NAWI Graz
Natural Sciences

TU Graz

Alexandra Weinberger, BSc

Simple Drawings and Rotation Systems Beyond the Complete Graph

MASTER'S THESIS

to achieve the university degree of

Master of Science

Master's degree programme: Mathematics

submitted to

Graz University of Technology

Supervisor

Assoc.Prof. Dipl.-Ing. Dr.techn. Oswin Aichholzer

Institute for Software Technology

Second Supervisor
Irene Parada, MSc

Graz, August 2019

Abstract

In this master thesis we study simple drawings. Simple drawings are drawings of graphs in the plane that fulfill some properties, which restrict their crossings.

We are interested in the information that rotation systems can offer about crossings. Rotation systems of simple drawings give us the order in which edges leave the vertices. It is known that rotation systems of simple drawings of complete graphs determine which edges cross. We study what information we can gain from rotation systems of other types of graphs. We show that rotation systems of simple drawings of graphs with one edge less than the complete graph determine the number of crossings. Moreover, we proof that rotation systems of simple drawings of graphs with $n \geq 5$ vertices and minimal degree of at least $(n-2)$ determine the number of crossings. Furthermore, we show that rotation systems of simple drawings of $K_{2,3}$ determine the parity of the number of crossings. It is known that the number of crossings in simple drawings of $K_{m,n}$ with m and n fixed and both odd always have the same parity.

We also focus on the question of whether simple drawings of complete bipartite graphs contain plane spanning trees. We show that simple drawings of $K_{2,n}$ and $K_{3,n}$ as well as some special types of simple drawings of $K_{m,n}$ do. Those special types are outer drawings, straight-line drawings, 2-page book drawings, and circular drawings. We show that all those simple drawings contain a particular type of plane spanning tree that we call shooting star. Shooting stars are plane spanning trees that contain all edges incident to one vertex.

Zusammenfassung

In dieser Master-Arbeit untersuchen wir Simple Drawings. Simple Drawings sind Zeichnungen von Graphen in der Ebene, die Eigenschaften erfüllen, welche ihre Kreuzungen beschränken.

Wir interessieren uns für die Information, die Rotationssysteme liefern können. Rotationssysteme von Simple Drawings geben die zyklische Reihenfolge an, in welcher die Kanten die Knoten verlassen. Es ist bekannt, dass Rotationssysteme von Simple Drawings von vollständigen Graphen vorgeben, welche Kanten sich schneiden. Wir untersuchen, welche Information wir von Rotationssystemen für andere Graphen gewinnen können. Wir zeigen, dass Rotationssysteme von Simple Drawings von Graphen, die eine Kante weniger haben als der vollständige Graph, die Anzahl von Kreuzungen vorgeben. Außerdem beweisen wir, dass auch Rotationssysteme von Graphen mit $n \geq 5$ Knoten und Minimalgrad $(n-2)$ die Anzahl von Kreuzungen vorgeben. Darüber hinaus beweisen wir, dass Rotationssysteme von Simple Drawings von $K_{2,3}$ die Parität der Anzahl von Kreuzungen vorgeben. Es ist bekannt, dass die Anzahl von Kreuzungen in Simple Drawings von $K_{m,n}$, in denen sowohl n als auch m gegeben und ungerade sind, dieselbe Parität haben.

Ein weiterer Fokus liegt auf der Frage, ob Simple Drawings von vollständig bipartiten Graphen kreuzungsfreie Spannbäume enthalten. Wir zeigen, dass Simple Drawings von $K_{2,n}$ und $K_{3,n}$ sowie einige spezielle Arten von Simple Drawings von $K_{m,n}$ das tun. Diese speziellen Arten sind Outer Drawings, geradlinige Graphen, 2-Page Book Drawings und Circular Drawings. Wir beweisen, dass all diese Simple Drawings eine spezielle Art von kreuzungsfreiem Spannbaum enthalten, die wir Shooting Star nennen. Shooting Stars sind kreuzungsfreie Spannbäume, die alle inzidenten Kanten eines Knotens enthalten.

Acknowledgements

I am profoundly grateful to my supervisor Oswin Aichholzer for his great support and patience during my work on this thesis. I would also like to express my deepest appreciation to my second supervisor Irene Parada for her commitment and helpful advice. Moreover, I want to thank Oswin Aichholzer and Irene Parada for reading through earlier versions of this thesis and giving me important feedback. I would also like to sincerely thank Birgit Vogtenhuber for her willingness to discuss ideas and her useful suggestions.

My deepest gratitude goes to Oswin Aichholzer, Birgit Vogtenhuber and Irene Parada for researching and discussing with me. I truly appreciate their help, enthusiasm, knowledge and guidance. I am also very grateful to them for the chance to attend EuroCG 2019 and the AJMS Workshop 2019.

Many thanks to Manfred Scheucher for his useful contributions to this thesis and the insight he provided.

Finally, I cannot begin to express my thanks to my parents, who never wavered in their support of me. Special thanks also to my brother, and all my family.

Contents

Chapter 1

Introduction

This chapter is dedicated to giving some motivation and background on the topic of the thesis. Before we can do that, we need to define some terms.

In Section 1.1 we give some basic definitions that we need throughout the thesis. Then we introduce the main points of the thesis, namely crossings (in Section 1.2), simple drawings (in Section 1.3), rotation systems (in Section 1.4) and spanning trees (in Section 1.5). Finally, we give an outline on how the thesis is structured.

1.1 Basic definitions

Let us first define some basic and essential terms. If not stated otherwise, the following definitions are taken from [6].

Definition 1.1.1. A *graph* $G = (V, E)$ is a pair of sets V and E. We call elements of V *vertices* and elements of E *edges*. Each edge is associated to a set of one or two vertices. We call those vertices its *endpoints*. We say the edge *joins* its endpoints.

Definition 1.1.2. A vertex v and an edge e are *incident* to each other if and only if v is an endpoint of e.

Two edges are *adjacent* if and only if they have a common endpoint. Two vertices are *adjacent* if and only if they are joined by an edge.

Definition 1.1.3. The *degree of a vertex* is the number of edges it is incident to.

The *minimal degree of a graph* G is the minimum of the degrees of all vertices in G.

Definition 1.1.4. We call an edge with a single endpoint a *self-loop*, that means the edge joins an endpoint with itself. We call a collection of two or more edges having identical endpoints *multi-edges*.

We call a graph that has neither self-loops nor multi-edges a *simple graph*.

In this thesis we work exclusively with simple graphs. From now on we always mean simple graphs, when we use the term “graph” without clarification.

Definition 1.1.5. A *drawing* of a graph on a surface is a representation of a graph on that surface in the following way:

- The vertices of the graph are represented by distinct points.
- The edges of the graph are represented by continuous arcs that connect two such points.
- The only points lying on an arc are its endpoints.

Definition 1.1.6. Two edges *cross* if the arcs intersect in their interior (meaning they intersect somewhere else than their endpoints). We call the point in which they cross a *crossing*.

A drawing without crossings is called *plane*.

A graph that has a plane drawing is called *planar*.

Definition 1.1.7. A graph in which every pair of vertices is joined by an edge is called *complete graph*. We write K_n for the complete graph with n vertices.

Definition 1.1.8. A graph whose vertices can be partitioned into two sets such that there are no edges joining a pair of vertices within the same set is called *bipartite graph*. We say the two sets are its sides of the bipartition.

A *complete bipartite graph* is a bipartite graph where every vertex is joined by edges to all vertices on the other side of the bipartition. We write $K_{m,n}$ for the complete bipartite graph where one side of the bipartition contains m vertices and the other side of the bipartition contains n vertices.

In our notation of K_n and $K_{m,n}$ the numbers n and m are always integers $n, m \in \mathbb{N}$, because they are the numbers of vertices. We will not take extra notice of this when speaking about K_n or $K_{m,n}$.

Definition 1.1.9. Let G be a graph with vertices V and edges E. A *walk* is a sequence of vertices $(v_1, v_2, ..., v_n)$ and edges $(e_1, e_2, ..., e_{n-1})$ such that e_i is joining the vertices v_i and v_{i+1}.

A walk in which edges are distinct and all interior vertices are distinct is called a *path*.

A path in which the first vertex is the same as the last vertex is called a *cycle*.

Definition 1.1.10. A graph is called *connected* if and only if for all pairs of vertices there is a path including both of them.

Definition 1.1.11. Let G be a graph with vertices V and edges E. A graph H with vertices V' and E' is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$.

If $V = V'$ then H is a *spanning* subgraph.

Definition 1.1.12. A graph is called a *tree* if and only if it is connected and does not contain any cycles.

For drawings these terms are defined analogously. A *subdrawing* of a drawing is a drawing that contains a subset of the vertices (represented as points) of that drawing and a subset of its edges (represented as curves). It is spanning if all vertices of the drawing are contained in the subdrawing. A drawing is a tree if the graph it represents is a tree.

Definition 1.1.13. [8] The *crossing number* of a graph is the minimal number of crossings that a drawing of that graph in the plane can have.

We will mainly talk about the *number of crossings* that a particular drawing has. However, in this first chapter, we also talk about crossing numbers. It is important to not confuse the terms.

1.2 About crossings

The problem of how to draw graphs in such a way that they contain as few crossings as possible has been thoroughly studied. In fact, it is one of the most popular problems in graph theory.

One of the first results that offered insight into when graphs can be drawn with few (in this case no) crossings was *Kuratowski's theorem.* In 1930 Kuratowski proofed that a graph is planar if and only if it does not contain a subgraph that is a subdivision of K_5 or $K_{3,3}$ [13]. (A *subdivision* of a graph is obtained by performing edge subdivisions. An edge subdivision of G is obtained the following way: First we delete an edge from G, say the edge joining u and v. Then we add a new vertex, say w, to the vertex set of G. Finally, we add an edge joining u and w, and add an edge joining w and v.)

The problem of finding the minimal crossing number was formalized by Turán for complete bipartite graphs. He started thinking about this question while he worked in a brick factory near Budapest during World War II. He and other workers had to bring bricks from kilns to storage yards. They did so by loading them on trucks and pushing the trucks along rails. The trucks were harder to push at points where railes crossed each other. Turán wished to find a way to have less crossings between the rails [22].

This problem became known as the *brick factory problem.* Mathematically, we can see the kilns and storage yards as vertices, and the rails between kilns and storage yards as edges. This gives us a complete bipartite graph with kilns on one side of the bipartition and storage yards on the other.

There is also much interest in the problem for the complete graph: What is the crossing number for complete graphs? Despite much research in this topic neither the crossing number of complete graphs nor the crossing number of complete bipartite graphs has been determined yet.

Zarankiewicz claimed in the early 1950s that the minimum crossing number of complete bipartite graphs is $\left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{m}{2} \right\rfloor \left\lfloor \frac{m-1}{2} \right\rfloor$. His proof that this is the exact number contained an error. However, he could show that this is an upper bound for the crossing number. That the claim is true could so far only be proven for small $\{m, n\}$ [7].

For the complete graph a conjecture about the crossing number was made later. Hill conjectured in [9] that it is $\frac{1}{4} \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{n-2}{2} \right\rfloor \left\lfloor \frac{n-3}{2} \right\rfloor$. Like in the case of complete bipartite graphs, this has been proven for small n and stays an open problem in general [8].

1.3 About simple drawings

We are interested in drawings of graphs where the edges are drawn as curves. There are infinite possibilities to draw a graph that way. We will make some restrictions to how we can draw the edges. The drawings we will work with are called *simple drawings*. We define them in this section. We also show that the requirements of simple drawings make sense and are no true restriction when doing research on the crossing number.

1.3.1 Definition of simple drawings

Definition 1.3.1. [16] A *simple drawing* is a drawing of a graph in the plane with the following additional properties:

1. An edge may not cross itself and may not be tangential to itself. That means it may not touch itself in any points.

2. Two edges may cross each other at most once and may not be tangential to each other.

3. Two adjacent edges may not cross each other.

4. There cannot be more than two edges crossing at the same point.

1.3.2 Why simple drawings are interesting

If a drawing is not a simple drawing it can easily be transformed into a simple drawing as can be seen in Figure 1.1. This transformation never creates more crossings. In fact, in most cases the transformation leads to fewer crossings. Drawings that are not simple could be seen as drawings that add crossings where they could be avoided very easily.

Consequently, studying simple drawings makes sense, when we are trying to find drawings with few crossings. In the rest of this thesis, we will only talk about simple drawings.

1.3.3 Some general results about simple drawings

The two following lemmas are well known facts about simple drawings. We will use them throughout the thesis.

Lemma 1.3.2. *Every simple drawing of a graph with at most three vertices is plane.*

Proof. A graph with at most 3 vertices cannot contain any edges that are not adjacent. Thus, there cannot be any edges that cross each other. □

Lemma 1.3.3. *Every simple drawing of a graph with at most four vertices contains at most one crossing.*

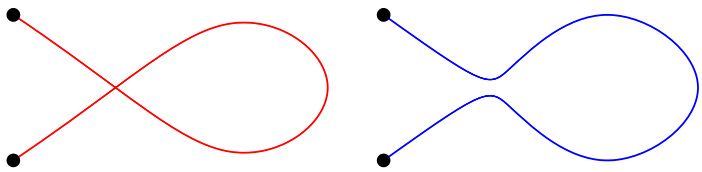
Proof. Let D be a simple drawing of a graph with four vertices.

Claim 1.3.4. *An edge in D can cross at most one other edge.*

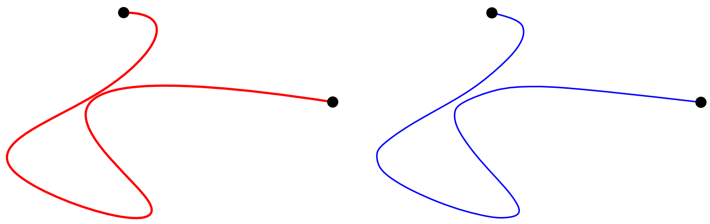
Proof of claim: Every edge is incident to two vertices. Let d be an arbitrary edge in D. We call the vertices that are incident to d vertices u and v. We call the other vertices in D vertices w and x. Then d is adjacent to all edges that are incident to either u or v. There is only one edge that is incident to neither u nor v, namely the edge joining w and x. It follows that every edge is adjacent to all but one other edges. In a simple drawing adjacent edges may not cross. ■

Let e and f be two edges that cross. We show that in this case there cannot be any other crossings.

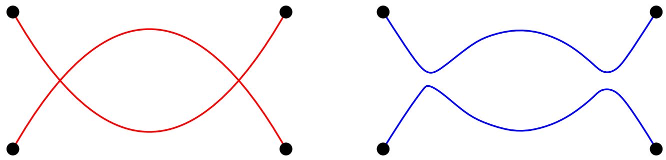
Let g be another edge. We call the vertices incident to g vertices v_1 and v_2, where v_1 is incident to e and v_2 is incident to f. We call the vertices not incident to g vertices v_3 and v_4, where v_3 is incident to e and v_4 is incident to f. By Claim 1.3.4 the edge g can cross neither e nor f. The part of the edge e that goes from v_1 to the crossing, the part of f that goes from v_2, and the edge g separates the plane into two areas. The vertices v_3 and v_4 are in the same area. We call this area A. It can be seen in Figure 1.2.



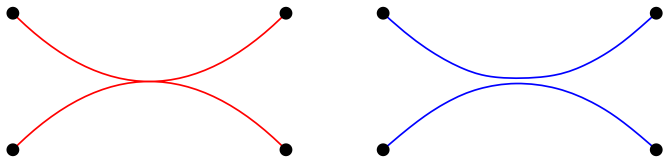
(a) How an edge that crosses itself may be transformed.



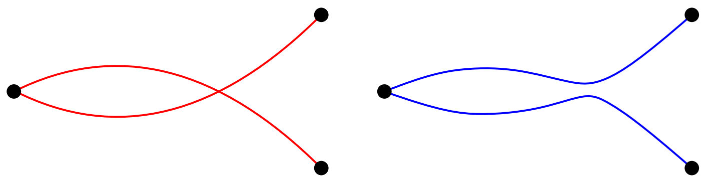
(b) How an edge that is tangential to itself may be transformed.



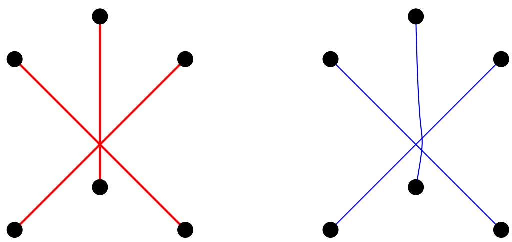
(c) How two edges that cross twice may be transformed.



(d) How two edges that are tangential may be transformed.



(e) How two adjacent edges that cross may be transformed.



(f) How three edges that cross in the same point may be transformed.

Figure 1.1: Transforming drawings that are not simple into simple drawings.

The only edge that can cross g is the edge joining v_3 and v_4. We call this edge h. Since endpoints of h lie in A, the edge starts and ends in A. If h crosses the boundary of A once, it will leave that area. It will have to cross the boundary another time in order to enter the area again. Since h is incident to edges e and f, the only part of the boundary that it may cross is g. Thus, if h crosses g, it has to cross g at least twice. This is not allowed in simple drawings. $\square$

Figure 1.2: Edges e and f cross. The edge g crosses neither e nor f. The left and right drawing show the two possible ways to draw g. The boundary of the area A is drawn in red and the area itself is shaded pink.

1.4 About the rotation system

Even with the restrictions that simple drawings have, it is hard to tell where their crossings are. In drawings of graphs where all edges have to be straight lines, the position of the points determines exactly which edges cross. In simple drawing this is no longer the case. In fact, the position of the points alone does not even determine the number of crossings. An example for that can be seen in Figure 1.3.

Figure 1.3: Two simple drawings of K_4 where the points have the same positions, but the drawings have different crossings.

There is a tool that can solve this problem for simple drawings of the complete graph, called *rotation systems*.

There are different ways to define rotation systems. The following definitions are taken from [14].

Definition 1.4.1. Let D be a simple drawing of a graph with vertex set V and edge set E. The *rotation* around a vertex $v \in V$ is the clockwise cyclical order in which the edges incident to v leave the vertices. The *rotation system* of D is the set of rotations around all its vertices.

By means of the rotation system, we define the term isomorphism for simple drawings.

Definition 1.4.2. [14] Two simple drawings D and D' are *weakly isomorphic* if and only if there exists an incidence preserving one-to-one correspondence between the vertex set of D and the vertex set of D' such that a pair of edges in D crosses if and only if the corresponding pair of edges in D' crosses.

Two simple drawings D and D' are *isomorphic* if and only if

1. D and D' are weakly isomorphic.
2. For each edge e of D the order of crossings with other edges is the same as the order of crossings on the corresponding edge e' of D'.
3. The rotation systems of D and D' are the same or inverse.

Rotation systems were already introduced in 1891. They were used by Heffter in his studies of the genus of graphs [10].

The usefulness of rotation systems in the study of crossings was shown in 2004. Then Pach and Tòth showed that the rotation system of a simple drawing of the complete graph determines which pairs of edges cross [17]. In 2005 Gioan showed that the other direction is true as well. The crossings in a simple drawing of the complete graph determine the rotation system [5].

For other graphs no such equivalences have been shown. But a tool that carries information about crossings would be desirable for all graphs, especially for complete bipartite graphs. After all, the brick factory problem is on complete bipartite graphs and it is in fact the first of the crossing number problems and possibly one of the most famous problems in graph theory. In Chapter 2, we study how much information the rotation system holds for graphs that are not complete.

1.5 On plane spanning trees and shooting stars

When studying the crossings of graphs and drawing them, we found ourselves wishing for some way to divide the graph into independent areas. That might simplify the finding of crossings. Something that would come in especially handy is knowing a plane spanning tree. We could start with drawing the tree and then add the other edges and try to find out when crossings appear. That would require the drawings we work with to contain plane spanning trees. But it is not known which simple drawings of graphs contain such substructures.

It is easy to see that simple drawings of the complete graph contain plane spanning trees. One such spanning tree consists of all edges that are incident to an arbitrary vertex. It is plane because the edges are adjacent and the drawing is a simple drawing. It is spanning because every vertex of a complete graph is adjacent to all other vertices in that graph.

