$\begin{cases} {M'}_\kappa^{\neg e} & = & M_\kappa^{\neg e} \\ {M'}_\kappa^{e} & = & \bigcup\limits_{M \in M_\kappa^e} \mathscr{M}'(M) \end{cases}$$

Having all these observations in mind, we can now compute the weight of κ in ${G'}_{k'}^{w'}$.

$$\begin{array}{rcllll} w(\kappa \text{ in } {G'}_{k'}^{w'}) & = & \sum\limits_{M' \in M'_\kappa} w(M') & & & \\ & = & \sum\limits_{M' \in {M'}_\kappa^{\neg e}} w(M') & + & \sum\limits_{M' \in {M'}_\kappa^{e}} w(M') \\ & = & \sum\limits_{M \in {M_\kappa}^{\neg e}} w(M) & + & \sum\limits_{M \in M_\kappa^e} \left(\sum\limits_{M' \in \mathscr{M}'(M)} w(M') \right) \\ & = & \sum\limits_{M \in {M_\kappa}^{\neg e}} w(M) & + & \sum\limits_{M \in M_\kappa^e} \left(\sum\limits_{M' \in \mathscr{M}'(M)} \frac{w(M)}{|w(e)|} \right) \\ & = & \sum\limits_{M \in {M_\kappa}^{\neg e}} w(M) & + & \sum\limits_{M \in M_\kappa^e} \left(|w(e)| \frac{w(M)}{|w(e)|} \right) \\ & = & \sum\limits_{M \in {M_\kappa}^{\neg e}} w(M) & + & \sum\limits_{M \in M_\kappa^e} w(M) \\ & = & w(\kappa \text{ in } G_k^w) & & \end{array}$$

So, since the weight of each feasible vertex colouring in G_k^w remains unchanged in ${G'}_{k'}^{w'}$, the monochromatic feasible vertex colourings have still a weight of 1, and the non-monochromatic feasible vertex colourings have still a weight of 0. So, ${G'}_{k'}^{w'}$ is still perfectly monochromatic and $\tilde{c}(G', k', w') = \tilde{c}(G, k, w)$. Now, we can rename G' as G and repeat the whole procedure while G_k^w has still edges with a weight different from $\{-1, 1\}$. The resulting graph has only edges that have signed unitary weights and has the same weighted chromatic index as the initial graph. So if any upper bound can be found on the weighted matching index of the final graph with signed unitary weights, it will still be valid for the weighted matching index of the initial one, with integer weights. □

The implication of this lemma is that there are actually 2 ways to reason about the Krenn's conjecture when we are interested in integer weights. The first way is to consider only non-redundant graphs (as defined in definition 2.0.2) and to try to find a bound on their weighted matching index. And the second way is to consider redundant graphs in which each edge has a weight included in $\{-1, 1\}$. Every result discovered in the second approach can be translated to the first approach, and vice versa.

3.1.2 Constraints' relaxation

As it was explained in the introduction, a simplified version of the conjecture was already proven thanks to Bogdanov.[Bog] This version, presented in lemma 2.1.1, is only valid when all the weights of a perfectly monochromatic graph G_k^w are positive. In this section, our main goal will be to relax these constraints.

Allowing one negative edge

Since the conjecture is proven to be true when all the weights are positive, it is natural to ask ourselves how the proof would be affected if this constraint was relaxed. The most simple case is the one where one edge is allowed to have a negative weight. The goal of this section is to answer that question. Let's begin by remarking the 2 following observations.

Observation 3.1.2 (Existence of a Hamiltonian cycle)**.** Let G_k^w be a perfectly monochromatic graph that respects the following properties.

- G_K^w is a simple graph, in the sense it has no multi-edges.
- G_K^w has no bicoloured edges.
- G_k^w has a weighted matching index $\tilde{c}(G,k,w) \geq 3$.
- It has at least 2 colour classes, denoted r, and g, such that all the edges coloured r or g have a real, positive weight.

Let M_r and M_g be 2 monochromatic perfect matchings of G_k^w coloured r and g respectively. Then, the union of M_r and M_g forms a Hamiltonian cycle of even length.

Proof. Since M_r and M_g are disjoint, they form a disjoint union of $\mathscr{C}$ cycles of even length. If $\mathscr{C} \geq 2$, we will denote by C_i the i^{th} cycle. Then, we can build the following non-monochromatic perfect matching :

$$N = (C_1 \cap M_r) \cup \left(\bigcup_{i=2}^{\mathscr{C}} C_i \cap M_g \right)$$

The construction of N is highlighted in figure 3.2.

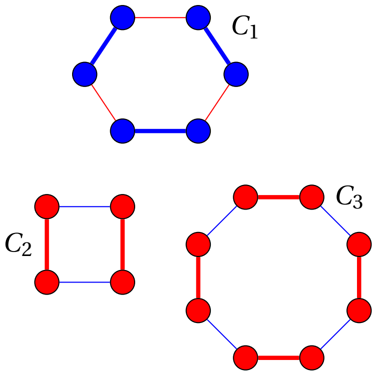

Figure 3.2: In this example, the non-monochromatic perfect matching N (represented by thick edges) is constructed from a red perfect matching and a blue one. The induced vertex colouring is also visible.

Since M_r and M_g include no negatively weighted edge, $w(N) > 0$. But, by definition 1.3.6 of a perfectly monochromatic graph, and using the notations introduced in definitions 1.3.2 and 1.3.4,

$$w(\kappa(N)) = \sum_{N_i \in \mathcal{M}_{\kappa(N)}} N_i = 0$$

Therefore, we know that $\exists N'$ such that $\kappa(N') = \kappa(N)$ and $w(N') < 0$. This is impossible, because the only way for N' to have a negative weight is to include at least one negative edge. And negative edges don't exist in colour r nor g. We can conclude that $\mathscr{C} = 1$, which means that M_r and M_g form a Hamiltonian cycle. □

Observation 3.1.3 (Parity of crossing edges)**.** Let G_k^w be a perfectly monochromatic graph that respects the following properties.

- G_K^w is a simple graph, in the sense it has no multi-edges.
- G_K^w has no bicoloured edges.
- G_k^w has a weighted matching index $\tilde{c}(G,k,w) \geq 3$.
- It has at least 2 colour classes, denoted r, and g, such that all the edges coloured r or g have a real, positive weight.

Let M_r, M_g be 2 monochromatic perfect matchings of colours r and g respectively. Let $H = (v_1, \ldots, v_n)$ be the Hamiltonian cycle of G_k^w formed by M_r and M_g. Let $e = (v_i, v_j)$ be an edge whose colour is b. Then, $j - i$ is even.

Proof. Let's assume by contradiction that $j - i$ is odd. Without loss of generality, we can assume that the colour of (v_i, v_{i+1}) is r (otherwise, inverse colours r and g in the following arguments). Then the following non-monochromatic perfect matching can be built.

$$N = e \cup (M_g \cap H_{i+1,j-1}) \cup (M_r \cap H_{j+1,i-1})$$

The construction of the non-monochromatic perfect matching N is shown in figure 3.3.

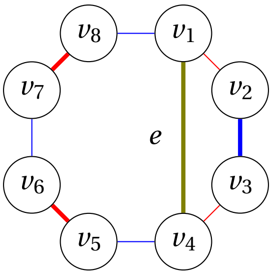

Figure 3.3: Construction of N from M_r, M_g and e if $j - i$ is odd.

