Adrian Fuchs, BSc.

On the number of monochromatic crossings in rectilinear embeddings of complete graphs

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium Mathematics

eingereicht an der

Technischen Universität Graz

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

Institut für Softwaretechnologie
8010 Graz, Inffeldgasse 16b/II

Graz, August 2019

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Abstract

This thesis considers the following combinatorial optimization problem in discrete geometry:

Given a set of finitely many points in the plane in general position, we want to draw line segments with one of two colors between each two of those points, such that the number of monochromatic crossings, i.e., the number of line segment intersections that are of the same color, is minimized.

In this thesis, several approaches will be discussed including an upper bound heuristics by local optimization and a lower bound by linear programming. By this approach, the problem has been solved for up to ten points in the plane by computer.

Zusammenfassung

In dieser Masterarbeit behandeln wir das folgende kombinatorische Optimierungsproblem aus der diskreten Geometrie:

Gegeben sei eine endliche Menge von Punkten in allgemeiner Lage in der Ebene. Wir wollen die Strecken zwischen je zwei dieser Punkte in einer von zwei Farben färben, sodass die Anzahl der monochromatischen Kreuzungen, also die Anzahl der Kreuzungen von Strecken in der gleichen Farbe, minimal ist.

In dieser Arbeit werden einige Ansätze betrachtet, z.B. wie man mit lokaler Optimierung eine obere Schranke findet, als auch eine untere Schranke, die man mittels linearen Programmen erhält. Mit den erwähnten Methoden wurde das Problem für alle Punktmengen in allgemeiner Lage bis zu zehn Punkten mit Computerunterstützung gelöst.

Motivation

Combinatorial optimization is one of the youngest and most active fields in discrete mathematics. It is related to the topics of graph theory, complexity theory and integer programming ([21], preface to the first English edition). The field of discrete geometry became popular due to the Hungarian mathematicians László Fejes Tóth, who formed the term “Intuitive Geometry”, by which he means geometry that “can be explained to and can appeal to the man of the street.” [19], and Paul Erdős, as well as the Canadian mathematician Harold Scott MacDonald Coxeter [26] who combined the theory of polytopes and of non-Euclidean geometry with group theory and combinatorics. Discrete geometry studies combinatorial geometric problems such as properties of finite point sets, triangulations, polytopes, drawings and many more in a topological space, typically in $\mathbb{R}^2$ and $\mathbb{R}^3$.

The motivation for the main problem that is discussed in this thesis comes from the intention to understand how a set of finitely many points in the plane behaves geometrically. A very basic idea is to connect all pairs of points by line segments. By this connection, there might appear crossings of line segments. An obvious question is: How many crossings do we expect from a point set? How do the crossings depend on the point set? How can we maximize or minimize the number of crossings?

It is easy to see that if the points lie in convex position, i.e., they are corners of their convex hull, all subsets of four points induce a crossing and hence the crossing number is maximized. However, it turns out that minimizing the number of crossings is not so easy. For up to 27 points, an exact minimum is known. For more points, we know a lower bound on the crossing number, but apart from the case of 30 points, we do not know a pointset that attains this bound, see [3].

The main question in this thesis goes one step further. We are not asking for the total number of crossings, but for the number of crossings that are in the same color class if we allow to choose one of two colors for each line segment. Again, we see that maximizing is easy. All we need to do is choose the same color for each line segment. So minimizing is the interesting task. Before asking for how to choose a point set for a minimal number of same-colored crossing line segments, we have to ask how to choose the colors. This problem is not trivial as well.

Contents

1 Basic definitions and notations

First a remark concerning structural notation: In this thesis, for enumerated paragraphs with title (such as Definition, Lemma, Theorem, etc.) a bar on the left side indicates their scope.

For a rigorous discussion of the problem of the number of monochromatic crossings in rectilinear embeddings of complete graphs, we need some basic definitions first.

Notation 1.1. There are some notations that are not uniform throughout mathematical literature. In this thesis, we will consistently use the following notations and symbols with the meaning as given in Table 1.

$\mathbb{N}$	The set of positive integers $\mathbb{N} = \{1, 2, \ldots\}$. We will distinguish $\mathbb{N}$ from the set of nonnegative integers $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$
$A \vee B$, $A \wedge B$	For A and B Boolean variables or statements, $A \vee B$ means A *or* B and $A \wedge B$ means A *and* B
$S \subset T$	For S and T sets, $S \subset T$ means S is a proper subset of T, i.e., $S \neq T$
$S \subseteq T$	For S and T sets, $S \subseteq T$ means $S \subset T \vee S = T$.
$S \dot{\cup} T$	For S and disjoint T sets, i.e., $S \cap T = \emptyset$, $S \dot{\cup} T$ denotes the union $S \cup T$. For non-disjoint sets, it is not defined. This notation emphasizes the disjointness of S and T.
$[a, b]$	The closed real interval $\{x \in \mathbb{R} : a \leq x \leq b\}$
$]a, b[$	The open real interval $\{x \in \mathbb{R} : a < x < b\}$
$\lceil x \rceil$, $\lfloor x \rfloor$	For $x \in \mathbb{R}$, $\lceil x \rceil = \min(\{a \in \mathbb{Z} : x \leq a\}$ denotes the *ceiling* of x and $\lfloor x \rfloor = \max(\{a \in \mathbb{Z} : x \geq a\}$ denotes the *floor* of x.
id_S	The identity map on a set S. $\mathrm{id}_S : S \to S, s \mapsto s$
$f\vert_{S'}$	If S and T are sets, $f : S \to T$ is a function, $S' \subseteq S$, then $f\vert_{S'}$ is the map $S' \to T$ defined by $f\vert_{S'}(s) = f(s)$ for all $s \in S'$.
$\vert S\vert$	For a set S, $\vert S\vert \in \mathbb{N}_0 \cup \{\infty\}$ denotes the number of elements of the set S, while for $z \in \mathbb{C}$, $\vert z\vert$ denotes the absolute value of z.
$\mathcal{P}(S)$	For a set S, $\mathcal{P}(S)$ denotes the power set of S, i.e., the set of all subsets of S.
$\binom{S}{k}$	For a set S and $k \in \mathbb{N}$, $\binom{S}{k}$ denotes the set of all k-element subsets of S, i.e., $\binom{S}{k} = \{S' \subseteq S \text{ such that } \vert S'\vert = k\}$. For $\vert S\vert < \infty$, this notation is consistent with the binomial coefficient $\vert\binom{S}{k}\vert = \binom{\vert S\vert}{k} = \frac{\vert S\vert!}{k!(\vert S\vert-k)!}$.
P	The complexity class of polynomially solvable combinatorial optimization or decision problems.
NP	The complexity class of non-deterministically polynomially solvable combinatorial optimization or decision problems.