However, we are interested in graphs beyond the complete graph and especially in complete bipartite graphs. Thus the question arises:

Question 1.5.1. *Do all simple drawings of complete bipartite graphs contain plane spanning trees?*

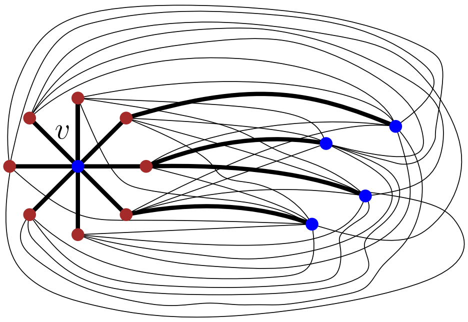
We are not the first who are interested in plane subdrawings of simple drawings. There has already been interesting research on them. In 1988 Rafla conjectured that every simple drawing of K_n contains a plane Hamiltonian cycle [19]. (A Hamiltonian cycle is a spanning subgraph that is a cycle.) This conjecture has been proven for some special types of simple drawings, but remains open for the general case. In 2005 Pach and Tóth showed that a simple drawing with n vertices and no k pairwise distjoint edges has at most $O(n \log^{4k-8} n)$ edges [18].

There is also very recent work on non-crossing substructures in simple drawings. In 2015 Ruiz-Vargas showed that every simple drawing of the complete graph K_n contains $\Omega(n^{1/2-\epsilon})$ pairwise disjoint edges [20]. This is an improvement to the result from Suk, who showed in 2013 that there are $\Omega(n^{1/3})$ pairwise disjoint edges [21]. In the same year Fulek and Ruiz-Vargas presented a new proof for this and showed that a simple drawing of the complete graph K_n contains at least $2n/3$ empty triangles [4].

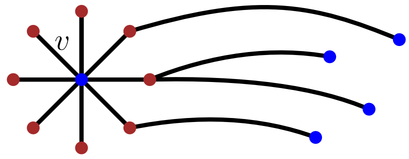
Most of the mentioned work deals with simple drawings of complete graphs. None of them answer our question of whether complete bipartite graphs contain plane spanning trees. It turned out that this is not an answer obtained trivially. Indeed the question of whether all simple drawings of $K_{m,n}$ contain plane spanning trees is still open. But we have shown that some simple drawings do. We present our research on this in Chapter 3. More precisely, we show that some simple drawings contain a particular kind of plane spanning tree that we call shooting star.

Definition 1.5.2 (Shooting star)**.** Let D be a simple drawing of a graph and v be a vertex of D. We call all the edges incident to v the *star of* v. A *shooting star rooted at* v is a plane spanning tree of D that contains the star of v.

In a simple drawing of a bipartite graph the star of v connects v with all vertices on the side of the bipartition that v is not part of. To obtain a spanning tree, there must be one additional edge for every vertex on the same side of the bipartition as v (except v). We could see those additional edges as tails of a shooting star, which motivates our term “shooting star”. Figures 1.4 and 1.5 display shooting stars.

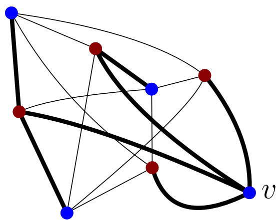


(a) A drawing of $K_{5,8}$ containing a shooting star rooted at v that is drawn in bold lines.

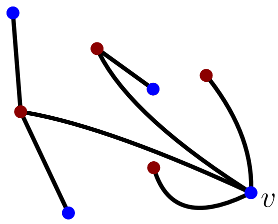


(b) The shooting star from the left drawing without the edges that are not contained in the shooting star.

Figure 1.4: A shooting star of a simple drawing of $K_{5,8}$. It is rooted at v.



(a) A drawing of a $K_{4,5}$ containing a shooting star rooted at v that is drawn in bold lines.



(b) The shooting star from the left drawing without the edges that are not contained in the shooting star.

Figure 1.5: An example for a shooting star of a simple drawing of $K_{4,4}$. It is rooted at v.

1.6 Outline of the thesis

Chapter 2 is dedicated to rotation systems of simple drawings. We study how much information about the number of crossings the rotation system contains. First we consider simple drawings of the complete graph K_n and take a look at a proof that the rotation system determines the pairs of crossing edges. Then we study simple drawings of graphs with an edge less than the complete graph. We show that the rotation system in this case determines the number of crossings.

Next, we show that this is also true for graphs with $n \geq 5$ vertices and a minimal degree of at least $(n-2)$. Then we study complete bipartite graphs. We show that for simple drawings of the complete bipartite graph $K_{2,3}$ the parity of the number of crossings is determined by the rotation system. We use our result for a proof that $K_{3,3}$ is not planar. It is already known that the parity of the number of crossings of all simple drawings of $K_{m,n}$ with m, n fixed and odd is the same. We briefly look at this result. Finally, we talk about graphs for which the rotation system does not give any information.

Chapter 3 is dedicated to finding plane spanning trees in complete bipartite graphs. We show that simple drawings of the complete bipartite graphs $K_{2,n}$ and $K_{3,n}$ contain shooting stars. Next, we look at special types of drawings of complete bipartite graphs that are called *outer drawings*. We define them and show that they contain plane spanning trees. Then we look at other special kinds of simple drawings of the complete bipartite graph $K_{m,n}$ that contain shooting stars. Those drawings are straight line drawings, 2-page book drawings and circular drawings. That all simple drawings of the complete bipartite graph $K_{m,n}$ contain plane spanning trees remains a conjecture.

In the last chapter, Chapter 4, we summarize what we have done and recall the results we have been able to prove. We also recall the problems which are still open and might be interesting to do further research on.

Chapter 2

Rotation systems and number of crossings

As mentioned in Section 1.4 the rotation system of drawings of complete graphs determines which pairs of edges cross [17]. We show that it is sufficient to know the rotation around all but one vertex. The observation that the rotation system carries more information than necessary, made us carefully optimistic that it might still bring results for non-complete graphs. Complete bipartite graphs are of particular interest to us. They share some features with complete graphs, which makes them promising candidates. Additionally, there has been much interest in crossings of complete bipartite graphs for a long time. As explained in Section 1.2 the first research on crossing numbers of complete bipartite graphs is older than that on complete graphs.

First, in Section 2.1, we look at the (already known) proof that the rotation system of simple drawings of the complete graph determines the crossings. Then, in Section 2.2, we show that the rotation system of graphs that have one edge less than the complete graph determines the number of crossings, but not exactly which edges cross. In Section 2.3 we proof that in graphs with at least five vertices where every vertex is adjacent to all but at most one other vertex the number of crossings is fixed as well. We first consider graphs with only five vertices and then generalize our result. We also notice that in general, the number of crossings is not determined by the rotation system. In Section 2.4 we study complete bipartite graphs in particular. We show that the parity of the number of crossings in the complete bipartite graph $K_{2,3}$ is determined by the rotation system. We also note which rotation system leads to an odd number of crossings and which to an even number. We use this to prove that the complete bipartite graph $K_{3,3}$ is not planar. Then we state that for two fixed, odd integers m and n, the parity of the number of crossings of all simple drawings of $K_{m,n}$ is the same. Finally, we show that the rotation system does not determine the parity in simple drawings of general graphs. Even graphs with five vertices that contain $K_{2,3}$ as subgraph may have simple drawings with the same rotation system, but different parities of their number of crossings.

2.1 Rotation system and crossings of simple drawings of K_n

In 2004 Pach and Tóth proved that the rotation systems of simple drawings determine exactly which pairs of edges cross [17]. In this section, we take a closer look at a proof for that. We then observe that not all information from the rotation system is needed to determine the crossings.

Theorem 2.1.1. *[17] The rotation system of a simple drawing of the complete graph determines which pairs of edges cross.*

Proof. We proof the theorem for K_4. Since all crossings of simple drawings are between pairs of edges with four distinct end vertices, the theorem then follows for the complete graph K_n.

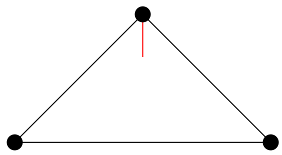
We construct our K_4 by drawing a subgraph of it that is a simple drawing of K_3. By Lemma 1.3.2 every simple drawing of K_3 is plane. The rotation system does not restrict drawings of K_3 as there are only two edges per vertex. So we can start by drawing an arbitrary plane triangle. We call the edges of that triangle $\triangle$-*edges* and the other edges *new edges.*

The rotation system then tells us at which place the new edge of each vertex is drawn. For every vertex of the simple drawing of K_3 there are two possibilities: The new edge may start in the bounded face (as it does in Figure 2.1a) or in the unbounded face (as it does in Figure 2.1b).

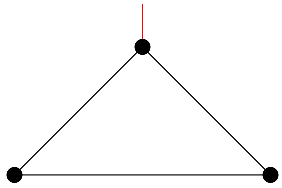
All the new edges are adjacent to each other, thus they cannot cross each other. Every new edge is also adjacent to two of the $\triangle$-edges. That means for every new edge there is only one edge it can cross. From Lemma 1.3.3 it follows further that there can be at most one crossing in total.

Since there are three new edges and only two faces, two of the new edges have to start in the same face. We say, without loss of generality, that they start inside the bounded face. As there can be at most one crossing, it is not possible that both edges cross a $\triangle$-edge. Both new edges end in the same new vertex. Hence, this vertex has to lie inside. This means the new vertex has to lie in the bounded face.

If the third edge starts in the bounded face as well, it is separated from the only edge it may cross by the other edges (as can be seen in Figure 2.2). Thus it cannot cross the edge and the drawing is plane.



(a) The beginning of the new edge is drawn in red. It starts in the bounded face.



(b) The beginning of the new edge is drawn in red. It starts in the unbounded face.

Figure 2.1: Drawings of the subgraph K_3 with the two possibilities of where a fourth edge incident to the top vertex can start.

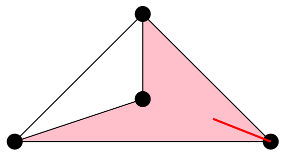


Figure 2.2: The new edge cannot leave the pink area.

If the third new edge starts in the unbounded face, it has to cross the only edge it may cross in order to be incident to the new vertex. That can be seen in Figure 2.3.

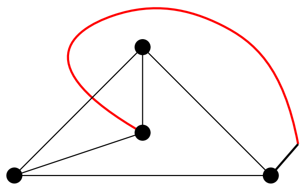


Figure 2.3: The new (red) edge has to cross the only edge it may cross.

In both cases, the pairs of edges that cross are determined. □

In our proof of Theorem 2.1.1 it can be seen that the rotation around all but one vertex is sufficient to determine which pairs of edges cross. The last vertex is determined by the orders of the other vertices.

Remark 2.1.2. The full rotation system of a simple drawing of the complete graph contains more information than needed to determine which edges cross.

Thus, the wish to still gain knowledge from the rotation system if we have less information is reasonable.

2.2 Rotation system and crossings of simple drawings with $\binom{n}{2} - 1$ edges

We are interested in graphs that are not complete. In this section we start with studying the question of what happens in graphs with exactly one edge less than complete graphs. This means we are interested in graphs with exactly two vertices that are not adjacent to each other.

Theorem 2.2.1. *The rotation system of a simple drawing of a graph with $\binom{n}{2} - 1$ edges determines the number of crossings of that drawing.*

Proof. Let D be a simple drawing of a graph with n vertices and $\binom{n}{2} - 1$ edges. As in the proof of Theorem 2.1.1, we look at the subdrawings of D that have four vertices. Those subdrawings either contain both vertices that are not adjacent or they do not contain both of them. If they do not contain both vertices that are not adjacent, they are simple drawings of K_4. It follows from the proof of Theorem 2.1.1 that the rotation system determines the pairs of edges that cross.

Let D' be a subdrawing of D where two of the vertices are not adjacent. The drawing D' without one of those two vertices is a drawing of K_3 as all the other vertices are adjacent. Thus, we can start by drawing a plane triangle as we did in the proof of Theorem 2.1.1.

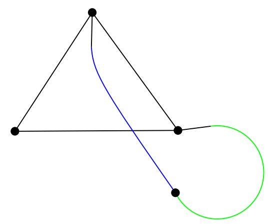
Like in the proof of Theorem 2.1.1 we call the edges not yet drawn new edges and look at whether they start in the bounded face or in the unbounded face. In the case that both start in the same face, we know from the proof of Theorem 2.1.1 that the fourth vertex has to lie in the same face and there are no crossings.

Let us consider the case that one new edge starts in the bounded face and the other edge starts in the unbounded face. Then one of the new edges has to end in a different face than it started. This means there is at least one crossing. Since there cannot be more crossings there is exactly one crossing.

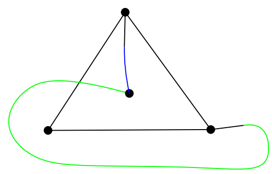
However, which edges cross and where the fourth vertex lies is not determined. There are two possibilities:

1. The fourth vertex lies in the unbounded face. This means that the edge that starts in the bounded face has to have a crossing with the only edge it is allowed to cross. This can be seen in Figure 2.4a.

2. The fourth vertex lies in the bounded face. This means that the edge that starts in the unbounded face has to have a crossing with the only edge it is allowed to cross. This can be seen in Figure 2.4b.



(a) The new edge that starts in the bounded face (the blue edge) has to cross an edge.



(b) The new edge that starts in the unbounded face (the green edge) has to cross an edge.

Figure 2.4: The fourth vertex may lie in the bounded face or in the unbounded face. Its position determines which two edges cross each other.

□

2.3 Graphs with $n \geq 5$ vertices and a minimal degree of $(n-2)$

In the last section we have shown that the rotation system of a simple drawing of a graph with one edge less than the complete graph determines the number of crossings in that drawing. In this section we show that this statement holds for graphs with $n \geq 5$ vertices having a minimal degrees of at least $(n-2)$. In order to do that, we prove the statement for graphs with five vertices. From that together with Theorem 2.1.1 and Theorem 2.2.1, we derive the theorem for graphs with n vertices.

2.3.1 Graphs with five vertices and a minimal degree of 3

We first consider graphs with only five vertices and a minimal degree of 3. We show that rotation systems of simple drawings of such graphs determine the number of crossings. To proof that we use the well known *Handshaking lemma.*

Lemma 2.3.1 (Handshaking lemma)**.** *Let G be a graph with vertices V and edges E. We denote with $|E|$ the size of set E. Then the sum of the degrees in G is $2|E|$.*

Proof. The degree of a vertex is the number of edges it is incident to. Every edge is incident to exactly two vertices. Thus, every edge adds exactly 2 to the sum of degrees. □

Theorem 2.3.2. *Let G be a graph with five vertices and a minimal degree of 3. Then the rotation system of simple drawings of G determine their number of crossings.*

Proof. Every vertex in G has degree 3 or 4. It follows from Lemma 2.3.1 that there cannot be an odd number of vertices with odd degree. Thus, either no vertices, two vertices, or four vertices have degree 3. If no vertex has degree 3, the theorem follows from Theorem 2.1.1. If two vertices have degree 3, the theorem follows from Theorem 2.2.1.

So let us assume that four vertices have degree 3. We call the vertices with degree 3 vertices v_1, v_2, v_3, and v_4. The other vertex has to have the full degree of 4. We call that vertex v_5.

We proof the theorem by looking at all possible non-isomorphic simple drawings of graphs with five vertices and a minimal degree of 3. Then we compare the number of crossings of all drawings that have the same or equivalent rotation systems. Two rotation systems are equivalent if they would be the same after relabeling the vertices.

Basics for constructing all simple drawings

We create our simple drawings step by step. First we consider the subgraph H that is obtained from G by deleting the vertex v_5 and all edges incident to it.

Claim 2.3.3. *The subgraph H is a 4-cycle, that means it is a cycle that contains exactly four vertices.*

<u>Proof of claim:</u> All vertices of H have degree 3 in G. The edges of H are the edges of G without the edges incident to v_5. As every vertex of H is adjacent to v_5 in G, every vertex of H has 1 degree less in H than it has in G. Thus, all vertices of H have degree 2 in H. It follows, that all vertices have to be in cycles.

Any cycle has at least three vertices. Hence, there cannot be more than one cycle. It follows that all vertices in H have to be in the same cycle in H. Hence, H has to be a 4-cycle. ■

Without loss of generality, we label the vertices so that they are in order v_1, v_2, v_3, v_4 if we go along the cycle. More formally, v_i is adjacent to v_{i+1} and v_{i-1} for i in $\{2, 3\}$ and v_1 is adjacent to v_4.

There are two non-isomorphic ways to draw H.

1. There is no crossing. Then the drawings in which vertex v_5 is drawn in the bounded face and the drawings in which it is drawn in the unbounded face are isomorphic. Thus, we only need to look at drawings where it lies in the bounded face.

    - *Case 1:* There is no crossing in H and the vertex v_5 lies in the bounded face. This can be seen in Figure 2.5.

2. There is a crossing. Then there are two bounded faces. We call the bounded face with v_1 and v_2 on its boundary F_1. We call the bounded face with v_3 and v_4 on its boundary F_2. We have to consider the different possibilities of where vertex v_5 may lie.

    - *Case 2*: The vertex v_5 lies in the unbounded face. This can be seen in Figure 2.6.
    - *Case 3:* The vertex v_5 lies in a bounded face. It does not matter in which bounded face it lies because the two cases are isomorphic. We draw it in F_1. This can be seen in Figure 2.7.

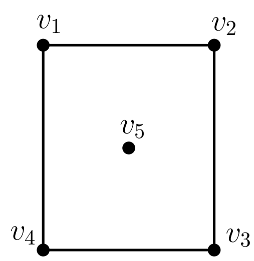


Figure 2.5: *Case 1.*

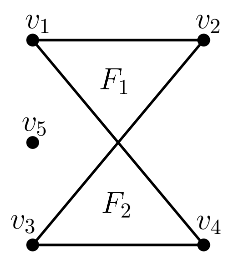


Figure 2.6: *Case 2.*

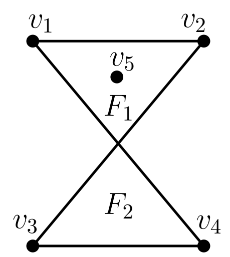


Figure 2.7: *Case 3.*

We call edges incident to v_5 *new* edges (because we add them "newly" after we drew H). We say a new edge incident to a vertex v_i in H starts in v_i and ends in v_5. As in the proofs of Theorem 2.1.1 and Theorem 2.2.1, we now distinguish where the new edges start.

Case 1 (There is no crossing between edges in H)

Claim 2.3.4. *It cannot happen that more than two new edges start in the unbounded face.*

Proof of claim: Assume that there are at least three vertices whose new edges start in the unbounded face. Then there exists a vertex that starts in the unbounded face and is adjacent to at least two vertices whose new edges start in the unbounded face. Without loss of generality, let that vertex be v_3, and let v_2 and v_4 be the other vertices whose new edges start in the unbounded face. That can be seen in Figure 2.8.

We call the new edge that is incident to v_3 edge e. The vertex v_5 is an endpoint of e and lies in the bounded face. As e does not lie in that face, it has to cross the boundary to reach v_5. It has to cross either the edge joining v_1 and v_2 or the edge joining v_1 and v_4. This follows from the fact that e is adjacent to the other two edges of the boundary.

If e crosses the edge joining v_1 and v_2, it prevents the new edge incident to v_2 from starting in the unbounded face, as can be seen in Figure 2.8a. If e crosses the edge joining v_1 and v_4, it prevents the new edge incident to v_4 from starting in the unbounded face, as can be seen in Figure 2.8b.

■

Thus there are four non-isomorphic possibilities how the edges of Case 1 start. They are drawn in Figure 2.9.

Case 2 (There is a crossing between the edges in H and the vertex v_5 lies in the unbounded face)

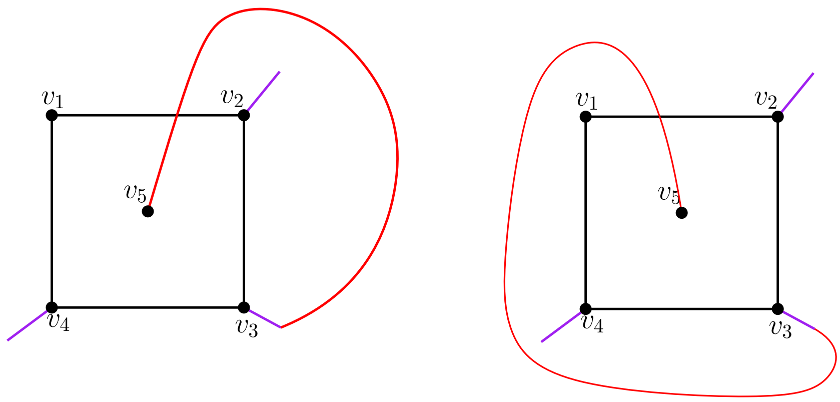
Claim 2.3.5. *Two new edges cannot start inside the same bounded face.*

Proof of claim: We assume, without loss of generality, that the new edge incident to v_4 and the new edge incident to v_3 start inside F_2. This can be seen in Figure 2.10. We call the new edge incident to v_4 edge e.