- If $w(e) > 0$, then $w(N) > 0$ by definition 1.3.2. But $w(\kappa(N)) = \sum_{N_i \in \mathcal{M}_{\kappa(N)}} N_i = 0$ by definitions 1.3.7 and 1.3.6 (using a notation introduced in definition 1.3.3). Therefore, $\exists$ another non-monochromatic perfect matching N' such that $\kappa(N') = \kappa(N)$ and $w(N') < 0$. To satisfy this constraint, $e \in N'$. But e is the only edge that has a colour different from r nor g in N', which means that the sign of $w(e)$ determines the sign of $w(N')$ by definition 1.3.2. Therefore, $w(N')$ can not be negative. This is a contradiction.

- If $w(e) < 0$, then $w(N) < 0$ by definition 1.3.2. But $w(\kappa(N)) = \sum_{N_i \in \mathcal{M}_{\kappa(N)}} N_i = 0$ by definitions 1.3.5 and 1.3.6 (using a notation introduced in definition 1.3.3). Therefore, $\exists$ another non-monochromatic perfect matching N' such that $\kappa(N') = \kappa(N)$ and $w(N') > 0$. To satisfy this constraint, $e \in N'$. But e is the only edge that has a colour different from r nor g in N', which means that the sign of $w(e)$ determines the sign of $w(N')$ by definition 1.3.2. Therefore, $w(N')$ can not be positive. This is a contradiction.

□

These observations may seem obscure at first, but they are necessary to prove the following lemma.

Lemma 3.1.4 (One negative edge allowed)**.** Let G_k^w be a perfectly monochromatic graph that respects the following properties.

- G_K^w is a simple graph, in the sense it has no multi-edges.
- G_K^w has no bicoloured edges.
- G_k^w has a weighted matching index $\tilde{c}(G,k,w) \geq 3$.
- $\forall e \in E(G_k^w)$, $w(e) \in \mathbb{R}$. Also, $E(G_k^w)$ has at most one edge that has a negative weight.

Then, if G_k^w is not isomorphic to K_4, $\tilde{c}(G,k,w) \leq 2$.

The sketch of the proof of lemma 3.1.4 goes as follow. Using observations 3.1.2 and 3.1.3, we build a Hamiltonian cycle that has two colours and build non-monochromatic perfect matchings in it that use a negative edge e. We then show that they create a disbalance in the weight of their feasible vertex colouring that can not be counterbalanced with another perfect matching. Let's dive into it.

Proof of lemma 3.1.4. Let assume by contradiction that the weighted matching index of G_k^w $\tilde{c}(G,k,w) \geq 3$.

1. If G_k^w has only positive weights, then we're done — this case is already solved by Bogdanov in [Bog] and was presented in the lemma 2.1.1 of this master thesis.

2. If G_k^w has exactly one negative weight: let M_r, M_g and M_b be three distinct monochromatic perfect matchings of G_k^w that have colour r, g and b respectively. They exist by definition 1.3.7 of the weighted matching index. Let e^- be the only negatively weighted edge of G_k^w. Without loss of generality, I will say that the colour of e^- is b. From observation 3.1.2, M_r and M_g form a Hamiltonian cycle $H = (v_1, v_2, \dots, v_n)$ of even length. In this new vertex ordering, $e^- = (v_i, v_j)$. We know from observation 3.1.3 that $j - i$ is even. Without loss of generality, I can assume that the color of (v_i, v_{i+1}) is r (otherwise we exchange colours r and g in the following reasoning). Let $e = (v_{i+1}, v_k) \in M_b$ (we are certain of the existence of e because v_{i+1} must be covered by M_b). Now, three situations can occur.

   2.1 If $v_k = v_j$: this situation is impossible from observation 3.1.3, because $k - (i + 1)$ is odd.

   2.2 If v_k appears in an edge from the arc $H_{j+1,i-1}$ (using the notation introduced in definition 1.1.9 of an arc), then we can form a non-monochromatic perfect matching as follows.

$$\begin{aligned} N &= e^- \\ &\cup e \\ &\cup (H_{j+1,k-1} \cap M_g) \\ &\cup (H_{i+2,j-1} \cap M_r) \\ &\cup (H_{k+1,i-1} \cap M_r) \end{aligned}$$

   This works because we are sure that $k - (i + 1)$ is even from observation 3.1.3. Since $e^- \in N$, $w(N) < 0$ by definition 1.3.2. But $w(\kappa(N)) = \sum_{N_i \in \mathcal{M}_{\kappa(N)}} N_i = 0$ by definitions 1.3.5 and 1.3.6. Therefore, $\exists$ a non-monochromatic perfect matching N' such that $\kappa(N') = \kappa(N)$ and $w(N') > 0$. This is possible only if $e^- \notin N'$. The only vertices that get colour b in $\kappa(N) = \kappa(N')$ are v_i, v_{i+1}, v_j and v_k. The only way to match these vertices in N' with b-coloured edges without using e^- is that $\exists e' = (v_i, v_k)$ and $e'' = (v_{i+1}, v_j)$ that have colour b and that are included in N'. But this is impossible, because $k - i$ and $j - (i + 1)$ are odd numbers, which is forbidden by observation 3.1.3. This reasoning is summarized in figure 3.4.

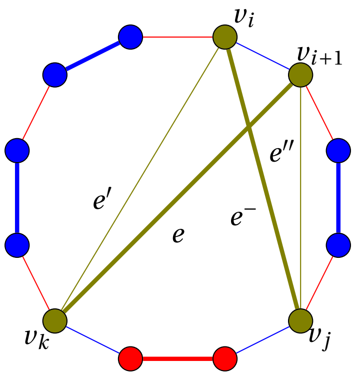

Figure 3.4: Illustration of the reasoning in case 2.2. In this figure, N is represented by thick edges, and the induced vertex colouring from N is visible. The edges e' and e'' that should be in N' are also represented. It is clear in this figure that e' and e'' can't exist using the parity argument.

2.3 If v_k appears in an edge from $H_{i+1,j-1}$, then it is possible to find an edge of M_b, called $e' = (v_l, v_m)$, that has both endpoints in $H_{i+1,k}(H)$, and such that the edge $e'' = (v_{l+1}, v_n) \in M_b$ has its v_n-endpoint in $H_{m+1,l-1}(H)$. This is true because it is known from observation 3.1.3 that $m - l$ is even. Actually, e' might be equal to e. Assuming without loss of generality that the colour of (v_l, v_{l+1}) is r, we find a new non-monochromatic perfect matching.

$$\begin{aligned} N &= e' \\ &\cup e'' \\ &\cup (H_{l+2,m-1} \cap M_r) \\ &\cup (H_{n+1,l-1} \cap M_r) \\ &\cup (H_{m+1,n-1} \cap M_g) \end{aligned}$$

The construction of N is illustrated in figure 3.5.