$\mathcal{O}(f(x))$	Landau notation: For a function $f : \mathbb{D} \to \mathbb{R}^+$ with $\mathbb{D} \in \{\mathbb{R}^+, \mathbb{N}\}$, the symbol $\mathcal{O}(f(x))$ describes the class of functions whose limiting behavior is bounded by f up to a constant, i.e., $\mathcal{O}(f(x)) = \{g : \mathbb{D} \to \mathbb{R}^+$ with $\exists c \in \mathbb{R}^+$ such that $\forall x \in \mathbb{D} : \frac{g(x)}{f(x)} < c\}$. For full formal exactness, we should write, $\mathcal{O}(f)$ (without variable), but plugging in a function, we allow the notation $g(x) \in \mathcal{O}(f(x))$ instead of $g \in \mathcal{O}(f)$. In many books, the notation $g(x) = \mathcal{O}(f(x))$ is used for this property. We will not stick to this notation as the equality sign "=" will be reserved for equality.

Table 1: Basic notations and symbols used in this thesis

Furthermore, we will give some basic definitions used in this thesis. Most of them are quite common in graph theory and they are defined like in [29]

Definition 1.2 (Graph)**.** As usual in graph theory, a *graph* is a pair (V, E) consisting of a set V and a set E of two-element subsets of V. The set V is called the set of *vertices* and E is called the set of *edges*. If $|V|$ is finite, we call G a *finite graph*. In this thesis, we will restrict our considerations to finite graphs and just call them *graphs*.

A vertex $w \in V$ is called a *neighbor* of a vertex $v \in V$ if $\{v, w\} \in E$. The set of neighbors of a vertex v is denoted by $\Gamma(v) = \{w \in V : \{v, w\} \in E\}$.

If $G' = (V', E')$ is a graph with $V' \subseteq V$ and $E' \subseteq E$, then G' is called a *subgraph* of G. If additionally it holds that $\forall\, e \in E : e \subseteq V' \to e \in E'$, i.e., all edges of G with vertices in V' are also in G', then G' is called an *induced subgraph* of G.

A graph is called *complete* if $E = \binom{V}{2} = \{\{u, v\} : u, v \in V, u \neq v\}$. For $k \in \mathbb{N}$, we will use the notation K_k for the complete graph on k vertices $V = \{1, \ldots, k\}$.

Two graphs $G = (V, E)$ and $G' = (V', E')$ are called *isomorphic* if there exists a bijective mapping $\varphi : V \to V'$ with the property that for all $\{u, v\} \in \binom{V}{2}$ it holds that $\{u, v\} \in E$ iff $\{\varphi(u), \varphi(v)\} \in E'$.

Next, we declare colorings.

Definition 1.3 (coloring of a graph)**.** Let $G = (V, E)$ be a graph. For $a \in \mathbb{N}$,

- an *a-edge coloring* of G is a mapping $E \to \{1, \ldots, a\}$.
- an *a-vertex coloring* of G is a mapping $V \to \{1, \ldots, a\}$.
- (even more general) an a-coloring of a set S is a mapping $S \to \{1, \ldots, a\}$.

The numbers $1, \dots, a$ are called *colors*. The numbers are just labels. We could call the colors also "red" or "blue" and so on.

An a-vertex coloring c_V is called *proper* if $\forall\, e \in E : |c_V(e)| = 2$, i.e., every edge consists of two differently colored vertices.

A graph G is called *a-partite* if there exists a proper a-vertex coloring of G. In the case $a = 2$ we call this property *bipartite*.

Definition 1.4 (Rectilinear drawing of a graph)**.** A *rectilinear drawing* of a graph $G = (V, E)$ is a map $\mathcal{D} : V \to \mathbb{R}^2$. The *drawing of an edge* $e = \{u, v\} \in E$ is given by the set $\mathcal{D}(e) = \{\alpha\, \mathcal{D}(u) + (1 - \alpha)\, \mathcal{D}(v) \text{ such that } \alpha \in [0, 1]\}$. For our considerations of rectilinear drawings in this thesis, we will restrict to those drawings, in which no three points in the image of $\mathcal{D}$ lie on a common straight line, i.e., $\forall u, v, w \in V$, the vectors $u - v$ and $u - w$ are linearly independent.

In this chapter, we will restrict our considerations to drawings of the complete graph. A short discussion on the non-complete case is given in Section 9.2.