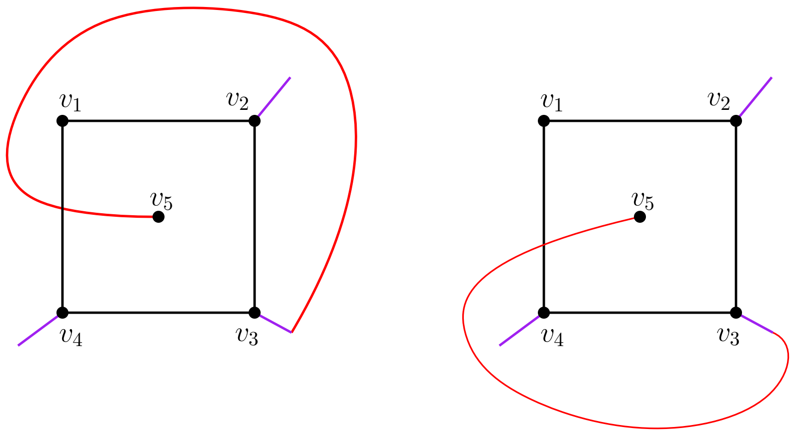
The vertex v_5 is an endpoint of e and does not lie in F_2. Since e starts inside F_2, it has to cross the boundary of F_2 to reach v_5. In particular, it has to cross the only edge on the boundary of F_2 that is not incident to v_4. This is the edge joining v_3 and v_2. As can be seen in Figure 2.10, the edge e then prevents the new edge incident to v_2 from starting in the unbounded face.

■

There are four non-isomorphic possibilities how the new edges can start. They are drawn in Figure 2.11.

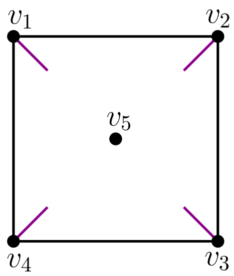


(a) There is no possibility to add the edge joining v_2 and v_5.

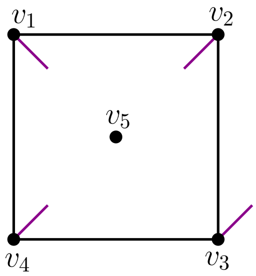


(b) There is no possibility to add the edge joining v_4 and v_5.

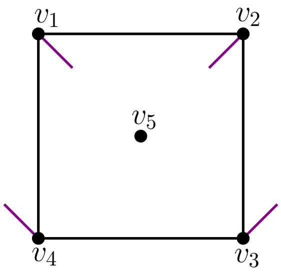
Figure 2.8: The purple lines show where the new edges have to start according to the rotation system. There are two ways to draw the new edge joining v_3 and v_5. In both drawings one of the other new edges cannot be drawn without violating the rotation system.



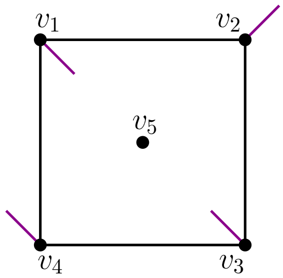
(a) *Case 1A:* All new edges start in the bounded face.



(b) *Case 1B:* One new edge starts in the unbounded face and all the others start in the bounded face. Here the vertex that is incident to the new edge starting in the unbounded face is v_3. All other ways to draw this are isomorphic.



(c) *Case 1C:* There are two new edges that start in the bounded face. The vertices (other than v_5) that are incident to those edges are adjacent to each other. The other new edges start in the unbounded face. Here the vertices incident to any new edges that start in the bounded face are labeled v_1 and v_2. All other ways to draw this are isomorphic.



(d) *Case 1D:* There are two new edges that start in the bounded face. The vertices (other than v_5) that are incident to those edges are not adjacent to each other. The other new edges start in the unbounded face. Here the vertices that start in the unbounded face are labeled v_2 and v_4. All other ways to draw this are isomorphic.

Figure 2.9: All possibilities for Case 1.

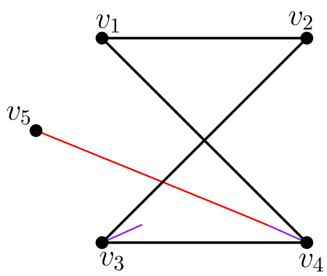
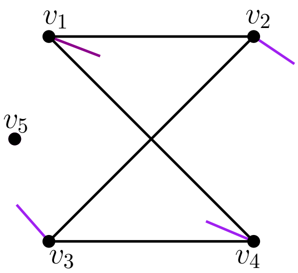
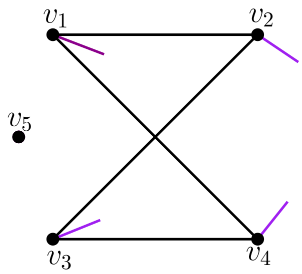


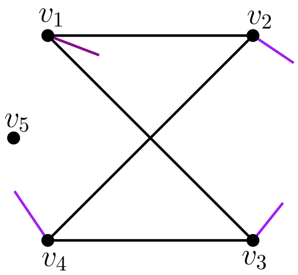
Figure 2.10: Two new edges cannot start in the same bounded face. The new edge incident to v_4 that is drawn in red traps the new edge incident to v_3.



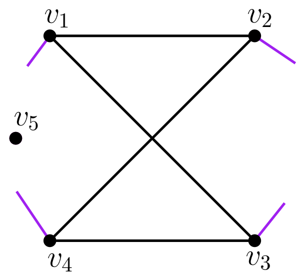
(a) *Case 2A*: Two new edges start in the unbounded face. They are adjacent to each other. The other new edges start inside two different bounded faces. Here the new edges that start in the unbounded face are incident to v_2 and v_3. The other drawing is isomorphic.



(b) *Case 2B*: Two new edges start in the unbounded face. They are not adjacent to each other. The other new edges start inside two different bounded faces. Here the new edges that start in the unbounded face are incident to v_2 and v_4. The other drawing is isomorphic.



(c) *Case 2C*: Exactly three new edges start in the unbounded face. Here the new edge that starts in the unbounded face is incident to v_1. The other drawings are isomorphic.



(d) *Case 2D*: All new edges start in the unbounded face.

Figure 2.11: All possibilities for Case 2.

Case 3 (There is a crossing between the edges in H and the vertex v_5 lies in a bounded face)

The new edge incident to v_3 and the new edge incident to v_4 cannot both start inside F_2. This follows from the proof of Claim 2.3.5. In the proof it is only relevant that v_5 does not lie in the same face as the edges that start inside that face. The information of whether it is in the unbounded face or in F_2 is not necessary. A drawing of how the edge incident to v_4 would trap the edge incident to v_3 can be seen in Figure 2.12.

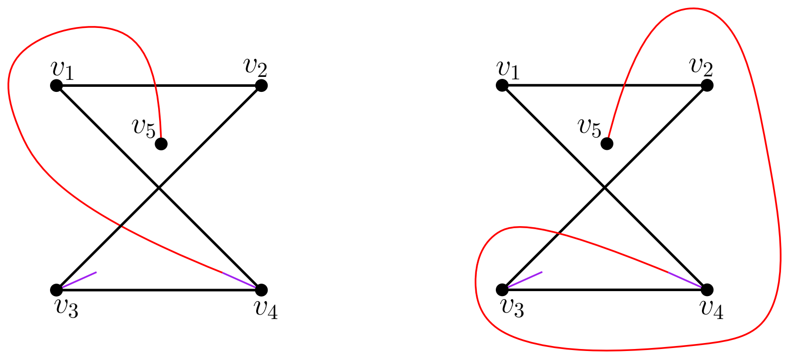


Figure 2.12: Two new edges incident to v_3 and v_4 cannot both start in F_2. The red edge incident to v_4 would trap the edge incident to v_3.

Claim 2.3.6. *Either the edge joining v_1 and v_5 or the edge joining v_2 and v_5 has to start inside F_1.*

<u>Proof of claim:</u> Assume that the edge joining v_1 and v_5, and the edge joining v_2 and v_5 do not start inside F_1. We call the edge joining v_1 and v_5 edge e. Similar to the proofs of the previous claims, the edge e has to cross the boundary of F_1 in order to reach v_5. In particular, it has to cross the one edge on the boundary of F_1 that e is not adjacent to. This is the edge joining v_2 and v_4. As can be seen in Figure 2.13, the edge e then prevents the new edge incident to v_2 from starting in the unbounded face.

■

Claim 2.3.7. *Either the edge joining v_5 and v_1, and the edge joining v_5 and v_2 both start inside F_1 or the edges joining v_5 with v_3, and v_4 both start in the unbounded face.*

<u>Proof of claim:</u> Assume that the edge joining v_4 and v_5 starts in F_2. We call that edge e. The boundaries of faces F_1 and F_2 intersect in only one common point. This point is a crossing of two edges and therefore cannot be intersected by any other edges. As v_1 lies in F_1, it follows that e has to cross two edges. It has to

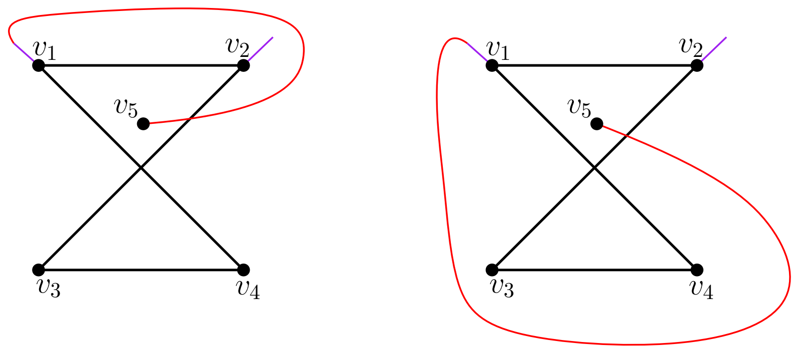


Figure 2.13: The new edges incident to v_1 and v_2 cannot both start in the unbounded face. The red edge incident to v_1 would trap the edge incident to v_2.

cross an edge on the boundary of F_2 to leave the face it started inside of, and an edge on the boundary of F_1 to reach v_5. There is only one edge on the boundary of F_2 that e is not adjacent to. This is the edge joining v_1 and v_4. There are two edges on the boundary of F_2 that e is not adjacent to. Those edges are the edge joining v_1 and v_4, and the edge joining v_1 and v_2. As any two edges can cross at most once, e cannot cross the edge joining v_1 and v_4 twice. It follows that e crosses the edge joining v_1 and v_4 on the boundary of F_2, and the edge joining v_1 and v_3 on the boundary of F_1. As can be seen in Figure 2.14 the edge e then prevents the new edge incident to v_1 and the new edge incident to v_2 from starting outside F_2. ■

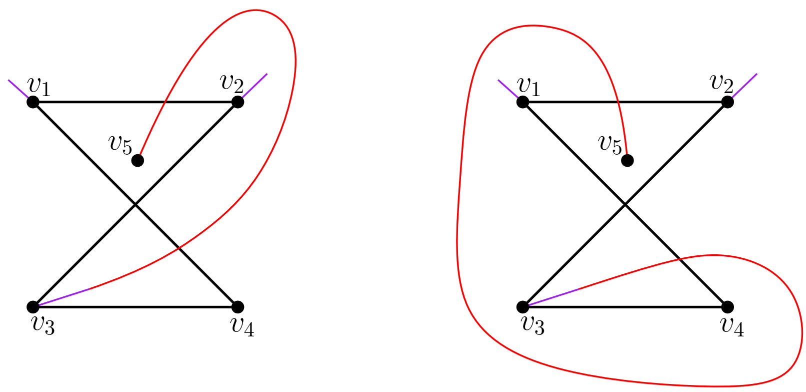
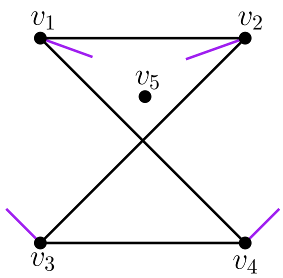
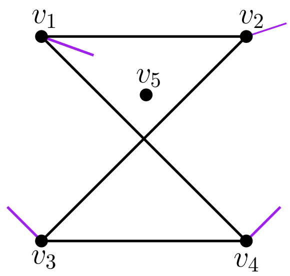


Figure 2.14: If a new edge incident to v_3 starts in F_2, neither the new edge incident to v_1 nor the edge incident to v_2 can start in the unbounded face. The red edge incident to v_4 would trap the edges incident to v_1 and v_2.

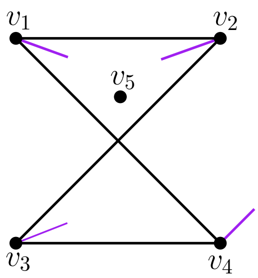
There are three non-isomorphic possibilities of how the new edges can start. They are drawn in Figure 2.15.



(a) *Case 3A*: The two new edges incident to v_1 and v_2 start inside a bounded face. The two new edges incident to v_3 and v_4 start in the unbounded face.



(b) *Case 3B*: The only new edge that starts inside a bounded face is the new edge incident to v_1. (The case where the only edge that starts inside a bounded face is the new edge incident to v_2 is isomorphic.)



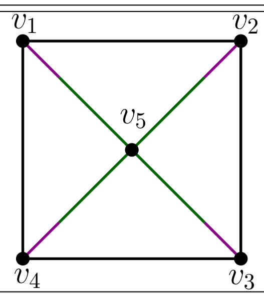
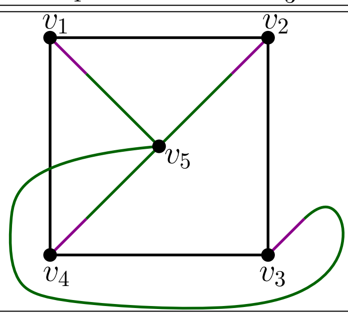
(c) *Case 3C*: The only new edge that starts in the unbounded face is the new edge incident to v_4. (The case where the only new edge starting in the unbounded face is incident to v_3 is isomorphic.

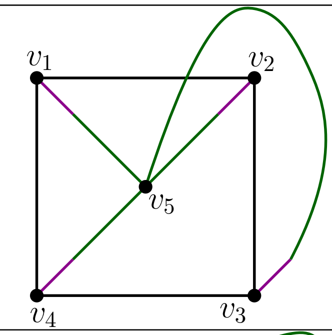
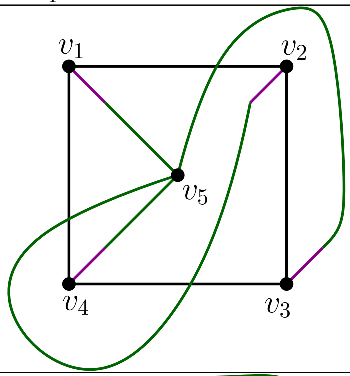
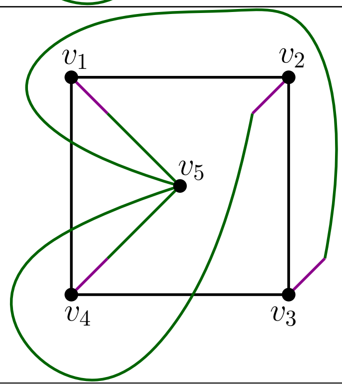
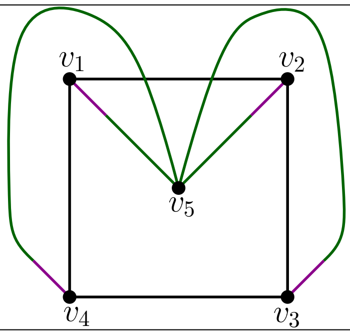
Figure 2.15: All possibilities for Case 3.

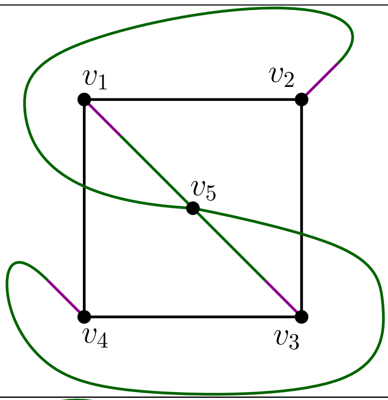
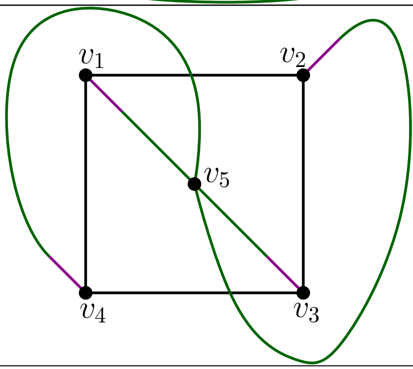
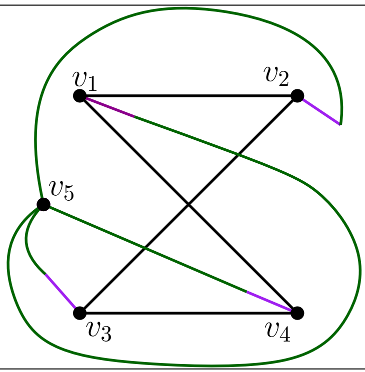
The simple drawings

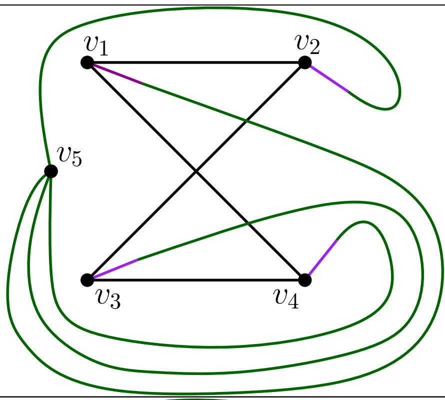
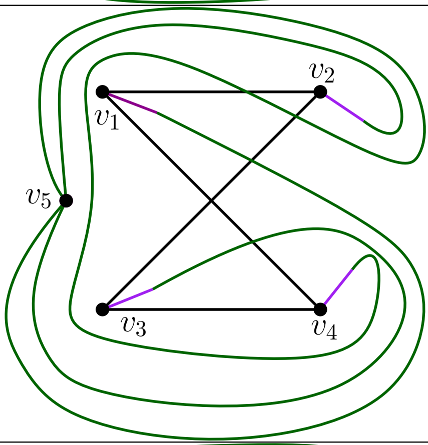
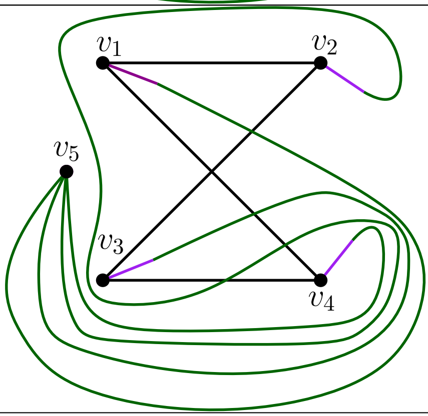
Using the unfinished drawings above, we create all possible drawings of graphs with five vertices of which one has full degree 4 and the others have degree 3. The drawings are listed in the table below. Each row represents a simple drawing. In the first column we write which case the drawing came from. The actual drawing is in the second column. We write the rotation system in the third column. Instead of writing down the actual labels of the vertices, we represent them by just their indices. This means we write i instead of v_i. We write the number of crossings of the drawing in the fourth column. In the header of the table we have used the symbol # as an abbreviation for "number of". In the last column we give the current rotation system a label.

