FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Filip Čermák

Generating simple drawings of graphs

Department of Applied Mathematics

Supervisor of the bachelor thesis: RNDr. Martin Balko, Ph.D.
Study programme: Computer Science
Study branch: General Computer Science

Prague 2021

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

I would like to mainly thank my supervisor RNDr. Martin Balko, Ph.D. for the leadership and correcting the whole thesis. I would also like to thank the Faculty of Mathematics and Physics for providing me all papers, articles, and software as Grammarly for free and to the Department of Applied Mathematics for providing computers with such a strong computational power. Next, I would like to thank Mgr. Martin Mareš, Ph.D. for the Latex template for bachelor theses. Lastly, I would like to thank my family and friends for understanding that I have not had so much time as I used to have.

Title: Generating simple drawings of graphs

Author: Filip Čermák

Department: Department of Applied Mathematics

Supervisor: RNDr. Martin Balko, Ph.D., Department of Applied Mathematics

Abstract: In this thesis, we study the crossing numbers of complete graphs. After introducing a long history of the old problem of determining the crossing number of K_n, we survey the recent progress on the Harary–Hill conjecture by compiling proofs of this conjecture for special classes of drawings of K_n. We also create a program for generating a database of all simple drawings of K_n with $n \leq 8$. We implement another program that visualizes these drawings and allows the user to create its own simple drawings of general graphs. The visualizer also captures the structure of crossings of the displayed drawings. We used our programs to verify a conjecture by Balko, Fulek, and Kynčl for small cases and we find a mistake in a paper by Mutzel and Oettershagen.

Keywords: crossing number, complete graph, k-edges, cumulated k-edges, the Harary–Hill conjecture

Contents

1. Introduction

1.1 Preliminaries

Let G be a graph without loops and multiple edges. A *drawing* D of G in the plane is the image of a mapping that maps vertices to distinct points and edges to continuous arcs connecting the images of their endpoints. We sometimes do not distinguish between a graph and its drawing, in particular, we identify edges and the arcs representing them, and vertices with the points representing them as well. For simplicity, we assume that the following four conditions are satisfied:

1. no edge goes through a vertex that is not its endpoint,
2. no two edges touch at an interior point,
3. no three edges meet at one common interior point,
4. any two edges share a finite number of intersections.

A *crossing* in D is an interior point of two edges in which they intersect. The *crossing number* $\mathrm{cr}(D)$ of a drawing D is the number of crossings in D. The *crossing number* $\mathrm{cr}(G)$ of a graph G is the minimum crossing number of D over all drawings D of G.

A drawing is *simple* if there are no two edges that intersect more than once. Here, a common endpoint of two edges is counted as their intersection. It is easy to prove that a drawing of every graph minimizing the number of crossings is simple.

In this thesis, we study the crossing numbers of complete graphs. First, we study them from a theoretical point of view. We present a compact survey about the recent progress on the Harary–Hill conjecture, including all the classes for which the conjecture is known to hold. We identify a wrong argument used in the latest article by Mutzel and Oettershagen [1]. Second, we create a program for generating and visualizing simple drawings of K_n. As a result of our program, we create a database of simple drawings of K_n with $n \leq 8$. We use this database to verify a conjecture of Balko, Fulek, and Kynčl for small simple drawings.

1.2 History

The problem to determine the minimum number of crossings of some graph is notoriously difficult in general. Even for special classes of graphs it is considered as one of the most well-known and oldest combinatorial problems about graph drawings.

This problem for the graph $K_{3,3}$ is known as the *House-and-utilities problem* [2] (or *Three utilities problem*) and it states,

> *“Assume there are three cottages on a plane and each cottage should be connected to the water, gas, and electricity companies by pipes. Is it possible to connect cottages with companies in such a way that no pipes cross?”*

(a) The three cottages (blue) and three companies (red).

(b) The cottages and the companies connected by pipes with only one intersection.

Figure 1.1: Three utilities problem.

This problem was described by mathematical puzzler Henry Dudeney "as old as the hills" and "an extinct volcano that burst into eruption in a surprising manner". In fact, it is much older than gas or electricity. However, it is easy to see that one crossing is needed and also sufficient here.

Paul Turán was one of the first mathematicians who dealt with this kind of problems. He thought about these questions during his forced labor in a brick factory during WWII in 1944 [3]. He had to carry bricks on small wheeled trucks from some kilns to storage yards. All the kilns, where the bricks where stored, were connected to all the storage yards by rails. The crossings of the rails made the labor most difficult, because that was the place where the bricks often fell down from the truck.

So Turán's idea was to minimize the number of crossing to save time and energy. Nevertheless, it was not clear what is the minimum number of crossings for an arbitrary number n of kilns and a number m of storage yards.

Formally, this problem is to determine $\mathrm{cr}(K_{m,n})$. Determining the crossing number of the complete bipartite graphs $K_{m,n}$ is over seventy years old and it was studied by polish mathematician Zarankiewicz in the 1950's. Zarankiewicz thought that he had proved that the minimum number of crossings in a drawing of $K_{m,n}$ is equal to

$$\left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{m}{2} \right\rfloor \left\lfloor \frac{m-1}{2} \right\rfloor .$$

However, his solution was flawed. The error was not discovered until eleven years after the publication. In fact, it remains an open problem even until now.

British artist Anthony Hill was interested in geometry and combinatorics and without any deeper knowledge of higher mathematics explored a huge range of geometrical and combinatorial objects. Unfamiliar with the Turán's brick factory problem, he drew some points in the plane, connected them all by arcs, and started finding out how many times these arcs must cross each another [3].

Together with Harary, Hill posed the *Harary–Hill conjecture* which states that the crossing number $\mathrm{cr}(K_n)$ of the complete graph on n vertices equals

$$Z(n) = \frac{1}{4} \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left\lfloor \frac{n-2}{2} \right\rfloor \left\lfloor \frac{n-3}{2} \right\rfloor .$$

This conjecture was proved for $n \leq 12$ [4] and it is known that for $n = 13$ we have only two possible values of $\mathrm{cr}(K_{13}) \in \{223, 225\}$ [5]. It was also proved for many special classes of drawings. However, we still do not know whether this is really the minimum value of the number of crossings. At least we know that $Z(n)$ is an upper bound, as there are some well-known classes of optimal drawings which exemplify the upper bound on the crossing number $\mathrm{cr}(K_n)$; see Figures 1.2, 1.3, and 1.4 for examples.

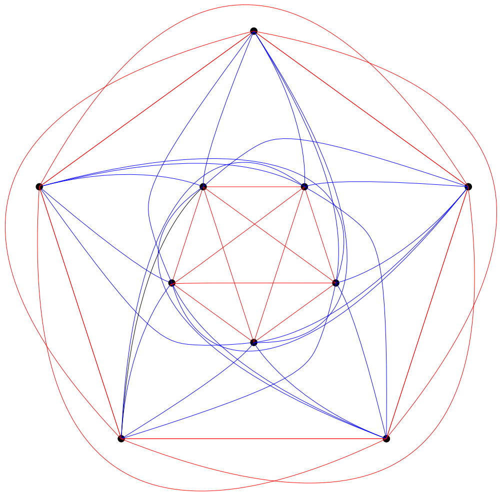


Figure 1.2: An example of an optimal Harary–Hill drawing of K_{10}.

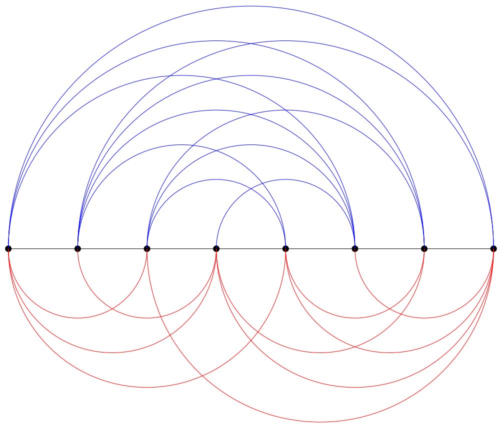


Figure 1.3: An example of an optimal 2-page of K_8 by Blažek and Koman [6].

It is also known that if the Harary–Hill conjecture is true for an odd $n-1$, then it is also true for n by an easy and well-known double-count argument [8]. However, the induction step from $n-1$ even to n odd does not seem to be that easy.

Proposition 1. *For every even positive integer n, if $\mathrm{cr}(K_{n-1}) \geq Z(n-1)$, then $\mathrm{cr}(K_n) \geq Z(n)$.*

Proof. We will prove the lower bound on $\mathrm{cr}(K_n)$ using any m and then substitute $m = n-1$. To prove the lower bound on $\mathrm{cr}(K_n)$, we consider all subgraphs K_m with $m < n$. Suppose $\mathrm{cr}(K_m) = Z(m)$. There are exactly $\binom{n}{m}$ subgraphs of K_n of size m. Now, we make the lower bound estimation on $\mathrm{cr}(K_n)$ using the smaller graphs. We know that we have at least

$$\frac{\binom{n}{m} Z(m)}{\binom{n-4}{m-4}}$$

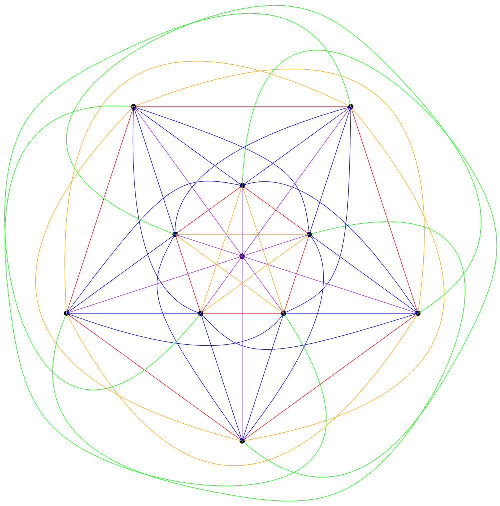


Figure 1.4: An example of an optimal drawing $N_{5,5,1}$ based on the Harary–Hill drawing with no two vertices sharing common face by Ábrego et al [7].

crossings, where we divided the product $\binom{n}{m}Z(m)$ by the factor $\binom{n-4}{m-4}$, because every crossing is induced by four vertices, which means that the crossing was counted $\binom{n-4}{m-4}$ times in the product.

Considering $n-1$ is odd then by substituting $m = n-1$ we get that $\mathrm{cr}(K_n) \geq Z(n)$, as

$$\begin{aligned}
\mathrm{cr}(K_n) \geq \frac{\binom{n}{n-1}Z(n-1)}{\binom{n-4}{n-5}} &= \frac{n \cdot \frac{1}{4}\left\lfloor\frac{n-1}{2}\right\rfloor\left\lfloor\frac{n-2}{2}\right\rfloor\left\lfloor\frac{n-3}{2}\right\rfloor\left\lfloor\frac{n-4}{2}\right\rfloor}{n-4} \\
&= \frac{n \cdot \frac{1}{4}\left\lfloor\frac{n-1}{2}\right\rfloor\left\lfloor\frac{n-2}{2}\right\rfloor\left\lfloor\frac{n-3}{2}\right\rfloor\frac{n-4}{2}}{n-4} \\
&= \left\lfloor\frac{n}{2}\right\rfloor \cdot \frac{1}{4}\left\lfloor\frac{n-1}{2}\right\rfloor\left\lfloor\frac{n-2}{2}\right\rfloor\left\lfloor\frac{n-3}{2}\right\rfloor \\
&= Z(n).
\end{aligned}$$

□

1.3 Our goals

Despite the fact that the Harary–Hill conjecture is open for decades, there has been a substantial progress on this problem recently. Many proofs of the Harary–Hill conjecture for special kinds of drawings were published in the past eight years. We think it is beneficial to sum up all the ideas of these proofs. Therefore our first goal is to survey this recent progress on the Harary–Hill conjecture.

Our second goal is to develop a software tool designed to analyze simple drawings of K_n in order to better understand the structure of these drawings. Such a tool might give us a valuable insight into the structure of the problem of determining $\mathrm{cr}(K_n)$ and potentially lead us to some further progress.

This tool consists of two parts. First, it contains a program to generate a database of all simple drawings of K_n for sufficiently small values of n. Using this database, we can verify several hypotheses about $\mathrm{cr}(K_n)$ for small values of n automatically. The second part of our software tool is a visualiser which serves the user to visualise and modify the drawings from the database as well as creating and analysing his/her own simple drawings of graphs. The visualiser is also equipped with several features that capture the underlined structure of the crossings of the visualised drawings.

2. Recent progress on the Harary–Hill conjecture

This chapter starts with preliminaries, where we introduce some basic definitions and results about the crossing numbers of complete graphs. This is followed by a survey of the recent progress on the Harary–Hill conjecture. This chapter is focused on the lower bounds on the crossing number of the following special classes of drawings of K_n: 2-page [9], cylindrical [10], x-monotone [11], x-bounded [10], shellable [10], bishellable [12], seq-shellable [13], and, lastly, semi-pair-shellable [1].

2.1 Preliminaries

The main concept in all the proofs we will go through are *k-edges*. OTheir origins come from discrete geometry, more specifically, they relate to problems about so-called *halving lines* and *k-sets*. The k-edges were first defined by Erdős et al. [14] who considered them in the following geometric setting. Let P be a set of n points and let (p_i, p_j) be a directed edge between two distinct points p_i and p_j of P. The line determined by the edge (p_i, p_j) separates the plane into two open half-planes, left and right. Let P_l be the set of points from P that lie in the left half-plane. Similarly, let P_r be the set of points from P in the right half-plane. We then call the edge (p_i, p_j) a *k-edge* if $k = \min\{|P_l|, |P_r|\}$.

Later Lovász et al. [15] used the *k-edges* for determining a lower bound on the crossing number of the rectilinear drawings of K_n. A drawing of K_n is *rectilinear* if each edge is represented by a line segment. Note that every rectilinear drawing is simple. The latest extension of the concept of *k-edges* was made by Ábrego et al. [9] who generalized k-edges from rectilinear drawings of K_n to general simple drawings of K_n. We now state this definition, but first we need to introduce some auxiliary terms.

Define $\mathcal{F}(D) = \mathbb{R}^2 \setminus D$ as the set of faces of a simple drawing D of K_n. Denote by $\mathcal{F}(D, v)$ the set containing exactly all the faces $F \in \mathcal{F}(D)$ incident to a vertex v of D. Denote by $F(v)$ the *superface* of F containing the face F in $D - v$. Consider an oriented edge $e = (u, v)$ in the drawing D and let $F \in \mathcal{F}(D)$ be any fixed face of D. We call such a face F a *reference face*. Taking any distinct third vertex w of D, the edge e together with w determines the triangle uvw. Since D is simple, this triangle forms a closed curve separating the plane into two parts. We can distinguish between the left and the right part of the separated plane by orienting the closed curve forming the triangle uvw according to the orientation of e. If the reference face F lies in the left part, we say that the triangle uvw has the *left* orientation, otherwise, it has the *right* orientation; see Figure 2.1.

It is easy to see that every edge e of D determines $n - 2$ triangles together with the remaining vertices of K_n. Each of these triangles has either left or right orientation. Denote by i the number of the triangles containing e with the right orientation. Then we know that there are $n - i - 2$ triangles containing e that have the left orientation. The edge e is then a *k-edge* for $k = \min\{i, n - i - 2\}$. Note that, as in the early point of view by Erdős et al. [15], every edge of K_n is

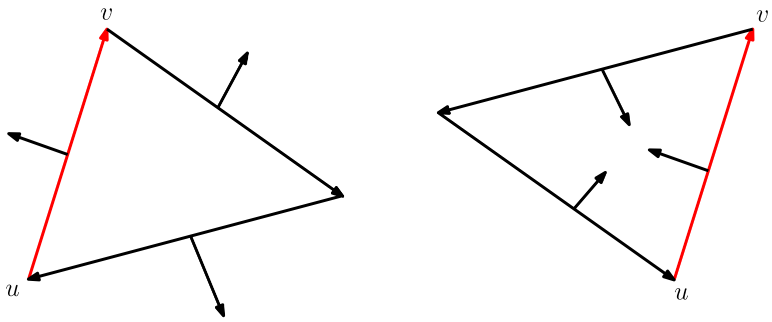


Figure 2.1: Example of a triangle (1) with the outer face on the left and (2) with the outer face on the right

a *k-edge* for some $k \in \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$.

When the reference face of a simple drawing D of K_n is not specified, then it is always the outer face of D. When the reference face is the outer face, a triangle uvw (traced in this order) has the left orientation if and only if it is oriented clockwise and it has the right orientation if and only if it is oriented counterclockwise. Also, there is always a spherical projection turning an arbitrary face of D to the outer face.

It is good to realise that for different reference faces we can have different numbers of k-edges. However, some properties of k-edges are preserved even when we change the reference face; see Lemma 7 for an example. The definition of k-edges can be extended even to so-called *semisimple* drawings [11], but we will not discuss these drawings here.

There is a really nice connection between the number of crossings in a simple drawing of K_n and the number of k-edges first shown by Lovász et al. [15] who used it to prove a lower bound on the crossing number of rectilinear drawings of K_n. We will show how Ábrego et al. [9] generalized this connection to simple drawings of K_n. The main idea is to use a clever double counting argument. We will now sketch the proof of this result because it is the basic building block of the entire recent progress on the Harary–Hill conjecture.