Definition 1.5 (Crossings of a rectilinear drawing of a graph)**.** Let $G = (V, E)$ be a graph and $\mathcal{D} : V \to \mathbb{R}^2$ a rectilinear drawing of G. Two distinct edges $e, f \in E$ *cross* if $\mathcal{D}(e) \cap \mathcal{D}(f) \setminus \mathcal{D}(V) \neq \emptyset$. In other words, two edges cross, if their corresponding line segments have a proper crossing that is not one of the end points. We denote by $\mathrm{Cr}(G, \mathcal{D})$ the set of all two-element subsets of the edge set that have crossing edges. If G is a complete graph, we will omit G and write $\mathrm{Cr}(\mathcal{D})$ instead.

$$\mathrm{Cr}(G, \mathcal{D}) = \left\{ \{e, f\} \in \binom{E}{2} : e, f \text{ cross} \right\}$$

Let $a \in \mathbb{N}$ and $c_E : E \to \{1, \dots, a\}$ be an a-edge coloring of G. Then we say that the edges e, f *cross mono-chromatically* or e and f form a *monochromatic crossing* with respect to c_E if e and f cross and $c_E(e) = c_E(f)$. We denote by $\mathrm{CrM}(G, \mathcal{D}, c_E)$ the set of all two-element subsets of the edge set that consist of mono-chromatically crossing edges. Again, if G is complete, we write $\mathrm{CrM}(\mathcal{D}, c_E)$ instead.

$$\mathrm{CrM}(G, \mathcal{D}, c_E) = \left\{ \{e, f\} \in \binom{E}{2} : e, f \text{ cross and } c_E(e) = c_E(f) \right\}$$

Now we can define the monochromatic crossing number as the number of monochromatic crossings

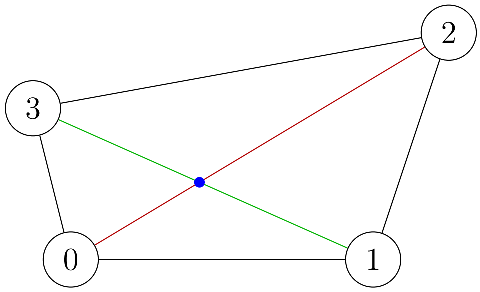


Figure 1: rectilinear drawing of K_4 in convex position: one crossing

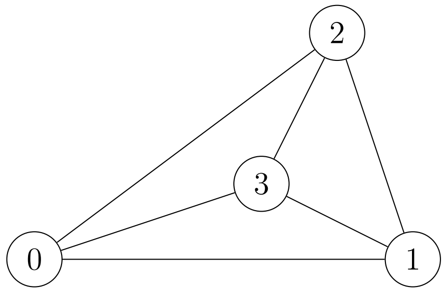


Figure 2: rectilinear drawing of K_4 with a triangular convex hull: no crossing

Definition 1.6 (monochromatic crossing number)**.** Let $a \in \mathbb{N}$, $G = (V, E)$ be a graph and $\mathcal{D}$ be a drawing of G. Then the *a-crossing number of $\mathcal{D}$*, denoted $\mathrm{cr}(G, \mathcal{D}, a)$, is defined by the minimum number of monochromatic crossings of $\mathcal{D}$ among all a-edge colorings of G, i.e.,

$$\mathrm{cr}(G, \mathcal{D}, a) = \min_{c_E : E \to \{1, \ldots, a\}} |\mathrm{CrM}(G, \mathcal{D}, c_E)|$$

As above, if G is complete, we write only $\mathrm{cr}(\mathcal{D}, a)$.

The *a-crossing number of G*, denoted $\mathrm{cr}(G, a)$, is defined by the minimum number of monochromatic crossings of any drawing of G.

$$\mathrm{cr}(G, a) = \min_{\mathcal{D} \text{ drawing of } G} \mathrm{cr}(G, \mathcal{D}, a)$$

Having introduced these terms, we can present the main problem that will be discussed in this thesis:

Problem 1.7. Given a rectilinear drawing $\mathcal{D}$ of a complete graph K_k for some $k \in \mathbb{N}$. Find a 2-edge coloring that minimizes the number of monochromatic crossings.

In other words, find a 2-edge coloring c_E with $|\mathrm{CrM}(\mathcal{D}, c_E)| = \mathrm{cr}(\mathcal{D}, 2)$.

Example 1.8 (rectilinear drawings of the complete graph on four vertices)**.**
A drawing of a complete graph on four vertices can be crossing free if the convex hull is a triangle (see Figure 2) or can have one crossing (see Figure 1). In the figures, vertices are enumerated starting with 0 and crossings are marked by small blue disks •. Edges are colored according to some 2-edge coloring in red or green. Edges that do not cross are not drawn in color as they may be colored arbitrarily without changing the monochromatic crossing number.

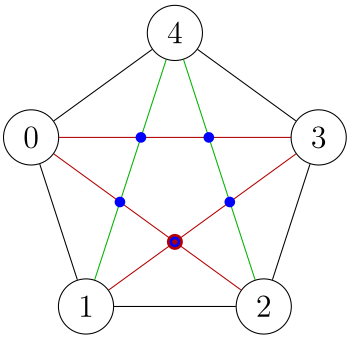


Figure 3: rectilinear drawing of K_5 in convex position: five crossings. For any 2-edge coloring there exists at least one monochromatic crossing

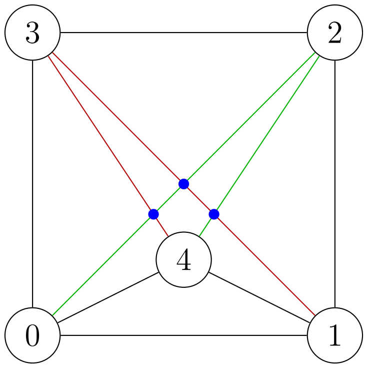


Figure 4: rectilinear drawing of K_5 with four extreme vertices: three crossings

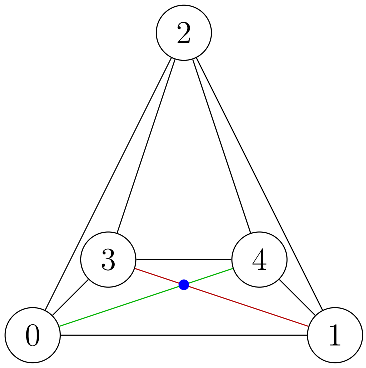


Figure 5: rectilinear drawing of K_5 with three extreme vertices: one crossing

Example 1.9 (rectilinear drawings of the complete graph on five vertices)**.**
For the complete graph on five vertices we see that the rectilinear crossing number can attain several different values, see Figures 3 to 5. If the five points lie in convex position, the diagonals in the induced pentagon intersect like in Figure 3. Since the number of diagonals is odd, we cannot color them alternately in a 2-edge coloring. Hence, any 2-edge coloring of this drawing has at least one monochromatic crossing, marked by • in Figure 3 .