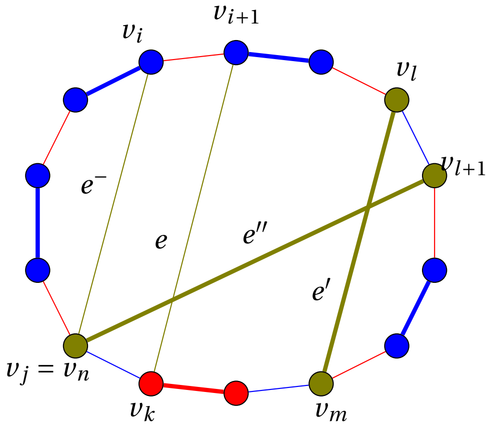

Figure 3.5: Illustration of the reasoning in case 2.3. In this figure, N is represented by thick edges, and the induced vertex colouring from N is visible. It this particular case, $v_j = v_n$. Nevertheless, it is impossible for v_i and v_j to be both green in $\kappa(N)$.

Since $e^- \notin N$, $w(N) > 0$ by definition 1.3.2. But $w(\kappa(N)) = \sum_{N_i \in \mathcal{M}_{\kappa(N)}} N_i = 0$ by definition 1.3.5 and 1.3.6. Therefore, ∃ another non-monochromatic perfect matching N' such that $\kappa(N') = \kappa(N)$ and $w(N') < 0$. For this condition to be satisfied, $e^- \in N'$. But it is impossible that both v_i and v_j get colour 3 in $\kappa(N) = \kappa(N')$ (only one of them maximum). This means that e^- can't be in N', which forms a contradiction.

□

Allowing all the classes to have arbitrary weights except 2

The main argument of the proof of the previous analysed case was that 2 colour classes had only positive weighted edges. This suggests that the structure of the proof might work as well if more than one single negatively weighted edge was present in the combination of the other colour classes. Such a result would be even more powerful since it would prove the conjecture to be true whenever 2 colour classes have only positive weighted edges, no matther the weights of the other colour classes. In this section, we will reuse the arguments from section 3.1.2 to verify them in the situation where multiple negative edge-weights are allowed.

Lemma 3.1.5 (2 positive colour classes)**.** Let G_k^w be a perfectly monochromatic graph that respects the following properties.

- G_K^w is a simple graph, in the sense it has no multi-edges.
- G_K^w has no bicoloured edges.
- ∃ two colour classes r and g such that all the edges coloured r or g have a real, positive weight.

Then, if G_k^w is not isomorphic to K_4, $\tilde{c}(G, k, w) \leq 2$.

Proof. By contradiction, let assume that $\tilde{c}(G, k, w) \geq 3$. Let M_r, M_g and M_b be 3 distinct monochromatic perfect matchings of G_k^w that have colour r, g and b respectively, and let assume that the colour classes r and g have only positive weighted edges. From observation 3.1.2, M_r and M_g form a Hamiltonian cycle $H = (v_1, v_2, \dots, v_n)$ of even length. We know from observation 3.1.3 that $\forall e = (v_i, v_j) \in M_b$, $j - i$ is even. Without loss of generality, let's assume that the color of (v_i, v_{i+1}) is r (otherwise we exchange colours r and g in the following arguments). Let $e = (v_i, v_j) \in M_b$ be a minimal crossing edge, i.e. be such that $e' = (v_{i+1}, v_k) \in M_b$ has its v_k endpoint in $H_{j+1,i-1}$. It is then possible to find a non-monochromatic perfect matching as follows.

$$\begin{aligned} N &= e \\ &\cup e' \\ &\cup M_1 \cap P_{i+2,j-1}(H) \\ &\cup M_1 \cap P_{k+1,i-1}(H) \\ &\cup M_2 \cap P_{j+1,k-1}(H) \end{aligned}$$

The construction of N is visualized in figure 3.6.

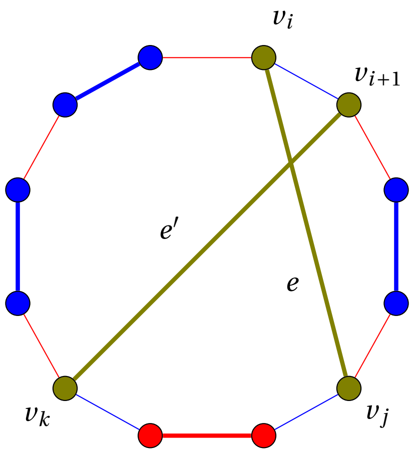

Figure 3.6: Construction of N from M_1, M_2, e and e'. N is represented by the thick edges. $\kappa(N)$ is also represented.

Centering the analysis around the potential signs of $w(e)$ and $w(e')$, 2 situations can occur.

1. if $w(e)$ and $w(e')$ have the same sign: then $w(N) > 0$ by definition 1.3.2. But $w(\kappa(N)) = \sum_{N_i \in \mathcal{M}_{\kappa(N)}} N_i = 0$ by definitions 1.3.5 and 1.3.6. This means that $\exists N'$ such that $\kappa(N') = \kappa(N)$ and $w(N') < 0$. Since the sign of N' is defined by the sign of its b-coloured edges, it can not contain e and e'. This last condition is satisfied only if $\exists e'' = (v_i, v_k)$ and $e''' = (v_{i+1}, v_j)$ of colour b. But this is forbidden by observation 3.1.3 because $(k-i)$ and $\big(j-(i+1)\big)$ are odd numbers.

2. if $w(e)$ and $w(e')$ have different signs: then $w(N) < 0$ by definition 1.3.2. But $w(\kappa(N)) = \sum_{N_i \in \mathcal{M}_{\kappa(N)}} N_i = 0$ by definitions 1.3.5 and 1.3.6. This means that $\exists N'$ such that $\kappa(N') = \kappa(N)$ and $w(N') > 0$. Since the sign of N' is defined by the sign of its b-coloured edges, it can not contain e and e'. This last condition is satisfied only if $\exists e'' = (v_i, v_k)$ and $e''' = (v_{i+1}, v_j)$ of colour b. But this is forbidden by observation 3.1.3 because $(k-i)$ and $\big(j-(i+1)\big)$ are odd numbers. This ends the proof of lemma 3.1.5.

□

3.2 Computational approach

In order to find a counter-example to the Krenn's conjecture in the best case scenario, or just to find interesting new properties in perfectly monochromatic graphs, having a more experimental approach has many interests. In this section, I am introducing a new tool I developed, called EGPI, that computes the weighted matching index of experiment graphs. Then, some of its potential uses are presented.

3.2.1 Motivation

Finding interesting examples of experimental graphs to study can be challenging due to the high degree of freedom experiment graphs can have, by definition 1.3.1. Also, the weighted matching index of experiment graphs as defined in definition 1.3.7 is hard to find by hand since it requires finding all perfect matchings of a graph. In my knowledge, no public access tool exists at the time of writing to easily encode experimental graphs and compute their weighted matching index. My hope is that such a tool could help me and other researchers to quickly verify some properties on instances of graphs they want to test, without losing more time on it. For this reason, I developed a new program that does exactly that. The program is called **EGPI**, which stands for **Experiment Graphs Properties Identifier**. I believe this name resumes the whole meaning of its utilization. The program is available on my Github repository [Han24].

3.2.2 Limitations of the experimental approach

Assuming that we successfully find a way to compute in a quick time the weighted matching index of an experiment graph, we still have to face some limitations. It may be tempting to use such a tool to prove the conjecture in some very restrained cases by generating and testing every possible experiment graph that respects some properties. While this idea is theoretically possible, the user must be aware that he is extremely limited by the computational time of such an algorithm. Indeed, the number of possible graphs grows exponentially with their degrees of freedom.