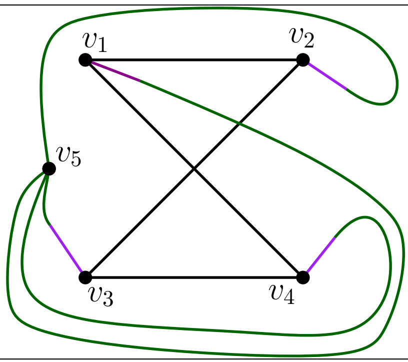
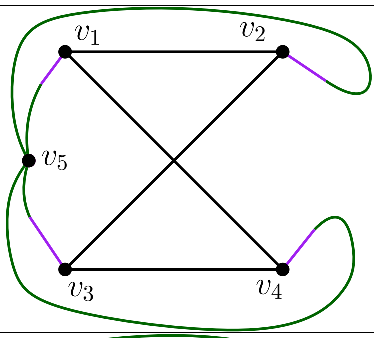
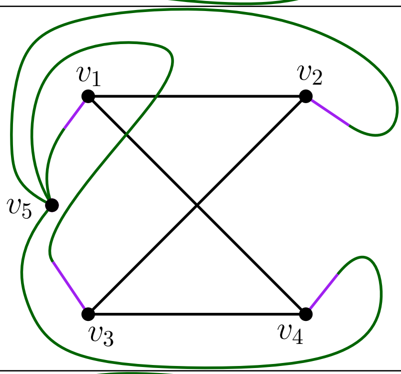
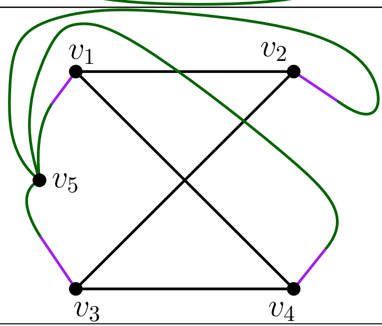
To confirm that these are all the possible drawings, we have included drawings that could not be finished in Figure 2.16. These are cases where an edge can be added in another way that does not break the rules of edges in simple drawings. However, it would be impossible to finish the drawing to a simple drawing.

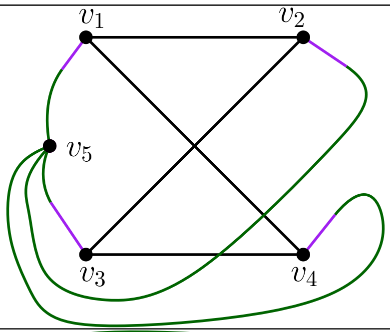
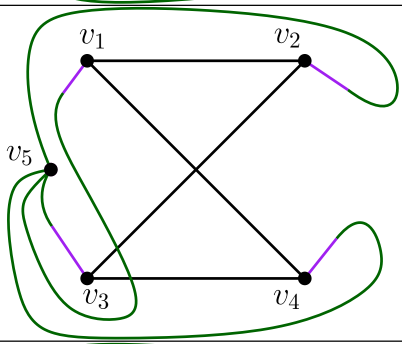
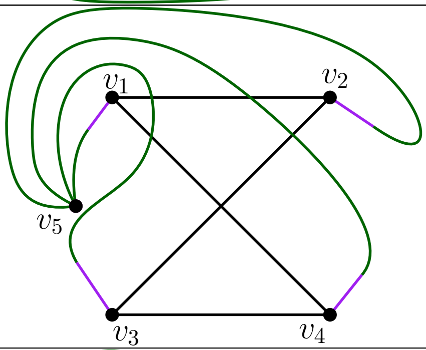
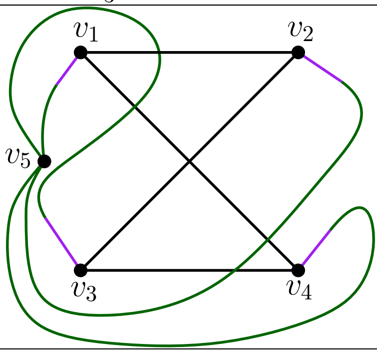
Case	**simple drawing**	**rotation system**	**#crossings**	**label**
1A		1 : 2 5 4 2 : 1 3 5 3 : 2 4 5 4 : 1 5 3 5 : 1 2 3 4	0	R1
1B		1 : 2 5 4 2 : 1 3 5 3 : 2 5 4 4 : 1 5 3 5 : 1 2 4 3	1	R2

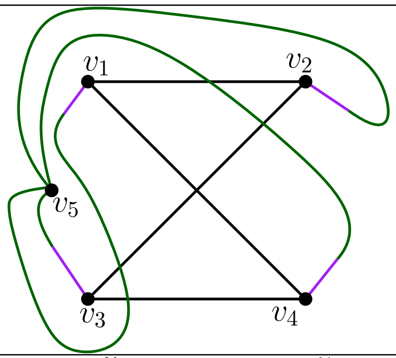
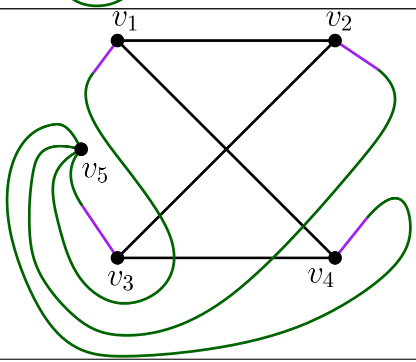
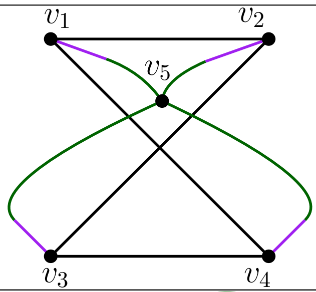
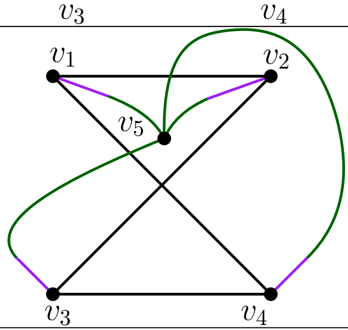
Case	simple drawing	rotation system	#crossings	label
1B		1 : 2 5 4 2 : 1 3 5 3 : 2 5 4 4 : 1 5 3 5 : 1 3 2 4	1	R3
1B		1 : 2 5 4 2 : 1 3 5 3 : 2 5 4 4 : 1 5 3 5 : 1 3 4 2	3	R4
1B		1 : 2 5 4 2 : 1 3 5 3 : 2 5 4 4 : 1 5 3 5 : 1 4 2 3	3	R5
1C		1 : 2 5 4 2 : 1 3 5 3 : 2 5 4 4 : 1 3 5 5 : 1 4 3 2	2	R6

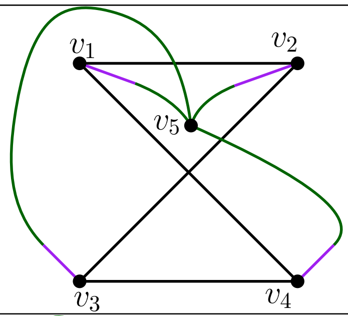
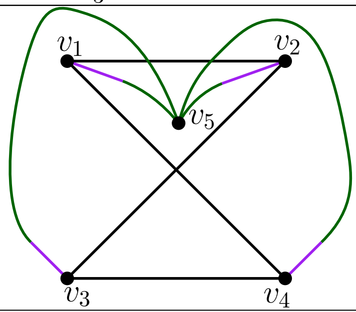
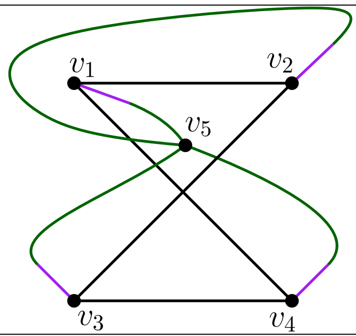
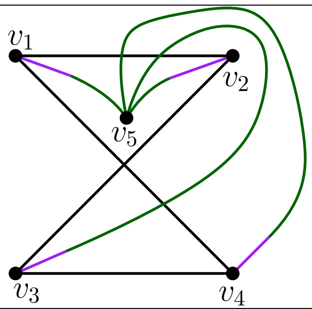
Case	**simple drawing**	**rotation system**	**#crossings**	**label**
1D		1 : 2 5 4 2 : 1 5 3 3 : 2 4 5 4 : 1 3 5 5 : 1 4 3 2	2	R7
1D		1 : 2 5 4 2 : 1 5 3 3 : 2 4 5 4 : 1 3 5 5 : 1 4 3 2	2	R7b
2A		1 : 2 5 4 2 : 1 5 3 3 : 2 4 5 4 : 1 3 5 5 : 1 2 4 3	3	R8

Case	**simple drawing**	**rotation system**	**#crossings**	**label**
2B		1 : 2 5 4 2 : 1 5 3 3 : 2 5 4 4 : 1 5 3 5 : 1 2 4 3	3	R9
2B		1 : 2 5 4 2 : 1 5 3 3 : 2 5 4 4 : 1 5 3 5 : 1 4 2 3	5	R10
2B		1 : 2 5 4 2 : 1 5 3 3 : 2 5 4 4 : 1 5 3 5 : 1 4 2 3	5	R10b

Case	**simple drawing**	**rotation system**	**#crossings**	**label**
2C		1 : 2 5 4 2 : 1 5 3 3 : 2 4 5 4 : 1 5 3 5 : 1 2 3 4	2	R11
2D		1 : 2 4 5 2 : 1 5 3 3 : 2 4 5 4 : 1 5 3 5 : 1 3 4 2	1	R12
2D		1 : 2 4 5 2 : 1 5 3 3 : 2 4 5 4 : 1 5 3 5 : 1 4 2 3	3	R13
2D		1 : 2 4 5 2 : 1 5 3 3 : 2 4 5 4 : 1 5 3 5 : 1 3 2 4	3	R14

Case	simple drawing	rotation system	#crossings	label
2D		1 : 2 4 5 2 : 1 5 3 3 : 2 4 5 4 : 1 5 3 5 : 1 3 2 4	3	R14b
2D		1 : 2 4 5 2 : 1 5 3 3 : 2 4 5 4 : 1 5 3 5 : 1 4 2 3	3	R13b
2D		1 : 2 4 5 2 : 1 5 3 3 : 2 4 5 4 : 1 5 3 5 : 1 2 4 3	5	R15
2D		1 : 2 4 5 2 : 1 5 3 3 : 2 4 5 4 : 1 5 3 5 : 1 2 4 3	5	R15b

Case	**simple drawing**	**rotation system**	**#crossings**	**label**
2D		1 : 2 4 5 2 : 1 5 3 3 : 2 4 5 4 : 1 5 3 5 : 1 2 4 3	5	R15c
2D		1 : 2 4 5 2 : 1 5 3 3 : 2 4 5 4 : 1 5 3 5 : 1 2 4 3	5	R15d
3A		1 : 2 5 4 2 : 1 3 5 3 : 2 4 5 4 : 1 5 3 5 : 1 2 4 3	3	R16
3A		1 : 2 5 4 2 : 1 3 5 3 : 2 4 5 4 : 1 5 3 5 : 1 4 2 3	3	R17

Case	simple drawing	rotation system	#crossings	label
3A		1 : 2 5 4 2 : 1 3 5 3 : 2 4 5 4 : 1 5 3 5 : 1 3 2 4	3	R18
3A		1 : 2 5 4 2 : 1 3 5 3 : 2 4 5 4 : 1 5 3 5 : 1 3 4 2	3	R19
3B		1 : 2 5 4 2 : 1 5 3 3 : 2 4 5 4 : 1 5 3 5 : 1 4 3 2	4	R20
3C		1 : 2 5 4 2 : 1 3 5 3 : 2 5 4 4 : 1 5 3 5 : 1 4 3 2	4	R21

Simple drawings with the same Rotation System

We now have to compare the number of crossings of simple drawings which have the same rotation system. We have calculated which rotation systems are equivalent

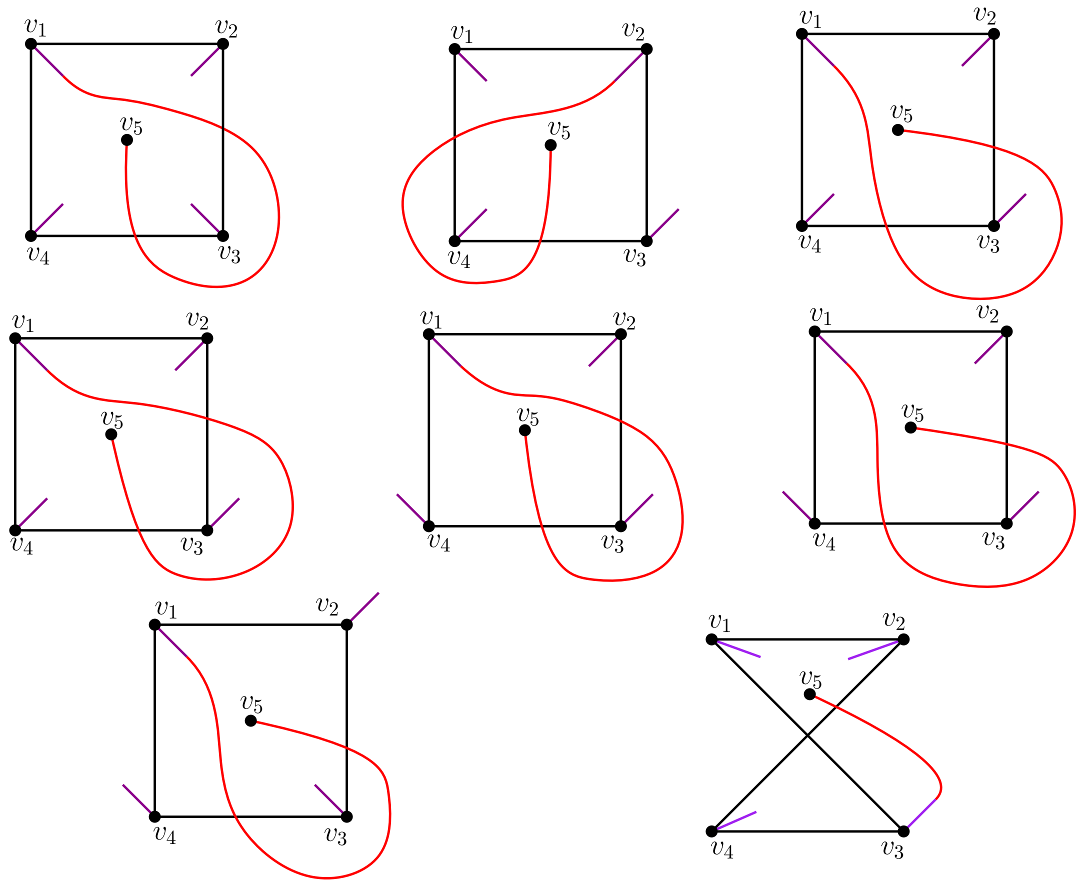


Figure 2.16: If we add the red edge as we do in these drawings, then the drawings cannot be finished.

to each other by trying all permutations of labels via computer. We used python's numpy to do that. The code can be found in Appendix A. Here we present how the program works.

The algorithm

We save all given rotation systems into matrices of the following *standard form*: Every row of the matrix corresponds to one vertex. The rows are sorted such that row i corresponds to vertex v_i.

The first four rows are as we have written them in the table above. The first entry we write is the index of the vertex it corresponds to. The following entries are the rotation around that vertex. When writing down the rotation we start with the smallest index (as second entry in the row).

In the last row we write only the rotation around v_5 without writing 5 in any of the entries. We again write down the rotation starting with the smallest entry, meaning in this case that the first entry of the row is 1.

The program works by first creating all matrices that can be obtained by permuting the labels. Then it brings the new matrices into standard form. It saves them together with an id that tells us of which drawing the original matrix is the rotation system. Finally, it looks for duplicates amongst the matrices. If there are duplicates with different ids, then the matrices they originated from are equivalent rotation systems.

Algorithm 1 Finding all equivalent rotation systems

```
Require: The standard form matrices that represent the rotation systems
perms= All permutations within the set {1, 2, 3, 4}.

function ALL_PERMS(a matrix M)
    for p ∈ perms do
        m= matrix created by performing permutation p on M;
        m= repair(m)    ▷ The matrix m is brought into standard form using the functions that are explained below.
    return m, p
function REPAIR_ROW(row, skip)    ▷ The value skip is either 0 (if dealing with the last row) or 1 (if dealing with one of the first four rows).
    Leave the first skip entries as they are.    ▷ For the first four rows this ensures that the first entry is not changed. It is the vertex the row corresponds to and not part of the rotation.
    For the other entries rotate the enumeration of the neighbours such that it starts with the lowest one.    ▷ This sorts the rotation such that it starts with the minimum index.
    return row
function REPAIR(m)
    for the first 4 rows do
        repair_row(row,1);    ▷ The considered row will be brought into standard form. Setting value skip to 1 is necessary to not change the first vertex and only fix the rotation.
        Sort rows such that their place is as in the standard form, meaning the first row has first entry 1, the second row has first entry 2, and so on.
    for the last row do
        repair_row(row,0);    ▷ The considered row will be brought into standard form. Setting value skip to 0 is necessary to fix the whole rotation, which starts at the first entry of the row.
    return m
for all matrices do
    Generate a global mapping of all matrices and the ids of the drawing they represent.    ▷ The ids tell us which drawing the original matrix came from
    Search for duplicates.
    Return the duplicates and the permutation p that was used to create that matrix.
```

Results

- The rotation systems of R1 to R6, R8, R11, and R12 are unique.
- The rotation systems of R7 and R7b are obviously the same. Both drawings have the same number of crossings.
- The rotation systems of R9, R13, R13b, R14 and R14b are equivalent. We can get from R9 to R13 if we swap the labels of vertices v_2 and v_3. We can get from R14 to R13 if we swap the labels of vertex v_1 and v_3 as well as the labels of vertices v_2 and v_4. It is obvious that R13 and R13b as well as R14 and R14b are the same. All of these drawings have the same number of crossings.
- The rotation systems of R10, R10b, R15, R15b to R15d are equivalent. We can get from R15 to R10 if we swap the labels from vertices v_2 and v_3. It is obvious that the rotation systems of R10 and R10b as well as those of R15 to R15d are the same. All of these drawings have the same number of crossings.
- The rotation systems of R16 to R19 are equivalent. We can get from R17 to R16 if we change the labels from R17 in this way: We label the former v_1 as v_3, the former v_2 as v_1, the former v_3 as v_4, and the former v_4 as v_2. We can get from R18 to R16 if we change the labels from R18 in the following way: We label the former v_1 as v_2, the former v_2 as v_4, the former v_3 as v_1, and the former v_4 as v_3. We can get from R19 to R16 if we swap the labels of vertices v_1 and v_4 as well as the labels of v_2 and v_3. All of these drawings have the same number of crossings.
- The rotation systems of R20 and R21 are equivalent. We can get from R21 to R20 if we change the labels from R21 in the following way: We label the former v_1 as v_3, the former v_2 as v_1, the former v_3 as v_4, and the former v_4 as v_2. Both drawings have the same number of crossings.

Conclusion

We have shown that all drawings with the same rotation system also have the same number of crossings. Thus the rotation systems of simple drawings of graphs with five vertices and a minimum degree 3 determine the number of crossings.

□

2.3.2 Graphs with $n \geq 5$ vertices and a minimal degree of $n - 2$

We can generalize Theorem 2.3.2 to all graphs with $n \geq 5$ vertices and a minimal degree of at least $n - 2$.

Theorem 2.3.8. *Let D be a simple drawing of a graph with at least five vertices and a minimal degree of $n - 2$. Then the rotation system of D determines its number of crossings.*

Proof. Every crossing in D consists of two edges and thereby is determined by a subdrawing of D with four vertices. Every such subdrawing of D is contained in exactly $(n - 4)$ bigger subdrawing of D with a vertex set of size 5. Those five vertices are the four vertices of the crossing and one other vertex. Consequently the number of crossings can be calculated directly from the number of crossings of the subdrawings. We add the number of crossings of all subdrawings of D that have five vertices and then divide it by $(n - 4)$, since we have counted each crossing exactly $(n - 4)$ times. Thus, it is sufficient to show that for every subdrawing of D with five vertices the number of crossings is determined.

Since D has minimal degree of $n - 2$ it follows that every vertex is adjacent to all but at most one other vertices. That means that in a subdrawing with 5 vertices all vertices have to be adjacent with at least 3 other vertices. Consequently, all such subdrawings fulfill the requirements of Theorem 2.3.2. It follows that the rotation system of the subdrawings determine their number of crossings. □

2.3.3 Rotation system and crossings of simple drawings of graphs with less than $\binom{n}{2} - 1$ edges

It would be desirable for the rotation system to always determine how many crossings there are for each drawing. Unfortunately, in general the rotation system does not determine the exact number of crossings. If a graph contains a vertex with degree $< n - 3$, the rotation system does not determine the crossings. As can be seen in Figure 2.17, this holds even if the graph has only two edges less than a complete graph. The rotation system of simple drawings of graphs with only four vertices and two edges less than the complete graph does not determine the number of crossings either. This is the case even if the graph has minimal degree of $n - 2 = 2$, as can be seen in Figure 2.18. Both figures proof Proposition 2.3.9.