Remark 1.10. In [2], it is shown that all drawings of K_4 are essentially equal to one of the drawings given in Figures 1 and 2 and all drawings of K_5 are essentially equal to one of the drawings given in Figures 3 to 5. The definition of the term *esentially equal* follows in Definition 3.4 on page 13.

2 Basic and geometric observations

Observation 2.1. In a rectilinear drawing, by definition it follows that:

1. An edge cannot cross itself.
2. Two distinct edges cross at most once.
3. Two edges that share a vertex cannot cross.

Observation 2.2 (Inverse approach)**.** Let $G = (V, E)$ be a graph and $\mathcal{D}$ a drawing of G. As the drawing already determines the number of crossings, Problem 1.7 can be solved maximizing the number of bichromatic crossings.

Observation 2.3 (Color number inequality)**.** Let $a, b \in \mathbb{N}$ such that $b \geq a$. Then any coloring of a graph $G = (V, E)$ with a colors is also a coloring of G with b colors (some colors may not show up). That is the set of colorings $\{c_E : E \to \{1, \ldots, a\}\}$ is contained in the set of colorings $\{c_E : E \to \{1, \ldots, b\}\}$. By Definition 1.6 of the crossing number as a minimum over all colorings, we have

$$\forall \mathcal{D} : V \to \mathbb{R}^2 : \mathrm{cr}(G, \mathcal{D}, a) \geq \mathrm{cr}(G, \mathcal{D}, b) \quad \text{and} \quad \mathrm{cr}(G, a) \geq \mathrm{cr}(G, b)$$

In other words, if we are allowed to use more colors, the monochromatic crossing number cannot increase.

Observation 2.4 (Subgraph inequality)**.** Let $G = (V, E)$ be a graph, $G' = (V', E')$ a subgraph of G, $\mathcal{D}$ a drawing of G, c_E an a-edge coloring of G for some $a \in \mathbb{N}$ and $c_{E'} = c_E|_{E'}$ the coloring restricted to the subgraph G. Then $\mathrm{Cr}(G', \mathcal{D}) \subseteq \mathrm{Cr}(G, \mathcal{D})$ and $\mathrm{CrM}(G', \mathcal{D}, c_{E'}) \subseteq \mathrm{CrM}(G', \mathcal{D}, c_E)$.

Minimizing the number of crossings over all colorings, we get:

$$\mathrm{cr}(G', \mathcal{D}, a) \leq \mathrm{cr}(G, \mathcal{D}, a)$$

Minimizing over all drawings of G yields:

$$\mathrm{cr}(G', a) \leq \mathrm{cr}(G, a)$$

In other words, in a subgraph there are at most as many (monochromatic) crossings as in the whole graph.

So far, we do not have discussed how to find a "good" coloring, i.e., how to find a coloring that minimizes the number of monochromatic crossings. But the other direction is easy. Given a graph $G = (V, E)$ and a drawing $\mathcal{D}$ of G, any constant coloring $c_{\text{const}} : E \to \{1, \dots, a\}$ with $c_{\text{const}}(e) = c$ for all edges $e \in E$ and some $c \in \{1, \dots, a\}$ maximizes the number of monochromatic crossings since every crossing is monochromatic. By this, we can find better upper bounds for the number of monochromatic crossings by the Probabilistic Method (see [11]).

Lemma 2.5 (Probabilistic upper bound)**.** *Let $a \in \mathbb{N}$, $G = (V, E)$ be a graph and $\mathcal{D}$ a drawing of G. Let $C_E = \{c_E : E \to \{1, \dots, a\}\}$ be the set of all a-edge colorings of G. We consider a probability space $(C_E, \mathcal{P}(C_E), \mathbb{P}(C_E))$, where $\mathbb{P}$ is a probability measure on $\mathcal{P}(C_E)$, i.e., a map $\mathcal{P}(C_E) \to [0,1]$ that is additive (i.e., $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B)$ for $A, B \in \mathcal{P}(C_E)$ with $A \cap B = \emptyset$) and $\mathbb{P}(C_E) = 1$. Then the minimal monochromatic crossing number is bounded from above by the "average" monochromatic crossing number, i.e., the expected number of monochromatic crossings.*

$$\operatorname{cr}(G, \mathcal{D}, a) = \min_{c_E \in C_E} |\operatorname{CrM}(G, \mathcal{D}, c_E)| \leq \mathbb{E}(|\operatorname{CrM}(G, \mathcal{D}, c_E)|) = \sum_{c_E \in C_E} \mathbb{P}(c_E)\, |\operatorname{CrM}(G, \mathcal{D}, c_E)|$$

Choosing equidistribution $\mathbb{P}(c_E) = \mathbb{P}(c'_E)$ for all $c_E, c'_E \in C_E$, i.e., $\mathbb{P}(c_E) = \frac{1}{|C_E|} = \frac{1}{n^{|E|}}$, we can give a more specific bound. In equidistribution, the colors of two different edges are independent random variables, each of them being equidistributed among $\{1, \dots, a\}$. Hence, the probability of two specific different edges having the same color is given by $\frac{1}{a}$. In this case, the term $\mathbb{E}(|\operatorname{CrM}(G, \mathcal{D}, c_E)|)$ simplifies to $\frac{1}{a}\, |\operatorname{Cr}(G, \mathcal{D})|$. This yields the inequality:

$$\operatorname{cr}(G, \mathcal{D}, a) \leq \frac{1}{a}\, |\operatorname{Cr}(G, \mathcal{D})|$$

This inequality is only tight if all a-edge colorings of G have the same number of monochromatic crossings. If we find an a-edge coloring of G with more than $\frac{1}{a}\, |\operatorname{Cr}(G, \mathcal{D})|$ monochromatic crossings, which we do if $|\operatorname{Cr}(G, \mathcal{D})| \geq 1$, then the above inequality is strict.