Let's consider a quick and simple example by trying to prove experimentally the following conjecture.

Conjecture 3.2.1. Let G_k^w be a non-redundant experiment graph of size $n = 6$ vertices, and let say that, for all edge $e \in E(G)$,

$$\begin{cases} Re(w(e)) & \in \{-1,0,1\} \\ Im(w(e)) & \in \{-1,0,1\} \\ w(e) & \neq 0 \end{cases}$$

Then, the weighted matching index $\tilde{c}(G,k,w) \leq 2$.

One way to prove the conjecture 3.2.1 is to use a brute force algorithm that generates every possible graph respecting the pre-conditions of the conjecture 3.2.1 with bicoloured edges that can take their colours in 3 defined colours (r,g,b), and check their weighted matching index. If none of their weighted matching index is 3, then the conjecture is true. Unfortunately, this is not possible because of the following observation.

Observation 3.2.2. Let G_k^w be an experiment graph that has n vertices, and let say that, for all edge $e=(u,v) \in E(G)$, the edge can have a value different from 0 chosen among W different values. Furthermore, $k(e,u)$ and $k(e,v)$ are chosen among K different colours. At last, $\forall e' = (u,v), k(e') \neq k(e)$. Then, the number of different possible graphs that G_k^w can be, denoted N_{graphs}, is

$$N_{graphs} = (1+W)^{\left(\frac{K^2 \cdot n \cdot (n-1)}{2}\right)}$$

Proof. Indeed, each edge e is characterized by the following properties.

- It has a position $\{u,v\}$, that can take $\frac{n\cdot(n-1)}{2}$ different values.
- It has 2 colours, one at each of its endpoint, that can take K different values. Then, in total, there are K^2 different ways to bicolour it.
- It has a complex weight that can have W different value.

A first observation is that the maximum number of edges between 2 different nodes of $V(G)$ is K^2. Indeed, more edges would imply that 2 edges at least would have the same positions and colours, which is forbidden.

Then, we observe that there are W^x possibilities to assign weights to an edge set of size x. Using these 2 last observations, it is possible to compute the number of different weighted bicoloured edges sets between 2 defined nodes u and v. Let's denote this number N_{edges}.

$$\begin{aligned} N_{edges} &= \binom{K^2}{0}W^0 + \binom{K^2}{1}W^1 + \binom{K^2}{2}W^2 + \cdots + \binom{K^2}{K^2}W^{K^2} \\ &= \sum_{x=0}^{K^2} \binom{K^2}{x} W^x \end{aligned}$$

It is possible to simplify this expression by using the binomial theorem [Wik].

$$\begin{aligned} N_{edges} &= \sum_{x=0}^{K^2} \binom{K^2}{x} W^x \\ &= \sum_{x=0}^{K^2} \binom{K^2}{x} 1^{(K^2-x)} W^x \\ &= (1+W)^{K^2} \end{aligned}$$

Having this number, it is easy to find the number N_{graphs} of possible experiment graphs. Indeed, we just have to choose one of the possible combinations of weighted bicoloured edges between 2 defined nodes for each possible pair of nodes. The number of possible pairs of nodes is $\frac{n\cdot(n-1)}{2}$. Therefore:

$$\begin{aligned} N_{graphs} &= N_{edges}^{\frac{n\cdot(n-1)}{2}} \\ &= \left((1+W)^{K^2}\right)^{\frac{n\cdot(n-1)}{2}} \\ &= (1+W)^{\left(K^2\cdot\frac{n\cdot(n-1)}{2}\right)} \end{aligned}$$

□

Returning to conjecture 3.2.1, let's compute as an example the number of graphs needed to verify to prove it by a brute force algorithm. The graphs described in conjecture 3.2.1 have $n = 6$ vertices, $K = 3$ different possible colours per edge's endpoint, and $W = 8$ possible weights different from 0. Therefore, in this case,

$$\begin{aligned} N_{graphs} &= (1+W)^{\left(\frac{K^2\cdot n\cdot(n-1)}{2}\right)} \\ &= (1+8)^{\left(\frac{3^2\cdot 6\cdot(6-1)}{2}\right)} \\ &= 9^{270} \end{aligned}$$

This number is absurdly big, and it is totally impossible to imagine being able to generate all of those graphs on a classical computer and compute their matching index, no matter of how efficient this operation is. Therefore, conjecture 3.2.1 cannot be solved by using a brute force algorithm, even if the graphs it is about are tiny and restricted compared to the space of all experiment graphs possible.

This observation leads us to consider that EGPI, in its implementation, should not try to generate all possible graphs. Instead, it should focus on generating as wisely as possible random graphs that have a high probability to have a big weighted matching index. This is the approach I took in the development of EGPI, and it is detailed in the next subsections.

3.2.3 Functionalities of EGPI

At the current state of its development, EGPI implements functions that allow the user to do the following:

1. **Discover all the perfect matchings of an encoded experiment graph**. The perfect matchings are encoded as edge sets. Their encoding includes access to their weights and to the feasible vertex colouring they induce.

2. **Discover all the feasible vertex colourings of an encoded experiment graph**. The set of all feasible vertex colourings of a graph is encoded as a Python dictionary, where the vertex colourings (Python tuples) are the keys and their weights (complex numbers) are the values.

3. **Find out if an encoded experiment graph is perfectly monochromatic or not.**

4. **Compute the weighted matching index of an encoded experiment graph.**

5. **Find out if an encoded experiment graph is bipartite.**

6. **Save an encoded experiment graph and its properties in a JSON file.** The JSON file contains all the properties of the graph, including its perfect matchings, its feasible vertex colourings, its weighted matching index, and the fact it is bipartite or not.

7. **Draw an encoded experiment graph in a pdf file.** This is done through the creation of a tex file containing the LaTeX representation of the graph. This LaTeX file is still available after the program ends, so that its code can be re-used if wanted so.

8. **Randomly create candidate experiment graph**, a process that can be defined as follows.

   Definition 3.2.3 (Random creation of a candidate experiment graph)**.** Given an even number of nodes $n \in \mathbb{N}$, a list of colours $L_{colours}$, a list of complex numbers $L_{weights}$, and a complexity bound $b \in \mathbb{N}$, the *random creation of a candidate experiment graph* respecting n, $L_{colours}$, $L_{weights}$ and b is the process described in algorithm 1.