Theorem 1. *Let D be a simple drawing of K_n and let $F \in \mathcal{F}(D)$ be the reference face. Then*

$$\mathrm{cr}(D) = 3\binom{n}{4} - \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor - 1} k(n-k-2)E_k(D).$$

Proof. One can show using a simple case analysis that there are only three pairwise non-isomorphic types of drawings of K_4. See Figure 2.2, where these three types A, B, and C are depicted. We denote by T_A, T_B, and T_C the numbers of subdrawings of K_4 of types A, B, and C, respectively, induced by D.

As remarked earlier, the key idea is to use a double counting argument. We double count so-called separations. We say that an edge uv of D separates distinct vertices w and t if the reference face F is, without loss of generality, on the left of the triangle uvw and on the right of the triangle uvt (otherwise we can swap the direction of the edge uv). The set $\{uv, w, t\}$ is then called a *separation*.

It easy to see that a k-edge belongs to exactly $k(n-2-k)$ separations, because it determines k triangles oriented one way and $n-2-k$ triangles oriented the

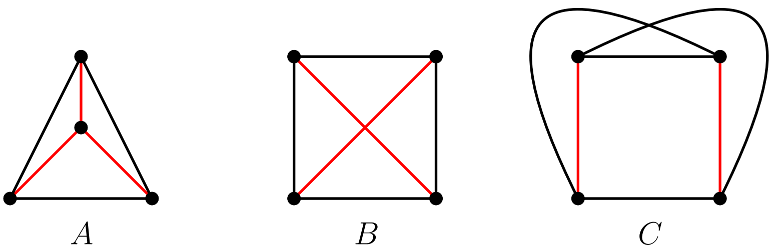


Figure 2.2: Types A, B, and C of simple drawings of K_4 with 3, 2, and 2 separations, respectively. The separations, which correspond to 1-edges with respect to the outer face, are coloured red.

other way with respect to F. Summing over all k-edges we thus get

$$3T_A + 2T_B + 2T_C = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor -1} k(n-k-2)E_k(D). \tag{2.1}$$

The total number of induced subdrawings of K_4 is $\binom{n}{4}$. In other words, we have $T_A + T_B + T_C = \binom{n}{4}$, which can be rewritten as $T_A = \binom{n}{4} - T_B - T_C$. We can use this equality to rewrite (2.1) as

$$3\left(\binom{n}{4} - T_B - T_C\right) + 2T_B + 2T_C = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor -1} k(n-k-2)E_k(D).$$

We clearly have $\mathrm{cr}(D) = T_B + T_C$, as only the subdrawings of K_4 of types B and C contain a crossing. Plugging this into the above expression, we obtain the desired equality

$$\mathrm{cr}(D) = T_B + T_C = 3\binom{n}{4} - \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor -1} k(n-k-2)E_k(D),$$

which concludes the proof.

□

It follows from Theorem 1 that we can use estimates on the numbers k-edges to obtain estimates on the number of crossings. Unfortunately, it turns out it is difficult to obtain useful estimates on $E_k(D)$. However, this can be overcome by considering appropriately weighted k-edges. This motivates the following definitions of cumulated k-edges.

Definition 2.1.1. Let D be a simple drawing of K_n and let $E_k(D)$ be the number of k-edges in D for $k \in \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$ with respect to a reference face $F \in \mathcal{F}(D)$. Then we denote by

$$E^1_{\leq k}(D) = \sum_{i=0}^{k} E_i(D)$$

the number of *cumulated k-edges*, by

$$E^2_{\leq k}(D) = \sum_{i=0}^{k} E^1_{\leq i}(D) = \sum_{i=0}^{k} (k+1-i)E_i(D)$$

the number of *double cumulated k-edges*, and by

$$E^3_{\leq k}(D) = \sum_{i=0}^{k} E^2_{\leq i}(D) = \sum_{i=0}^{k} \binom{k+2-i}{2} E_i(D)$$

the number of *triple cumulated k-edges*.

The cumulated k-edges were first used by Lovász et al. [14]. The double cumulated k-edges were introduced by Ábrego et al. [9] and the triple cumulated ones by Balko, Fulek, and Kynčl [11]. In the literature, the number of cumulated, double cumulated, and triple cumulated k-edges was typically denoted by $E_{\leq k}(D)$, $E_{\leq\leq k}(D)$, and $E_{\leq\leq\leq k}(D)$, respectively.

The expression from Theorem 1 can be rewritten in terms of double cumulated k-edges to the following form.

Corollary 1.1. Let D be a simple drawing of K_n. With respect to a reference face $F \in \mathcal{F}(D)$, we have

$$\operatorname{cr}(D) = 2 \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor - 2} E^2_{\leq k}(D) - \frac{1}{2}\binom{n}{2}\left\lfloor \frac{n-2}{2} \right\rfloor - \frac{1}{2}(1 + (-1)^n) E^2_{\leq \lfloor \frac{n}{2} \rfloor - 2}(D).$$

For triple cumulated edges, we get the following result.

Corollary 1.2. Let D be a simple drawing of K_n. With respect to a reference face $F \in \mathcal{F}(D)$, we have

$$\operatorname{cr}(D) = 2E^3_{\leq \lfloor \frac{n}{2} \rfloor - 2}(D) - \frac{1}{8} n(n-1)(n-3)$$

for odd n and

$$\operatorname{cr}(D) = E^3_{\leq \lfloor \frac{n}{2} \rfloor - 2}(D) + E^3_{\leq \lfloor \frac{n}{2} \rfloor - 3}(D) - \frac{1}{8} n(n-1)(n-2)$$

for even n.

When expressing $\operatorname{cr}(D)$ by k-edges and their cumulations, it is easy to see that if we were able to give some lower bounds on the values of k-edges or their cumulations, then it would give us some lower bound on the crossing number. The idea, which is common to all the proofs, is to prove suitable lower bounds on the number of k-edges and their cumulations for all k. It is not difficult to show that if $E_k(D) \geq 3(k+1)$ for each $k \in \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 2\}$, then the Harary–Hill conjecture is true for D. However, the lower bound $E_k(D) \geq 3(k+1)$ does not hold even for rectilinear drawings. Therefore, we consider only the following cumulated versions.

Claim 2. *Let D be a simple drawing of K_n and $E^1_{\leq k}(D)$ be a number of cumulated k-edges in D with respect to a reference face $F \in \mathcal{F}(D)$. If the lower bounds*

$$E^1_{\leq k}(D) \geq 3\binom{k+2}{2}$$

hold for each $k \in \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 2\}$, then $\operatorname{cr}(D) \geq Z(n)$.

Claim 3. *Let D be a simple drawing of K_n and $E^2_{\leq k}(D)$ be a number of double cumulated k-edges in D with respect to a reference face F. If the lower bounds*

$$E^2_{\leq k}(D) \geq 3\binom{k+3}{3}$$

hold for each $k \in \{0, \dots, \left\lfloor \frac{n}{2} \right\rfloor - 2\}$, then $\mathrm{cr}(D) \geq Z(n)$.

Claim 4. *Let D be a simple drawing of K_n and $E^3_{\leq k}(D)$ be a number of triple cumulated k-edges in D with respect to a reference face F. If the lower bounds*

$$E^3_{\leq k}(D) \geq 3\binom{k+4}{4}$$

hold for $k = \left\lfloor \frac{n}{2} \right\rfloor - 2$ if n is odd and $k \in \{\left\lfloor \frac{n}{2} \right\rfloor - 2, \left\lfloor \frac{n}{2} \right\rfloor - 3\}$ if n is even, then $\mathrm{cr}(D) \geq Z(n)$.

2.2 The 2-page and monotone drawings of K_n

Lovász et al. [15] proved a lower bound on the crossing numbers of rectilinear drawings of complete graphs using estimates on cumulated k-edges and proved the lower bound from Claim 2. However, there are many simple drawings of K_n which do not satisfy Claim 2. The first such a class of simple drawings for which the Harary–Hill conjecture was shown to be true were 2-page drawings [9].

Now we will proceed through the common part of all the proofs of the Harary–Hill conjecture for restricted classes of drawings of K_n. Then we show the proof for monotone drawings of complete graphs and we discuss so called 2-page drawings afterwards.

2.2.1 The structure of simple drawings

We want to prove the inequalities from Claim 3 by double induction on k and on the number of vertices n. We want to remove one "nice" vertex, in this case meaning a vertex incident to the outer face, and we also want to estimate the number of double cumulated k-edges if we know the number double cumulated $(k-1)$-edges. It is easy to see that after removing one vertex v, the number k for each edge can either remain the same or it can be reduced by one.

Let D be a simple drawing of K_n and let v be a vertex of D. An edge of D is *invariant* with respect to v if it is a k-edge in D and also in $D - v$. From the definition of $E^2_{\leq k}(D)$ we know that an i-edge contributes to $E^2_{\leq k}(D)$ with $(k+1-i)$. On the other hand, after removing the vertex v, an i-edge becomes either an i-edge or an $(i-1)$-edge. This means that it will contribute to $E^2_{\leq k-1}(D - v)$ with either $(k - i)$ or $(k - (i - 1)) = (k + 1 - i)$. In other words, an invariant edge reduces the value it contributes to $E^2_{\leq k-1}(D - v)$ by one when compared to the value it contributes to $E^2_{\leq k}(D)$. A non-invariant edge contributes to $E^2_{\leq k-1}(D-v)$ with the same value as to $E^2_{\leq k}(D)$.

So there are three terms we need to know in order to express $E^2_{\leq k}(D)$. First, the number $E^2_{\leq k-1}(D)$ of double cumulated $(k-1)$-edges of the drawing $D - v$,

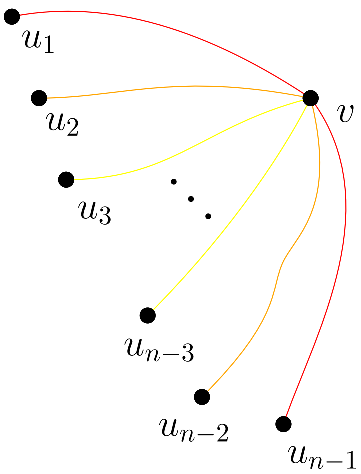


Figure 2.3: The structure of k-edges around a vertex v on the outer face.

second, the number $E^1_{\leq k}(D, D-v)$ of cumulated invariant k-edges, and, third, the contribution $E^2_{\leq k}(v)$ of edges incident to the vertex v. In other word we have

$$E^2_{\leq k}(D) = E^2_{\leq k-1}(D-v) + E^1_{\leq k}(D, D-v) + E^2_{\leq k}(v). \tag{2.2}$$

Now, we estimate the three terms on the right side of (2.2) by induction on k and on the number n of vertices. For the base of the induction, we show that for $k = 0$ and for arbitrary simple drawing D' of K_n with more than two vertices, we have $E^2_{\leq 0}(D') \geq 3$. This is true because every such a drawing contains at least three edges incident to the outer face and the triangles containing such edges are all oriented the same way. This shows that we have at least three 0-edges in D' and thus $E^2_{\leq 0}(D') \geq 3$.

Now for the induction step, let $k \geq 1$ be an integer and let D be a simple drawing of K_n with $k \leq \left\lfloor \frac{n}{2} \right\rfloor - 2$. From the induction hypothesis, the first term $E^2_{\leq k-1}(D-v)$ can be estimated from below by $3\binom{(k-1)+3}{3} = 3\binom{k+2}{3}$ because $D-v$ is also monotone or 2-page but this will be discussed later.

To estimate the third term $E^2_{\leq k}(v)$ of (2.2), we apply the following lemma.

Lemma 5. *Consider a simple drawing D of K_n and a vertex v on the outer face of D. If we label vertices of $D-v$ by $u_1, \ldots, u_{n-1}$ in the counter-clockwise order around v, then the edges vu_i and uv_{n-i} are $(i-1)$-edges for every $i \in \{1, \ldots, \left\lfloor \frac{n}{2} \right\rfloor\}$; see Figure 2.3.*

Proof. When considering the edge vu_i of D, all the triangles vu_iu_j for $j < i$ are oriented clockwise and are oriented counter-clockwise for $j > i$. Otherwise v is not on the outer face of D, which is a contradiction; see Figure 2.4. Thus vu_i and vu_{n-i} are indeed $(i-1)$-edges. □

This means that the whole contribution of the edges incident to v in D to $E^2_{\leq k}(D)$ is $E^2_{\leq k}(v) = 2\sum_{i=0}^{k}(k+1-i) = 2\binom{k+2}{2}$ because vu_1,..., vu_{k+1} are 0-,..., k-edges and vu_{n-k-1},..., vu_{n-1} are k-,..., 0-edges for every $k \leq \left\lfloor \frac{n}{2} \right\rfloor - 2$.

Remark. We know that it is sufficient for Claim 3 to have k in $\{0, \ldots, \left\lfloor \frac{n}{2} \right\rfloor - 2\}$. For $k = \left\lfloor \frac{n}{2} \right\rfloor - 1$, the lower bound on $E^2_{\leq k}(v)$ does not have to hold because for n even the edges vu_{k+1} and vu_{n-k-1} merge.

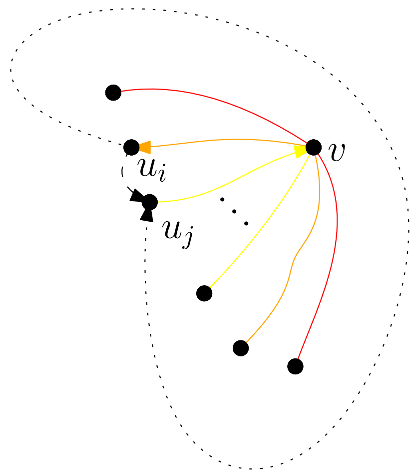


Figure 2.4: The orientation of every triangle vu_iu_j with $j > i$ is counter-clockwise (dashed curve u_iu_j). On the other hand, since the drawing is simple and v is on the outer face, the clockwise orientation (dotted curve u_iu_j) is a contradiction because then v is not on the outer face.

2.2.2 Monotone drawings

Now, we show the proof of the Harary–Hill conjecture for so-called monotone drawings of complete graphs by Balko, Fulek, and Kynčl [11].

Definition 2.2.1. A simple drawing D of a graph is *monotone* if each edge is crossed by every vertical line at most once.

The last missing piece in the proof of the Harary–Hill conjecture for monotone drawings is the estimate on the number of invariant edges. Until now, we did not use the assumption that the drawing D is monotone, we only assumed that D is simple. For a monotone drawing D of K_n, we sort the vertices by their x-coordinates from left to right and we pick the rightmost vertex $v = u_n$, which is obviously on the outer face.

Definition 2.2.2. Consider a vertex u_i of D. The j *topmost* (*bottommost*) edges at the vertex u_i is the set of the first j edges of D with the left endpoint u_i in the clockwise (counter-clockwise), respectively, order around u_i.

Lemma 6. *Let D be a simple drawing of K_n and $k \leq \left\lfloor \frac{n}{2} \right\rfloor - 2$. Consider the vertices $u_1, \ldots, u_{k+1}$. Then, for each $i \in \{1, \ldots, k+1\}$, at least $k+2-i$ topmost or bottommost edges at u_i are invariant.*

Proof. As we can see in Figure 2.5, the edge vu_i is either not among the $k+2-i$ topmost or not among $k+2-i$ bottommost edges at u_i. First, consider the vertex u_1 on the outer face. By Lemma 5, there are 0-,…, k-edges and then k-, …, 0-edges around u_1. We also know, from the proof of Lemma 5, that an edge e among the topmost (bottommost) $k+1$ edges around u_1 determines a counter-clockwise (a clockwise) oriented triangle together with any other edge which is above (below) e and a clockwise (a counter-clockwise) oriented triangle with any other edge below (above) e. Without loss of generality, we assume that vu_1 is not among the $k+1$ bottommost edges. Then every j-edge among the

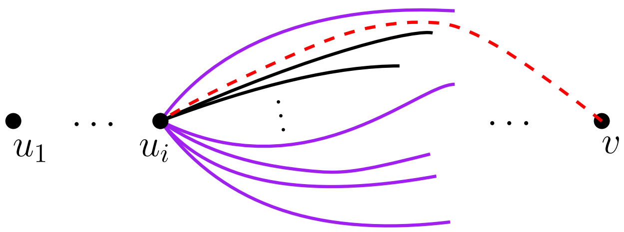


Figure 2.5: The red edge u_iv and the $k+2-i$ topmost and the bottommost edges at v are all at most k-edges in D. The edge u_iv in the upper half, which means that we have at least $k+2-i$ edges going from left to right at u_i in the lower half and all of them are invariant. Some edges in the upper part can also be invariant. For example, the topmost one is invariant here.

$k+1$ bottommost edges at u_1 remains a j-edge after removing v. Thus we have at least $k+1$ invariant edges at u_1 with the left endpoint at u_1.