Observation 2.6 (Crossing family lower bound)**.** Let $a < b \in \mathbb{N}$, $G = (V, E)$ be a graph, $\mathcal{D}$ a drawing of G, and $\{e_1, e_2, \dots, e_b\} \subseteq E$ a *b-crossing family*, i.e., a set of b edges that pairwise cross. Then by the pigeonhole principle, there exist for any a-edge coloring c_E of G at least two different indices $i, j \in \{1, \dots, b\}$ such that the corresponding crossing edges are in the same color $c_E(e_i) = c_E(e_j)$. In this case, we conclude $\operatorname{cr}(G, \mathcal{D}, a) \geq 1$.

We can slightly improve this bound. Since in a crossing family edges pairwise cross, if c_i is the number of edges among $\{e_1, e_2, \dots, e_b\}$ having color i, we have $\binom{c_i}{2}$ monochromatic crossings in color i. As this holds for all colors, we have $S_{a,b} := \sum_{i=1}^{a} \binom{c_i}{2}$ monochromatic crossings with $\sum_{i=1}^{a} c_i = b$. As the function $x \mapsto \binom{x}{2}$ is a convex function, the minimum of $S_{a,b}$ over all colorings is taken if c_i is equal to $\frac{b}{a}$ for all i due to Jensen inequality, see [18]. So we have at least $n\binom{\frac{b}{a}}{2} = \frac{a\frac{b}{a}\left(\frac{b}{a}-1\right)}{2} = \frac{b(b-a)}{2a}$ monochromatic crossings.

As c_i are integers, we can improve this in the following way: Let $p \in \mathbb{N}, q \in \{0, \ldots, a-1\}$ such that $pa + q = b$. Then the minimum of $S_{a,b}$ over all colorings is taken for $c_i = p+1$ for $i \in \{1, \ldots, q\}$ and $c_i = p$ for $i \in \{q+1, \ldots, a\}$. So the bound reads $q\binom{p+1}{2} + (a-q)\binom{p}{2}$. As an example, an optimally 2-colored 5-crossing family is given in Figure 6, illustrating that a 2-colored 5-crossing family has at least four monochromatic crossings. Although we can guarantee by this a better bound for big crossing families, this bound is quite weak compared to the bound given in Lemma 2.7 below because big crossing families are quite rare in general.

By the subgraph inequality given in Observation 2.4, we conclude $\mathrm{cr}(G, \mathcal{D}, a) \geq l$ if we find l pairwise disjoint $a+1$-crossing families in $\mathcal{D}$. In fact, this inequality also works for l $a+1$-crossing families $f_1, \ldots f_k$ if $|f_i \cap f_j| \leq 1$ for all $i, j \in \{1, \ldots, l\}$ with $i \neq j$, because as in this case two of the mentioned $a+1$-crossing families have at most one edge in common, the guaranteed monochromatic crossing is a different one for each of the $a+1$-crossing families.

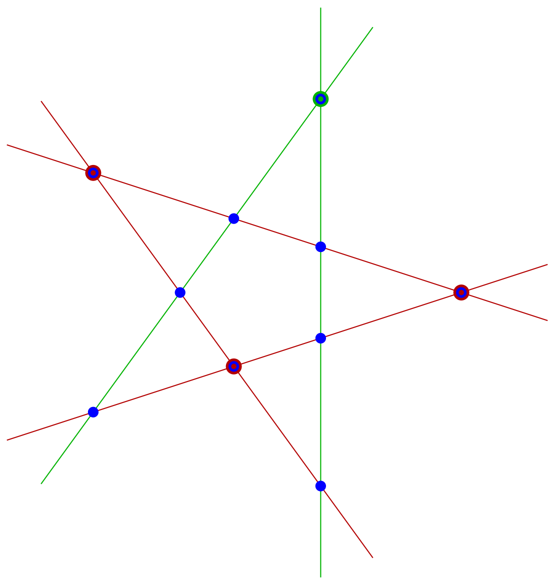


Figure 6: A 2-coloring of a 5-crossing family has at least 4 monochromatic crossings.

Lemma 2.7 (Lifting crossing family bounds)**.** *Assume we have proven for some $k, a \in \mathbb{N}$ that for any rectilinear drawing of K_k there exists an $a+1$-crossing family. Then for $h \in \mathbb{N}$*

with $h \geq k$ *it holds that*

$$\operatorname{cr}(K_h, a) \geq \frac{\binom{h}{k}}{\binom{h}{k} - S_0 - (a+1)S_1}$$

where

$$S_0 = \sum_{j=0}^{a+1} 2^j \binom{a+1}{j} \binom{h-2a-2}{k-j}$$

and

$$S_1 = \sum_{j=0}^{a} 2^j \binom{a}{j} \binom{h-2a-2}{k-2-j}$$

Proof. Let $\mathcal{D}$ be a drawing of the complete graph $K_h = (V, E)$. Since an induced subgraph of a complete graph is complete, every set $S \in \binom{V}{k}$ induces a complete subgraph on k vertices, whose sub-embedding $\mathcal{D}|_S$ contains an $a+1$ crossing family by assumption. There are $\binom{h}{k}$ k-element subsets of the vertex set V. Fixing an $a+1$-crossing family F, we want to count the number of k-element subsets of V containing at most one edge of F.

First we count the number of k-element subsets not containing an edge of F. Let $j \in \{0, \dots, a+1\}$ be the number of vertices of F contained in the chosen k-element subset. We may choose j edges of F and for each chosen edge we may choose one of its two vertices. So there are $2^j \binom{a+1}{j}$ ways to choose these vertices. Furthermore, there are $\binom{h-2a-2}{k-j}$ ways to choose the vertices not in F. In total, there are $S_0 = \sum_{j=0}^{a+1} 2^j \binom{a+1}{j} \binom{h-2a-2}{k-j}$ k-element subsets not containing an edge of F.