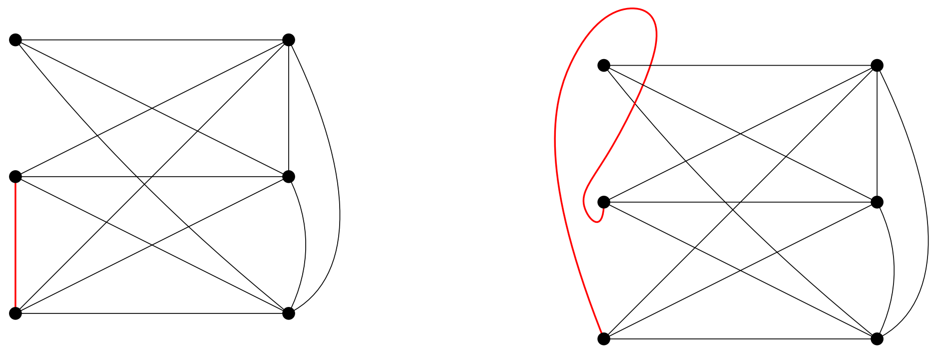


Figure 2.17: Two simple drawings of the same graph with 6 vertices. The black edges are drawn the same way in both drawings, the red edges are drawn differently. Both drawings have the same rotation system, but they have different number of crossings.

Figure 2.18: Two simple drawings of the same graph with 4 vertices. They have the same rotation system, but different number of crossings.

Proposition 2.3.9. *In general simple drawings, the rotation system alone does not determine the number of crossings.*

2.4 Complete bipartite graphs

As stated in Proposition 2.3.9, the rotation system does not determine the number of crossings in general simple drawings. We are interested in still obtaining some information from the rotation system for special kinds of graphs. As mentioned before, complete bipartite graphs seem especially interesting in this context. We study them in this section.

First we show that for simple drawings of the complete bipartite graph $K_{2,3}$ the rotation system does not determine the number of crossings, but it does determine its parity. Once we have shown this we use the result for a proof that the complete bipartite graph $K_{3,n}$ cannot be drawn plane. It is already known that the parity of the number of crossings is the same for all drawings of the complete bipartite graph $K_{m,n}$ with m and n even. We briefly discuss that result here.

2.4.1 Simple drawings of $K_{2,3}$

We show that the rotation system determines the parity of the number of crossings of simple drawings of the complete bipartite graph $K_{2,3}$. We do so by looking at all non-isomorphic simple drawings of $K_{2,3}$ and then comparing the number of crossings of those with equivalent rotation systems.

Theorem 2.4.1. *Let D be a simple drawing of $K_{2,3}$. The rotation system of D determines the parity of its number of crossing. The number is odd if the vertices within one side of the bipartition have the same rotation and even otherwise.*

Proof. We call the smaller side of the bipartition of $K_{2,3}$ side U and the other side of the bipartition V. We first make some general observations about the crossings in simple drawings of $K_{2,3}$.

Claim 2.4.2. *Any simple drawing of $K_{2,3}$ has at most 3 crossings.*

Proof of claim: Each edge has an endpoint in U and one in V. If two edges cross, they cannot be adjacent and thus all their vertices have to be different. If follows that two edges can only cross if they are incident to two different vertices in U and two different vertices in V. As U contains only two vertices, there is only one possibility to choose two different vertices. There are $\binom{3}{2} = 3$ possibilities to choose two different vertices in V. Each of those possibilities is a subdrawing with four vertices and thus can have at most one crossing by Lemma 1.3.3. ■

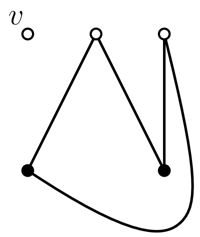
Claim 2.4.3. *An edge of a simple drawing of $K_{2,3}$ can cross at most 2 other edges.*

Proof of claim: Each edge is adjacent to 3 other edges. (Its endpoint in U is incident to 2 other edges. Its endpoint in V is incident to 1 other edge.) The complete bipartite graph $K_{2,3}$ has $(3 \cdot 2) = 6$ edges. An edge may not cross itself or its 3 incident edges. Hence it may cross at most $(6 - 1 - 3) = 2$ edges. ■

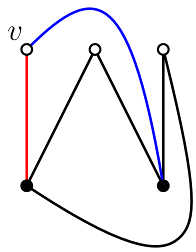
Let D be a simple drawing of $K_{2,3}$. As stated in the proof of Claim 2.4.2, every subdrawing of D that contains a crossing in G has to contain both vertices of U and two vertices of V. If there is a crossing in D, let H be a subdrawing of D that contains two crossing edges, exactly the four vertices they are incident to and all edges in D that join two vertices in H. If D does not contain any crossings, let H be a subdrawing of D that contains both vertices in U, exactly two vertices in V, and all edges in D that join two vertices in H. The only vertex of D that is not in H has to be in V. We call that vertex v. (We will relabel the vertex later.) As H is complete bipartite and both sides of the bipartition contain exactly two vertices, it has to be a 4-cycle. Similar to the proof of Theorem 2.3.2 we start with that drawing H and then look at where v lies.

We part the possible drawings into three cases:

1. *Case 1*: There is no crossing in D. Then H is plane. The vertex v can lie in the unbounded face or inside the area enclosed by H. Both cases are isomorphic. We consider the case where it lies in the unbounded face. As there are no crossings in the whole drawing, the edges incident to v cannot cross anything. Hence, there is a unique (up to isomorphism) way to draw them. Case 1 can be seen in Figure 2.19.



(a) *Case 1*: There is no crossing in H.



(b) *Drawing 1*: The edges of H are drawn in black. The edges incident to v are drawn in red and blue.

Figure 2.19: There is a unique drawing resulting from Case 1.

2. If there exists at least one crossing, there are two non-isomorphic possible drawings. *Case 2:* The vertex v lies in the unbounded face. Case 2 can be seen in Figure 2.20.

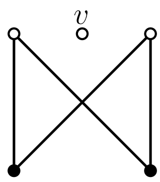


Figure 2.20: *Case 2*: There is a crossing in H. The vertex v lies in the unbounded face.

3. *Case 3*: The vertex v lies in a bounded face. Case 3 can be seen in Figure 2.21.

We then look at all the possibilities to add one new edge. This can be seen in Figure 2.22.

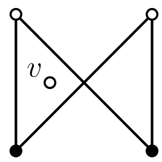
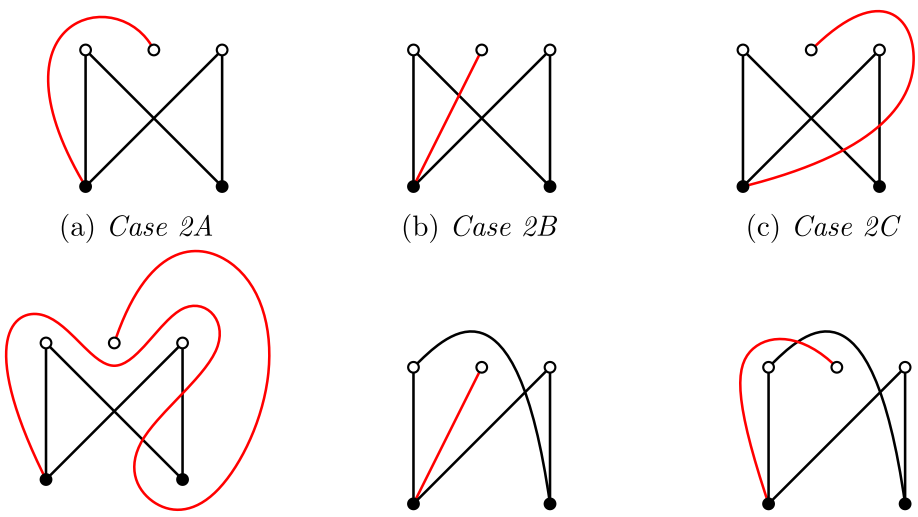


Figure 2.21: *Case 3*: There is a crossing in H. The vertex v lies in a bounded face.

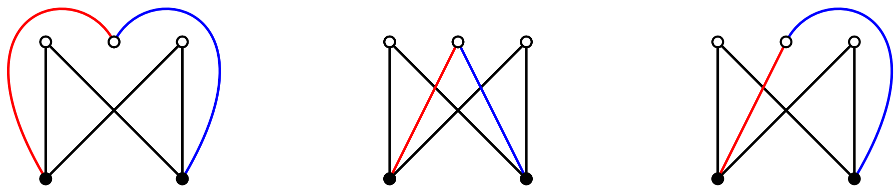


(a) *Case 2A* (b) *Case 2B* (c) *Case 2C*

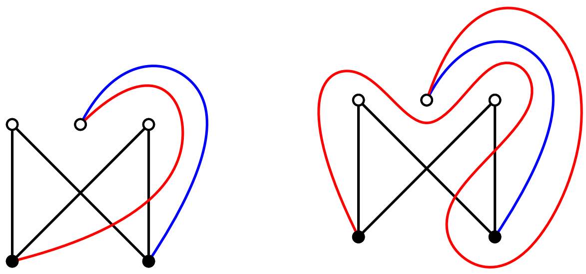
(d) *Case 2D* (e) *Case 3A* (f) *Case 3B*

Figure 2.22: We add a new edge (drawn in red).

Then we draw the last edge. Without loss of generality, we only consider those drawings where that edge has at most as many crossings as the one we inserted before. By the symmetry of the drawings, the other drawings would be isomorphic. The resulting simple drawings can be seen in Figure 2.23. None of those drawings results from Case 3A because we cannot add the last edge without crossing something. This would mean the last edge has more crossings than the one we drew before, which we excluded.



(a) *Drawing 2A.1*; 1 crossing. (b) *Drawing 2B.1*; 3 crossings. (c) *Drawing 2B.2*; 2 crossings.



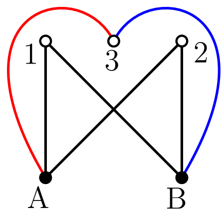
(d) *Drawing 2C.1*; 3 crossings. (e) *Drawing 2D.1*; 3 crossings. (f) *Drawing 3B.1*; 3 crossings.

Figure 2.23: We add a new edge (drawn in blue). The only drawings we consider are the drawings where the blue edge crosses at most as many edges as the red one.

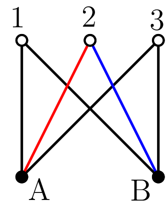
We now label our vertices. We call the vertices of U vertices A and B, and the vertices of V vertices 1,2, and 3. The vertices in V are adjacent to only two vertices and therefore have the same rotation $\{A, B\}$. For the vertices in U there are two possible rotations $\{1, 2, 3\}$ or $\{1, 3, 2\}$. All different labelings of one drawing are isomorphic. We label the vertices so that A has rotation $\{1, 2, 3\}$. There are two possibilities for the complete rotation system:

1. B has the same rotation as A, namely $\{1, 2, 3\}$. These drawings can be seen in Figure 2.24. Their number of crossings are 1 or 3 and therefore all odd.

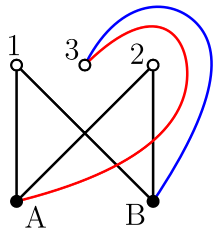
2. B has a different rotation than A, in particular B has rotation $\{1, 3, 2\}$. Those drawings can be seen in Figure 2.25. Their number of crossings are 0 or 2 and therefore all even.



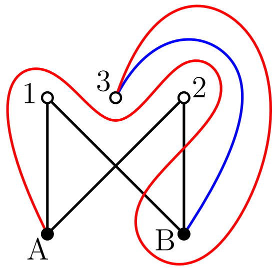
(a) *Drawing 2A.1*;
1 crossing.



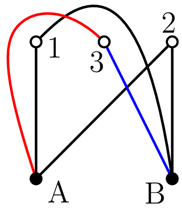
(b) *Drawing 2B.1*;
3 crossings.



(c) *Drawing 2C.1*;
3 crossings.

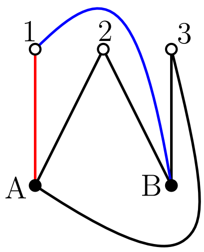


(d) *Drawing 2D.1*;
3 crossings.

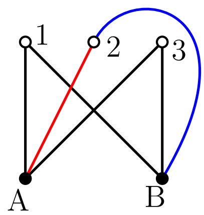


(e) *Drawing 3B.1*;
3 crossings.

Figure 2.24: A and B have the same rotation. There are 1 or 3 crossings.



(a) *Drawing 1*;
0 crossings.



(b) *Drawing 2B.2*;
2 crossings.

Figure 2.25: A and B have different rotations. There are 0 or 2 crossings.

In conclusion, the drawings with the same rotation systems have the same parity of their number of crossings. In particular, the number is odd if all vertices have the same rotation and even otherwise. □

Remark 2.4.4. It can be easily observed that while the parity of the number of crossings of all simple drawings with the same rotation system is the same, the numbers themselves differ.

2.4.2 $K_{3,3}$ is not planar

We can use Theorem 2.4.1 to show a very famous result in graph theory.

Theorem 2.4.5. *There is no plane drawing of the complete bipartite graph $K_{3,3}$.*

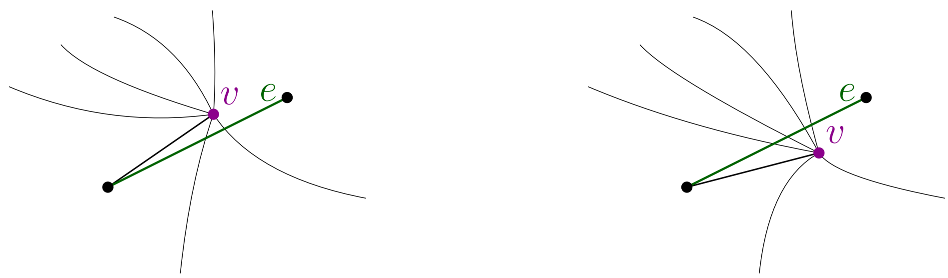
There exist several proofs of this result already. However, Theorem 2.4.1 offers a completely new approach.

Proof. We call the sides of the bipartition U and V and label the vertices in V with 1, 2, and 3 as before. There are only two possible rotation for the vertices in U (namely $\{1, 2, 3\}$ or $\{1, 3, 2\}$). Hence at least two of the vertices in U have the same rotation. From Theorem 2.4.1 it follows that the subdrawing consisting of those two vertices, all vertices in V, and the edges between them, has at least one crossing. Consequently, the drawing of $K_{3,3}$ has at least one crossing. $\square$

2.4.3 $K_{m,n}$ with m, n odd

In simple drawings of $K_{m,n}$ where m and n are fixed and both odd, the parity of the number of crossings is fixed. This result was first used and shown by Kleitman in 1971 when he studied the crossing number of $K_{5,n}$ [11]. Since some found his proof unconvincing, Kleitman published another proof in 1976 [12].

Both proves work with a continuous movement of edges to get from one simple drawing to another. Each time the crossings change there is an even number of changes. In Figure 2.26 it can be seen how such a change looks like. In the first drawing (Figure 2.26a) the vertex v is at "another side" of the edge e than in the second drawing (Figure 2.26b). As a consequence, all edges incident to v that cross e in the left drawing do not cross e in the right drawing and vice versa. The exception is the edge that is incident to v and adjacent to e. It may never cross e in any simple drawing.



(a) The (purple) vertex v is "on the left side" of the (green) edge e.

(b) The (purple) vertex v is "on the right side" of the (green) edge e.

Figure 2.26: The vertex v moves over the edge e, causing the crossings to change.

Let, without loss of generality, v be on the side of the bipartition with m vertices. That means v is incident to n edges. Let a be the number of edges incident to v that crossed e before the movement and do not cross it after the movement. Let b be the number of edges incident to v that did not cross e before and cross it now. Then $a + b + 1 = n$ and therefore $a + b$ is even. Thus, $b - a$ is even. That means that every-time the number of crossings changes there is an even number added or subtracted. Hence, the parity stays the same.

A more formal and complete proof done by Kleitman can be found in [12]. An alternative, more recent proof, can be found in [15]. It was published by McQuillan and Richter in 2010. Their proof is inductive and works mainly with counting arguments.

All mentioned authors call this result the *parity argument*.

Theorem 2.4.6. *[11] The number of crossings of all simple drawings of the complete bipartite graph $K_{m,n}$ with m, n fixed and odd have the same parity.*

2.4.4 Drawings with complete bipartite subdrawings

We have shown that the rotation system of simple drawings of the complete bipartite graphs $K_{2,3}$ contains the information about the parity of the number of crossings. The question arises whether it is sufficient if we have those drawings as subdrawings of the graph. Does the rotation system of simple drawings of graphs that contain $K_{2,3}$ determine the parity of the number of crossings?

In general, the answer is no. Figure 2.27 shows two simple drawings of the same graph with the same rotation system. The complete bipartite graph $K_{2,3}$ is a subgraph of the drawn graph. The drawn graph contains only one edge more than $K_{2,3}$. There are no additional vertices. Still the number of crossings of these simple drawings have different parities. This means that one additional edge is sufficient to loose the information that the rotation system contains for simple drawings of $K_{2,3}$.

We could add another edge like we did in Figure 2.28. The complete bipartite graph $K_{2,3}$ is a subgraph of that graph as well. The drawn graph now has only 2 edges less than the complete graph K_5. Still both drawings have the same rotation system, but their number of crossings have different parities.

Figure 2.27: Two simple drawings of the same graph. The black edges stay the same. The red edges are drawn differently, but the drawings still have the same rotation system. The parity of the number of crossings is different.

Figure 2.28: We add a (green) edge to the drawings of Figure 2.27. We obtain two simple drawings with the same rotation system and a different number of crossings.

Proposition 2.4.7. *In general simple drawings, the rotation system alone does not determine the parity of the number of crossings.*

Chapter 3

Shooting stars in simple drawings of complete bipartite graphs

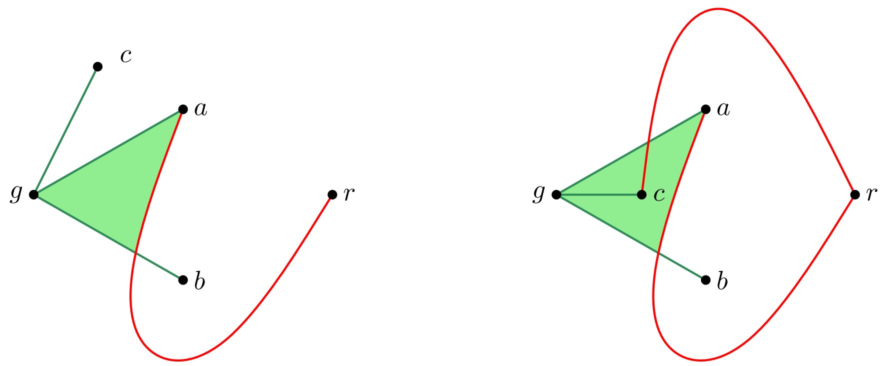
In this chapter we look for plane spanning trees in simple drawings of complete bipartite graphs. First we look at a proof that simple drawings of the complete bipartite graph $K_{2,n}$ contain spanning trees and in particular shooting stars. (Shooting stars are defined in Definition 1.5.2 on page 9.) Next we show that simple drawings of the complete bipartite graph $K_{3,n}$ contain shooting stars as well. We then consider a special kind of simple drawings, called outer drawings. We show that all outer drawings of the complete bipartite graph $K_{m,n}$ contain shooting stars. We present three different proofs for this. Then we consider further special kinds of simple drawings. Those drawings are straight-line drawings, 2-page book drawings, and circular drawings. We show that those drawings contain shooting stars. Finally, we conjecture that shooting stars are also contained in general simple drawings of the complete bipartite graph $K_{m,n}$.

Parts of this chapter has been turned into a paper jointly with O. Aichholzer, M. Scheucher, I. Parada, and B. Vogtenhuber. It has been published in [2].

3.1 Existence of a plane spanning subgraph in $K_{2,n}$

The existence of plane spanning subgraphs in the complete bipartite graph $K_{2,n}$ has been proven by I. Parada and M. Scheucher and given to the author in personal communication. It has also been published in our common paper [2]. For the sake of completeness it is written in this thesis as it is printed in the paper.

We prove that every simple drawing of $K_{2,n}$ contains plane spanning trees of a certain structure. In order to do so, we introduce some notions and provide some auxiliary results.



(a) Case 1 (b) Case 2

Figure 3.1: An illustration of the two cases of base case $k = 3$ from Lemma 3.1.1. The area $\triangle$ is colored light green.