Now, assume that all the vertices $u_1, \ldots, u_{i-1}$ were removed. Then u_i is on the outer face because D is monotone. We can apply the same reasoning as before for u_1 to show that we have at least $k+2-i$ invariant edges at u_i with the left endpoint at u_i. Since after removing one vertex of D, each j-edge either stays a j-edge or become a $(j-1)$-edge, we have at most $(i-1)$-, $\ldots$, k-edges and then k-, $\ldots$, $(i-1)$-edges around the vertex u_i in the drawing $D - \{u_1, \ldots, u_{i-1}\}$. Without loss of generality, we assume that vu_i is not among the $k+2-i$ bottommost edges around the vertex u_i. Then these $k+2-i$ edges are invariant with respect to v. We also need to realize that $k+2-i$ topmost and bottommost edges at u_i in $D - \{u_1, \ldots, u_{i-1}\}$ are disjoint sets. Therefore we need to check that $2(k+2-i) \leq (n-1)-(i-1)$, but this is trivial because $k \leq \left\lfloor \frac{n}{2} \right\rfloor - 2$.

□

Using Lemma 6 we immediately obtain the following bound.

Corollary 6.1. For every $k \leq \left\lfloor \frac{n}{2} \right\rfloor - 2$, we have

$$E^1_{\leq k}(D, D-v) \geq \sum_{i=1}^{k+1} (k-(i-1)+1) = \sum_{i=1}^{k+1} (k+2-i) = \binom{k+2}{2}.$$

Now, we have suitable lower bounds on all three terms on the right side of (2.2). If we put them together, we obtain

$$\begin{aligned} E^2_{\leq k}(D) &= E^2_{\leq k-1}(D-v) + E^1_{\leq k}(D, D-v) + E^2_{\leq k}(v) \geq \\ &\geq 3\binom{k+2}{3} + 2\binom{k+2}{2} + \binom{k+2}{2} = \\ &= 3\binom{k+3}{3}, \end{aligned}$$

which is exactly the estimate in Claim 3. So the assumption from Claim 3 is true for monotone drawings and, consecutively, the Harary–Hill conjecture holds for monotone drawings.

We recall three steps where we used the assumption that D is monotone. The first step is considered when we wanted to apply induction hypothesis so we needed that $E^2_{\leq k-1}(D-v) \geq 3\binom{k+2}{3}$. Therefore we need to know that $D-v$ is also monotone which is trivial and that $k-1 \leq \left\lfloor \frac{n-1}{2} \right\rfloor - 2$ which is also obvious because $k \leq \left\lfloor \frac{n}{2} \right\rfloor - 2$.

The second step was when we found a vertex on the outer face. Here the assumption was not really necessary as we can always pick a vertex and face $F \in \mathcal{F}(D)$ incident to it and consider F to be the outer face.

The last time we used the assumption that D is monotone was in the part about invariant edges. We needed to know that after removing the vertices $u_1, \ldots, u_{i-1}$, the vertex u_i is on the outer face F. Suppose for a contradiction, there is an edge e separating u_i from the outer face. Then, however, there is no vertex to the left of u_i, which means that e should go around the vertex u_i, in other words, the vertical line going though u_i intersects e at least twice, which contradicts the fact that D is monotone.

In fact the proof works because monotone drawings contain the following sequence of vertices u_i that helps us to find sufficiently many invariant edges.

Definition 2.2.3. Let D be a simple drawing of K_n with a reference face F. Let k be a non-negative integer, let v be a vertex incident to F, and let $S_v = (u_1, \ldots, u_{k+1})$ be a sequence of distinct vertices where $u_i \in V \setminus \{v\}$. If for each $i \in \{1, \ldots, k+1\}$ the vertex u_i is incident to the superface of F in the drawing $D - \{u_1, \ldots, u_{i-1}\}$, then we call such a sequence S_v *simple*.

2.2.3 2-page drawings

In this subsection, we consider so-called 2-page drawings of complete graphs. This class of drawings is contained in the class of simple monotone drawings of complete graphs and the Harary–Hill conjecture was known to be true for this class even before simple monotone drawings [9]. However, the proof for 2-page drawings uses a different notation and since these drawings are more specialized than the monotone ones, we decided to state the proof for the monotone drawings first.

Definition 2.2.4. A simple drawing D of K_n is *2-page* if all vertices of D lie on a line ℓ (called the *spine*) and each edge of D is fully contained in one of the half-planes (called *pages*) determined by ℓ; see Figure 1.3.

Ábrego et al. [9] described the structure of 2-page drawings using a matrix representation, which describes whether an edge is going through the upper or the lower page separated by the spine. Although the structure of 2-page drawings is represented differently, all parts of the proof for monotone drawings work, except for the part about invariant edges discussed at the end of Subsection 2.2.2. We can again suppose for a contradiction that after removing $u_1, \ldots, u_{i-1}$, there is an edge e separating u_i from the outer face. However, this means that e goes around the vertex u_i. In other words, the spine is crossed by e, which is a contradiction. So we have found a simple sequence also here and therefore the Harary–Hill conjecture holds for 2-page drawings.

Ábrego et al. [9] also studied properties and the structure of optimal 2-page drawings. They showed that there exists a unique optimal 2-page drawing of K_n for any even n, while there are exponentially many optimal 2-page drawings of K_n for n odd, up to sphere-homeomorphism.

2.3 Shellable drawings of K_n

It turned out quickly that finding invariant edges is not always easy, but the proof for monotone drawings can be strengthened. Therefore the family of drawings for which the Harary–Hill conjecture is known to be true started to broaden up. We have already seen that the assumption that our drawings are monotone or 2-page was needed only in one step and otherwise it was not really necessary. The next authors Ábrego et al. [10] generalized the monotone drawings to a class of so-called shellable drawings and proved that the Harary–Hill conjecture holds for such drawings. To state the definition of shellable drawings, we need to state some auxiliary terms.

Definition 2.3.1. ([10]) Let D be a simple drawing of K_n with a reference face $F \in \mathcal{F}(D)$ and let s be a positive integer. The drawing D is *s-shellable* if there exists a sequence $v_1, \dots, v_s$ of distinct vertices of D such that, for all pairs of positive integers r and t with $1 \leq r < t \leq s$, the vertices v_r and v_t are both incident to the superface of F in the drawing $D - \{v_1, \dots, v_{r-1}, v_{t+1}, \dots, v_s\}$.

For $s \geq \left\lfloor \frac{n}{2} \right\rfloor$ and an s-shellable drawing of K_n, Ábrego et al. [10] proved the Harary–Hill conjecture. Also, to simplify the discussion, the drawings with which are s-shellable for $s \geq \left\lfloor \frac{n}{2} \right\rfloor$ are called *shellable*.

Ábrego et al. [10] proceed with the proof little bit differently than we do. They fix an s-shellable drawing D of K_n with a sequence $v_1, \dots, v_s$ and $k \leq \min(s-2, \left\lfloor \frac{n-1}{2} \right\rfloor - 1)$. For each such k and D they apply induction over $i \leq k$. In every step they remove $k - i$ vertices $v_s, \dots, v_{s-(k-i)+1}$ and prove the estimate $E^2_{\leq i}(D - \{v_s, \dots, v_{s-(k-i)+1}\} \geq 3\binom{i+3}{3}$ using the induction step based on equality (2.2). For $i = 0$, the estimate is trivial. The structure here is little bit different since in our proof we do not need to fix k and do induction for each $i \leq k$. We immediately proceed using double induction based on equality (2.2).

Now, we prove the Harary–Hill conjecture for shellable drawings using simple sequences. First, if D is s-shellable, then for each $k \leq min\{s-2, \left\lfloor \frac{n}{2} \right\rfloor - 2\}$ we have $E^2_{\leq k}(D) \geq 3\binom{k+3}{3}$. This is trivial for $k = 0$ because then there are always three 0-edges incident to F. Now, we show that s-shellability of D implies the $(s-1)$-shellability of $D - v$ for $v = v_s$. However, this is obvious, as we can only forget v_s, the last term of the sequence $(v_1, \dots, v_s)$, and shorten the sequence by v_s to witness the $(s-1)$-shellability of the drawing $D - v_s$. Finally, to use the induction step, we also need to know that $k \leq min\{s-2, \left\lfloor \frac{n}{2} \right\rfloor - 2\}$ implies $k - 1 \leq min\{s-2, \left\lfloor \frac{n}{2} \right\rfloor - 2\} - 1 \leq min\{(s-1)-2, \left\lfloor \frac{n-1}{2} \right\rfloor - 2\}$. Therefore we can use the induction step and get $E^2_{\leq k-1}(D) \geq 3\binom{k+2}{3}$.

Now we can use the equality (2.2). The third term of (2.2) is estimated because considering $v = v_s$ together with s-shellability implies that v_s is incident to the reference face F. To estimate the second term of (2.2), it suffices to show that there is a simple sequence for every shellable drawing. We can consider v to be the vertex v_s incident to F in D, since considering $r = 1$ and $t = s$ we get that v_1 and v_s are both on the face F. Now it is easy to see that by taking $r = 1, \dots, k+1$ and $t = s$ we get our simple sequence $(v_1, \dots, v_{k+1})$. Now we need to show that the index r is always at most $s - 1$ because v_s is taken as v in order to use the definition of s-shellability. Then all vertices v_i are incident to the superface of F in $D - \{v_1, \ldots, v_{i-1}\}$. Since $k \leq \left\lfloor \frac{n}{2} \right\rfloor - 2$ and $s \geq \left\lfloor \frac{n}{2} \right\rfloor$, we indeed have $s - 1 \geq r$, as $s - 1 \geq \left\lfloor \frac{n}{2} \right\rfloor - 1 \geq k + 1 \geq r$.

We can now see that the class of shellable drawings is not the most general one we can consider because we needed to use only the conditions for a subset of pairs r, t and not for all of them. However, the class of s-shellable drawings with $s \geq \left\lfloor \frac{n}{2} \right\rfloor$ contains many well-known classes of simple drawings of K_n for which the Harary–Hill conjecture is now known to be true. In particular, it contains the following two classes.

Definition 2.3.2. A simple drawing of K_n is *x-bounded* if for any two vertices v_i and v_j the edge $v_i v_j$ does not cross neither the vertical line going through v_i nor the vertical line going though v_j.

We can easily see that the argument that worked for monotone drawings easily translates to x-bounded ones.

Definition 2.3.3. A simple drawing of K_n is *cylindrical* if there are two concentric circles that contain all the vertices of the graph K_n and no edge of K_n crosses any of the circles.

Ábrego et al. [10] proved that simple drawings of K_n containing a cycle that satisfies certain crossing-constraints is shellable. However, this structure is, in our opinion, unnecessary because we only need to show that there is a simple sequence in such drawings. To show that, consider a cylindrical drawing D of K_n. By the pigeonhole principle, there is a circle C in D with at least $s \geq \left\lfloor \frac{n}{2} \right\rfloor$ vertices. Consider some vertex v_0 on C to be the vertex that we are about to remove. Let $v_1, \ldots, v_{k+1}$ be consecutive vertices on C that follow v_0. We know that the edge $v_i v_{i+1}$ is not crossed by any other edge of D since $v_i v_{i+1}$ and the circle C form a closed curve C' that does not enclose any vertex of D. This is because v_i and v_{i+1} are consecutive vertices on C. Therefore any edge of D should either go through the part of C' formed by C, which is impossible as D is cylindrical, or it crosses the edge $v_i v_{i+1}$ at least twice, which is also impossible as D is simple. Therefore, we can choose the face F as the face determined by the closed curve formed by $v_0 v_1$ and C. Now, after removing $v_1, \ldots, v_{i-1}$, the vertices v_0 and v_i become consecutive on C and it is easy to see that the superface of F in $D - \{v_1, \ldots, v_{i-1}\}$ is the face formed by C and by $v_0 v_i$ and therefore it is incident to v_i, which is exactly what we wanted to get a simple sequence. The part about induction step is the same as for shellable ones.

2.4 Bishellable drawings of K_n

Ábrego et al. [12] further generalized the class of shellable drawings to the following more natural class of drawings of K_n.

Definition 2.4.1. ([12]) Let D be a simple drawing of K_n with a reference face $F \in \mathcal{F}(D)$ and let s be a non-negative integer. The drawing D is *s-bishellable* if there are two sequences $a_0, \ldots, a_s$ and $b_s, \ldots, b_0$, each containing distinct vertices of D, such that

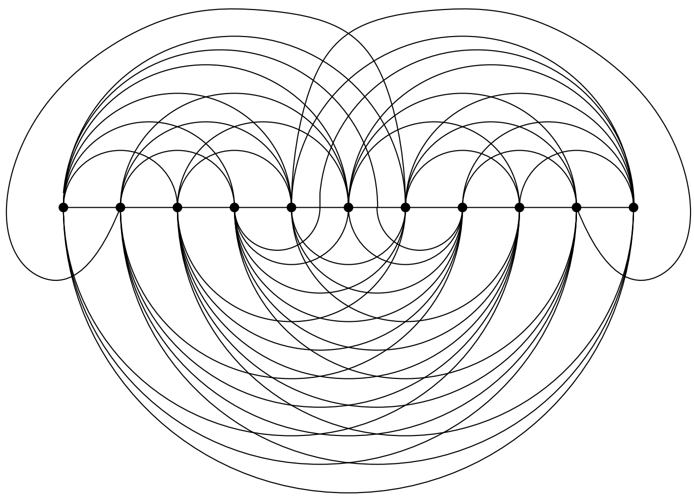


Figure 2.6: An optimal drawing of K_{11} which is bishellable but not s-shellable for any $s \geq 5$.

1. for each $i \in \{0, \ldots, s\}$, the vertex a_i is incident to the superface of F in the drawing $D - \{a_0, \ldots, a_{i-1}\}$,
2. for each $i \in \{0, \ldots, s\}$, the vertex b_i is incident to the superface of F in the drawing $D - \{b_0, \ldots, b_{i-1}\}$, and
3. for each $i \in \{0, \ldots, s\}$, the set $\{a_0, \ldots, a_i\} \cap \{b_{s-i}, \ldots, b_0\}$ is empty.

In contrast to the s-shellable drawings of K_n, if the drawing D is s-bishellable, then it is also $(s-1)$-bishellable by shortening the sequences $(a_0, \ldots, a_s)$ by a_s and $(b_s, \ldots, b_0)$ by b_s. Additionally, if D is s-bishellable, then the drawing $D - v$ for $v = a_0$ is $(s-1)$-bishellable when we shorten the sequence $(b_s, \ldots, b_0)$ by b_s.

The Harary–Hill conjecture was proved for $(\left\lfloor \frac{n}{2} \right\rfloor - 2)$-bishellable drawings, which are called just *bishellable*. Ábrego et al. [12] constructed infinitely many bishellable drawing which are not shellable; see Figure 2.6.

Similarly as in the previous section about the shellable drawings, we would like to prove the argument using simple sequences also here. First, we again need to check whether we can apply induction on the first term of equality (2.2). We state that if a drawing D of K_n is s-bishellable for $s \leq \left\lfloor \frac{n}{2} \right\rfloor - 2$, then $E^2_{\leq s} \geq 3\binom{s+3}{3}$. The case for $s = 0$ is trivial because we have at least three 0-edges incident to F as D is simple. Then, as we said before, if D is s-shellable, then $D - a_0$ is $(s-1)$-shellable. Then again, since $s \leq \left\lfloor \frac{n}{2} \right\rfloor - 2$, we also have $s - 1 \leq \left\lfloor \frac{n}{2} \right\rfloor - 2 - 1 \leq \left\lfloor \frac{n-1}{2} \right\rfloor - 2$. And therefore we can apply the induction.

We can again easily see that for $v = a_0$, and the sequence $(b_s, \ldots, b_0)$ our argument for counting invariant edges also works. We can choose $b_0, \ldots, b_s$ to be our simple sequence $(v_1, \ldots, v_{s+1})$. The last term of equality (2.2) is determined therefore the proof of the Harary-Hill conjecture for bishellable drawings works the same.

Remark. If D is s-shellable witnessed by a sequence $v_1, \ldots, v_s$, then it is also $(s-2)$-bishellable. To see that, it suffices to set $(a_0, a_1, \ldots, a_{s-2}) = (v_1, v_2, \ldots, v_{s-1})$ and $(b_0, b_1, \ldots, b_{s-2}) = (v_s, v_{s-1}, \ldots, v_2)$.

2.5 Seq-shellable drawings

Both shellable and bishellable drawings were probably motivated by extending the family of drawings for which the Harary–Hill conjecture holds. However, the structure of shellable or bishellable drawings was created to prove the conjecture for some known drawings which have not belong to any known class yet and the structure we need was not the most general we could get. Therefore, Mutzel and Oettershagen in [13] introduced the simple sequences (Definition 2.2.3) and used them to extend the class of drawings for which the Harary–Hill conjecture is true.

Definition 2.5.1. ([13]) Let D be a simple drawing of K_n. We call D *s-seq-shellable* for some non-negative integer s if there exists a reference face $F \in \mathcal{F}(D)$ and a sequence of distinct vertices $(a_0, \ldots, a_s)$ such that

1. for each $i \in \{0, \ldots, s\}$, the vertex a_i is incident to the face containing F in the drawing $D - \{a_0, \ldots, a_{i-1}\}$ and
2. for each $i \in \{0, \ldots, s\}$, the vertex a_i has a simple sequence $S_i = (u_0, \ldots, u_{s-i})$ with $u_j \in V \setminus \{a_0, \ldots, a_i\}$ for all $j \in \{0, \ldots, s-i\}$ in the drawing $D - \{a_0, \ldots, a_{i-1}\}$.