Now we count the number of k-element subsets containing exactly one edge of F. For this, we first choose one of $a+1$ edges. For the rest of the vertices, we have to count the number of $(k-2)$-element subsets not containing an edge of the crossing family that consists of F without the removed edge. So index shifting of the formula in the first case gives the result: $S_1 = \sum_{j=0}^{a} 2^j \binom{a}{j} \binom{h-2a-2}{k-2-j}$.

Summing up, there are $\binom{h}{k} - S_0 - (a+1)S_1$ k-element subsets of V containing at least two edges of F. Hence, we may choose at least $\frac{\binom{h}{k}}{\binom{h}{k} - S_0 - (a+1)S_1}$ many $a+1$-crossing families that pairwise have at most one edge in common, each of them resulting in a different monochromatic crossing. □

Remark 2.8 (Bounds for rectilinear drawings of small graphs).

1. In any rectilinear drawing of a graph G with four vertices, there can be at most one crossing. This can be shown by some easy case distinction, see also [3].

2. On the other hand, we deduce from Kuratowski's theorem [22, 30] that the complete graph on five vertices K_5 is not planar, i.e., any rectilinear drawing of K_5 has at least one crossing (a 2-crossing family).
3. By analyzing all essentially equal rectilinear drawings, we see that there exists a rectilinear drawing of K_8 that has a 2-coloring without a monochromatic crossing, see [5] and Figure 7.
4. By analyzing all essentially equal rectilinear drawings, we find out that in any rectilinear drawing of K_{10} there exists a 3-crossing family (see [2]).
5. By extending the point sets of the data base on 11 points and case distinction, we find out that in every rectilinear drawing of K_{15}, there exists a 4-crossing family (see [2]).

Corollary 2.9 (General bounds for rectilinear drawings)**.** By 2.8 and 2.7 we can deduce lower bounds on the monochromatic crossing numbers of rectilinear drawings of the complete graph up to three colors. A table for up to 30 vertices is given in Table 2.

k	$a=1$	$a=2$	$a=3$	k	$a=1$	$a=2$	$a=3$	k	$a=1$	$a=2$	$a=3$
1	0	0	0	11	66	1	0	21	1197	11	2
2	0	0	0	12	99	2	0	22	1463	13	2
3	0	0	0	13	143	2	0	23	1771	15	2
4	0	0	0	14	201	3	0	24	2126	18	3
5	1	0	0	15	273	3	1	25	2530	22	3
6	3	0	0	16	364	4	1	26	2990	25	3
7	7	0	0	17	476	5	1	27	3510	30	3
8	14	0	0	18	612	6	2	28	4095	34	4
9	26	0	0	19	776	7	2	29	4751	39	4
10	42	1	0	20	969	9	2	30	5481	45	5

Table 2: Lower bounds on the monochromatic crossing numbers of rectilinear drawings deduced by Remark 2.8 and Lemma 2.7

In fact, there are much better bounds for the crossing number for rectilinear drawings with $a = 1$ known, see [3].

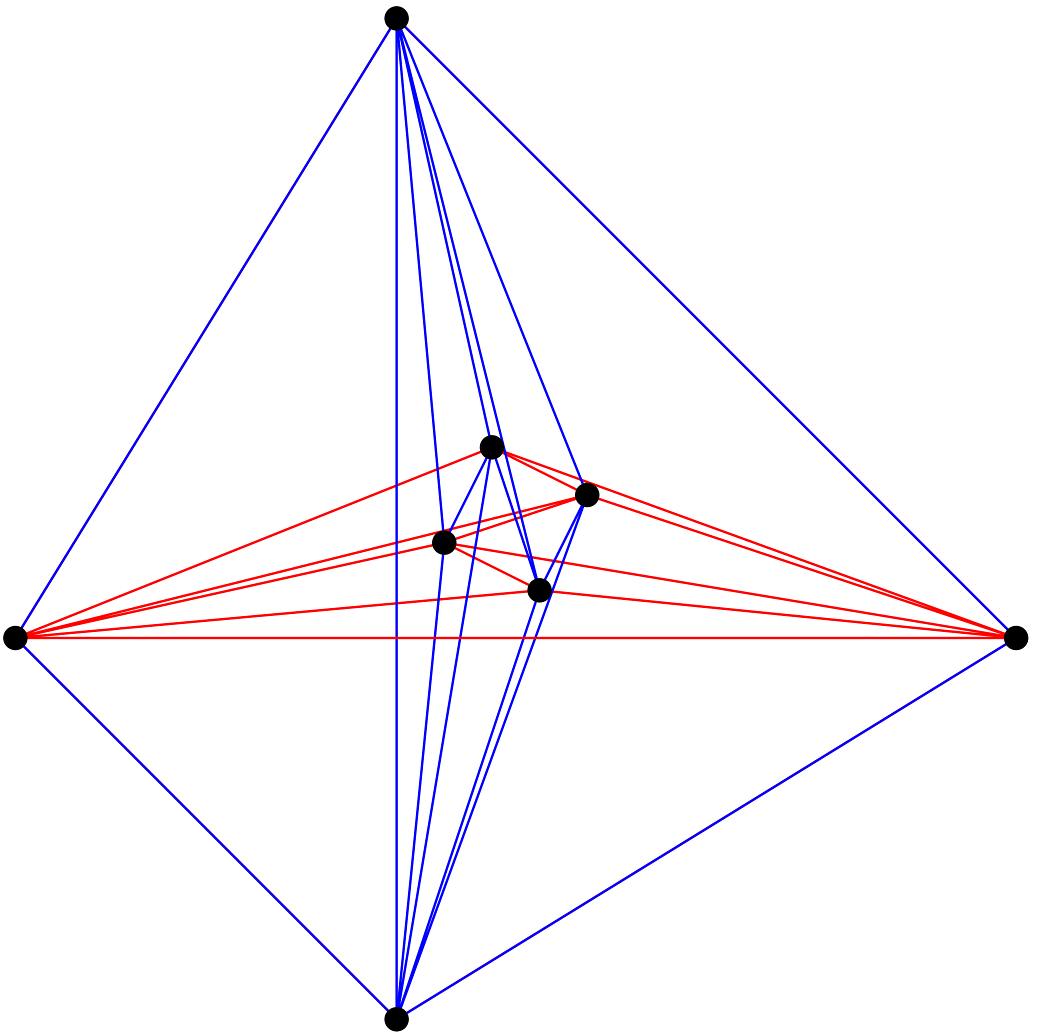


Figure 7: A 2-edge colored rectilinear drawing of K_8 without a monochromatic crossing (Figure taken from [5])

3 The crossing Graph

It is quite hard to see further geometric properties of a drawing of a graph. One approach is to reformulate Problem 1.7 into an equivalent one.