   The denomination *candidate* experiment graph comes from the fact that graphs built that way have higher chances to have big weighted matching index than completely random graphs. Also, this construction ensures that each of the edges included in a candidate graph is part of a perfect matching. Furthermore, they do

Algorithm 1 Random creation of a candidate experiment graph

```
Require: $n$ is a positive even integer
Require: $L_{colours}$ is a list of colours
Require: $L_{weights}$ is a list of complex numbers different from 0
Require: $b \in \mathbb{N}$
  $G_k^w \leftarrow$ an empty graph with $n$ vertices and no edge
  for $k \in L_{colours}$ do
      $S \leftarrow \frac{n}{2}$ non-touching edges at random positions of $G_k^w$
      Colour each edge $\in S$ with colour $k$
      Assign random weights $\in L_{weights}$ to each edge $\in S$
      Add $S$ to $E(G_k^w)$          ▷ These edges form a monochromatic perfect matching
  end for
  for $i \in \{1, 2, \ldots, b\}$ do
      $S \leftarrow \frac{n}{2}$ non-touching edges at random positions of $G_k^w$
      for each pair of vertices $(u, v) \in S$ do
          if there is no edge between $u$ and $v$ in $G_k^w$ then
              Add an edge $e$ between $u$ and $v$
              Colour $e$ with a bicolour $\in L_{colours}^2$
              Assign a weight $\in L_{weights}$ to $e$
          else
              $add\_edge \leftarrow True$ with probability $\frac{1}{(\text{#edges between u and v})+1}$
              if $add\_edge$ then
                  Add an edge $e$ between $u$ and $v$
                  Colour $e$ with a bicolour $\in L_{colours}^2$ different from the bicolour of the other edges between $u$ and $v$
                  Assign a weight $\in L_{weights}$ to $e$
              end if
          end if
      end for
  end for
```

not have multiple edges of the same bicolour between 2 vertices. Assuming $0 \in L_{weights}$, this makes them non-redundant, according to the definition 2.0.2. All of these observations make them indeed good *candidates* to study.

9. **Perform a random experiment graphs research process**, defined as follows.

   Definition 3.2.4 (Random Experiment Graphs Research Process)**.** Given a number of nodes $n \in \mathbb{N}$, a list of colours $L_{colours}$, a complexity bound $b \in \mathbb{N}$, a list of complex numbers $L_{weights}$, and a number of trials $m \in \mathbb{N}$, a *Random Experiment Graphs Research Process* is the process of randomly creating m candidate experiment graphs respecting n, $L_{colours}$, $L_{weights}$ and b, and analyse them. More specifically, each of these graphs G_k^w gets verified to check if it is perfectly monochromatic and if it has at least one non-monochromatic feasible vertex colouring. If it is, G_k^w is saved as a JSON file containing all its properties (including its perfect matchings, its feasible vertex colourings, its weighted matching index, and the fact it is bipartite or not). Then, G_k^w is drawn in another pdf file.

   In short, this experiment allows the user to generate a big number of experiment graphs and analyse properties on them. The condition that the graphs must have at least one non-monochromatic feasible vertex colouring allows the user to focus on non-trivial graphs. Indeed, if it is not satisfied, then the graph is monochromatic, according to definition 1.2.1, and this case of the Krenn's conjecture is already proven thanks to Bogdanov [Bog].

3.2.4 Details of implementation

This section explains all the technical aspects of the implementation of EGPI. Readers only interested in its use can pass this section if they want to.

EGPI was implemented in Python, this choice was motivated by two reasons. Firstly, Python is a high-level programming language that offers many functionalities and data types by default [YH21]. This made the implementation a lot easier and compacter than if it had to be done in other programming languages. Secondly, Python is easy to read and to learn for people who are not familiar with programming. This may allow future researchers interested in re-using EGPI to perform modifications to it without too many difficulties if they want to.

The main purpose of EGPI is to encode and apply operations on experiment graphs, rigorously defined in definition 1.3.1. It is therefore important to build a good data structure of experiment graphs. The architecture of the program to do so is detailed in a state diagram available in figure 3.7.

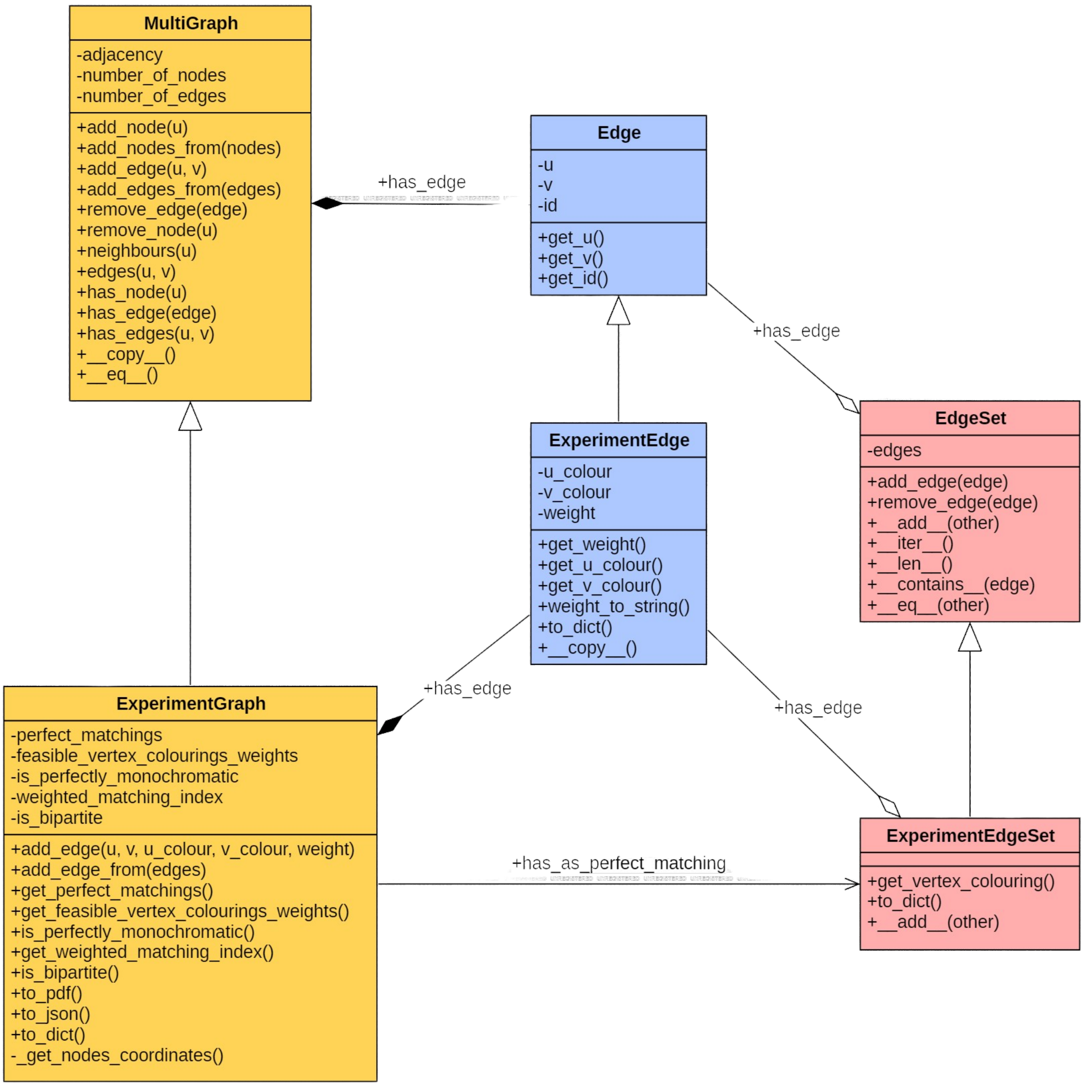

Figure 3.7: Structure diagram of EGPI showing the important relations between the different implemented data structures. Most of the implemented methods of the program are shown in the diagram.

Here is an exhaustive list of all non-trivial experiment graphs' properties we are interested to compute, and the algorithms we use in EGPI to do so.

1. **Their perfect matchings:** to find all the perfect matchings of a graph, EGPI uses an algorithm written in pseudocode in algorithm 2.