As we can see, this is the definition we used all the time for the simple sequence generating invariant edges. As in the s-bishellable drawings, also the s-seq-shellability of the drawing D implies $(s-1)$-seq-shellabity for the drawings D and $D - v$ after choosing $v = a_0$ and forgetting the simple sequence S_0. For simplicity, we call $(\left\lfloor \frac{n}{2} \right\rfloor - 2)$-seq-shellable drawings just *seq-shellable*. The proof of the Harary-Hill conjecture works the same for seq-shellable drawings as for bishellable drawings since s-seq-shellability of a drawing D implies $(s-1)$-seq-shellability of $D - a_0$. The base of the induction is the same and, again, $s \leq \left\lfloor \frac{n}{2} \right\rfloor - 2$ implies $s - 1 \leq \left\lfloor \frac{n}{2} \right\rfloor - 2 - 1 \leq \left\lfloor \frac{n-1}{2} \right\rfloor - 2$ and the simple sequence S_0 is given.

Remark. Note that every s-bishellable drawing is also a s-seq-shellable drawing. We can either consider $(a_0, \ldots, a_s) := (a_0, \ldots, a_s)$ and simple sequences $S_i = (b_0, \ldots, b_{s-i})$, or $(a_0, \ldots, a_s) := (b_0, \ldots, b_s)$ and simple sequences $S_i = (a_0, \ldots, a_{s-i})$. Here we can see, that s-bishellability is a stronger constraint than the s-seq-shellability; see Figure 2.7.

This concludes all the known progress about the classes for which Harary–Hill conjecture is true due to induction argument on double cumulated k-edges (using Claim 3). However, there are still many drawings for which there exists a face for which Claim 3 does not hold; see Figure 2.8. It easy to see that this claim is stronger than the Harary–Hill conjecture. However, when we rewrite Theorem 1 into Corollary 1.2 we can see that, at least for some number k, Corollary 1.2 and the Harary–Hill conjecture are of the same strength. There is no known example of a simple drawing of K_n that does not satisfy the inequalities from Claim 4 for any $k \leq \left\lfloor \frac{n}{2} \right\rfloor - 2$.

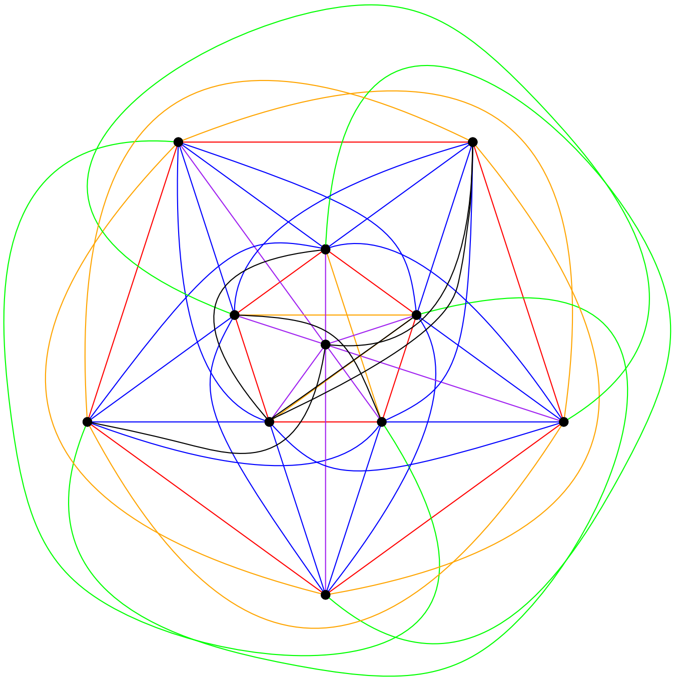


Figure 2.7: A drawing of K_{11} which is seq-shellable but it is not bishellable.

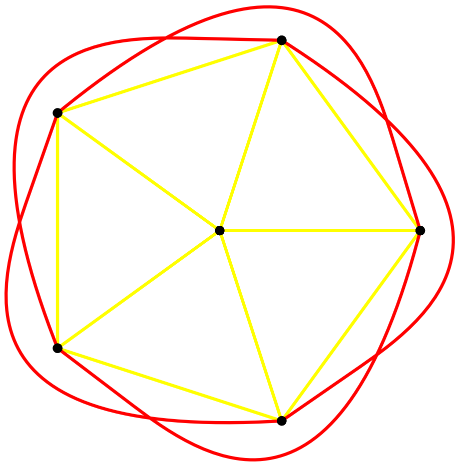


Figure 2.8: A simple drawing of K_6 with five 0-edges (denoted red) and five 2-edges (denoted yellow). In this drawing D, we have $E^2_{\leq 1}(D) = 5(1+1-0) = 5 \cdot 2 = 10 < 12 = 3\binom{1+3}{3}$.

2.6 Semi-pair-shellable drawings of K_n

In this section, we will show a potentially useful lemma for further progress and also a further generalization of seq-shellable drawings by Mutzel and Oettershagen [1]. We sketch their proof of the claim that the Harary–Hill conjecture is true for semi-pair-shellable drawings and we show that this proof is not correct. As we stated above, there are some drawings for which we can find a reference face, for which Claim 3 does not hold. To overcome this issue, we consider the triple cumulated edges instead of the double cumulated ones.

We can easily modify our previous proof based on (2.2) to use the triple cumulated edges. The modified equality (2.2) states

$$E^3_{\leq k}(D) = E^3_{\leq k-1}(D - v) + E^2_{\leq k}(D, D - v) + E^3_{\leq k}(v). \tag{2.3}$$

By Lemma 5 applied to the face F incident to the vertex v, the term $E^3_{\leq k}(v)$ is equal to $2\binom{k+3}{3}$ and the lower bound on $E^3_{\leq k-1}(D - v)$ is equal to $\binom{k+3}{4}$ by the induction argument from Claim 4 applied for all $k \leq \left\lfloor \frac{n}{2} \right\rfloor - 2$.

Now we will state a crucial definition from [1].

Definition 2.6.1. Let D be a simple drawing of K_n and let $v \in V$. Let $P_v = (u_0, \dots, u_{\lfloor \frac{n}{2} \rfloor - 2})$ be a sequence of distinct vertices from $V \setminus \{v\}$. We call P_v a *pair-sequence* of v if it satisfies the following two conditions

1. if n is odd then, for all even $j \in \{0, \dots, \left\lfloor \frac{n}{2} \right\rfloor - 2\}$, the vertex u_j is incident to a face $F \in \mathcal{F}(D - \{u_0, \dots, u_{j-1}\}, v)$. If $j + 1 \leq \left\lfloor \frac{n}{2} \right\rfloor - 2$, then u_{j+1} is incident to the superface of F in the drawing $D - \{u_0, \dots, u_j\}$.
2. If n is even, then u_0 is incident to a face $F' \in \mathcal{F}(D, v)$ and for all odd $j \in \{1, \dots, \left\lfloor \frac{n}{2} \right\rfloor - 2\}$, the vertex u_j is incident to the a face $F \in \mathcal{F}(D - \{u_0, \dots, u_{j-1}\}, v)$. If $j + 1 \leq \left\lfloor \frac{n}{2} \right\rfloor - 2$, then u_{j+1} is incident to the superface of F in the drawing $D - \{u_0, \dots, u_j\}$.

Definition 2.6.1 is written slightly differently than the one in [1], because Mutzel and Oettershagen [1] require u_{j+1} to be incident to some face even if there is no vertex u_{j+1}. Although the definition from [1] is more compact, we consider two cases, as our definition exactly states how the sequences should look like depending on the parity.

We can easily see that Definition 2.6.1 is a generalization of the simple sequence from Definition 2.2.3 for $k = \left\lfloor \frac{n}{2} \right\rfloor - 2$, where we can now change the face in "every second" step. We need to show that even if we change the face, there are enough invariant edges. The reason why we are considering changing the face is the following lemma.

Lemma 7. *Let D be a simple drawing of K_n with a reference face F and let $v \in V$ incident to F. For n odd, the number $E^2_{\leq \lfloor \frac{n}{2} \rfloor - 2}(D, D - v)$ of double cumulated invariant edges is the same with respect to any face $F \in \mathcal{F}(D, v)$ and the superface of F in $D - v$.*

Proof. We will consider equality (2.3) and we put the term with double cumulated invariant edges on the left hand side to obtain

$$E^2_{\leq \lfloor \frac{n}{2} \rfloor - 2}(D, D - v) = E^3_{\leq \lfloor \frac{n}{2} \rfloor - 2}(D) - E^3_{\leq \lfloor \frac{n}{2} \rfloor - 3}(D - v) - E^3_{\leq \lfloor \frac{n}{2} \rfloor - 2}(v). \tag{2.4}$$

Now we would like to argue that the right side of (2.4) remains the same for every face $F \in \mathcal{F}(D, v)$. By Corollary 1.2, the first term is the same for all faces $F \in \mathcal{F}(D)$ because it depends only on $\mathrm{cr}(D)$ which is determined by the drawing D since n is odd. The second term is the same through all faces $F \in \mathcal{F}(D, v)$ because after removing the vertex v each F is contained in the common superface $F(v)$. The last term is trivial, because we consider only faces around the vertex v where the term is equal to $2\binom{\lfloor \frac{n}{2} \rfloor + 1}{3}$. It means that all the terms remains the same for all faces in $\mathcal{F}(D, v)$. $\square$

This is where the motivation for Definition 2.6.1 of a pair-sequence came from. Mutzel and Oettershagen [1] tried to prove that a pair-sequence also guarantees enough invariant edges. We now sketch their proof and later we point out a mistake in their argument.

Lemma 8. *Let D be a simple drawing of K_n and let $v \in V$. If v has a pair-sequence with the reference face $F \in \mathcal{F}(D)$, then $E^2_{\leq \lfloor \frac{n}{2} \rfloor - 2}(D, D - v) \geq \binom{\lfloor \frac{n}{2} \rfloor + 1}{3}$.*

Proof. Let $k = \left\lfloor \frac{n}{2} \right\rfloor - 2$. We will consider n odd. For n even, the proof is analogous except we start by removing only a single vertex u_0.

As in the proofs of Claim 3 for the previous classes of drawings, we consider the vertex u_0 which contributes at least $\binom{k+2}{2}$ invariant edges because it is incident to the reference face F_0. The vertex u_1 contributes with at least $\binom{k+2-1}{2}$ to $E^2_{\leq k}(D, D - u_0)$ because every j-edge in $D - u_0$ is at most $(j+1)$-edge in D. Now, the number of vertices in $D - \{u_0, u_1\}$ is again odd, so we can apply Lemma 7 and change the face F_0, since the number $E^2_{\leq k-1}(D - \{u_0, u_1\}, D - \{u_0, u_1, v\})$ is the same for every reference face $F_1 \in F(D - \{u_0, u_1\}, v)$. After picking such a new reference face F_1, we count the invariant edges in the same way. Again, we pick two vertices u_i and u_{i+1} incident to F_1, as in the simple sequence, we note that there are at least $\binom{k+2-i}{2}$ in invariant edges at the vertex u_i and then we change the face again to F_2 and we repeat. Generally, after removing vertex u_i, where i is odd, we can change to face $F_{\frac{i+1}{2}}$ The lower bounds sum up to

$$\sum_{i=0}^{k} \binom{k+2-i}{2} = \binom{k+3}{3} = \binom{\left\lfloor \frac{n}{2} \right\rfloor + 1}{3}.$$

□

The definition of semi-pair-shellable drawings is then following.

Definition 2.6.2. Let D be a simple drawing of K_n for odd n with a reference face F. We call D an *semi-pair-shellable* drawing if there is vertex v which has pair-sequence and $D - v$ is seq-shellable with respect to the reference face F.

We can easily see that if the proof for pair-sequence works, then the Harary–Hill conjecture is also true for this class of drawings because we satisfy the conditions from Claim 4. In Figure 2.9 there is semi-pair-shellable drawing which is not seq-shellable. The only new step with respect to the proof using simple sequences was to change the face every “second” step. However, we claim that this step is not correct. We will now show three reasons why Lemma 7 cannot be used in this way. We focus on the first change of faces, namely F_0 to F_1 when n is odd. Nevertheless, the argument can be applied during a change of other faces analogously, and also for even n. Now we will discuss the three reasons in detail.

Consider the first change to F_1. After this change, we would like to make estimates on invariant edges for vertices u_2 and u_3. The estimates are based on the fact that after removing the vertices u_0, u_1 and u_0, u_1, u_2, the vertices u_2 and u_3, respectively, are incident to the reference face F_1. Therefore in the drawings $D - \{u_0, u_1\}$ and $D - \{u_0, u_1, u_2\}$ we have 0-, …, $(k-1)$- and then $(k-1)$-, …, 0-edges by Lemma 5. So our lower bound on the contribution of the vertex u_i to $E^2_{\leq k}(D, D - v)$ is $\binom{k+2-i}{2}$ because we know that after removing i vertices, the j-value of an edge can be reduced by at most i.

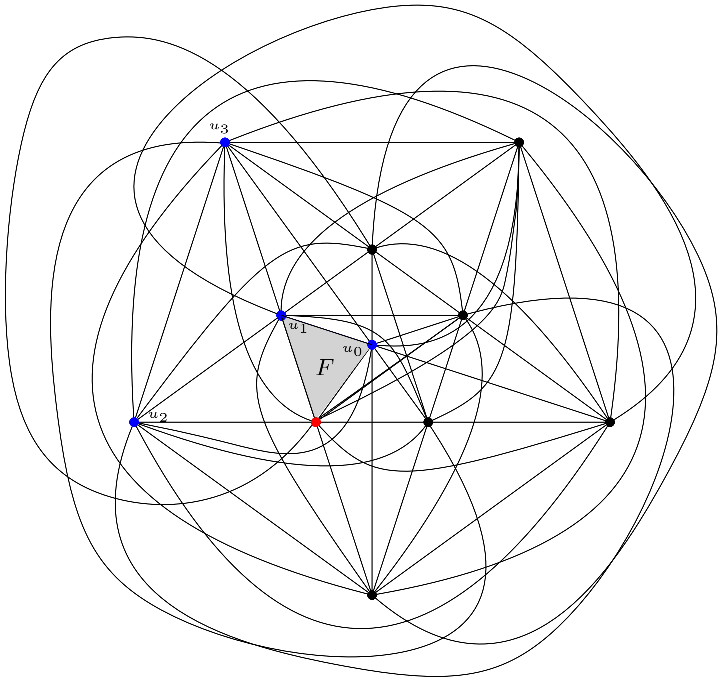


Figure 2.9: Semi-pair-shellable drawing of K_{11} that is not seq-shellable with v having pair-sequence (u_0, u_1, u_2, u_3).

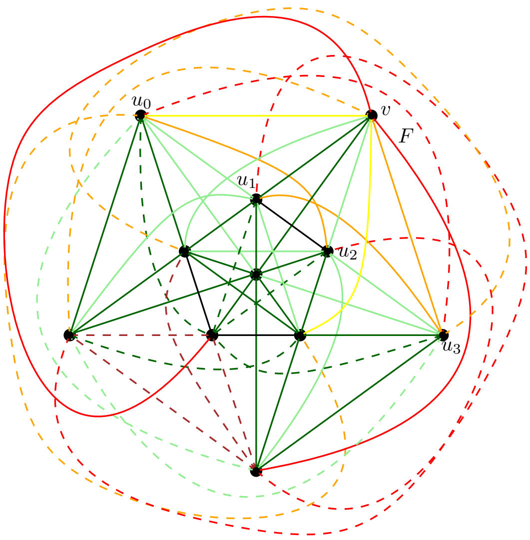
1. However, the argument for changing the face is that the number $E^2_{\leq k-1}(D - \{u_0, u_1\}, D - \{u_0, u_1, v\})$ is the same for all faces in $\mathcal{F}(D - \{u_0, u_1\}, v)$. Nevertheless, there can be some $(k-1)$-edges in $D - \{u_0, u_1\}$ which are $(k+1)$-edges in D, we call them *bad*, and therefore are not counted in $E^2_{\leq k}(D, D - v)$. This is an issue because although the value $E^2_{\leq k-1}(D - \{u_0, u_1\}, D - \{u_0, u_1, v\})$ remains the same, we know that the invariant edges at vertices u_0, u_1 could contribute more to $E^2_{\leq k}(D, D - v)$ for the reference face F_0 than for the reference face F_1. If there are more bad edges for the face F_0 than for F_1 we can, in fact, consider more invariant edges contained in $E^2_{\leq k}(D, D - v)$ than there truly are if we count them in a more precise way. Otherwise it just not indicative value for changing the face.