For a given simple drawing of K_n with vertex set V and two fixed vertices $g \neq r \in V$, we define a relation $\rightarrow_{gr}$ on the remaining vertices $V \backslash \{g, r\}$, where $a \rightarrow_{gr} b$ if and only if the arc ra properly crosses gb. In the following, we simply write $a \rightarrow b$ if the two vertices g and r are clear from the context.

Lemma 3.1.1. *The relation $\rightarrow$ is asymmetric and acyclic, that is, there are no vertices $v_1, v_2, \ldots, v_k$ ($k \in \mathbb{N}$) with $v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_k \rightarrow v_1$.*

Proof. We give a proof by induction on k.
Induction basis: The case $k = 1$ is trivial. The case $k = 2$ follows from the fact that there is at most one proper crossing in every 4-tuple in a simple drawing – if ra crosses gb then rb cannot cross ga. For the case $k = 3$ assume there are three vertices a, b, c with $a \rightarrow b \rightarrow c \rightarrow a$. Let $\triangle$ denote the area bounded by the edges ga, gb, ra and not containing the vertex r, as illustrated in Figure 3.1. We distinguish the following two cases:

Case 1: $c \notin \triangle$. Since $c \rightarrow a$ holds, the arc rc crosses ga, and therefore the boundary of $\triangle$. Since $r \notin \triangle$ and since rc cannot cross ra, rc must also cross gb. Thus we have $c \rightarrow b$, which is a contradiction to $b \rightarrow c$.

Case 2: $c \in \triangle$. Since $a \rightarrow b$, the arc rb cannot cross ga. Moreover, since rb can neither cross ra nor gb, it is therefore completely outside of $\triangle$. Since gc is completely contained in $\triangle$, rb and gc cannot cross, and therefore, $b \not\rightarrow c$. Contradiction.

Since c can neither be inside nor outside $\triangle$, the statement is proven.

Induction step: Suppose – towards a contradiction – that there exist $v_1, \ldots, v_k$ with $k \geq 4$ and $v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_k \rightarrow v_1$. We write $a = v_1$, $b = v_2$, $w = v_{k-1}$, and $z = v_k$. Let $\triangle$ denote the area bounded by the edges ga, gb, and ra that does not contain the vertex r. We distinguish the following two cases:

Case 1: $z \notin \triangle$. We continue analogously to Case 1 of base case $k = 2$. Since $z \to a$ holds, rz crosses ga, and therefore the boundary of $\triangle$. Since $r \notin \triangle$ and since rz cannot cross ra, rz must also cross gb. Thus we have $z \to b$.

Case 2: $z \in \triangle$. Since $w \to z$ holds, rw crosses gz at some point inside $\triangle$. Since $r \notin \triangle$ and since rw cannot cross ra, it must cross ga or gb (or both). Thus we have $w \to a$ or $w \to b$.

In both cases, we can find $v'_1, \ldots, v'_l$ for some $l < k$ with $v'_1 \to \ldots \to v'_l \to v'_1$, which is a contradiction. This completes the proof of the lemma. □

Theorem 3.1.2. *Let D be a simple drawing of the complete bipartite graph $K_{2,n}$ with sides of the bipartition $\{g, r\}$ and P. Then, for every $k \in \{1, \ldots, n\}$, D contains a plane spanning tree with k edges incident to g and $n - k + 1$ edges incident to r.*

Proof. According to Lemma 3.1.1, we can find a labeling $v_1, \ldots, v_n$ of the vertices in P such that $v_i \to_{gr} v_j$ only holds if $i < j$. Let S_1 be the star with center g and children $\{v_1, \ldots, v_k\}$ and let S_2 be the star with center r and children $\{v_k, \ldots, v_n\}$. By definition of relation $\to_{gr}$, the edges of S_1 and S_2 do not cross, and hence we have a plane spanning tree.

□

Corollary 1. Let D be a simple drawing of the complete bipartite graph $K_{2,n}$ with sides of the bipartition $\{g, r\}$ and P. Then for each $c \in \{g, r\}$, D contains a shooting star rooted at c.

Proof. Consider again the proof of Theorem 3.1.2. With the according labeling of P, no edge rv_i can cross the edge gv_1. Hence, the plane spanning tree consisting of all the edges incident to r together with the edge gv_1 gives the desired shooting star rooted at r. Similarly, the tree with all edges incident to g and the edge rv_n is a shooting star rooted at g. □

3.2 Remarks to a possible generalization of the proof for $K_{2,n}$ to a proof for $K_{m,n}$

Remark 3.2.1. In $K_{m,n}$ for $m \geq 3$ there might be a spanning subgraph that cannot be obtained by an order as we used it in the proof for $K_{2,n}$. Figure 3.2 shows such

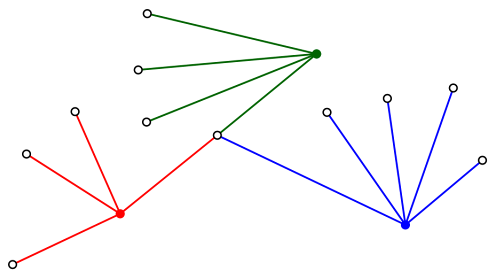


Figure 3.2: A plane spanning subgraph that cannot be found by the ordering.

a subgraph. It is a valid plane spanning subgraph. Since one of the vertices has degree 3 in the subgraph, that subgraph can never be obtained by the ordering.

It follows, that trying to find a spanning tree by extending the order, might not be enough.

Remark 3.2.2. Another possibility to find a plane spanning tree in a drawing of $K_{m,n}$ that we considered is to extend a plane spanning tree we found for $K_{2,n}$. Every simple drawing of $K_{m,n}$ contains simple drawings of $K_{2,n}$ as subdrawings. Those subdrawings contain plane trees that are spanning the simple drawing of $K_{2,n}$. We would only need to add one edge for every vertex of the side of the bipartition that is not yet covered (in $K_{3,n}$ one in total) to obtain a spanning subgraph for the whole graph. However, there are drawings in which this approach can fail even for $m = 3$. An example for that can be found in Figure 3.3. This means it is not always possible to extend a subgraph of the complete bipartite graph $K_{2,n}$ to a plane subgraph of the complete bipartite graph $K_{3,n}$.

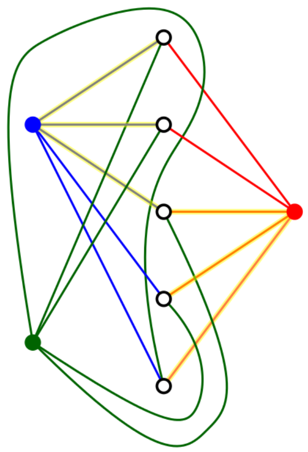


Figure 3.3: The subdrawing consisting of blue and red vertices together with the blue and red edges are a plane drawing of $K_{2,5}$. The yellow marked edges are one possibility for a spanning tree in that subdrawing. Every green edge crosses an edge of that spanning tree.

3.3 Shooting stars in simple drawings of $K_{3,n}$

As shown in the last section, every simple drawing of the complete bipartite graph $K_{2,n}$ contains a shooting star. In this section we proof that every simple drawing of the complete bipartite graph $K_{3,n}$ contains a shooting star as well. The following theorem has been published without proof in [2].

Theorem 3.3.1. *Let D be a simple drawing of the complete bipartite graph $K_{3,n}$ with sides of the bipartition $\{g, r, b\}$ and $P = \{v_1, \ldots, v_n\}$. Then for each $c \in \{g, r, b\}$, D contains a shooting star rooted at c.*

Proof. We color the vertices r, b, and g with colors red, blue, and green, respectively, and all edges of D in the color of the vertex in $\{r, b, g\}$ they are incident to. Then we choose one vertex $c \in \{g, r, b\}$, without loss of generality g, as root of the shooting star ★ that we are constructing. We fix all green edges as part of ★. Note that due to the simplicity of D, those edges are pairwise non-crossing.

From Theorem 3.1.2 it follows that there is a blue edge and a red edge that do not cross any of the green edges. In the rest of this proof, we call these two edges *candidate edges*. If they do not cross each other, we can add them to ★ and hereby complete our shooting star.

Hence assume that the candidate edges cross. Let $v_b \in \{v_1, \ldots, v_n\}$ be the vertex that is incident to the blue candidate edge. We denote with r_2 the edge $v_b r$. Analogously, let $v_r \in \{v_1, \ldots, v_n\}$ be the vertex that is incident to the red candidate edge and let b_2 be the edge $v_r b$. The two edges r_2 and b_2 cannot cross the candidate edges, because they are incident to them. They also cannot cross each other, since in a simple drawing every subdrawing with at most 4 vertices contains at most one crossing (and since the candidate edges cross). Thus, r_2 and b_2 together with the candidate edges form two triangular regions[1] as depicted in Figure 3.4. We denote the region incident to b as $\triangle_b$ and the one incident to r as $\triangle_r$.

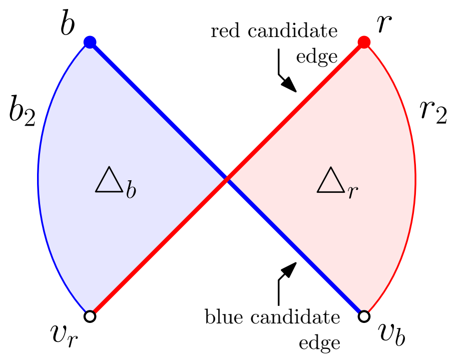


Figure 3.4: The two crossing candidate edges together with the edges r_2 and b_2 form two triangular regions.

If either b_2 or r_2 does not cross any of the green edges, we obtain our shooting star ★ by adding that edge and the candidate edge of the other color.

So assume next that both edges cross at least one green edge. We will prove that in this case there has to be another blue or red edge which does not cross any green edges and does not cross the candidate edges either. Since green edges do not cross the candidate edges and can only cross the other edges at most once, it follows that there has to be at least one vertex in each of the triangles $\triangle_b$ and $\triangle_r$. Further, the green vertex g can lie either inside one of the triangles or outside of both of them. As the case in which g lies in $\triangle_r$ and the case in which g lies in $\triangle_b$ are symmetric, among those we only consider the case in which g lies in $\triangle_b$. Hence, there are two cases to be considered:

Case 1: g lies in $\triangle_b$. In this case, g lies in $\triangle_b$ and there has to be at least one vertex that lies in $\triangle_r$, as shown in Figure 3.5 (left). Actually, all green edges that lie partly inside $\triangle_r$ have one endpoint in $\triangle_r$ (since green edges do not cross the candidate edges and can cross r_2 at most once). We denote this set of vertices inside $\triangle_r$ as V_r, as shown in Figure 3.5 (right).

Claim 3.3.2. *All edges from a vertex in V_r to the red vertex r lie completely inside $\triangle_r$.*

[1]Note that "triangular" here is not meant in a straight-line way, but in the sense that the regions are bounded by a 3-cycle of (parts of) the bounding edges.

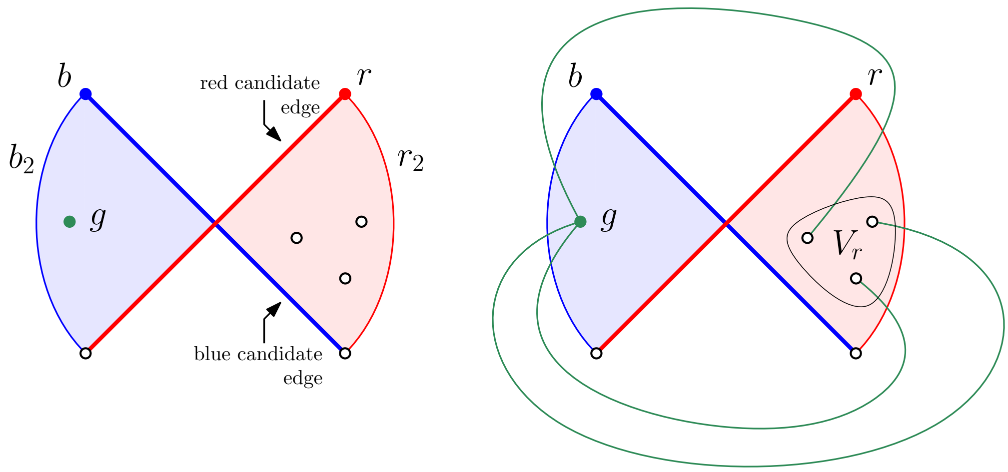


Figure 3.5: Case 1: The green vertex g lies inside a triangle.

Proof of claim: Red edges cannot cross any of the other red edges and may only cross every blue edge at most once. Furthermore, no red edge can cross a green edge adjacent to it. Thus, any red edge that has an endpoint in V_r is forced by the green edge adjacent to it, the candidate edges, and b_2 and r_2 to stay completely inside $\triangle_r$.

That can be seen in Figure 3.6.

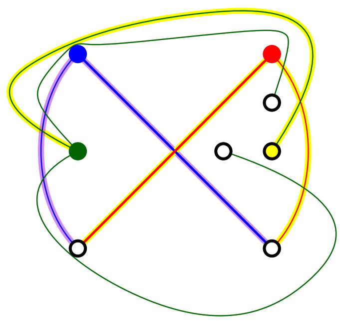


Figure 3.6: The red edge incident to the yellow marked vertex may not cross the yellow marked edges and may only cross the purple marked edges at most once.

■

Claim 3.3.3. *There is a (red) edge from a vertex in V_r to the red vertex r that does not cross any green edges and does not cross the blue candidate edge.*

Proof of claim: Consider the subdrawing D' of D that has V_r, the red vertex r, and the green vertex g as its vertex set and all green and red edges incident to the vertices in V_r as its edge set. Notice that D' is a simple drawing of $K_{2,|V_r|}$. From

Theorem 3.1.2 it follows that there is a red edge in that subdrawing that does not cross any green edge within the subdrawing. That red edge lies completely inside $\triangle_r$, because of Claim 3.3.2. Also, no green edges outside the subdrawing can lie (partly) inside $\triangle_r$. Thus, the red edge we just found does not cross any green edges outside the subdrawing either, and as it lies completely inside $\triangle_r$, it also cannot cross the blue candidate edge. ■ Adding the red edge from Claim 3.3.3 and the blue candidate edge to ★ completes our shooting star for Case 1.

Case 2: g lies outside $\triangle_b$ and $\triangle_r$. In this case, the green vertex g lies outside both triangles, and in both triangles there is at least one vertex, as depicted in Figure 3.7 (left). Again, it holds that all green edges that lie partly inside one of the two triangles have one endpoint in the corresponding triangle.

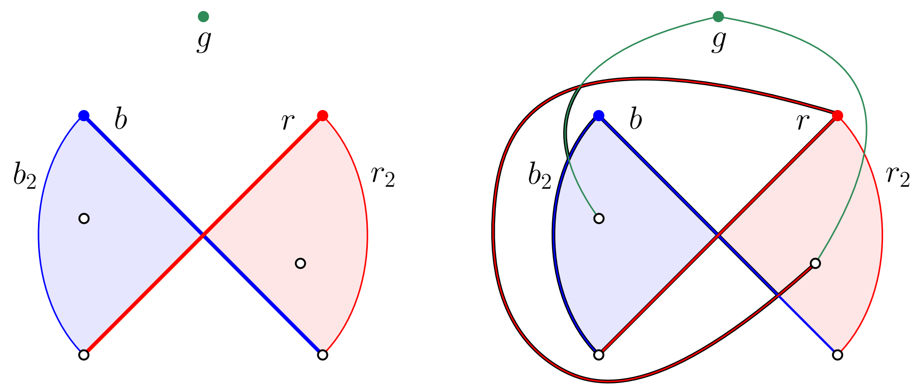


Figure 3.7: Case 2: The green vertex g lies outside both triangles. Right: case in which a red edge from a vertex in $\triangle_r$ to the r does not stay completely inside $\triangle_r$.

In drawings in which Claim 3.3.2 also holds, we can obtain a plane spanning tree as we do in Case 1. However, it does not hold in general. It is possible that a red edge incident to a vertex in V_r lies only partly inside $\triangle_r$ and crosses the blue candidate edge. That can bee seen in Figure 3.7 (right) and Figure 3.8.

Denote with V_r the set of vertices inside $\triangle_r$ and with V_b the set of vertices inside $\triangle_b$. If all red edges incident to a vertex of V_r lie completely in $\triangle_r$, we can complete our shooting star ★ in the same way as in Case 1. Hence assume that this is not the case. We will show that all blue edges lie completely inside $\triangle_b$. Once we have proven that, we can apply Theorem 3.1.2 to the simple drawing of $K_{2,|V_b|}$ formed by g, b, V_b, and their incident edges, by this obtaining our shooting star ★ with similar arguments as in Case 1 (just with swapped colors).

Note that all red edges incident to a vertex of V_r that do not lie completely inside $\triangle_r$ connect to r from outside $\triangle_r$, avoid $\triangle_b$, and intersect all edges from g to a vertex in V_b as well as the edge b; see Figure 3.7 (right). Hence every blue edge

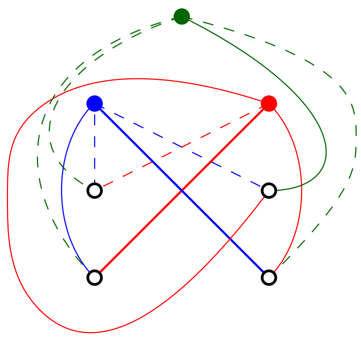


Figure 3.8: In Case 2, there might be a red edge lying only partly inside $\triangle_r$.

incident to a vertex in $\triangle_b$ is forced to stay completely inside $\triangle_b$ by (i) the green edge adjacent to it, (ii) the blue edges from the triangle, (iii) the red candidate edge, and (iv) a red edge that goes from a vertex of V_r to the red vertex r and does not stay completely inside $\triangle_r$. The only edges out of the listed ones that the blue edge may cross are the red ones. But if the blue edge crossed one of those red edges, it would have to cross one of them at least twice (in order to still be an edge from inside $\triangle_b$ to the blue vertex), and that is not allowed in a simple drawing. □

Corollary 2. Let D be a simple drawing of the complete bipartite graph $K_{3,n}$ with sides of the bipartition $\{g, r, b\}$ and P, and let e be an edge of D. Then D contains a shooting star containing e.

Proof. Every edge in the complete bipartite graph $K_{3,n}$ is incident to a vertex $c \in \{g, r, b\}$. By Theorem 3.3.1 there is a shooting star containing all edges incident to that vertex c. □

3.4 Plane spanning subgraphs in outer drawings

In this section we study the problem of finding plane spanning trees in a special kind of simple drawings of bipartite graphs, namely outer drawings. The term outer drawings for complete bipartite graphs was introduced in [3]. They are defined as follows:

Definition 3.4.1. [3] A simple drawing of the complete bipartite graph $K_{m,n}$ in which all the m vertices of one side of the bipartition lie on the outer boundary of the drawing is called *outer drawing*.

We denote by P the side of the bipartition whose (*perimeter*) vertices must lie on the outer boundary of the drawing. The other side of the bipartition is denoted by S (as it contains the *surrounded* vertices).

Note that points of S may also lie on the outer boundary but are not required to.

Theorem 3.4.2. *Let D be an outer drawing of the complete bipartite graph $K_{m,n}$ with sides of the bipartition P and S (where the vertices of P lie on the outer boundary). Let p be an arbitrary vertex in P. Then D contains a shooting star rooted at p.*

In this section we show three different proofs for Theorem 3.4.2. Proof 1 is the proof we could show first. It proofs a statement that is slightly stronger than Theorem 3.4.2. The other two proofs only show Theorem 3.4.2, but are more compact than Proof 1. The last proof we found, Proof 3, is more elegant than the other proofs.

3.4.1 Proof 1

In our first proof of Theorem 3.4.2 we revisit the proof of Theorem 3.3.1. We simplify it for outer drawings and derive a stronger theorem for them. By means of this stronger result for outer drawings of $K_{3,n}$ we proof the general statement for outer drawings of $K_{m,n}$.

Theorem 3.3.1 revisited for outer drawings of $K_{m,n}$

As outer drawings are simple drawings, Theorem 3.3.1 holds. In the case of outer drawings the proof for the original statement can be simplified. Moreover, a slightly stronger statement holds.

Theorem 3.4.3. *Let D be an outer drawing of the complete bipartite graph $K_{3,n}$ with sides of the bipartition $\{g, r, b\}$ and S, where the vertices of $\{g, r, b\}$ lie on the outer boundary. Then for each $c \in \{g, r, b\}$, D contains a plane subgraph that contains all edges incident to c. Moreover, for every edge e that does not cross any edges incident to c, there is a shooting star that contains the edge e and all edges incident to c.*

Proof. We will not repeat the proof of Theorem 3.3.1 in detail. Instead we point out what new properties hold, what can be concluded from that, and which points of the original proof can be omitted.