   Complexity of algorithm 2: The algorithm is a depth first search (DFS) algorithm. In the worst case scenario, G_k^w is a complete graph with n vertices. In other words,

Algorithm 2 Find all perfect matchings of an experiment graph G

```
Require: $G$ is an experiment graph
  if $G$ has no node then
    The only perfect matching is $\varnothing$
  else if $G$ has nodes then
    $PMs \leftarrow$ empty list
    choose a random node $u \in V(G)$
    for all $v \in$ neighbours of $u$ do
      $subPMs \leftarrow$ all perfect matchings of $G$ without $u$ and $v$
      for all $subPM \in subPMs$ do
        for all $e \in$ edges between $u$ and $v$ do
          add $subPM \cup e$ to $PMs$
        end for
      end for
    end for
  end if
  return PMs
```

each node has $n-1$ neighbours in G_k^w. Also, at each recursive step of the algorithm, we remove 2 nodes from the graph. This means that the depth of the search tree is $\frac{n}{2}$. The running time of this search algorithm is then $O\left(n^{\frac{n}{2}}\right)$.

2. **The weights of their feasible vertex colourings:** according to their definition 1.3.4, finding the feasible vertex colourings of an experiment graph requires to find their perfect matchings. The next steps to find them and their weights are simpler and are described in algorithm 3.

Algorithm 3 Find all feasible vertex colourings of an experiment graph G_k^w

```
Require: $G_k^w$ is an experiment graph
  $PMs \leftarrow$ all perfect matchings of $G_k^w$
  $FVCs \leftarrow$ empty Python dictionary
  for all $PM \in PMs$ do
    $FVC \leftarrow$ feasible vertex colouring induced by $PM$
    $w \leftarrow$ weight of $PM$
    $FVCs[FVC] \leftarrow w$
  end for
  return $FVCs$
```

The hardest step of this algorithm is to find all the perfect matchings of G_k^w. Therefore, the complexity of this algorithm is the same as the one of algorithm 2, which is $O\left(n^{\frac{n}{2}}\right)$ in the worst case.

3. **If the graph is perfectly monochromatic:** by definition 1.3.6, finding if an experiment graph is perfectly monochromatic or not just requires looking at all its feasible vertex colourings. This is done in algorithm 4.

   The only hard step of this algorithm is to find all the feasible vertex colourings of

Algorithm 4 Check if an experiment graph G_k^w is perfectly monochromatic

```
Require: $G_k^w$ is an experiment graph
  $FVCs \leftarrow$ all feasible vertex colourings of $G$
  $isPM \leftarrow$ True
  for all $FVC \in FVCs$ do
      if FVC is monochromatic and $w(FVC) \neq 1$ then
          $isPM \leftarrow$ False
      else if FVC is not monochromatic and $w(FVC) \neq 0$ then
          $isPM \leftarrow$ False
      end if
  end for
  return $isPM$
```

G_k^w. Therefore, the complexity of this algorithm is the same as the one of algorithm 3, which is $O\left(n^{\frac{n}{2}}\right)$ in the worst case.

4. **The weighted matching index of the graph:** defined in definition 1.3.7, the weighted matching index of a perfectly monochromatic experiment graph is the number of monochromatic feasible vertex colourings in this graph. If the graph is not perfectly monochromatic, the weighted matching index is 0 by definition. EGPI uses the algorithm 5 to compute the weighted matching index of a graph.

Algorithm 5 Compute the weighted matching index of an experiment graph G_k^w

```
Require: $G_k^w$ is an experiment graph
  $FVCs \leftarrow$ all feasible vertex colourings of $G_k^w$
  $isPM \leftarrow$ is $G_k^w$ perfectly monochromatic ?
  if $isPM$ then
      $c \leftarrow 0$
      for all $FVC \in FVCs$ do
          if $w(FVC) = 1$ then
              $c \leftarrow c + 1$
          end if
      end for
      return $c$
  else
      return 0
  end if
```

Again, the complexity of this algorithm is determined by the one of algorithm 3, which is $O\left(n^{\frac{n}{2}}\right)$ in the worst case.

5. **If the graph is bipartite:** finding if a graph is bipartite or not is a well-known problem in graph theory. EGPI uses the following greedy algorithm to find if a graph is bipartite or not. The algorithm is written in algorithm 6.

   This algorithm visits all the nodes of G_k^w only once. Therefore, its complexity is $\mathcal{O}(n)$, where n is the number of vertices of G_k^w.

Algorithm 6 Check if an experiment graph G_k^w is bipartite

```
Require: $G_k^w$ is an experiment graph
  $Q \leftarrow$ empty queue
  add $v_0 \in V(G_k^w)$ to $Q$
  colour of $v_0 \leftarrow$ red
  $isBipartite \leftarrow$ True
  while $Q$ is not empty do
      $u \leftarrow$ pop $Q$
      for all $v \in$ neighbours of $u$ do
          if colour of $v$ is not defined then
              colour of $v \leftarrow$ opposite colour of $u$
              add $v$ to $Q$
          else if colour of $v$ is the same as colour of $u$ then
              $isBipartite \leftarrow$ False
          end if
      end for
  end while
  return $isBipartite$
```

3.2.5 Realized experiments with EGPI

In this section, I present some of the experiments I realized with EGPI. The goal of these experiments was to find interesting properties of experiment graphs, and to try to find a counter-example to the Krenn's conjecture.

Research for counter-examples

The first experiment I realized with EGPI was to try to find a counter-example to the Krenn's conjecture. To do so, I performed different random experiment graphs research processes, defined in definition 3.2.4. Each of these processes used the following parameters.

- The number of trials was $m = 10^6$.
- The possible colours of the edges were $L_{colours} = \{red, green\}$.
- The possible complex weights of the edges were $L_{weights} = \{-1, 1, -i, i\}$.

I performed 21 different random experiment graphs research processes, with $n \in 6, 8, 10$ and $b \in \{1, 2, 3, 4, 5, 6\}$.

On all the 21 millions generated candidate graphs, none of them was perfectly monochromatic. This result is a new argument in favour of the Krenn's conjecture, at least for graphs that have less than 10 vertices. Nevertheless, it does not constitute a proof of it in any case, and I encourage future researchers to continue looking for counterexamples to it alongside their researches (by using EGPI or any other tool).

Research for perfectly monochromatic graphs with a weighted matching index of 2

The second experiment I realized with EGPI was to try to find perfectly monochromatic graphs that have a weighted matching index of 2. These graphs are authorized by the Krenn's conjecture, and their properties are interesting to study. Indeed, understanding better the structure of these graphs might help researchers to find new ideas in their seek for a proof to the conjecture.

But this is not the main and only interest of the experiment. Exploring this class of experiment graphs is a great way to test the different functionalities of EGPI, and to experimentally check the impact of the different parameters of the program. Using the following parameters,

- The number of trials was $m = 10^6$.
- The possible colours of the edges were $L_{colours} = \{red, green\}$.
- The possible complex weights of the edges were $L_{weights} = \{-1, 1, -i, i\}$.

I performed 15 different random experiment graphs research processes, with $n \in 6, 8, 10$ and $b \in \{1, 2, 3, 4, 5\}$. I got results summarized in figure 3.8.

Results of the experiment						
n	$b = 1$	$b = 2$	$b = 3$	$b = 4$	$b = 5$	total
6	103	18	4	0	0	125
8	39	1	0	0	0	40
10	3	0	0	0	0	3

Figure 3.8: Results of 3 different random experiment graphs research process with $n = 6$, $n = 8$ and $n = 10$ respectively.