   First, we denote by $E^2_{\leq k}(D, D - v)(u)$ the number of double cumulated invariant edges in D with respect to v that are incident to u. To state the more precise way of counting the invariant edges, we rewrite the number $E^2_{\leq k}(D, D - v)$ of double cumulated invariant edges inductively to obtain the following equality similar to (2.2):

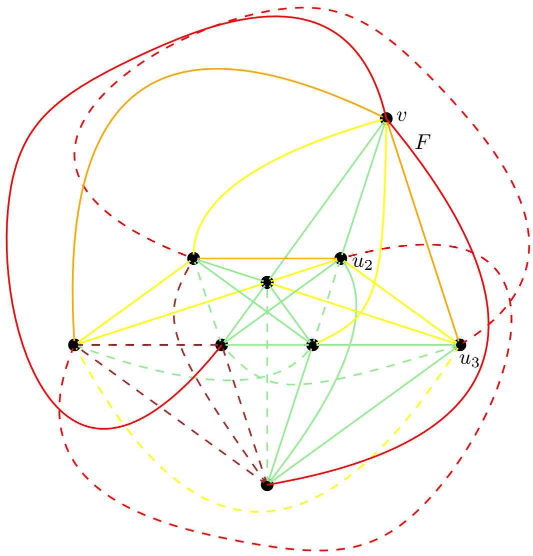
$$\begin{aligned} E^2_{\leq k}(D, D - v) = E^2_{\leq k-1}(D - u, D - \{u, v\}) + \\ + E_{\leq k}(D, D - \{u, v\}) + E^2_{\leq k}(D, D - v)(u). \end{aligned} \tag{2.5}$$

   The first term on the right side is just the number of invariant double cumulated $(k-1)$-edges in $D - u$ with respect to v because we know that every edge contained in $E^2_{\leq k-1}(D - u, D - \{u, v\})$ contributes with at least the same value to $E^2_{\leq k-1}(D - u, D - \{u, v\})$ as to $E^2_{\leq k}(D, D - v)$. The first and the last term are usually used in the proofs to give a lower bound on the left-hand side. Now, assume that we are also able to count the middle term $E_{\leq k}(D, D - \{u, v\})$ representing the invariant edges with respect to both u and v. Those are the edges that contribute to $E^2_{\leq k}(D, D - v)$ with one more than to $E^2_{\leq k-1}(D - u, D - \{u, v\})$. This term is usually hard to count. However, in the following counter-example, it can contradict the proof of Lemma 8.

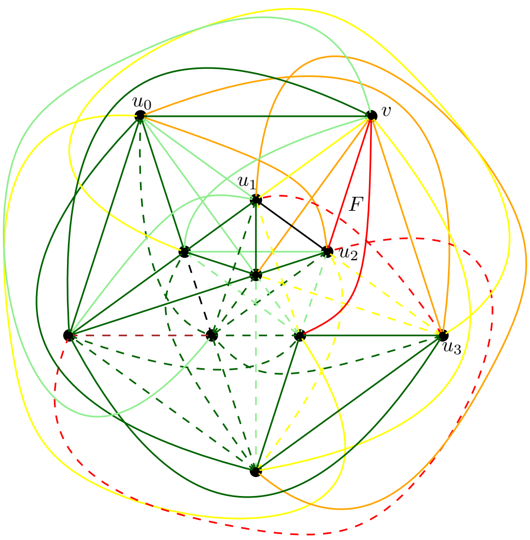
In Figure 2.10, we can see a drawing of K_{11} with $k = 3$. The dashed edges are invariant with respect to v. The bad edges are brown. The 0-, 1-, 2-, 3-, 4-edges are colored red, orange, yellow, light green, dark green, respectively.



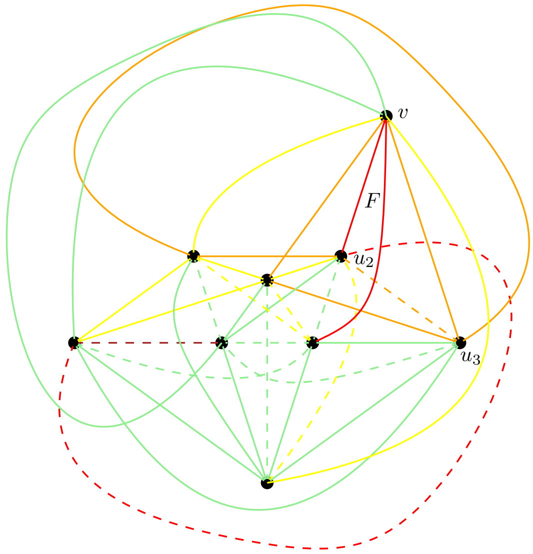
(a) The drawing of K_{11} with the face F_0 before removing vertices u_0, u_1.



(b) The drawing of K_{11} with the face F_0 after removing vertices u_0, u_1.



(c) The drawing of K_{11} with the face F_1 before removing vertices u_0, u_1.



(d) The drawing of K_{11} with the face F_1 after removing vertices u_0, u_1.

Figure 2.10: A counter-example for changing the reference face in semi-pair-shellable drawings.

We can see, that the contribution to $E^2_{\leq k}(D, D - v)$ for edges incident to u_0 and u_1 is greater for F_0 than for F_1. Then after removing those vertices, there are more bad edges for F_0 than for F_1; see Subfigures 2.10b and 2.10d. This means that we are considering more invariant edges that remain invariant even in D in the term $E^2_{\leq k-1}(D - \{u_0, u_1\}, D - \{u_0, u_1, v\})$ for the reference face F_1 than for F_0.

Now we want to have a look at the invariant $\leq k$-edges in D. Specifically, the invariant edges incident to u_0, u_1 with respect to the face F_0 are 0-, 1-, 3-edge and 0-edge, respectively. The remaining invariant edges for v are 0-, 1-, 3-edge and therefore $E^2_{\leq k}(D) = 20$; see Subfigure 2.10a. Then, after the change to F_1 the invariant edges incident to u_0, u_1 are none and 0-, 2-edge, respectively, and the remaining invariant edges are 0-, 2-, 2-, 2-, 3-, 3-, 3-, 3-edge; see Subfigure 2.10c. Thus, we get more invariant edges with respect to F_1 in the drawing $D - \{u_0, u_1\}$ even though the term $E^2_{\leq k-1}(D - \{u_0, u_1\}, D - \{u_0, u_1, v\})$ remains the same.

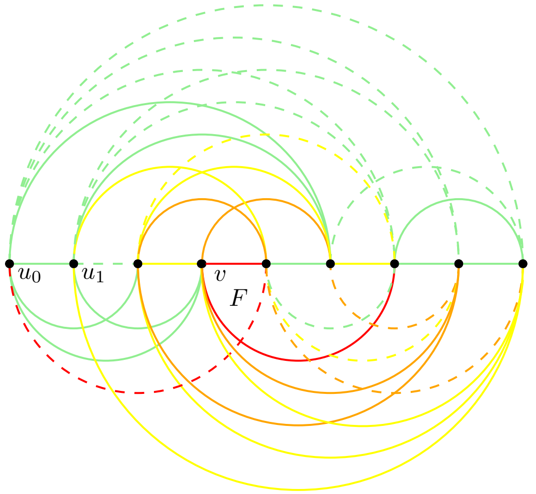
If we count invariant edges incident to u_0, u_1 for the reference face F_0 and for u_2 and u_3 for the reference face F_1, then the term $E^2_{\leq k}(D, D - v)$ sums up to $(4+3+1)+(4)+(4+2+2+1)+(2) = 22 > 20$, where each bracket represents the contribution of vertices u_0, u_1, u_2, u_3 when we do not omit the middle term of equality (2.5).

2. In the proofs with the simple sequences or their similar variants, we considered at least $(k - i + 1)$ invariant edges after removing i vertices. Therefore, the value we really cared about was $E_{\leq k-i}(D - \{u_0, \ldots, u_{i-1}\}, D - \{u_0, \ldots, u_{i-1}, v\})$ because we knew that these edges are also invariant in D with respect to v. We could also consider all $\leq k$-edges in D which are invariant in D and also in $D - \{u_0, \ldots, u_{i-1}\}$ as in equality (2.5) but as we said before, this is usually hard to count.

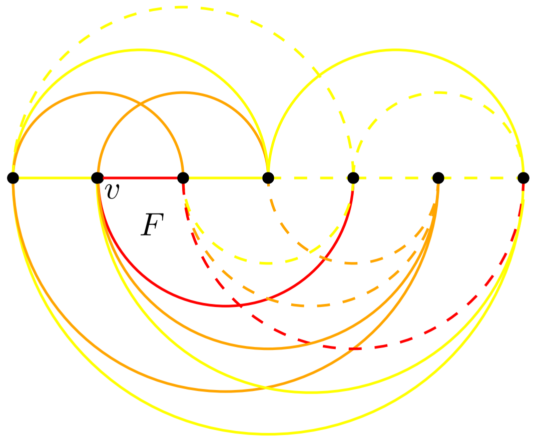
   As we mentioned in the previous paragraph, if we knew that the number $E_{\leq k-2}(D - \{u_0, u_1\}, D - \{u_0, u_1, v\})$ remains the same, everything would work fine. However, when only the term $E^2_{\leq k-1}(D - \{u_0, u_1\}, D - \{u_0, u_1, v\})$ remains the same, we can redistribute the values after the change so that we can overestimate the value $E^2_{\leq k-2}(D - \{u_0, u_1\}, D - \{u_0, u_1, v\})$, which is done in the proof of Lemma 8. In Figure 2.11, there is K_9 ($k = 2$) where we overestimate the term $E^2_{\leq k-2}(D - \{u_0, u_1\}, D - \{u_0, u_1\} - v)$ after changing the face from F_0 to F_1. For the reference face F_0, see Subfigure 2.11a, is the term $E^2_{\leq k-2}(D - \{u_0, u_1\}, D - \{u_0, u_1\} - v) = 1$ and for the reference face F_1 we get the term $E^2_{\leq k-2}(D - \{u_0, u_1\}, D - \{u_0, u_1\} - v) = 2$.

3. The last problem is, in fact, a combination of the two previous ones. Consider a situation when we redistribute the edges in $E^2_{\leq k-1}(D - \{u_0, u_1\}, D - \{u_0, u_1, v\})$ so that we overestimate the term $E^2_{\leq k-2}(D - \{u_0, u_1\}, D - \{u_0, u_1, v\})$ as in part 2, so we can count invariant edges around some vertex incident to F_1 more easily. After the redistribution, there are less $(k-1)$-edges in $D - \{u_0, u_1\}$. This means that, before the change of face, there are even more $(k-1)$-edges and more of them are bad (as in part 1) than after the change, so we overestimate the term $E^2_{\leq k}(D, D - v)$ for the reference face F_0.

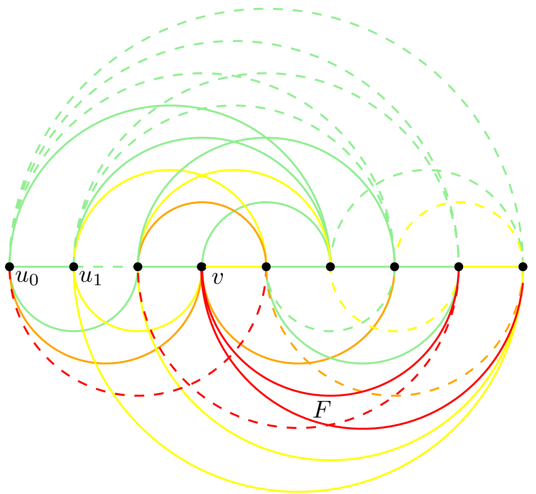
   As we said before, we usually omit the middle term in equality (2.5) when estimating the number of double cumulated invariant edges by simple se-



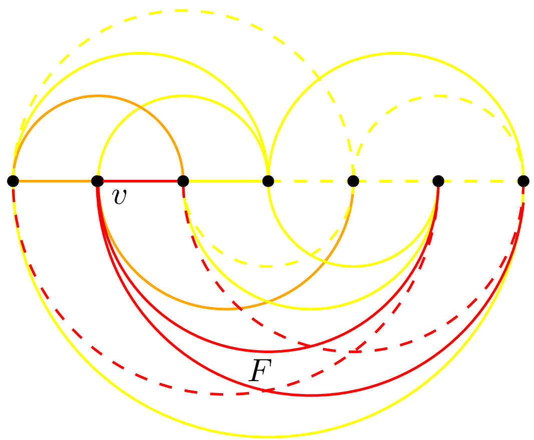
(a) The drawing of K_9 with the face F_0 before removing vertices u_0, u_1.



(b) The drawing of K_9 with the face F_0 after removing vertices u_0, u_1.



(c) The drawing of K_9 with the face F_1 before removing vertices u_0, u_1.



(d) The drawing of K_9 with the face F_1 after removing vertices u_0, u_1.

Figure 2.11: A counter-example on the redistribution of invariant edges during the first change of a reference face in semi-pair-shellable drawings.

quences. Therefore, we have not found a counter-example which gives a higher value of $E^2_{\leq k}(D, D-v)$ just by counting redistributed edges counted in $E^2_{\leq k-2}(D\{u_0, u_1\}, D - \{u_0, u_1, v\})$ yet but we believe that there is such a counter-example for larger drawings.

Thus, the proof for pair-sequences for odd n is incorrect. The problem remains the same for even n, but only for the second and later changes of the face. The difference is that when we consider a pair-sequence for n even, we can change the face after removing $\{u_0, \dots, u_{j-1}\}$ for j odd. In other words, we can change the face first after removing only the vertex u_0. The term $E_{\leq k-1}(D - u_0, D - \{u_0, v\})$ should be the same so that we can change the face and everything works as for previous classes of drawings. However, $E_{\leq k-1}(D - u_0, D - \{u_0, v\})$ is the same as the term in Lemma 7 for the drawing $D - u_0$, because then $n-1$ is odd and therefore $E_{\leq \lfloor \frac{n-1}{2} \rfloor -2}(D - u_0, D - \{u_0, v\}) = E_{\leq k-1}(D - u_0, D - \{u_0, v\})$.

3. Generating simple drawings

In this chapter, we describe our programs for generating databases of simple drawings of K_n for small values of n and their visualization. We describe the high-level idea of how they work and then we show some results obtained as an application of our programs. We implemented three programs. First, we have a *checker*, which finds rotation systems that can be realized as a simple drawing. The second program is a *generator*, which constructs simple drawings determined by the realizable rotation systems that are produced by the checker. Finally, we have a *visualizer*, which is used to visualize the drawings produced by the generator and which captures the structure of k-edges of a given drawing. The ideas used to implement the checker and the generator are based on the approach of Pammer [16].

3.1 Checker of realizable drawings

Since there are infinitely many simple drawings of K_n, we encode them to distinguish only finitely many classes of drawings so that the combinatorial properties of all drawings of the same class are the same. In order to describe the structure of simple drawings of K_n we consider so-called rotation systems and fingerprints. In particular, we distinguish the drawings only up to weak-isomorphism.

Definition 3.1.1. Two simple drawings D and D' of K_n are *weakly isomorphic* if there is a bijection f from $V(D)$ to $V(D')$ such that two edges ab and cd in $E(D)$ cross if and only if the two corresponding edges $f(a)f(b)$ and $f(c)f(d)$ cross in D'.

We can see that $\mathrm{cr}(D)$ and $\mathrm{cr}(D')$ are the same for weakly isomorphic D and D'. Therefore, there are indistinguishable from the point of view of the Harary–Hill conjecture. This is why it is sufficient for us to consider the simple drawings up to weak isomorphism.

3.1.1 Creating rotation systems and fingerprints

Now we will look at rotations of vertices which are crucial in describing the weak isomorphism of two drawings.

Definition 3.1.2. Let D be a simple drawing of K_n. The *rotation* of a vertex v of D is a counter-clockwise cyclic order of edges incident to v. A rotation $\rho(v)$ of v is represented by a cyclic sequence of vertices incident to v. The *rotation system* is then the set of rotations of all vertices in D.

The checker is based on the following result whose proof can be found in [16].

Claim 9. *Two simple drawings of K_n with the same or inverse rotation system are weakly isomorphic. Also, any two weakly isomorphic simple drawings of K_n have either the same or inverse rotation system.*

Thus, if we want to check the Harary–Hill conjecture by computer, we only need to go through all rotation systems and there are only finitely many of them. Instead of going through all the rotation systems of a drawing, we can pick some canonic one. Since a rotation is a cyclic sequence, we can choose, for example, the lexicographically minimal rotation $\rho_{min}(v)$ at every vertex v. In other words, we consider a so-called *fingerprint* $FP_{RS} = (\rho_{min}(0), \rho_{min}(1), \dots, \rho_{min}(n-1))$ where rotation $\rho_{min}(v)$ is lexicographically minimal.

This way, we reduce the number of rotation systems we need to generate. However, there are still many possible relabelings of a given fingerprint. Therefore, we choose the lexicographically minimal one to be our canonic fingerprint CFP_{RS} of FP_{RS}. We can always relabel the vertices so that $\rho_{min}(0)$ is equal to $1 \cdots (n-1)$. We also know that if we are able to do that, then every canonic fingerprint starts with the rotation $\rho_{min}(0)$ of the vertex 0. Sometimes in the literature, the fingerprint is denoted only by $FP_{RS} = (\rho_{min}(1), \dots, \rho_{min}(n-1))$. On the other hand, when we want to create the inverse rotation system and also relabel it, we also need to invert and relabel $\rho_{min}(0)$. Therefore, we consider the canonic fingerprint to be $CFP_{RS} = (\rho_{min}(0), \rho_{min}(1), \dots, \rho_{min}(n-1))$ although we know that $\rho_{min}(0)$ is determined uniquely.