Let us color D as before, and again fix the green edges as the ones we want to have in the spanning tree. Let us further fix the blue candidate edge as the edge we want to have additionally in the spanning tree.

As before, if the two candidate edges do not cross, we can use both of them together with the green edges, and we have the desired plane shooting star.

If the two candidate edges cross, we need to find a red edge that crosses neither the blue candidate edge, nor any green candidate edges. To achieve that, we consider $\triangle_r$. If there is no vertex in $\triangle_r$, we can use the red edge incident to the blue candidate edge. The proof that in that case it cannot be crossed by any green edge is the same as in the proof of Theorem 3.3.1.

If there is at least one vertex in $\triangle_r$, then there is at least one edge completely inside $\triangle_r$ that does not cross any green edges and does not cross the blue candidate edge. We will show that by using special properties of outer drawings.

The green vertex has to lie on the outer boundary. In particular, it cannot lie inside $\triangle_r$ or $\triangle_b$. Thus, we are always in Case 2 of the proof for Theorem 3.3.1. As before all green edges that lie partly inside the triangle, have to have an endpoint there.

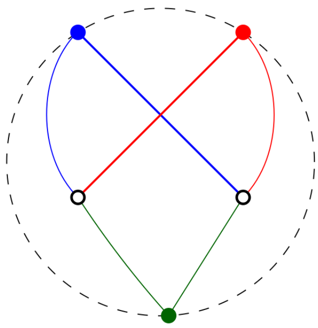


Figure 3.9: The green vertex has to lie on the outer boundary, that is drawn as dashed circle.

In outer drawings every red edge incident to a vertex in $\triangle_r$ has to lie completely within $\triangle_r$. That is because any such edge may not cross the green edge incident to it, the red candidate edge, the red edge incident to the blue candidate edge, and the outer boundary. As marked in Figure 3.10 those lines together with the blue candidate edge, that it may only cross once, force the red edge to stay completely within $\triangle_r$.

As before, we can conclude that there has to be a red edge which crosses no green edge and does not cross the blue candidate edge. (We again do so by using Corollary 1 on the complete bipartite graph consisting of the vertices inside $\triangle_r$, the red and green edges incident to them, and the red and green vertex.) Thereby we have found a plane spanning subgraph using all green edges and the blue candidate edge.

□

Remark 3.4.4. Theorem 3.4.3 does not hold for general simple drawings.

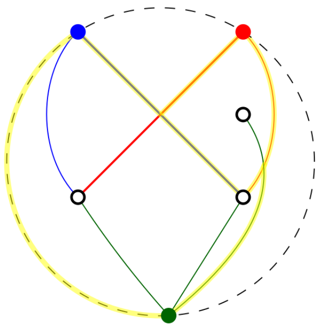


Figure 3.10: Any red edge incident to a vertex in $\triangle_r$ is forced by the yellow (parts of) edges and the yellow part of the outer boundary to stay completely inside $\triangle_r$.

Proof. In Figure 3.11 no green edge crosses the marked red edge. However, there cannot be any plane spanning subgraph that contains that red edge and all green edges, because all blue edges cross at least one of them. □

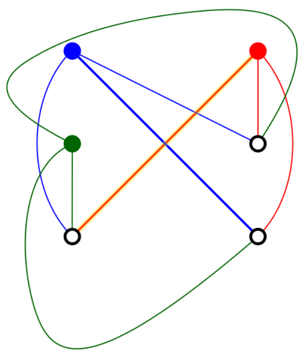


Figure 3.11: Theorem 3.4.3 does not hold in this simple drawing.

Shooting stars in outer drawings of $K_{m,n}$

Using Theorem 3.4.3 we can now proof that every outer drawing of the complete bipartite graph $K_{m,n}$ contains a shooting star. First, we introduce an idea of how we can find such a drawing. Then we proof that this construction does indeed work.

We proof a generalization of Theorem 3.4.3, making our new Theorem slightly stronger than Theorem 3.4.2.

Theorem 3.4.5. *Let D be an outer drawing of the complete bipartite graph $K_{m,n}$ with sides of the bipartition P and S (where the vertices of P lie on the outer boundary). Let p be an arbitrary vertex in P. Then D contains a shooting star* *rooted at p. Moreover, if H is a plane subgraph of D that contains all edges incident to v and is a tree, then H can be extended to a plane spanning tree of G.*

With Theorem 3.4.5 we can construct a shooting star.

Let D be a given outer drawing of $K_{m,n}$. Color all vertices of P with colors c_1, $c_2 \dots c_m$. As before, color the edges incident to them with the same color.

Start with a subgraph H of D that is a tree which contains all edges of color c_1. (There is always at least one such subgraph, namely the one that contains exactly all edges of color c_1.)

Then consider the colors c_2 to c_m successively. In the step where c_i is considered, add an edge to H that has color c_i and does not cross any edges already in H. If there are several such edges, choose an arbitrary one, and add it to H.

What needs to be shown is, that this edge of color c_i that does not cross any edge already in H truly exists.

Lemma 3.4.6. *Let D be an outer drawing of the complete bipartite graph $K_{m,n}$ with sides of the bipartition P and S (where the vertices of P lie on the outer boundary). Let p be an arbitrary vertex in P. If every subdrawing of D that is a simple drawing of $K_{m',n}$ with $m' \in \mathbb{N}$, $m' < m$ and contains p has a plane spanning tree that contains all edges incident to p, then any such tree can be extended to a plane spanning tree of D.*

Proof. As described before we color p with color c_1 and the other vertices of P with colors c_2, c_3, $\dots$, c_m.

We proof the theorem by induction on m.

Induction basis: For $m = 1$ all edges are of color c_1. They do not cross because it is a simple drawing. For $m = 2$ Lemma 3.4.6 follows from Corollary 1, and for $m = 3$ it follows from Theorem 3.4.3.

Induction hypothesis: Assume Lemma 3.4.6 is true for $K_{l,n}$ with $3 \leq l \leq m - 1$.

Induction step: Let H be a plane tree that is a subgraph of D that contains all edges of color c_1. If H has an edge of every other color, it already is a plane spanning tree. So assume H does not contain any edge of (at least) one color, say c_m.

Let D_m be the subgraph of D that contains all edges and vertices of D but the ones of color c_m.The graph D_m is $K_{m-1,n}$, and all its subgraphs are also subgraphs of D. Thus, D_m fulfills the spanning subgraph requirement of Lemma 3.4.6. The graph H is a subgraph of D_m. Hence, the induction hypothesis can be applied. It yields a plane spanning tree of D_m that contains all the edges of H. We call this tree H_1. Since H_1 contains all edges of color c_1 and is a tree, it can only contain at most one edge of any other color. Otherwise there would be a cycle (see

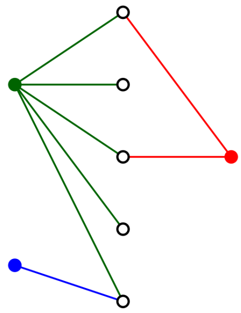


Figure 3.12: If all edges incident to one vertex and at least two edges of another vertex on the other side of the bipartition are contained in the subgraph, then there is a cycle.

Figure 3.12). If we can find an edge of color c_m in D that does not cross any edge in H_1, we have the desired spanning tree of D.

Let H_2 be the subgraph of H_1 that contains all its edges but the one of color c_{m-1}. Let D_{m-1} be the subgraph of D that contains all edges and vertices of D but the ones of color c_{m-1}.The graph D_2 is a subgraph of D_{m-1}. Applying the hypothesis on H_2 yields a plane spanning tree of D_{m-1} that contains all edges of H_2. Call it H_3.

Consider now the union of H_3 and H_1. Since both are plane and contain H_2, the edges in H_2 do not cross anything in the union. That means that the edges of color c_1 and the candidate edges of color c_2 to c_{m-2} cross no edge in the union. The only possible crossing is between the candidate edge of color c_{m-1} and the one of color c_m.

As stated before, we wish to keep the edges of H_1 and add another edge of color c_m. If the candidate edge of color c_m, that we just found, does not cross the candidate edge of color c_{m-1}, we are done. Otherwise we proceed similar to the proof of Theorem 3.3.1, to find another edge of color c_m that we can add.

Assume that the candidate edges of color c_m and c_{m-1} cross. Together with the edges of color c_m and c_{m-1} that are incident to them, they form two triangles. We call the triangle that has the vertex of color c_m on its boundary $\triangle_{c_m}$, as we did with 3 colors.

Observe that what was true for green edges in the proof of Theorem 3.3.1, is now true for the edges of color c_1 and for all candidate edges of colors c_2 to c_{m-2}. That is, they are not allowed to cross the candidate edges, they are only allowed to cross other edges at most once, they cannot go over the outer boundary. Hence, such edges can only cross the edge of color c_m that is incident to the candidate edge of color c_{m-1} if they have an endpoint in $\triangle_{c_m}$.

- If $\triangle_{c_m}$ does not have any vertices inside, then the edge of color c_m that is incident to the candidate edge of color c_{m-1} is not crossed by any candidate edges or any green edges. Consequently, H_1 together with that edge is a plane spanning tree.

- All green edges that lie partly inside $\triangle_{c_m}$, and all candidate edges that lie partly inside $\triangle_{c_m}$ have to have an endpoint there.

All vertices inside $\triangle_{c_m}$ have to have edges to the vertex of color c_m.

Those edges of color c_m have to stay completely inside $\triangle_{c_m}$. Like in Theorem 3.4.3 and Figure 3.10 (with c_m is red, c_{m-1} is blue, and c_1 is green), they are bounded by the edges that form $\triangle_{c_m}$, the c_1 edge they are incident with, and the outer boundary.

Consider now the subgraph F that has all the vertices inside $\triangle_{c_m}$, the vertex of color c_1, the vertex of color c_2, and all the vertices whose candidate edges lie partly inside $\triangle_{c_m}$ as vertex set, and all edges between those vertices as edge set (as seen in Figure 3.13). It is a complete bipartite graph. The vertex of color c_{m-1} is not contained in the subgraph, because its candidate edge is a boundary of the triangle. Hence, F is a simple drawing of $K_{l,n}$ with $l \leq m-1$. Let H_F be the graph

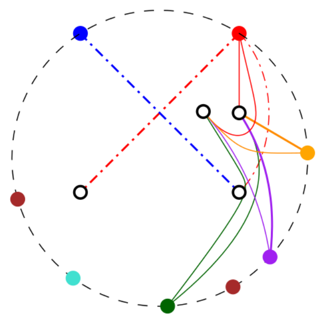


Figure 3.13: Here the color c_m is red, c_{m-1} is blue, and c_1 is green. We consider the complete bipartite graph consisting of all vertices in $\triangle_{c_m}$ (the red triangle), the vertex of color c_1 (green), and the vertices whose candidate edges are inside the triangle (purple and orange).

that contains all edges and vertices in the intersection of H_1 and F. In particular, it contains all edges of color c_1 in F and all candidate edges of H_1 that lie in F. Those candidate edges do not cross each other (because the candidate edges of colors c_{m-1} and c_m are not in F). Thus, H_F is a plane subgraph of F which fulfills all the requirements of Lemma 3.4.6.

We can use the hypothesis to obtain a plane spanning subgraph that contains all the edges of color c_1 incident to vertices inside the triangle, all the candidate edges incident to vertices inside the triangle, and an edge of color c_m that does not cross any of them. That edge also crosses no other edge of color c_1 and no other candidate edge, because it lies completely inside the triangle. Thus, that edge of color c_m together with H_1 is a plane spanning tree of D.The graph H is a subgraph of that spanning tree, as H is a subgraph of H_1. □

Theorem 3.4.5 follows immediately from Lemma 3.4.6.

Proof. We proof Theorem 3.4.5 via induction on m, using Corollary 1 and Theorem 3.4.3 as base and Lemma 3.4.6 as induction step.

Induction basis: For simple drawings of $K_{1,n}$ the edges incident to p (which are all edges) are the spanning subgraph. For $K_{2,n}$ and $K_{3,n}$ the Theorem follows immediately from Corollary 1 and Theorem 3.4.3.

Induction hypothesis: Theorem 3.4.5 is true for outer drawings of any $K_{l,n}$ with $l \leq m - 1$.

Induction step: Let D be an outer drawing of $K_{m,n}$. It follows from the induction hypothesis, that D fulfills the requirements of Lemma 3.4.6. Hence, any tree that is a plane spanning tree and contains all edges incident to p can be extended to a plane spanning tree of D. The subgraph that has the vertex p and all the vertices of the other side of the bipartition as vertex set, and all the edges incident to p as edge set, is such a tree. Thus, it can be extended and therefore a plane spanning tree exists. □

Remark 3.4.7. Theorem 3.4.5 and Lemma 3.4.6 do not hold in general simple drawings.

Proof. Figure 3.11 from Remark 3.4.4 is not an outer drawing, but fulfills all other requirements of Theorem 3.4.5 and Lemma 3.4.6. The subgraph that consist of all green edges and the marked red edge is a plane subgraph that is a tree and cannot be extended to a plane spanning subgraph of D. □

3.4.2 Proof 2

When revisiting the first proof and having a careful look at Figure 3.9 we can find another proof for Theorem 3.4.5.

Proof. We follow the proof of Lemma 3.4.6 until we consider the crossing of candidate edges c_m and c_{m-1}. As stated before, we are finished if they do not cross. So let us consider the crossing.

If the two edges cross, then the part of the candidate edge of color c_m that goes from the vertex of color c_m to the crossing, and the part of the candidate edge of color c_{m-1} that goes from that crossing to the vertex of color c_{m-1} split the area bounded by the outer boundary into two areas. The vertices of S that are incident to the candidate edges of color c_m and c_{m-1} lie in one of those areas. We call the other area the *outer triangle*. It can be seen in Figure 3.14.

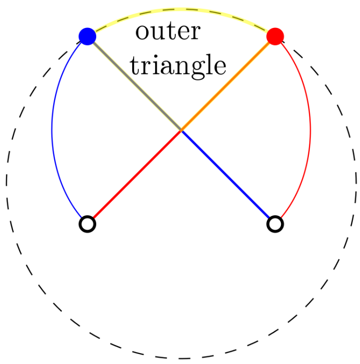


Figure 3.14: The candidate edges together with the outer boundary form a triangle whose bounds are marked in yellow.

Claim 3.4.8. *Either the vertex p and all the vertices of S lie inside the outer triangle or none of those vertices lie inside the outer triangle.*

Proof of claim: Any edge that lies partly inside the outer triangle has to either lie completely inside it or cross a candidate edge of color c_m or c_{m-1}. The edges of color c_1 cannot cross any candidate edges.

Thus either p and all vertices incident to edges of color c_1 lie inside the outer triangle or all of them lie outside the outer triangle. As D is complete bipartite, all vertices in S are adjacent to p and therefore incident to edges of color c_1. Since the vertices of S that are incident to the candidate edges of color c_m and c_{m-1} do not lie in the outer triangle, it follows that neither p nor the vertices of S can lie inside the outer triangle.

The vertices of P that are not p have candidate edges that do not cross the candidate edge of color c_m or c_{m-1}. Therefore, the candidate edges either lie completely inside the triangle or completely outside of the triangle. Since there is no vertex of S inside the triangle, they cannot lie completely inside. Thus, they have to lie completely outside. Hence, the vertices of P have to lie outside the outer triangle as well. ■

Assume that the candidate edges of colors c_{m-1} and c_m cross. It follows from Claim 3.4.8 that they form an outer triangle that is empty. That means that there is one direction in which we can follow the outer boundary from c_{m-1} to c_m and encounter no vertex.

We look at the root of our shooting star, vertex p (which is colored in c_1). Starting in p we follow the outer boundary clockwise. We colour the first vertex in P that we encounter with color c_{m-1}. We again start in p and follow the outer boundary counter-clockwise. We colour the first vertex in P that we encounter with color c_m. The side of the bipartition P contains at least four vertices. (The cases in which P contains less vertices are the induction basis.) This together with our choice of coloring implies that we cannot follow the outer boundary from c_{m-1} to c_m in any direction without encountering vertices in P. (In particular, we encounter p if we go counter-clockwise and all other vertices of P if we go clockwise.) Hence, the candidate edges of colors c_m and c_{m-1} cannot cross. □

3.4.3 Proof 3

This final proof works inductively as well. But it constructs the shooting star in a smarter way, leading to a shorter and more elegant proof. This proof has also been published in [2].

Proof. First, we label the vertices in P. We start in $p_1 := p$ and go clockwise along the outer boundary and denote the vertices of P by p_2 to p_m following the order in which they occur. Let T_1 be the subgraph that is induced by all edges incident to p_1. Notice that T_1 is a plane tree. We will add edges to T_1 until it becomes a spanning tree. We do so inductively by first adding an edge incident to p_2, then an edge incident to p_3 and so on until we add an edge incident to the vertex p_m. We denote by T_i the tree that we get by adding to T_{i-1} the selected edge incident to p_i for $2 \leq i \leq m$. We will show that it is possible to add edges such that T_i is always plane. After adding the last edge the statement then follows.

In the first step, for T_2, we need to find an edge that is incident to p_2 and does not cross any edge incident to p. We know from Theorem 3.1.2 that there is at least one such edge. We add that edge to T_1 and get a plane tree T_2. For T_i we need to find an edge that is incident to p_i and does not cross any of the edges of T_{i-1}. We denote by e_{i-1} the edge in T_{i-1} that is incident to p_{i-1} and by s_{i-1} the vertex in S that e_{i-1} is incident to. We also denote the edge that is incident to s_{i-1} and p by e_p. See Figure 3.15 for an illustration. The part of the boundary that goes from p clockwise until p_{i-1} together with the edges e_{i-1} and e_p encloses a region that we call R_1. The vertices p_2 to p_{i-1} all lie on that part of the boundary, because of the way we labeled them. We call the rest of the area inside the outer boundary R_2.

Claim 3.4.9. *All edges in T_{i-1} that are not incident to p lie completely inside R_1.*

<u>Proof of claim:</u> Since the boundary of R_1 consists of edges in T_{i-1} and the outer boundary, all edges of T_{i-1} that lie partly inside R_2 have to lie completely inside it. The edges in T_{i-1} that are not incident to p are incident with the vertices p_2 to p_{i-1}. As they have to lie on the part of the outer boundary that is also part of the boundary of R_1, the edges incident to these vertices have to lie partly inside R_1. Thus these edges have to lie completely inside R_1. ■

Let us now consider the region R_2. The subdrawing induced by p, p_i, and all vertices of S that lie in R_2 is a simple drawing of $K_{2,n'}$ with $n' \in \mathbb{N}$. By

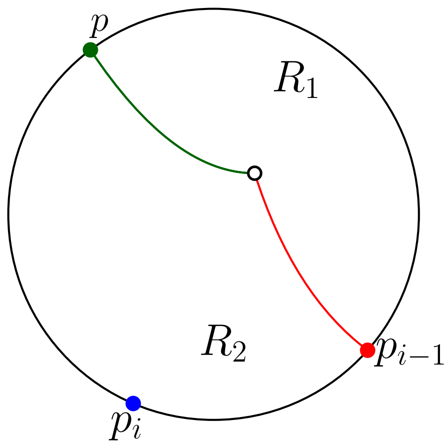


Figure 3.15: The edges e_p and e_{i-1} together with the outer boundary form two regions.

Theorem 3.1.2 there is an edge incident to p_i that does not cross any edges incident to p. This edge can neither cross the outer boundary nor e_p and it can only cross e_{i-1} once. Since the edge has both endpoints in R_2, it follows that the edge has to lie completely in R_2. From Claim 3.4.9 it follows that it does not cross any of the edges of T_{i-1} that are not incident with p. As it doesn't cross any edges incident with p either, it follows that it doesn't cross any of the edges of T_{i-1}. Thus, we can add that edge and obtain a plane tree T_i. We continue to do so until we added an edge for every vertex in P. The plane spanning tree T_m is a shooting star. □

3.5 Other drawings

3.5.1 Straight-line drawings

Theorem 3.5.1. *Let D be a straight-line drawing of the complete bipartite graph $K_{m,n}$ and let v be an arbitrary vertex. Then D contains a shooting star rooted at v.*

Proof. If one side of the bipartition has cardinality 2 , the theorem follows from Corollary 1. So we assume that both sides of the bipartition contain at least three vertices. Let V be the side of the bipartition that contains v and let U be the other side of the bipartition.