Here is an example of discovered perfectly monochromatic graph by the program.

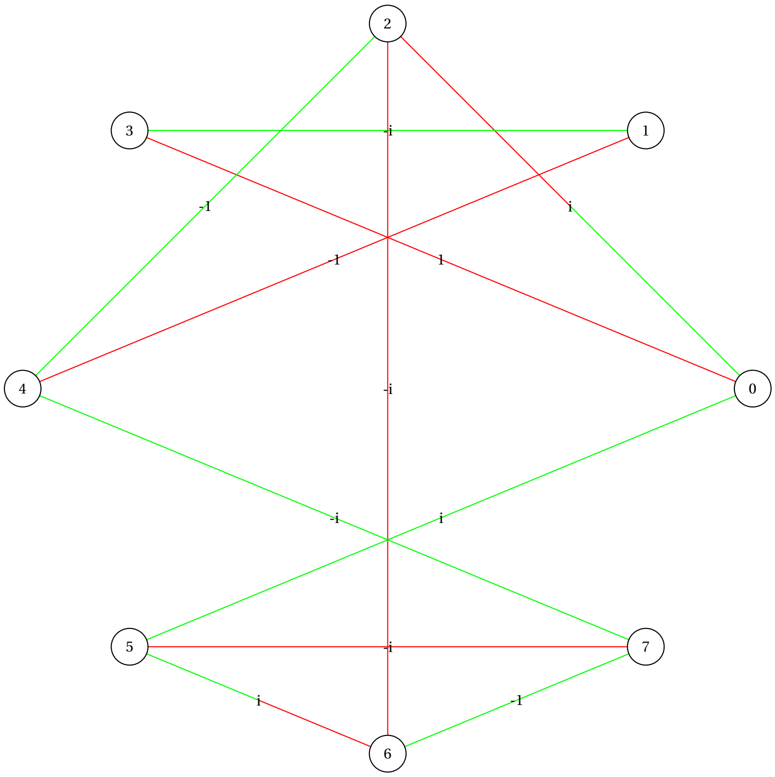

Figure 3.9: A perfectly monochromatic graph of size 8 vertices, with a weighted matching index of 2, discovered by EGPI during the experiment.

From the results in figure 3.8, we can do the following observations. Firstly, it is a lot harder to find perfectly monochromatic graphs in these conditions when their size is bigger. Indeed, the number of possibilities grows exponentially with the number of vertices, and the probability to find a perfectly monochromatic graph decreases. Secondly, the complexity bound has a big impact on the number of found graphs: a higher complexity bound results in a smaller number of found graphs. This was expected, since the graphs generated with a higher complexity bound have higher chances to have more edges and perfect matchings. The probability that all these randomly generated edges and perfect matchings together satisfy the definition 1.3.6 are small.

The last thing I am interested in is to count bipartite graphs among all the perfectly monochromatic graphs found. The result is the following: **none** of the perfectly monochromatic graphs found were bipartite. This is an alluring new observation: indeed, it is consistent with the arguments we used in the proofs of lemmas 3.1.4 and 3.1.5. One of the observations these proofs used is that, in these particular cases, the perfectly monochromatic graphs found were not bipartite. This allows me to formulate the following conjecture.

Conjecture 3.2.5. Let G_k^w be a non-redundant perfectly monochromatic graph that respects the following properties.

- $\tilde{c}(G,k,w) \geq 2$
- $\forall e \in E(G_k^w), w(e) \in \{-1, 1, -i, i\}$
- G_k^w has at least one non-monochromatic feasible vertex colouring.

Then, G_k^w is not bipartite.

This conjecture might constitute a new interesting subcase of the Krenn's conjecture to investigate in the future.

Possible improvements for EGPI

EGPI is a tool that can still be improved in many ways.

Firstly, currently it does not offer the functionality of generating only non-isomorphic graphs. This functionality could be interesting if the user wants to count the number of different graphs generated by the tool. This could be coded from scratch, or by using an external program. For instance, in Python, the package *NetworkX* offers a function to check if 2 graphs are isomorphic or not [HSS08]. A more common program used a lot in the field of graph theory is Nauty, a program originally designed by Brendan McKay [MP14]. Using it would offer a lot of advantages, but it would require to write a wrapper around it to use it in Python. Indeed, Nauty is written in C.

A second improvement that could be brought to EGPI is to improve the efficiency of the algorithms used to find the perfect matchings of a graph. Indeed, this step bounds the efficiency of most of the tasks performed by EGPI. Improving this algorithm would allow the program to search more graphs in a given time, and to find more results in general. One idea to improve it could be to look at the Edmonds' Blossom algorithm, an algorithm to find a single perfect matching in a graph [Edm65]. It has to be determined whether this algorithm can be adapted to find all perfect matchings of a graph.

Conclusion

In this master thesis, we started by understanding and summarizing the concept of the Krenn's conjecture: a conjecture about perfect matchings and their intersections in edge-weighted edge-bi-coloured multigraphs.[Kre] We dived deeper in the connection between this conjecture and quantum physics, investigated for the first time by Mario Krenn in [KGZ17]. Mario Krenn indeed discovered that the (non-)existence of certain specific graphs could lead to the (non-)feasibility to create specific quantum states, called GHZ-states, through the use of certain experiments. Being able to create such states is a key point in quantum computing [GK20] and cryptography [PHM18]. This discovery was the starting point of its conjecture in graph theory, which remains unsolved.

We then focused on dressing a state of the art of the research on the Krenn's conjecture. In it, we could see that the conjecture was already solved for real, positive weights by Bogdanov in [Bog]. We also spoke about Chandran and Gajjala, who proved the conjecture in the special case of graphs that have a matching index (defined in 1.2.2) different from 1 [CG22]. In that same paper, they could find some upper bounds on the weighted matching index in terms of minimum degree and of edge connectivity.
Their researches were extremely valuable for us, as they introduced a lot of notations and concepts that we used in our own work.

After that, we adopted an algorithmic approach to find new interesting results related to the problem. We first tried to perform problem reductions to find out what restrained cases were interesting to study. By doing so, we discovered that the study of non-redundant experiment graphs (defined in 2.0.1) with integer weights could be reduced to the study of redundant experiment graphs that have weights included in $\{-1, 1\}$. We then focused on proving some new special cases of the Krenn's conjecture. Our main results are the following: we could extend Bogdanov's results[Bog] by allowing one weight to be negative. This attempt was a success, and encouraged us to extend the used arguments to a more general case. This is how we came by with what I consider to be the main result of this master thesis, formulated in Lemma 3.1.5.

Lemma 3.2.6 (2 positive colour classes)**.** Let G_k^w be a simple perfectly monochromatic graph that has only real weights, and that has at least 2 colour classes that have only positive weighted edges. Then, if G is not isomorphic to K_4, $\tilde{c}(G,k,w) \leq 2$.