Definition 3.1 (crossing graph)**.** Let $G = (V, E)$ be a graph and $\mathcal{D}$ be a drawing of G. Then the *crossing graph*, denoted by $\times(G, \mathcal{D}) = (V_\times, E_\times)$, is defined to be the graph with vertex set $V_\times = E$, and edge set $E_\times = \{\{e, f\} \in \binom{E}{2} : e \text{ and } f \text{ cross}\}$.

Lemma 3.2 (Basic properties of the crossing graph)**.** *Let $G = (V, E)$ be a graph , $\mathcal{D}$ a rectilinear drawing of G and $\times(G, \mathcal{D}) = (V_\times, E_\times)$ the crossing graph of $\mathcal{D}$. Then*

1. $|V_\times| = |E|$
2. *For edges $e, f \in E$ that share a vertex, we have $\{e, f\} \notin E_\times$*
3. *A tuple of four vertices can induce at most one crossing. Hence, for $t, u, v, w \in V$ with $\{\{t, u\}, \{v, w\}\} \in E_\times$, it holds that $\{\{t, v\}, \{u, w\}\} \notin E_\times$ and $\{\{t, w\}, \{u, v\}\} \notin E_\times$*
4. $|E_\times| \leq \binom{|V|}{4}$
5. *If G is a complete graph, we have $|E_\times| \geq \frac{1}{5}\binom{|V|}{4}$. In fact, much better bounds are known.*

Proof. 1. This is clear by the definition of the crossing graph.

2. This directly follows by the definition of the crossing graph, see Definition 3.1, and the property of $\mathcal{D}$ being a rectilinear drawing.

3. This follows from Remark 2.8 and Definition 3.1

4. This follows directly from 3. There are $\binom{k}{4}$ four-element subsets of V, each of them induces at most one crossing.

5. This follows from the lifted two-crossing family lower bound (see Lemma 2.7) for $k = 5$, $a = 1$ and $h = |V|$. □

Observation 3.3 (Number of crossing graphs)**.** Let $G = (V, E)$ be a graph. In any drawing $\mathcal{D}$, for every two-element subset $\{e, f\} \in \binom{E}{2}$, the edges e and f cross or they don't cross. In other words, for every two vertices of the crossing graph, they are connected by an edge or not. Hence, we observe that there are at most $2^{\binom{|V_\times|}{2}} = 2^{\binom{|E|}{2}}$ crossing graphs of G. In fact, we can derive a better bound from Lemma 3.2, Property 3. Since for any four points, there are only three possibilities for a crossing and one for no crossing, there are at most $4^{\binom{|V|}{4}}$ different crossing graphs.

Since there are infinitely (even uncountably) many rectilinear drawings of a graph G, we want to identify those drawings which have the "same" crossings, i.e., which lead to the same or an isomorphic crossing graph.

Definition 3.4 (Essentially equal and essentially different drawings)**.** Let $G = (V, E)$ be a graph and $\mathcal{D}_1$ and $\mathcal{D}_2$ two drawings of G. $\mathcal{D}_1$ and $\mathcal{D}_2$ are called *essentially equal*, denoted by $\mathcal{D}_1 \sim \mathcal{D}_2$, if there exists a *graph automorphism* $\varphi : G \to G$, i.e., a bijective map $\varphi : V \to V$ with the property $\forall\ u, v \in V : \{u, v\} \in E \leftrightarrow \{\varphi(u), \varphi(v)\} \in E$, such that for all edges $e, f \in E$ it holds that e and f cross with respect to the drawing $\mathcal{D}_1$ iff $\varphi(e)$ and $\varphi(f)$ cross with respect to the the drawing $\mathcal{D}_2$.

In other words, two drawings are essentially equal if after a possible relabeling of the vertices, the crossing graphs of those drawings coincide.

Observation 3.5 (Partition into finitely many essentially equal drawing classes)**.** Let $G = (V, E)$ be a graph. Obviously, the relation $\sim$ on drawings of G is an equivalence relation. By Observation 3.3, there are only finitely many equivalence classes. We write $G_\sim$ for the set of classes of essentially equal drawings $\{[\mathcal{D}]_\sim | \mathcal{D}$ rectilinear drawing of $G\}$.

By definition, if two drawings $\mathcal{D}_1$ and $\mathcal{D}_2$ are essentially equal, then their crossing graphs $\times(G, \mathcal{D}_1)$ and $\times(G, \mathcal{D}_2)$ are isomorphic. Hence, minimizing over all drawings for a given graph can be considered as a finite and thus combinatorial optimization problem and can be done operating on the crossing graph.