We say that a rotation system ρ of K_n is *realizable* if there is a simple drawing of K_n with the rotation system ρ. We generate realizable rotation systems inductively. If we have all realizable (canonic) fingerprints of K_{n-1}, we can create all realizable fingerprints of K_n by adding the new vertex $n-1$ on every position in each realizable fingerprint of K_{n-1}. This is because the subdrawing of K_n on vertices $0, \dots, (n-2)$ determines a realizable fingerprint of K_{n-1}. We create a new potentially realizable fingerprint, by adding the vertex $n-1$ on some positions in all rotations $\rho(0), \dots, \rho(n-2)$ and then we create $\rho(n-1)$ arbitrarily.

Then, because we could create some of the rotation systems many times, we find canonical fingerprints by trying all relabelings of vertices. In other words, we try only those n relabelings that produce the rotation $1 \cdots (n-1)$. Then we also invert the fingerprint and also try all relabelings producing the rotation $1 \cdots (n-1)$, since we know by Claim 9 that drawings with inverse rotation systems are weakly isomorphic and they are the same from our point of view. In this way, we can find all potentially realizable fingerprints, and then we can proceed with checking the realizability of a fingerprint.

3.1.2 Checking realizability of fingerprints

In this subsection, we describe a method how to check whether a given rotation system is realizable. We combine the method by Pammer [16] with results by Richter and Sullivan [17] to speed up the computations.

Constructing part

To check whether a fingerprint of K_n is realizable, we will try to construct a simple drawing of K_n with such a fingerprint. Each simple drawing D is stored in the *half-edge structure*. In this structure, every edge of D is represented by pairs of oriented edges, where each pair is formed by opposite edges. This is because we want to be able to easily move over an edge of D when creating D by iteratively adding edges of K_n. Each edge contains an information about **next**, **previous**,

opposite edges, the end points, vertex **to** and **from** and lastly **face** incident to it. A vertex u contains an information consisting of a single **edge** (v, u). A face contains an information consisting of a single **edge** incident to it; see Figure 3.1.

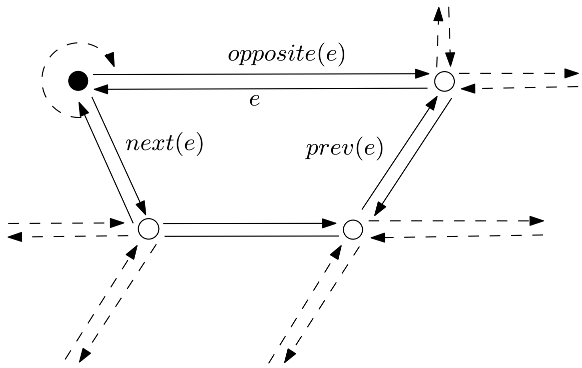
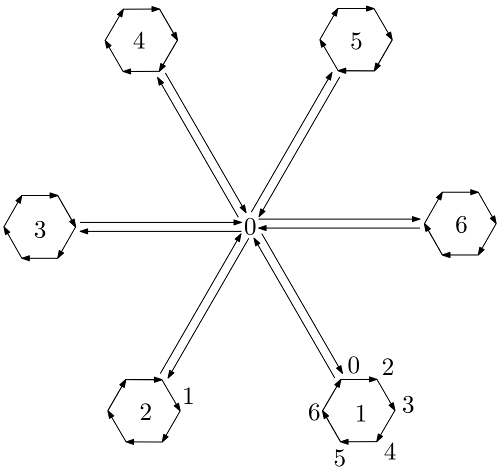
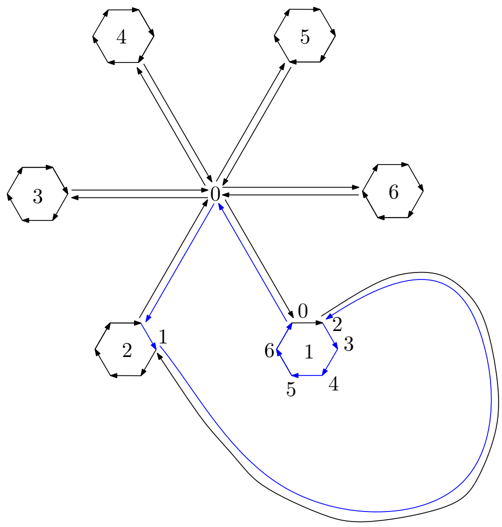


Figure 3.1: The half-edge data structure.

The half-edge structure helps us to easily represent the drawing D. Our goal is to create a drawing with a given fingerprint if it is realizable. Therefore we need to force the edges of D to respect the rotation of a vertex of D. In order to preserve the order, we represent every vertex v of D by a circle of dummy $n-1$ vertices. Every dummy vertex contains the label determining which edge from $\{v0, \ldots, v(n-1)\} \setminus \{vv\}$ it is attached to; see Figure 3.2a. Since the drawing D is simple, any of the edges cannot cross. In other words, the rotation around v agrees with the given fingerprint.



(a) A base star.



(b) A base star with next edge between vertices one and two.

Figure 3.2

We consider three arrays/stacks to store the information about which edge should go where. The stack **segments[index]** stores all the edges. It is a stack because we proceed by recursion, and therefore pushing and popping edges to/from the top is $\mathcal{O}(1)$ operation. The array **starts[a][b]** contains the information on which index in segments the edge determining the starting point of edge ab of D is. The last array is four-dimensional **blocked[a][b][c][d]** for checking whether edges ab and cd intersect. The values for edges that are on the circle representing vertices are set to true because these are special cases that we do not want to intersect anytime. Also, two edges sharing an endpoint are considered intersecting because we consider only simple drawings of K_n.

We start with a *base star* which contains all edges incident to the vertex 0; see Figure 3.2. Then we recursively try all possible ways how the edge 12 can be drawn until we find the first realizable drawing of the edge 12. Due to the heuristics we use and describe in the paragraph *Heuristic part*, we first want to create K_{n-1} and then pull all the edges incident to the vertex $n-1$. Since the base star creates the edge $0(n-1)$, all the edges incident to $n-1$ are meant except of the edge $0(n-1)$; see Figure 3.3. The decisions on lines 6, 9, 12, 15, and 18 of Algorithm 2 create edges of K_{n-1} and then the edges incident to $n-1$ as we wanted. In contrast, Pammer [16] generates the edges incident to the vertices 1, 2, and so on first. His code is then little bit simpler, but the heuristics cannot be applied that easily.

Algorithm 1 Initialization followed by realization of a fingerprint.

```
create_base_star()
find_the_way(starts[1][2], starts[2][1], 1, 2)
```

Heuristic part

To make the process of checking the realizability of K_n faster, we modified the method by Pammer [16] using some results by Richter and Sullivan [17]. The resulting heuristic slightly improved the running time of our program. First, we use the following claim.

Claim 10 ([17])**.** *Let D be a simple drawing of K_n with $n \geq 3$ and let C be the simple closed curve bounding a face of a simple drawing of K_n. If e is any closed edge of K_n then $e \cap C$ is either connected or consists of the two vertices incident to e.*

Due to Claim 10 we know that when we already realized a drawing of K_{n-1} any edge e incident to the vertex $n-1$ can be either connected or it consists of the two vertices incident to e. Therefore $e \cap C$ has to be already connected or consisting of the endpoints for any closed curve C of a face of K_{n-1}, because of the simplicity of the drawing. Therefore if e already has two disjoint parts (not only endpoints) on some closed curve C of any face in D then we can drop the next calling of **find_the_way** function and immediately step back.

The second result from [17] that we use is the following one.

Claim 11 ([17])**.** *Let D be a simple drawing of K_n with $n \geq 4$ and let C be the simple closed curve bounding any face of D. If e_1, e_2, e_3 are distinct (open) edges of D incident to a common vertex v of D then at least one of the intersections $C \cap e_1$, $C \cap e_2$, and $C \cap e_3$ is empty.*

Algorithm 2 find_the_way(s, t, a, b)

```
function FIND_THE_WAY(s, t, a, b)
    seg ← segments[s] →next_
    while seg ≠ segments[s] do
        if seg = segments[t] then
            add_edge(segments[s]→from_,segments[t]→from_, a, b)
            if b < n − 2 then
                find_the_way((starts[a][b + 1], starts[b + 1][a], a, b + 1)
                if (done) then return
            else if (b = n − 2) and (a < n − 3) then
                find_the_way(starts[a+1][a+2], starts[a+2][a+1], a+1, a+2)
                if (done) then return
            else if (b = n − 2) and (a = n − 3) then
                find_the_way(starts[1][b + 1], starts[b + 1][1], 1, b + 1)
                if (done) then return
            else if (b = n − 1) and (a < n − 2) then
                find_the_way(starts[a + 1][b], starts[b][a + 1], a + 1, b);
                if (done) then return
            else
                realized ← realized + 1
                done ← true
                return
            delete_edge_back()
        first_end ← seg → first_end
        second_end ← seg → second_end
        if !blocked[a][b][first_end][second_end] then
            blocked[a][b][first_end][second_end] ← true
            intersect(-)
            find_the_way(segments[s] → opposite_, t, a, b)
            if done then
                blocked[a][b][first_end][second_end] ← false
                return
            undo_intersect(-)
            blocked[a][b][first_end][second_end] ← false
        seg ← seg → next_
```

Similarly as before, we know that if we have already realized K_{n-1}, then any edge e incident to the last vertex $n-1$ cannot be on the boundary of the face where two other edges incident to $n-1$ already are, because the drawing we are creating is simple. After realizing that we found "a wrong face" we can again immediately step back.

Pre-detection part

As we build fingerprints by induction based on the realizable fingerprints of K_{n-1} we also first check whether all induced copies of K_4 are realizable because we want to reveal a non-realizable fingerprint as soon as possible to speed up the computation. The detection of a non-realizable induced K_4 is based on the following fact. We can consider $\rho(i) = (j, k, l)$ of a vertex i of a K_4 induced by vertices i, j, k, l and we rotate $\rho(i)$ so that $j = \min\{j, k, l\}$. If $k > l$, we call $\rho(i)$ *negative*, and *positive* otherwise. It can be shown by a simple case analysis that the rotation system of K_4 is realizable if and only if the number of negative rotations is even. This can also be done by checking whether it is one of the fingerprints of K_4 that we have from the induction. This approach can also be generalized to arbitrary induced drawings of K_m with $m \leq n-1$. However, the time complexity is then $\Omega(n^m)$. Therefore we consider only $m = 4$ to reveal "the most obvious" non-realizable fingerprints.

The checker is divided into t threads. We divide the file with fingerprints for K_{n-1} into t files with approximately the same number of lines. Then we run t almost separate threads, except that the threads share a dictionary, which consists of all already checked fingerprints, to check every fingerprint once.

3.2 Generator of drawings

The *generator* part of the program takes already realizable fingerprints of K_n. The backbone is similar to the one in the checker part. In other words, we try to construct a realization of a given fingerprint. There are some technical parts about generating coordinates, but this is described in the attached programmer documentation in more detail. Nevertheless, we now give a high-level overview of these parts.

We need to create coordinates for each vertex. Since the vertex is represented by a circle, we need to decide whether to have coordinates for each element or to have one for the whole circle and identify them during the generating of the realization. We applied the second approach.

The next technical step is creating an edge. More precisely, creating one part of the edge between vertices or intersections a and b so that it does not cross any other edge. When we want to create such an edge segment going only inside the given face F, we first triangulate F using a triangulation T. Then we build the graph where the vertices are the midpoints of the sides of triangles in T as well as a and b. For each triangle t of T, we connect all midpoint of the edges of t. If one of the vertices of t is either a or b then a or b is also connected to the midpoints. Then we find the shortest path from a to b using the Dijkstra algorithm. We know that since T is triangulation, the newly created edge segment stays in the face F. We used library mapbox/earcut.hpp [18] for triangulation.

Collinear points cause some troubles in our implementation of the triangulation. In that case, our triangulation algorithm does not consider the middle point in this collinear configuration as the vertex on the face F. For example, if the middle point is one of a or b, our algorithm for creating such an edge segment fails.

In order to avoid this situation, we consider two approaches. When we are creating the line, we find some segment which we want to go through; see Algorithm 2. This segment is divided into two halves and then we try to connect the vertex or the intersection a with this new intersection b. In this situation, we have clearly created three collinear points. In order for the triangulation algorithm to consider all three points, we move the intersection in both directions perpendicular to the line b it lies on. This solves our problem.

The second approach is whenever the collinear points are created during pulling the edge segment through the face F, then the middle one is considered redundant and therefore it is deleted.

When the face F is the outer one, we have no boundary for the triangulation, so an auxiliary boundary is created. In our case, we create a quadrilateral, which is far away from the drawing itself.

For this generating, as well as for checking, we use the multithreading approach. When we have n threads, we divide the file with fingerprints into files with approximately the same number of lines. Then we consider n independent threads and each of them generates coordinates for the corresponding file. This distributing computing led to a significant improvement of the running time, the generator is faster up to n times.

For each realizable fingerprint of K_n, checked by the checker part, the generator prints out $\binom{n}{2}$ lines, where every line represents the coordinates of one edge. The coordinates of vertices are considered as the endpoints of the edges.

3.3 Visualizer of drawings

The *visualizer* has three conceptual modes in which we can use them. First, we introduce the structure that is common for the whole program. For each drawing, we create a list of vertices, edges, and a dictionary that represents the neighborhood. Since each edge created by the generator is composed of line segments and can be quite long and twisted, we first try to make the realization nicer using a force-directed algorithm.

3.3.1 Force-directed algorithm

We try to redraw the realized drawing by a force-directed algorithm with crossing preserving properties introduced by Bertault [19]. The crossing preserving property is crucial because we need to preserve rotations and also the weak-isomorphism class. To see the difference between the drawings before and after the redrawing algorithm has been applied, see Figure 3.4.

Counting forces

Let D be a drawing of a graph. Let $(x(v), y(v))$ denote the coordinates of a vertex v and let $d(a,b)$ be the Euclidean distance between vertices a and b. The parameter $\delta > 0$ is the optimal length of edges. In each iteration, the algorithm computes a force $F(v) \in \mathbb{R}^2$ for every vertex v and then moves v in the direction $F(v)$. The movement is restricted so that the crossings are preserved.

We consider three kinds of forces to redraw the drawing D. First, the attraction force between the vertices connected by an edge, second, the repulsion force between all pairs of vertices, and lastly, the repulsion force between edges and vertices.

The force applied to a vertex v is $F(v) = (F_x(v), F_y(v))$. The attraction and repulsion forces of the vertex v, when we consider the other vertex u fixed, are

$$F_x^a(u,v) = \frac{d(u,v)}{\delta}(x(u) - x(v)) \quad \text{and} \quad F_x^r(u,v) = \frac{-\delta^2}{d(u,v)^2}(x(u) - x(v)).$$

To count the repulsion force $F^e(v, (a,b))$, we consider the orthogonal projection i_v of the vertex v onto the line (a,b). The force is applied to the vertex v if the projection i_v lies inside the segment (a,b) and if the distance between v and i_v is smaller than parameter γ. For the vertex v and an edge (a,b) disjoint with v, we have

$$F_x^e(v,(a,b)) = \begin{cases} -\dfrac{(y - d(v,i_v))^2}{d(v,i_v)}(x(i_v) - x(v)) & \text{if } i_v \in (a,b), d(i,i_v) < \gamma \\ 0 & \text{otherwise.} \end{cases}$$

The overall force applied to the vertex v is

$$F_x(v) = \sum_{(u,v)\in E} F_x^a(u,v) + \sum_{u\in V} F_x^e(u,v) + \sum_{(a,b)\in E} F_x^e(v,(a,b)) - \sum_{\substack{u\in V, w\in W,\\ (v,w)\in E}} F_x^e(u,(v,w)).$$

We compute the coordinate $F_y(v)$ analogously.

Counting zones

We know that it is crucial to preserve the rotation systems and therefore crossing properties so we need to restrict the area where every vertex is allowed to move. Bertault [19] introduced a *zone* $Z(v)$ related to the vertex v. The zone $Z(v)$ consists of eight circular sectors $Z_0(v), \ldots, Z_7(v)$ with radii $R_0(v), \ldots, R_7(v)$, respectively. The zone $Z(v)$ is the area in which v is allowed to move to preserve the crossing properties.

To count the radii $R_i(v)$, we consider one vertex v and one edge (a,b) at time. We distinguish two cases depending on whether the orthogonal projection i_v lies inside or outside the line segment (a,b).

Case 1: If i_v lies on a line segment (a,b). We consider the sector $Z_s(v)$ which contains the ray (v, i_v). Then we update the values by letting

$$
\begin{aligned}
R_i(v) &= \min\left(R_i(v), \frac{d(v,i_v)}{3}\right), & i &= r(s-2),\ldots,r(s+2),\\
R_i(a) &= \min\left(R_i(a), \frac{d(v,i_v)}{3}\right), & i &= r(s+2),\ldots,r(s+6),\\
R_i(b) &= \min\left(R_i(b), \frac{d(v,i_v)}{3}\right), & i &= r(s+2),\ldots,r(s+6),
\end{aligned}
$$

where $r(j) = j \pmod 8$.