At last, we adopted a more experimental approach to the problem by looking for counter-examples to the Krenn's conjecture. In practice, this was done through the implementation of a new program called EGPI (Experiment Graphs Properties Identifier) — a program that generates random experiment graphs and checks properties related to the Krenn's conjecture on them. In my opinion, such a tool was currently missing in the researchers' community. Indeed, finding interesting graphs to analyse or showcase by hand is hard, and the EGPI program could help in this task. Also, even though I strongly believe the conjecture to be true at this point, the EGPI program could still be used to find counter-examples to it by chance. Using EGPI, I looked for such counter-examples and could not find any, which reinforces my belief in the conjecture's truth. EGPI also allowed me to analyse a big number of random perfectly monochromatic graphs with a weighted matching index of 2, and to discover that none of them were bipartite. Therefore, I finished my work by stating a new conjecture about the non-existence of bipartite perfectly-monochromatic graphs with a weighted matching index of 2.

Acknowledgements

First, I would like to thank my two supervisors, Gwenaël Joret and Yelena Yuditsky, for their help and guidance throughout this master thesis. We worked a lot together, and I am very grateful for the time they spent on me. Most of the results I discovered were found thanks to their help, and I am very thankful for that.

Then, I would like to thank my old colleagues from my previous physics science studies. They helped me a lot in understanding the basics of quantum optics that I was lacking, and that were crucial in the context of this master thesis. They have all my gratitude for their time and patience.

I would also like to thank the members of the jury for having accepted to evaluate this master thesis. I am very grateful for the time they will spend on it, and I hope they will find it interesting.

Finally, a huge thank you to my family and friends for their support during this year. Their presence and advises were always appreciated, and I thank them for being the guarantors of my mental health in the last weeks preceding the deadline. Completing this master thesis would have been impossible without them.

List of theorems and definitions

Bibliography

[Bel64] Bell, John S (1964). “On the einstein podolsky rosen paradox”. In: *Physics Physique Fizika* 1.3, p. 195.

[Bog] Bogdanov, Ilya (n.d.). *Answer to : Graphs with only disjoint perfect matchings.* Accessed: 2023-04-19. URL: `https://mathoverflow.net/a/267013/63938`.

[BM+76] Bondy, John Adrian, Uppaluri Siva Ramachandra Murty, et al. (1976). *Graph theory with applications.* Vol. 290. Macmillan London.

[Bou+99] Bouwmeester, Dik et al. (Feb. 1999). “Observation of Three-Photon Greenberger-Horne-Zeilinger Entanglement”. In: *Physical Review Letters* 82.7, pp. 1345–1349. ISSN: 1079-7114. DOI: `10.1103/physrevlett.82.1345`. URL: `http://dx.doi.org/10.1103/PhysRevLett.82.1345`.

[CG22] Chandran, L. Sunil and Rishikesh Gajjala (2022). “Perfect Matchings and Quantum Physics: Progress on Krenn’s Conjecture”. In: arXiv: `2202.05562 [cs.DM]`.

[CG23] — (2023). *Graph-theoretic insights on the constructability of complex entangled states.* arXiv: `2304.06407 [quant-ph]`.

[Die17] Diestel, Reinhard (2017). *Graduate Texts in Mathematics.* Vol. Graph theory. Berlin, Heidelberg: Springer.

[Dir39] Dirac, Paul Adrien Maurice (1939). “A new notation for quantum mechanics”. In: *Mathematical proceedings of the Cambridge philosophical society.* Vol. 35. 3. Cambridge University Press, pp. 416–418.

[Edm65] Edmonds, Jack (1965). “Paths, Trees, and Flowers”. In: *Canadian Journal of Mathematics* 17, pp. 449–467. DOI: `10.4153/CJM-1965-045-4`.

[EPR35] Einstein, A., B. Podolsky, and N. Rosen (May 1935). “Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?” In: *Phys. Rev.* 47

(10), pp. 777–780. DOI: 10.1103/PhysRev.47.777. URL: https://link.aps.org/doi/10.1103/PhysRev.47.777.

[GHZ89] Greenberger, Daniel M, Michael A Horne, and Anton Zeilinger (1989). “Going beyond Bell’s theorem”. In: *Bell’s theorem, quantum theory and conceptions of the universe*. Springer, pp. 69–72.

[GK20] Gu, Xuemei and Mario Krenn (2020). “Compact Greenberger—Horne—Zeilinger state generation via frequency combs and graph theory”. In: *Frontiers of Physics* 15.6, p. 61502.

[HSS08] Hagberg, Aric, Pieter Swart, and Daniel S Chult (2008). *Exploring network structure, dynamics, and function using NetworkX*. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

[Han24] Hannon, Merlin (2024). *GitHub - mhannon/EGPI: Experiment Graphs Properties Identifyer*. https://github.com/mhannon/EGPI. [Accessed 23-05-2024].

[Has+20] Hassija, Vikas et al. (2020). “Forthcoming applications of quantum computing: peeking into the future”. In: *IET Quantum Communication* 1.2, pp. 35–41.

[J12] J, Matson (2012). “Quantum teleportation achieved over record distances”. In: *Nature*. URL: https://doi.org/10.1038/nature.2012.11163.

[Kre] Krenn, Mario (n.d.). *Inherited Vertex Coloring of Graphs*. Accessed: 2023-05-07. URL: https://mariokrenn.wordpress.com/graph-theory-question/.

[KGZ17] Krenn, Mario, Xuemei Gu, and Anton Zeilinger (Dec. 2017). “Quantum Experiments and Graphs: Multiparty States as Coherent Superpositions of Perfect Matchings”. In: *Physical Review Letters* 119.24. DOI: 10.1103/physrevlett.119.240403. URL: https://doi.org/10.1103%2Fphysrevlett.119.240403.

[KG21] Kumar, Ajay and Sunita Garhwal (2021). “State-of-the-art survey of quantum cryptography”. In: *Archives of Computational Methods in Engineering* 28, pp. 3831–3868.

[MP14] McKay, Brendan D. and Adolfo Piperno (2014). “Practical graph isomorphism, II”. In: *Journal of Symbolic Computation* 60, pp. 94–112. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2013.09.003. URL: https://www.sciencedirect.com/science/article/pii/S0747717113001193.

[Ott+17] Otterbach, Johannes S et al. (2017). “Unsupervised machine learning on a hybrid quantum computer”. In: *arXiv preprint arXiv:1712.05771*.

[PHM18] Pivoluska, Matej, Marcus Huber, and Mehul Malik (2018). “Layered quantum key distribution”. In: *Physical Review A* 97.3, p. 032312.

[Sci22] Sciences, The Royal Swedish Academy of (2022). *The Nobel Prize in Physics 2022 — nobelprize.org*. `https://www.nobelprize.org/prizes/physics/2022/press-release/`. [Accessed 18-04-2024].

[Sie20] Siegel, Ethan (2020). “No, We Still Can’t Use Quantum Entanglement To Communicate Faster Than Light”. In: *Forbes*. URL: `https://www.forbes.com/sites/startswithabang/2020/01/02/no-we-still-cant-use-quantum-entanglement-to-communicate-faster-than-light/`.

[Wan+16] Wang, Xi-Lin et al. (2016). “Experimental ten-photon entanglement”. In: *Physical review letters* 117.21, p. 210502.

[Wik] Wikipedia (n.d.). *Binomial theorem - Wikipedia — en.wikipedia.org*. `https://en.wikipedia.org/wiki/Binomial_theorem`. [Accessed 12-05-2024].

[YH21] Yuill, Simon and Harry Halpin (2021). “Python”. In: *Python releases for windows* 24.