We extend the edges incident to v to infinity such that v is still an endpoint of the extended lines. Those lines part the plane into sections. We split those sections into two using the angle bisectors between two edges that are following each other in the rotation around v. (An angle bisector divides the angle into two angles with equal measures.) This construction can be seen in Figure 3.16. The new sections are bounded by a bisector on one side and the extension of an edge on the other side. All sections have exactly one vertex of U on their boundary. By construction the angle around v of any section has an angle of degree at most 180.

For our shooting star, we add all edges incident to v. We need to connect the vertices in V. Let w be a vertex in V. It can lie inside one section, or on the boundary of two section. Let S be a section that w lies inside of or on the boundary of. We add the edge connecting v to the vertex on the boundary of S.

As the drawing uses only straight lines, these edges lie within one section and therefore cannot cross any edges outside its section. By our construction all edges within one section are connected with the same vertex. Thus they cannot cross any other edges within the same section. Therefore, we have obtained a shooting star.

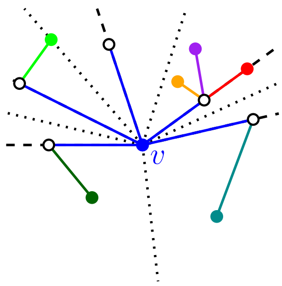


Figure 3.16: A shooting star in a straigt-line drawing of $K_{7,5}$, and our construction to find the shooting star. The vertices in U are black, empty circles. The vertex v is a filled blue circle. The other vertices in V are circles that are filled in other colors. The edges are colored in the same color as the vertex in V they are incident with. The section bounds obtained by extending edges are drawn in dashed, black lines. The section bounds obtained by angle bisectors are drawn in dotted, black lines.

□

3.5.2 2-page book drawings

In this subsection we look at a special type of simple drawing called 2-page book drawing.

Definition 3.5.2. [1] A simple drawing where all vertices lie on a line, called the spine, and every edge lies completely on one side of the spine is called *2-page book drawing.*

It is possible that different edges lie on different sides of the spine, but no edge may lie on both sides and thus no edge can cross the spine.

Theorem 3.5.3. *Let D be a 2-page book drawing of the complete bipartite graph $K_{m,n}$ and let v be an arbitrary vertex. Then D contains a shooting star rooted at v.*

Proof. We denote the side of the bipartation that contains v as V and the other side of the bipartition U. Again we start by adding all edges incident to v. We then have to add edges for the other vertices in V. Let w be a vertex in V. There are three possibilities where w can lie:

1. *Case 1*: There are vertices in U between v and w. Let u be the closest of those vertices with respect to w. Closest here means that there are no other vertices in U between u and w. Then we add the edge joining w and u. The yellow, and the purple vertex in Figure 3.17 are examples for that case.

2. *Case 2*: There are no vertices in U between w and v, but there are vertices in U on the same side of v as w. Then we add an edge joining w and the closest of those vertices with respect to w. The red, and the blue vertex in Figure 3.17 are examples for that. (If w lies right of v, that means we look at the vertices in U that lie right of v as well. Then we choose the leftmost of those vertices.)

3. *Case 3*: There are no vertices in U between w and v, and no vertices in U lie on the same side of v as w. Then we add the edge joining w and the vertex in U that is farthest away from w. The gray, and the brown vertex of Figure 3.17 are examples for that. (If w lies right of v that would mean we add the edge joining w and the leftmost vertex in U.)

We proof that this construction works by means of an observation:

Claim 3.5.4. *In a 2-page book drawing two edges can only cross if one endpoint of one of the edges lies in between the endpoints of the other edge and the other endpoint does not.*

Proof of claim: Let a and b be adjacent vertices joined by an edge e_1. Let e_2 be an edge that crosses e_1. We denote by A the area that is enclosed by e_1 and the spine. In order for the edges to cross, e_2 has to lie partly in A. As the drawing is simple, e_1 and e_2 may only cross once. By the definition of 2-page book drawings, the spine may not be crossed. Thus one vertex incident to e_2 has to lie inside A and the other vertex has to lie outside A. ■

In Case 1 and Case 2 only vertices in V can lie between w and the vertex it is connected to in the shooting star. By our construction these vertices are connected to the same vertex as w and in a simple drawing such edges cannot cross.

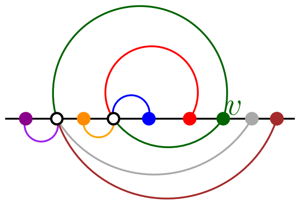


Figure 3.17: The shooting star rooted at the (green) vertex v that is contained of a 2-page book drawing.

In Case 3 the only vertices that do not lie between w and the vertex it is connected to, are vertices in V. By our construction these vertices are connected to the same vertex as w and in a simple drawing such edges cannot cross.

□

3.5.3 Circular drawings

Finally, we look at a special kind of simple drawing that is very closely related to outer drawings.

Definition 3.5.5. A simple drawing of the complete bipartite graph $K_{m,n}$, in which the m vertices of one side of the bipartition lie on a closed curve that is not crossed by any edges, the vertices of the other side of the bipartition do not lie on the curve is called a *circular drawing*.

We denote by P the side of the bipartition whose (*perimeter*) vertices must lie on the closed curve.

Theorem 3.5.6. *Let D be a circular drawing of the complete bipartite graph $K_{m,n}$, and let v be a vertex in P. Then D contains a shooting star rooted at v.*

Proof. We can construct a spanning tree as we did with outer drawings. We add all edges incident to v. The only vertices that are not yet connected with the tree have to lie on the curve. We connect them in the same way as we connected vertices on the boundary of the outer drawings. An example for such a spanning tree is Figure 3.18. □

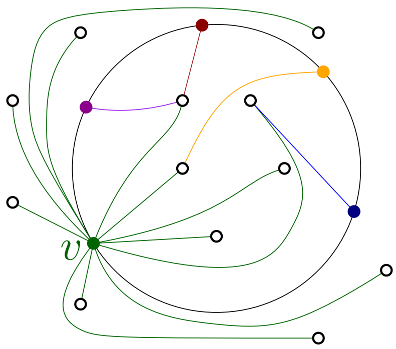


Figure 3.18: The shooting star rooted at the (green) vertex v that is contained of a circular drawing.

3.6 Plane spanning trees in $K_{m,n}$

The question of whether every simple drawing of a completely bipartite graph contains a plane spanning tree remains open. It is not even clear for simple drawings of $K_{4,n}$.

However, as it is true for all special drawings we looked at as well as for simple drawings of $K_{2,n}$ and $K_{3,n}$ it seems plausible that all simple drawings of $K_{m,n}$ contain plane spanning trees. In fact, after working with the drawings and trying long and hard to find a counter example, we strongly believe that this is the case.

Conjecture 3.6.1. *Let D be a simple drawing of the complete bipartite graph $K_{m,n}$. Then D contains a plane spanning tree.*

We also think it is highly possible that all simple drawings of $K_{m,n}$ contain shooting stars.

Conjecture 3.6.2. *Let D be a simple drawing of the complete bipartite graph $K_{m,n}$. Then D contains a shooting star.*

It might even be, that an arbitrary vertex can be chosen as root, and there is a shooting star rooted at that vertex.

Conjecture 3.6.3. *Let D be a simple drawing of the complete bipartite graph $K_{m,n}$, and let v be a vertex of D. Then D contains a shooting star rooted at v.*

Note that Conjecture 3.6.3 would include Conjecture 3.6.2, which would include Conjecture 3.6.1.

Chapter 4

Conclusion

We considered questions about simple drawings beyond the complete graph. A particular focus was on the complete bipartite graph. We were interested in crossings, number of crossings, rotation systems, and plane spanning trees.

Pach and Tòth showed that rotation systems of simple drawings of complete graphs determine which pairs of edges cross [17]. We started by studying the question: What can the rotation system tell us about simple drawings beyond the complete graph? We presented our research on rotation systems in Chapter 2.

We showed that the rotation system of simple drawings of graphs with only one edge less than compete graphs (that means $\binom{n}{2} - 1$ edges) determines the number of crossings. Then we proved that in graphs with $n \geq 5$ vertices and a minimal degree of at least $(n-2)$ the rotation system determines the number of crossings as well. Additionally, we showed examples of a graph with minimal degree of $(n-3)$, and a graph with only 4 vertices (but minimal degree of 2), where this no longer holds.

Then we turned to complete bipartite graphs. We showed that the rotation system of simple drawings of the complete bipartite graph $K_{2,3}$ determines the parity of the number of crossings. We looked at the question which rotation system leads to an even number of crossings and which to an odd. Then we used that result to present a proof that drawings of $K_{3,3}$ cannot be plane. We recalled that if n and m are fixed and both odd, all simple drawings of the complete bipartite graph $K_{m,n}$ have the same parity of number of crossings. Then we showed that containing the complete bipartite graph $K_{2,3}$ as subgraph is not sufficient for the rotation system to carry the information about the parity. We showed that even graphs with five vertices that contain the complete bipartite graph $K_{2,3}$ and have only two or three edges less than K_5 can have simple drawings which have the same rotation system, but different parities of their number of crossings.

Our research on crossings in complete bipartite graphs led us to wish for a plane structure that we can use to divide the drawings and to start drawing them.

This motivated the question we studied in Chapter 3: Do all simple drawings of complete bipartite graphs contain plane spanning trees?

We answered that question to the affirmative for some types of simple drawings of complete bipartite graphs. We proved that simple drawings of the complete bipartite graphs $K_{2,n}$ and $K_{3,n}$ contain shooting stars. Furthermore, we showed for special types of simple drawings that they contain shooting stars. Those simple drawings were outer drawings, straight-line drawings, 2-page book drawings, and circular drawings of the complete bipartite graph $K_{m,n}$.

We conjectured that every simple drawing of the complete bipartite graph contains a shooting star. However, it is still an open problem whether every simple drawing of the complete bipartite graph contains even a plane spanning tree.

List of Figures

Bibliography

[1] Bernardo M. Ábrego, Oswin Aichholzer, Silvia Fernández-Merchant, Pedro Ramos, and Gelasio Salazar. The 2-page crossing number of K_n. In *Discrete & Computational Geometry*, volume 49/4, pages 747–777, 2013.

[2] Oswin Aichholzer, Irene Parada, Manfred Scheucher, Birgit Vogtenhuber, and Alexandra Weinberger. Shooting stars in simple drawings of $K_{m,n}$. In *Proceedings of the 34th European Workshop on Computational Geometry (EuroCG19)*, pages 59:1–59:6, 2019.

[3] Jean Cardinal and Stefan Felsner. Topological Drawings of Complete Bipartite Graphs. In *Proceedings of the 24th International Symposium on Graph Drawing and Network Visualization (GD 2016)*, 2016.

[4] Radoslav Fulek and Andres J. Ruiz-Vargas. Topological graphs: empty triangles and disjoint matchings. In *Proceedings of the 29th Annual Symposium on Computational Geometry (SoCG'13)*, pages 259–266, 2013.

[5] Emeric Gioan. Complete graph drawings up to triangle mutations. In *Graph-Theoretic Concepts in Computer Science*, pages 139–150, 2005.

[6] Jonathan L. Gross and Jay Yellen. Fundamentals of graph theory. In *Handbook of Graph Theory*, pages 2–19, 2004.

[7] Richard K. Guy. The decline and fall of Zarankiewicz's theorem. In *Proceedings of the Second Ann Arbor Graph Theory Conference*, pages 63–69, 1968.

[8] Richard K. Guy. Crossing numbers of graphs. In *Graph Theory and Applications*, pages 111–124, 1972.

[9] Frank Haray and Anthony Hill. On the number of crossings in a complete graph. In *Proceedings of the Edinburgh Mathematical Society*, volume 13/4, pages 333–338, 1963.

[10] Lothar Heffter. Über das Problem der Nachbargebiete. In *Mathematische Annalen*, volume 38, pages 477–508, 1891. (In German).

[11] Daniel Kleitman. The crossing number of $K_{5,n}$. In *Journal of Combinatiorial Theory*, volume 9, pages 315–323, 1971.

[12] Daniel Kleitman. A note on the parity of the number of crossings of a graph. In *Journal of Combinatorial Theory (Series B)*, volume 21, pages 88–89, 1976.

[13] Kazimierz Kuratowski. Sur le problme des courbes gauches en topologie. In *Fundamenta Mathematicae*, volume 15, pages 270–283, 1930 (In French).

[14] Jan Kynčl. Simple realizability of complete abstract topological graphs in p. In *Discrete & Computational Geometry*, volume 45/3, pages 383–399, 2011.

[15] Dan Mcquillan and R. Bruce Richter. A parity theorem for drawings of complete and complete bipartite graphs. In *The American Mathematical Monthly*, volume 117/3, page 267273, 2010.

[16] János Pach, József Solymosi, and Géza Tóth. Unavoidable configurations in complete topological graphs. In *Discrete & Computational Geometry*, volume 30/2, pages 311–320, 2003.

[17] János Pach and Géza Tóth. How many ways can one draw a graph? In *Proceedings of Graph Drawing (GD'04)*, pages 47–58, 2004.

[18] János Pach and Géza Tóth. Disjoint edges in topological graphs. In *Combinatorial Geometry and Graph Theory*, pages 133–140, 2005.

[19] Nabil H. Rafla. *The good drawings D_n of the complete graph K_n*. PhD thesis, McGill University, Montreal, 1988.

[20] Andres J. Ruiz-Vargas. Many disjoint edges in topological graphs. In *Proceedings of the 8th Latin-American Algorithms, Graphs and Optimization Symposium (LAGOS'15)*, volume 50, pages 29–34, 2015.

[21] Andrew Suk. Disjoint edges in complete topological graphs. In *Discrete & Computational Geometry*, volume 49/2, pages 280–286, 2013.

[22] Paul Turán. A note of welcome. In *Journal of Graph Theory*, volume 1, pages 7–9, 1977.

Appendix A

Program to check equivalences

This is the code of the program that we use in Section 2.3.1 to find equivalent rotation systems. In the proof of Theorem 2.3.2 from Section 2.3.1, we obtain the rotation systems of all non-isomorphic drawings of graphs with five vertices and minimal degree 3. We can represent those rotation systems with matrices of the standard form explained in the proof. These standard form matrices are the matrices of the code. The order in which the matrices are written in the code is the order in which they are listed in the table of the proof. Rotation systems that are identical are only written once in the code. (It is obvious that two identical rotation systems are equivalent to each other.) The output of the program indicates which of the matrices represent equivalent rotation systems, and which permutations we can use to get from one rotation system to another. More details on the program can be found in Section 2.3.1.

```
import numpy
import itertools

perms = [(x + (5,)) \
        for x in itertools.permutations(range(1, 5))]

def all_perms(m0):
  """all_perms returns all isomorphic \
     representations of m0."""
  for p in perms:
    m = numpy.matrix([p[x-1] for x in \
        m0.flat]).reshape(m0.shape)
    m = repair(m)
    yield (m, p)
```

```
def repair_row(row, skip):
  """repair_row rotates the enumeration of the neighbours,
   starting with the lowest one."""
  i = numpy.argmin(row[skip:]) + skip
  return row[:skip] + row[i:] + row[skip:i]

def repair(m):
  """repair fixes the matrix m \
     so that it is given in standard form"""
  for i in range(4):A
    m[i] = repair_row(list(m[i].flat), 1)
  m[4] = repair_row(list(m[4].flat), 0)
  m[:4] = m[numpy.argsort(m[:4,0].flat)]  # sort rows
  return m

ms = [

    ("1.1.1", [
      [1, 2, 5, 4],
      [2, 1, 3, 5],
      [3, 2, 4, 5],
      [4, 1, 5, 3],
      [1, 2, 3, 4],
    ]),
    ("1.2.2", [
      [1, 2, 5, 4],
      [2, 1, 3, 5],
      [3, 2, 5, 4],
      [4, 1, 5, 3],
      [1, 2, 4, 3],
    ]),
    ("1.2.3", [
      [1, 2, 5, 4],
      [2, 1, 3, 5],
      [3, 2, 5, 4],
      [4, 1, 5, 3],
      [1, 3, 2, 4],
    ]),
    ("1.2.4", [
      [1, 2, 5, 4],
      [2, 1, 3, 5],
      [3, 2, 5, 4],
      [4, 1, 5, 3],
      [1, 3, 4, 2],
    ]),
    ("1.2.5", [
      [1, 2, 5, 4],
      [2, 1, 3, 5],
      [3, 2, 5, 4],
      [4, 1, 5, 3],
      [1, 4, 2, 3],
    ]),
    ("1.3.6", [
      [1, 2, 5, 4],
      [2, 1, 3, 5],
      [3, 2, 5, 4],
      [4, 1, 3, 5],
      [1, 4, 3, 2],
    ]),
    ("1.4.7", [
      [1, 2, 5, 4],
      [2, 1, 5, 3],
      [3, 2, 4, 5],
      [4, 1, 3, 5],
      [1, 4, 3, 2],
    ]),
    ("2.1.8", [
      [1, 2, 5, 4],
      [2, 1, 5, 3],
      [3, 2, 4, 5],
      [4, 1, 3, 5],
      [1, 2, 4, 3],
    ]),
    ("2.2.9", [
      [1, 2, 5, 4],
      [2, 1, 5, 3],
      [3, 2, 5, 4],
      [4, 1, 5, 3],
      [1, 2, 4, 3],
    ]),
```

```
    ("2.2.10", [
      [1, 2, 5, 4],
      [2, 1, 5, 3],
      [3, 2, 5, 4],
      [4, 1, 5, 3],
      [1, 4, 2, 3],
    ]),
    ("2.3.11", [
      [1, 2, 5, 4],
      [2, 1, 5, 3],
      [3, 2, 4, 5],
      [4, 1, 5, 3],
      [1, 2, 3, 4],
    ]),
    ("2.4.12", [
      [1, 2, 4, 5],
      [2, 1, 5, 3],
      [3, 2, 4, 5],
      [4, 1, 5, 3],
      [1, 3, 4, 2],
    ]),
    ("2.4.13", [
      [1, 2, 4, 5],
      [2, 1, 5, 3],
      [3, 2, 4, 5],
      [4, 1, 5, 3],
      [1, 4, 2, 3],
    ]),
    ("2.4.14", [
      [1, 2, 4, 5],
      [2, 1, 5, 3],
      [3, 2, 4, 5],
      [4, 1, 5, 3],
      [1, 3, 2, 4],
    ]),
    ("2.4.15", [
      [1, 2, 4, 5],
      [2, 1, 5, 3],
      [3, 2, 4, 5],
      [4, 1, 5, 3],
      [1, 2, 4, 3],
    ]),
    ("3.1.16", [
      [1, 2, 5, 4],
      [2, 1, 3, 5],
      [3, 2, 4, 5],
      [4, 1, 5, 3],
      [1, 2, 4, 3],
    ]),
    ("3.1.17", [
      [1, 2, 5, 4],
      [2, 1, 3, 5],
      [3, 2, 4, 5],
      [4, 1, 5, 3],
      [1, 4, 2, 3],
    ]),
    ("3.1.18", [
      [1, 2, 5, 4],
      [2, 1, 3, 5],
      [3, 2, 4, 5],
      [4, 1, 5, 3],
      [1, 3, 2, 4],
    ]),
    ("3.1.19", [
      [1, 2, 5, 4],
      [2, 1, 3, 5],
      [3, 2, 4, 5],
      [4, 1, 5, 3],
      [1, 3, 4, 2],
    ]),
    ("3.2.20", [
      [1, 2, 5, 4],
      [2, 1, 5, 3],
      [3, 2, 4, 5],
      [4, 1, 5, 3],
      [1, 4, 3, 2],
    ]),
    ("3.3.21", [
      [1, 2, 5, 4],
      [2, 1, 3, 5],
      [3, 2, 5, 4],
      [4, 1, 5, 3],
      [1, 4, 3, 2],
    ]),

]

# generate global mapping of all possible matrices and the
# ids (and perms) that lead to that matrix
seen = {}
  """mapping from matrix to a set of ids \
     (including the perm)"""
for id, m0 in ms:
  for m, p in all_perms(numpy.matrix(m0)):
    key = str(m)
```

```
    lst = seen.get(key)
    if not lst:
      lst = dict()
    lst[id] = p
    seen[key] = lst

# print the global mapping
for key, lst in seen.iteritems():
  print(key, lst)

# find all duplicates
dupl = set()
for key, lst in seen.iteritems():
  if len(lst) > 1:
    dupl.add(' '.join(sorted(lst)))

# print all duplicates
for x in dupl:
  print(' - %s' % x)
```