Case 2: If i_v does not lie on a line segment (a,b) we update the values as follows

$$
\begin{aligned}
R_i(v) &= \min\left(R_i(v), \frac{\min(d(a,v),d(b,v))}{3}\right), & i &= 0,\ldots,7\\
R_i(a) &= \min\left(R_i(a), \frac{d(a,v)}{3}\right), & i &= 0,\ldots,7\\
R_i(b) &= \min\left(R_i(b), \frac{d(b,v)}{3}\right), & i &= 0,\ldots,7.
\end{aligned}
$$

The correctness and the time complexity of the algorithm are described in [19]. As we can see in Figure 3.4, the algorithm helps to see the structure of drawings a lot better.

3.3.2 Modes

The visualizer has three conceptual modes. The first one, based on the generator part, reads the coordinates from the input files and transforms the stored representation into the structure we described at the beginning of Section 3.3. The visualizer then subdivides the edges, tries to redraw the realized drawing by the force-directed algorithm described above, and then shortens edges that are too long. The whole algorithm produces drawings with colored k-edges so that we can see their structure; see Figure 3.4. The second mode allows the user to produce his own drawings. The user can add vertices or edges in the drawings already created by the generator. The third mode serves to verify our conjectures. The visualizer is described in more detail in Chapter 4.

3.4 Applications

Balko, Fulek, and Kynčl [11] came with the following conjecture.

Conjecture 12. *([11]) Let $k \geq 0$ be an integer and let D be a simple drawing of a graph with at least $\binom{2k+3}{2}$ edges. Then*

$$E^3_{\leq k} \geq 3\binom{k+4}{4}$$

with respect to all the faces of D as the reference faces.

Conjecture 12 is a generalized version of Claim 4 even for K_n because it considers every k up to $\left\lfloor \frac{n}{2} \right\rfloor - 1$ or $\left\lfloor \frac{n}{2} \right\rfloor - 2$ depending on the parity. We verified this conjecture for all simple drawings D of K_n up to weak isomorphism and for all faces of D for $n \leq 7$. It would be also better to verify the conjecture for all simple drawings up to (only) isomorphism because there can be some configuration of faces we have not captured.

Although we have seen before that Claim 3 does not have to hold for all faces of given drawing D of K_n (see Figure 2.8), we still think that it holds for at least one face of D. Therefore, we state the following conjecture.

Conjecture 13. *Let $k \geq 0$ be an integer and let D be a simple drawing of a graph with at least $\binom{2k+3}{2}$ edges. Then*

$$E^2_{\leq k} \geq 3\binom{k+3}{3}$$

for at least one face of D as the reference face.

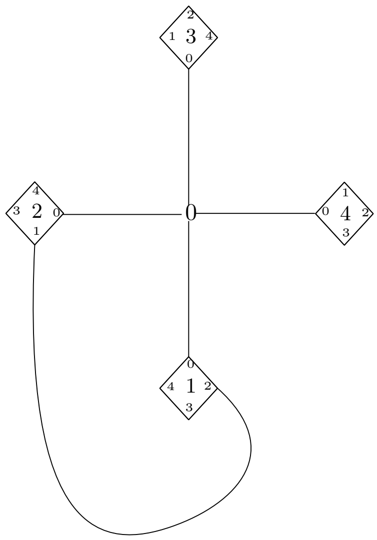
Similarly, we verified Conjecture 13 for all simple drawings of K_n up to weak isomorphism for $n \leq 7$. As we mentioned before, it would be better to verify the conjecture for all simple drawings up (only) isomorphism. It also seems that there is only a small number of faces that do not satisfy Claim 3. We are currently trying to verify these conjectures also for $n = 8$ but the computation have not finished yet.

We also used the visualizer to see the structure of the drawings and to find a concrete counter-example to the argument by Mutzel and Oettershagen [1]; see Figures 2.10 and 2.11.

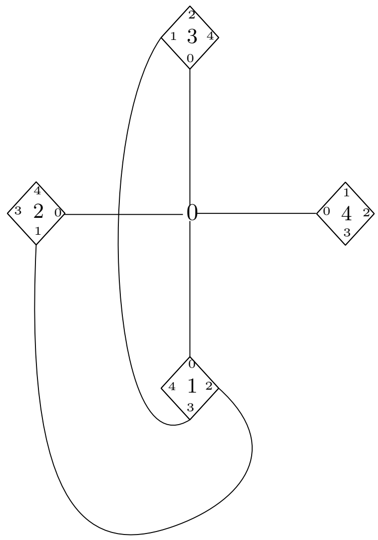
We also used the database of simple drawings to know the numbers of simple drawings of K_n up to weak isomorphism. The numbers we obtained match the numbers by Pammer [16]; see Table 3.1.

n	the number of non-weakly isomorphic drawings of K_n
3	1
4	2
5	5
6	102
7	11556
8	5370725
9	running

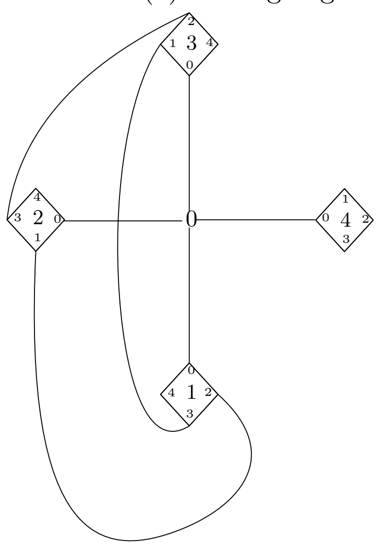
Table 3.1: The numbers of simple drawings of K_n with $3 \leq n \leq 8$ up to weak isomorphism.



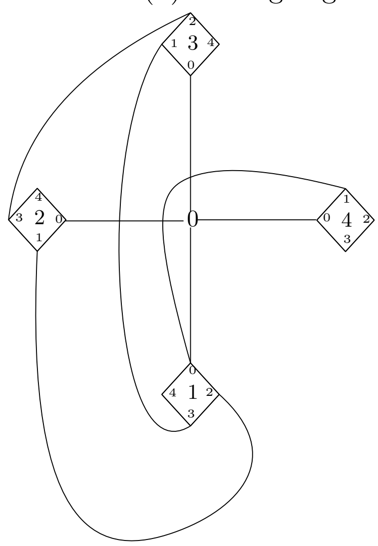
(a) Adding edge 12.



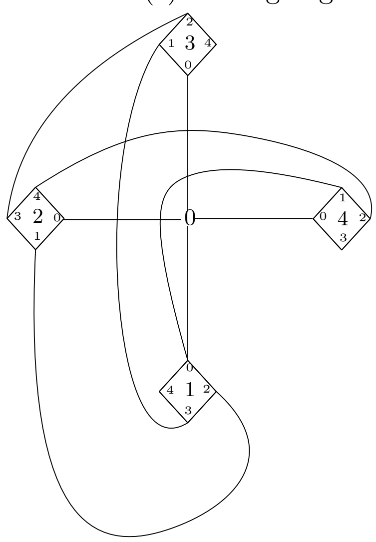
(b) Adding edge 13.



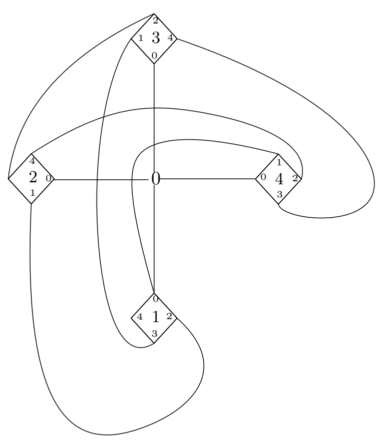
(c) Adding edge 14.



(d) Adding edge 23.

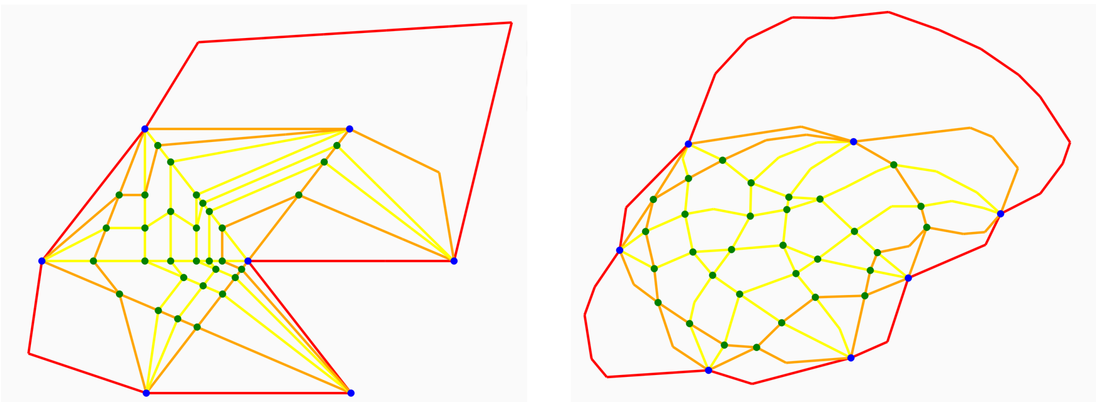


(e) Adding edge 24.



(f) Adding edge 34.

Figure 3.3: Whole process of adding edges of K_5 drawing when base star is prepared.



(a) Before redrawing (b) After redrawing

Figure 3.4: A drawing of K_7 before and after redrawing using force-directed algorithm.

4. User's guide

As we mentioned in Chapter 3, the program consists of three parts: the checker, the generator, and the visualizer. The first two parts mainly serve as a basis for the visualizer, which is the main application for users. In this chapter, we describe the visualizer from the user's point of view. The programmer documentation can be found in the file generated by the code, which is contained among attachments of the thesis and it is also recommended to read `README.md` file in order to install everything necessary. The visualizer is designed for Windows.

The visualizer is displayed in one window that contains the canvas on the left, where the drawing D can be drawn, and the buttons and the values of $E_k(D)$, $E_{\leq k}(D)$, $E^2_{\leq k}(D)$, $E^3_{\leq k}(D)$ on the right; see Figure 4.1.

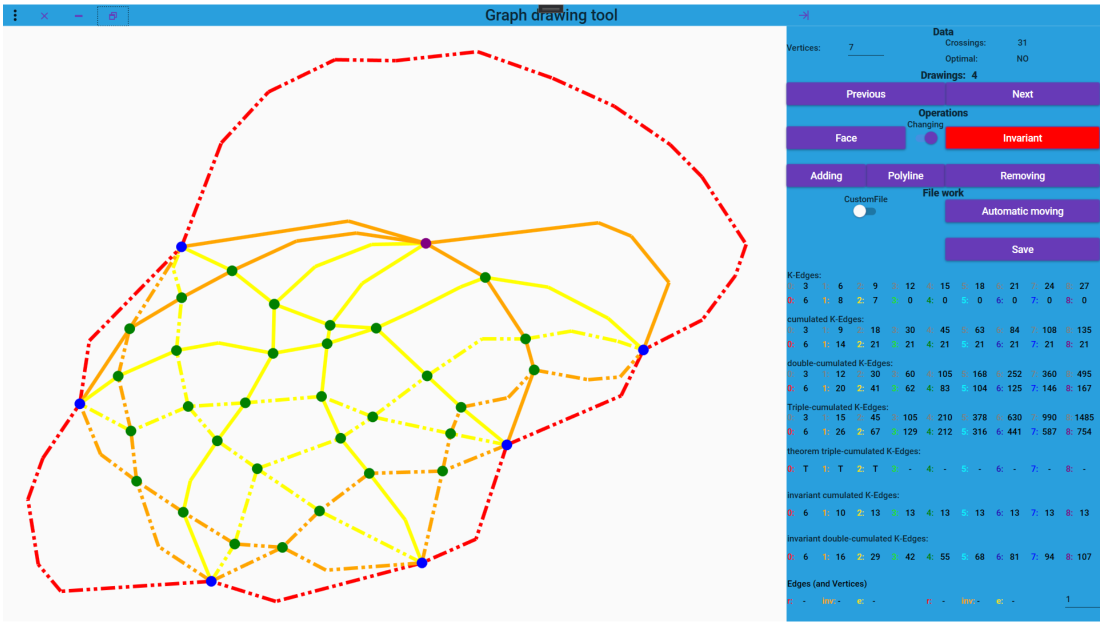


Figure 4.1: The design of the visualizer.

4.1 Canvas

Now we will have a more detailed look at the visualizer. We will start with the canvas; see Figure 4.2. The canvas contains the drawing that we generated or created manually. The vertices are colored blue, the intersections are colored green, and the vertices for which we count invariant edges are colored purple. The edges are colored depending on their k value. The color legend will be discussed in Section 4.4. Finally, the invariant edges are the dashed ones.

4.2 Top bar

The top bar is divided into three parts, left, middle, and right.

- **The left part** consists of four buttons; see Figure 4.3a. After clicking on the first one, you can set the **smoothing constant** on the first line, which

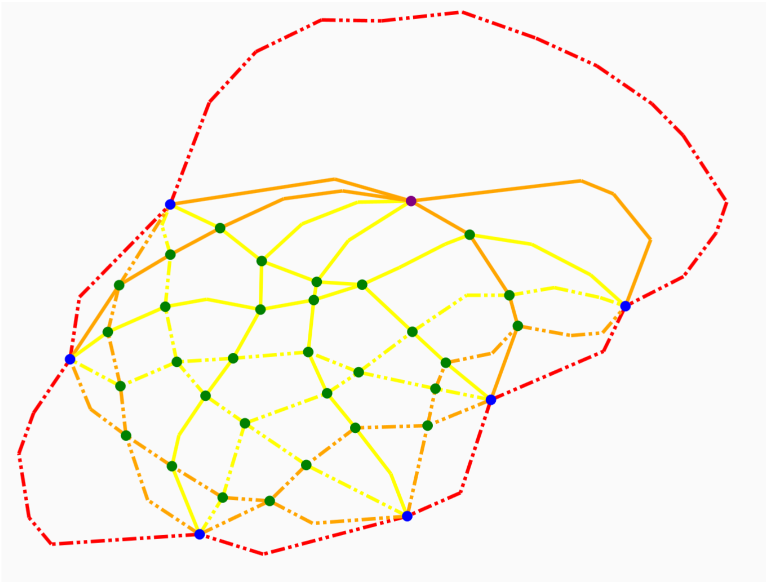


Figure 4.2: The canvas with a drawing of K_7.

determines the number of iterations of the redrawing force-directed algorithm. There is a checker that accepts only non-negative integers. When the text of the smoothing textbox is empty, the smoothing constant is set to zero, and the default value is set to 10. The second line is for setting any decimal number d to the **automatic moving constant** which allows you to see d drawings per second. When there is either an invalid text or an empty text, the value is set 1.0, which is also the default value.

To the right of the first button, there are the **close** button, the **minimize** button, and the **resize** button, which passes between two states. First, when the window is maximized, and second when the window is set to the "normal" size.

- **The middle part** is the title of the program.
- **The right part** of the top bar contains either the **close menu** button, see Figure 4.3b, or the **open menu** button, see Figure 4.3c, depending on whether the menu is opened or closed. These buttons are here mainly to enlarge the canvas so the drawing is better visible.

(a) Top left corner - Setting long-term constants.

(b) Opened menu with the close menu button in the top left corner.

(c) Closed menu with the open menu button in the top left corner.

Figure 4.3: Parts of the top bar.

4.3 Controlling part

Now we describe the *controlling* part. The controlling part is divided horizontally into four blocks: the **data**, **drawings**, **operations**, and **file work**; see Figure 4.4.

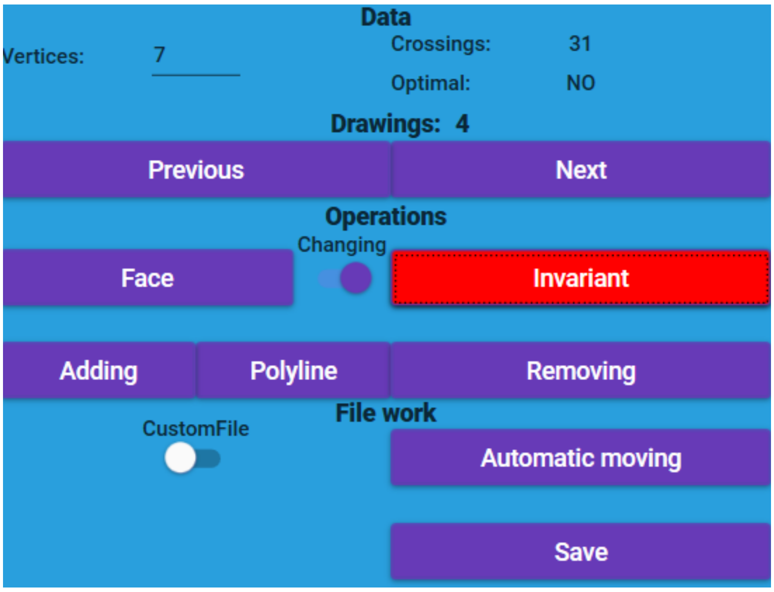


Figure 4.4: Controlling part divided into four blocks.

4.3.1 Data

The data part contains a textbox on the left, where we can set the size of the complete graph we would like to see. In Figure 4.4, we can see a drawing of K_7. The textbox is limited to accept only sizes of the already generated drawings. In other words, we can write only numbers from 4 to 8 there. The default value is four. On the right, there are two lines showing the number of crossings of our drawing and the word YES if the current drawing contains exactly $Z(n)$ crossings and NO otherwise.

4.3.2 Drawings

The drawings part starts with the title, followed by the number of already displayed drawings of K_n. The next line consists of two buttons that generate the previous and the next drawing. When there is no previous drawing, a message box shows up. When there is no next drawing, the empty canvas appears so that we can draw our own drawings on it.

4.3.3 Operations

The following part, titled **operations**, allows using various operations on the drawings or on the empty canvas.

- The first line contains the **face** button, which serves to change a reference face when the button is activated. We can click on any free place of the canvas and the face that contains the selected point becomes the reference face. The default reference face is the outer one.

- The **changing toggle** button on the right of the face button is by default set to the checked state. This means that when we want to display the next or the previous drawing, the reference face is again set to the outer face. When the toggle button is not checked, then every time we display the next drawing, the face containing the point we selected last time is considered to be the reference face. The main advantage of this toggle button is when you have many similar consecutive drawings and you want to find out the changes in the k-edges values.

- The **invariant** button, which is activated in Figure 4.4, is used to detect the invariant edges with respect to some set of vertices on the canvas. We can pick and unpick any vertex on the canvas. After picking it turns purple. If the edge is invariant with respect to chosen (purple) vertices, it is dashed, if it changes the k-value by one then it stays the same, and if it changes the k-value by two it becomes dotted.

- The next line consists of three buttons. The first two are **adding** and **adding polyline** buttons. The adding button creates a new vertex on the canvas by clicking on any white place of the canvas. Also, when we click on two already created vertices, it creates a line segment between them. With the adding polyline button, we can create an edge consisting of many consecutive line segments by clicking on some vertex, then by clicking anywhere on the white canvas, and lastly by clicking on another vertex. The edges are created by connecting the selected points by line segments. We cannot add two edges between some pair of vertices, because we consider only simple drawings. Therefore when this happens, a message box shows up.

- The remaining button of the operations part is the **remove** button, which serves for removing an edge, a vertex with all the edges incident to it, or an intersection with its two incident edges, when the button is activated.

The common part is that only one of the adding, adding polyline, removing, and invariant buttons can be chosen at one time. When you try to pick another one, the one chosen before will be deactivated. Also, as you can see in Figure 4.4, when any of the control buttons is activated it becomes red.

4.3.4 File work

The last part consists only of three buttons and eases the custom work.

- The first **custom file toggle** button can switch the states, if we want to visualize drawings generated by the generator or if we want to visualize the drawings created by ourselves and saved in a file. If the button is checked, we work with the custom file. Otherwise, we visualize the generator's data. The default value is unchecked.

- To the right of the custom file toggle button, there is the **automatic moving** button. When the button is activated, the visualizer starts to display as many drawings per second as the frequency is set by automatic moving constant. The redrawing takes some time on bigger graphs so the frequency can become smaller.
- The **save** button serves only to save the current drawing into a file. After changing the state to the custom one we can see our custom drawings in the order of how we saved them.

4.4 The value part

The value part consists of eight horizontal blocks which contain values of k-edges, invariant k-edges, and their cumulations; see Figure 4.5.

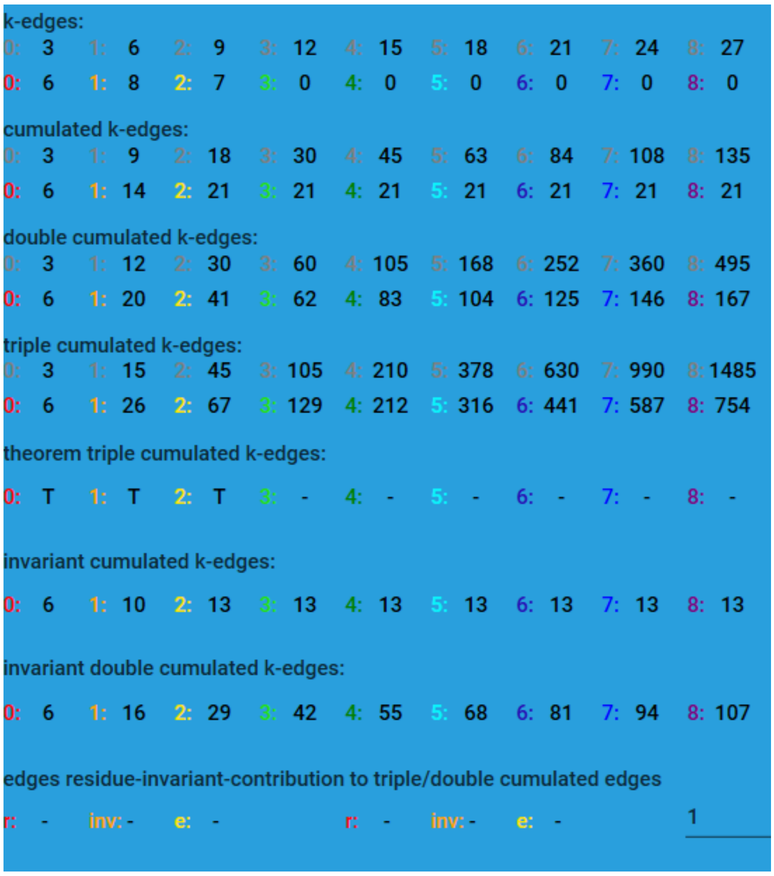


Figure 4.5: The value part consisting of eight horizontal blocks.

- The first four blocks are the same. They always contain the title (***k*-edges, cumulated *k*-edges, double cumulated *k*-edges and triple cumulated *k*-edges**). Then there is a line with the values from Claims 2, 3, 4 and a line with the analogous values for k-edges.

  The following values are the ones we have in the current drawing. The colors of the numbers $0, 1, \ldots, 8$ correspond with the colors we used for the k-edges on the canvas.

- The fifth block titled **triple cumulated theorem** checks whether the drawing on the canvas satisfies Conjecture 12 for a given k. There are also blank characters $-$ for values that exceed the threshold from Conjecture 12.
- The following two lines are the true values of the **invariant cumulated and double cumulated k-edges** for the drawing on the canvas. The last line shows the three terms from the equalities (2.3) and (2.2) modified for removing an edge (which we selected after the invariant button was activated) instead of a vertex v. The textbox in the right bottom corner contains the number k (set to 1 by default) determining which k is considered in the equalities (2.3) and (2.2). The textbox accepts only numbers between 0 to 8 as we only count the values of k-edges for $k \in \{0, \ldots, 8\}$.

There are only nine columns in each of eight blocks of value part because we thought it is enough to have that many k-values. Nevertheless, we can always enlarge the controlling panel or make some moving windows to see more values.

5. Conclusion

In this thesis, we surveyed the recent progress and the recent proofs of the Harary–Hill conjecture for restricted classes of drawings of K_n. We used simple sequences (Definition 2.2.3) to make the exposition simpler and more intuitive. We showed that Mutzel and Oettershagen [1] used an incorrect argument in Lemma 8 to extend the class of drawings for which the Harary–Hill conjecture is true.

We tried to generalize the proof by induction to prove the conjecture for a richer class of drawings. However, we think that it cannot be generalized without a deeper understanding of the inner structure of the drawings. The issue is that when we use the equality (2.3) to estimate the left hand side by induction, we estimate all three terms separately. However, the third term is determined uniquely because we choose v to be incident to F. This means that we have a degree of freedom only in two of these three terms on the right side.

Now, consider $k = \left\lfloor \frac{n}{2} \right\rfloor - 2$ for n odd and a simple drawings D of K_n with $Z(n)$ crossings. We know that having $Z(n)$ crossings implies that $E^3_{\leq k}(D) = \binom{k+4}{4}$ by Corollary 1.2 for n odd. On the other hand, we could choose v and F so that the drawing $D - v$ is not optimal after removing v. This is because Proposition 1 does not hold when we want to apply the step from even $n-1$ to n odd. Therefore, there has to be some subdrawing $D - v$ that is not optimal when $\mathrm{cr}(D) = Z(n)$.

This also means that the sum $E^3_{\leq k}(D - v) + E^3_{\leq k-1}(D - v)$ is greater than $Z(n)$ due to Corollary 1.2 for n even. Therefore, either $E^3_{\leq k}(D - v) > \binom{k+3}{4}$ or $E^3_{\leq k-1}(D - v) > \binom{k+2}{4}$. Now, consider the case $E^3_{\leq k-1}(D - v) > \binom{k+2}{4}$. Then we know, from the equality (2.3), where the term first term is exactly $E^3_{\leq k-1}(D - v)$, that the number $E^2_{\leq k}(D, D - v)$ of invariant double cumulated edges is less than $\binom{k+3}{3}$. In other words, we cannot use the induction in such a straightforward way. We have tried removing all vertices from known odd optimal drawings shown in Figures 1.2, 1.3, and 1.4. However, these types of drawings make only the first term $E^3_{\leq k}(D - v)$ in the sum $E^3_{\leq k}(D - v) + E^3_{\leq k-1}(D - v)$ greater. Therefore, we have not found an example with the second term larger yet, although we think there should be some. We have been finding some new classes of optimal drawings, but for all of them only the first term was greater than it should be. We also know that both terms $E^3_{\leq k}(D - v), E^3_{\leq k-1}(D - v)$ are the same for all faces $F \in \mathcal{F}(D, v)$ because they result in common superface $F(v)$.

We also think that Conjectures 12 and 13 hold, but we have not found any good direction how to prove them.

Our program for generating simple drawings is still running, so we are still extending our database of realizable fingerprints and our database of coordinates for these realizable fingerprints. We also think that the visualizer helps in understanding of the structure of simple drawings. We are using it to count the number of double and triple cumulated edges. Additionally, we verify Conjectures 12 and 13 and we think about generalization also for all simple drawings only up to isomorphism. If the reader wants to add some new features to our visualizer, do not hesitate to write us an email and we will consider it.

Bibliography

[1] Petra Mutzel and Lutz Oettershagen. The crossing number of semi-pair-shellable drawings of complete graphs. In *CCCG*, 2018.

[2] A. Bogomolny. 3 utilities puzzle. http://www.cut-the-knot.org/do_you_know/3Utilities.shtml.

[3] L. Beineke and R. Wilson. The early history of the brick factory problem. *Math. Intelligencer*, 32(2):41–48, 2010.

[4] S. Pan and R. B. Richter. The crossing number of K_{11} is 100. *J. Graph Theory*, 56(2):128–134, 2007.

[5] B. M. Ábrego, O. Aichholzer, S. Fernández-Merchant, T. Hackl, J. Pammer, A. Pilz, P. Ramos, G. Salazar, and B. Vogtenhuber. All good drawings of small complete graphs. In *Proc. 31st European Workshop on Computational Geometry (EuroCG)*, pages 57–60, 2015.

[6] J. Blažek and M. Koman. A minimal problem concerning complete plane graphs. In *Theory of Graphs and its Applications (Proc. Sympos. Smolenice, 1963)*, pages 113–117. Publ. House Czechoslovak Acad. Sci., Prague, 1964.

[7] B. Ábrego, O. Aichholzer, S. Fernández-Merchant, P. Ramos, and B. Vogtenhuber. Non-shellable drawings of kn with few crossings. In *Proc. 26th Annual Canadian Conference on Computational Geometry CCCG 2014*, pages online–only, 2014.

[8] R. K. Guy. Crossing numbers of graphs. In *Graph theory and applications (Proc. Conf., Western Michigan Univ., Kalamazoo, Mich., 1972; dedicated to the memory of J. W. T. Youngs)*, pages 111–124, 1972.

[9] B. M. Ábrego, O. Aichholzer, S. Fernández-Merchant, P. Ramos, and G. Salazar. The 2-page crossing number of K_n. *Discrete Comput. Geom.*, 49(4):747–777, 2013.

[10] B. M. Ábrego, O. Aichholzer, S. Fernández-Merchant, P. Ramos, and G. Salazar. Shellable drawings and the cylindrical crossing number of K_n. *Discrete Comput. Geom.*, 52(4):743–753, 2014.

[11] M. Balko, M. Fulek, and J. Kynčl. Crossing numbers and combinatorial characterization of monotone drawings of K_n. *Discrete Comput. Geom.*, 53(1):107–143, 2015.

[12] B. M. Ábrego, O. Aichholzer, S. Fernández-Merchant, D. McQuillan, B. Mohar, P. Mutzel, P. Ramos, R. B. Richter, and B. Vogtenhuber. Bishellable drawings of K_n. *SIAM J. Discrete Math.*, 32(4):2482–2492, 2018.

[13] P. Mutzel and L. Oettershagen. The crossing number of seq-shellable drawings of complete graphs. In *Combinatorial algorithms*, volume 10979 of *Lecture Notes in Comput. Sci.*, pages 273–284. Springer, Cham, 2018.

[14] P. Erdős, L. Lovász, A. Simmons, and E. G. Straus. Dissection graphs of planar point sets. In *A survey of combinatorial theory (Proc. Internat. Sympos., Colorado State Univ., Fort Collins, Colo., 1971)*, pages 139–149, 1973.

[15] L. Lovász, K. Vesztergombi, U. Wagner, and E. Welzl. Convex quadrilaterals and k-sets. In *Towards a theory of geometric graphs*, volume 342 of *Contemp. Math.*, pages 139–148. Amer. Math. Soc., Providence, RI, 2004.

[16] J. Pammer. Rotation systems and good drawings. Master's thesis, Institute for Software Technology Graz University of Technology, Austria, 2014.

[17] R. B. Richter and M. Sullivan. Remarks on the structure of simple drawings of K_n, 2020.

[18] Mapbox. `https://github.com/mapbox/earcut.hpp`, 2020.

[19] F. Bertault. A force-directed algorithm that preserves edge crossing properties. In *Graph drawing (Štiřín Castle, 1999)*, volume 1731 of *Lecture Notes in Comput. Sci.*, pages 351–358. Springer, Berlin, 1999.

A. Attachments

In addition to the files mentioned below, every folder also contains files generated by the Visual studio and a file `documentation.html` and documentation contained in a folder `html`. We do not include them in the list of attachments. First, we recommend to read `README.md` in order to install everything what is needed.

A.1 drawing_of_cliques

This is a folder containing files to check the fingerprints:

- `drawing_of_cliques.cpp` - main program running a parallel computation for checking fingerprints,
- `function.hpp` - library containing all functions.

Then there is a folder containing fingerprints for different sizes of graphs.

- `graph3.txt`
- `graph4.txt`
- `graph5.txt`
- `graph6.txt`
- `graph7.txt`

A.1.1 Tests

A folder containing files for testing the half-edge structure and for working with the fingerprints.

- `pch.h` and `pch.cpp` - header and main file for running tests
- `test_adding_deleting_edges.cpp`
- `test_all_special_vertices.cpp`
- `test_canonic_fingerprint.cpp`
- `test_graph_properties.cpp`
- `test_special_vertex.cpp`
- `test_vertices_changes.cpp`

A.2 coordinates_generator

A folder containing files for generating the realizable fingerprints.

- `drawings_of_cliques.cpp` - main program running a parallel computations for generating realizable fingerprint,
- `functions.hpp` - library containing all functions except of triangulations,
- `triangulation.hpp` - library for triangulating polygons.

And also again folder `data` containing fingerprints for different sizes of graphs.

- `graph3.txt`
- `graph4.txt`
- `graph5.txt`
- `graph6.txt`
- `graph7.txt`
- `graph8.txt`

A.3 VisualizerWPF

A folder containing all files for the visualizer.

- `App.xaml`
- `App.xaml.cs`
- `AssemblyInfo.cs`
- `CollisionDetection.cs`
- `CustomMath.cs`
- `Edge.cs`
- `EdgeListExtensions.cs`
- `ForceDirectedAlgorithm.cs`
- `GraphCoordinates.cs`
- `GraphGenerator.cs`
- `HashSetExtension.cs`
- `MainWindow.xaml`
- `MainWindow.xaml.cs`

- `PointExtensions.cs`
- `Vertex.cs`

Then the folder `data` containing coordinates for different sizes of graphs.

- `graph3.txt`
- `graph4.txt`
- `graph5.txt`
- `graph6.txt`
- `graph7.txt`
- `savedGraphs.txt` - file containing custom drawings
- `savedGraphBackUp.txt` - an auxiliary file for temporary storing of custom drawings

A.4 ConjectureChecker

This folder contains the files for verifying Conjectures 12 and 13.

- `App.xaml`
- `App.xaml.cs`
- `AssemblyInfo.cs`
- `CollisionDetection.cs`
- `ConjectureChecker.cs`
- `CustomMath.cs`
- `Edge.cs`
- `EdgeListExtensions.cs`
- `ForceDirectedAlgorithm.cs`
- `GraphCoordinates.cs`
- `GraphGenerator.cs`
- `HashSetExtension.cs`
- `PointExtensions.cs`
- `Vertex.cs`

The folder `data` contains the coordinates for different sizes of graphs.

- `graph3.txt`
- `graph4.txt`
- `graph5.txt`
- `graph6.txt`
- `graph7.txt`
- `savedGraphs.txt` - file containing custom drawings
- `savedGraphBackUp.txt` - an auxiliary file for temporary storing of custom