Lemma 3.6. *Problem 1.7 can be solved by solving the following combinatorial optimization problem:*

Problem 3.7 (Minimum vertex coloring problem on the crossing graph)**.** Given $\times(\mathcal{D}) = (V_\times, E_\times)$ the crossing graph of a rectilinear drawing of a complete graph. Find a 2-vertex coloring that minimizes the number of edges whose vertices have the same color. We will call those edges *monochromatic edges.*

$$\min_{c_{E_\times} \text{ is a 2-coloring of } E_\times} \left|\{\{e, f\} \in E_\times : c_{E_\times}(e) = c_{E_\times}(f)\}\right|$$

Proof. By definition, the crossing graph translates crossings in the drawing $\mathcal{D}$ into edges in the crossing graph $G_\times$. The property of a crossing to be monochromatic translates to the property of edges to be monochromatic in the crossing graph if we just carry over the coloring from the edge set of the graph to the vertex set of the crossing graph. □

Lemma 3.8. *Problem 1.7 can polynomially be reduced to Problem 3.7.*

Proof. As the coloring of the crossing graph carries over to the drawing canonically, it suffices to show that we can construct the crossing graph in polynomial time. Given k points in the plane in general position, we have to check for every choice of four points if there are two line segments that cross. We have already seen that there can be at most one crossing within four points $t = (t_1, t_2), u = (u_1, u_2), v = (v_1, v_2), w = (w_1, w_2)$ in the plane, i.e., there is exactly one crossing if they lie in convex position and there is no crossing if they do not lie in convex position. We can assume that a test if two line segments $\{\alpha t + \beta u : \alpha, \beta \in [0, 1], \alpha + \beta = 1\}$ and $\{\gamma v + \delta w : \gamma, \delta \in [0, 1], \gamma + \delta = 1\}$ intersect can be done in constant time. Either one solves the linear equation system or, even easier, one checks if the orientation of the vectors $t - u, t - v$ and $t - u, t - w$ as well as those of $v - w, v - t$ and $v - w, v - u$ are equal, i.e., it suffices to evaluate the sign of four 2×2 determinants. Actually, this is equivalent to two conditions on 3×3 determinants:

$$\begin{vmatrix} t_1 & u_1 & v_1 \\ t_2 & u_2 & v_2 \\ 1 & 1 & 1 \end{vmatrix} \cdot \begin{vmatrix} t_1 & u_1 & w_1 \\ t_2 & u_2 & w_2 \\ 1 & 1 & 1 \end{vmatrix} < 0$$

$$\begin{vmatrix} v_1 & w_1 & t_1 \\ v_2 & w_2 & t_2 \\ 1 & 1 & 1 \end{vmatrix} \cdot \begin{vmatrix} v_1 & w_1 & u_1 \\ v_2 & w_2 & u_2 \\ 1 & 1 & 1 \end{vmatrix} < 0$$

Performing $\binom{k}{4} \in \mathcal{O}(k^4)$ times a constant time operation leads to a complexity of $\mathcal{O}(k^4)$, which is polynomial. □

The benefit of the problem formulation in Problem 3.7 is, that we can use methods and theorems from graph theory to tackle the problem. The downside of this approach is that the geometry of the embedding, which is implicit by the construction of the crossing graph, is not directly visible.

Lemma 3.9. *Problem 3.7 is equivalent to the maximal bipartite subgraph problem on crossing graphs.*

Problem 3.10 (Maximal bipartite subgraph problem)**.** Given a graph $G = (V, E)$. Find a maximal bipartite subgraph of G, i.e., find a graph $G' = (V', E')$ with the following properties:

1. $V' = V$
2. $E' \subseteq E$
3. G' is *bipartite.*
4. $|E'|$ is maximal among all graphs $G' = (V', E')$ satisfying properties 1, 2 and 3.

Proof. First consider an instance of Problem 3.7 on a graph $G = (V, E)$ (not necessarily a crossing graph). Let $G' = (V', E')$ be a maximal bipartite subgraph of G. Then by breath first search (see e.g. [29], pages 71 ff), we can find a proper 2-vertex coloring $c_{V'}$ of G'. Since G' is a maximal bipartite subgraph, the number of removed edges $E \setminus E'$, that do not necessarily have differently colored end vertices, is minimal. Hence, all edges in $E \setminus E'$ are monochromatic. So c_V is a 2-vertex coloring of G that minimizes the number of monochromatic edges.

Conversely, consider a graph $G = (V, E)$ as an instance of Problem 3.10. Let c_V be a 2-vertex coloring of G that minimizes the number of monochromatic edges. Let $\bar{E} = \{\{u, v\} \in E : c_V(u) = c_V(v)\}$ be the set of monochromatic edges. Then $G' = (V, E \setminus \bar{E})$ is a bipartite subgraph since by construction, c_V is a proper 2-vertex coloring of G'. Since $|\bar{E}|$ is minimal, $|E \setminus \bar{E}|$ is maximal. Hence, G' is a maximal bipartite subgraph of G. □

Remark 3.11. For general graphs, Problem 3.10 is NP-hard, see [13].

But the decision problem, if a graph is bipartite can be solved in polynomial time using breath first search. This gives hope that we can deduce reasonable heuristics.

Furthermore, we have the characterization that a graph G is bipartite iff it does not contain an odd cycle, i.e., iff for every odd $u \geq 3$, no subgraph of G is isomorphic to $C_u = (\{1, \ldots, u\}, \{\{1, 2\}, \{2, 3\}, \ldots, \{u - 1, u\}, \{u, 1\}\})$ due to Kőnig, see [23].

Remark 3.12. Because we only want to find a 2-vertex coloring, we can omit all singletons, i.e., all vertices that do not have neighbors in the crossing graph. These vertices correspond to edges in the drawing that do not cross another edge. They can be colored arbitrarily not changing the monochromatic crossing number.

Furthermore, all vertices in the crossing graph that have only one edge correspond to edges in the drawing that cross exactly once. If the rest of the graph is colored, there is a unique way to color them avoiding a monochromatic crossing, i.e., color them in the other color than the crossing edge. So for Problem 1.7, we can iteratively remove all the edges that cross at most one edge for our consideration. This translates to Problem 3.7 as to consider only the 2-*core* of the crossing graph, i.e., the graph that remains from the crossing graph after iteratively removing vertices with degree (number of neighbors) smaller than 2. This is also consistent with Problem 3.10, because vertices with degree at most one cannot appear in odd cycles.

4 Equivalence to an integer linear program

For a graph $G = (V, E)$, we formulate Problem 3.10 (and analogously Problem 3.7) as an integer linear program. Let $G' = (V', E')$ be a maximal bipartite subgraph of G and c_V a proper 2-vertex coloring on G'. For each edge $e = \{u, v\} \in E$, we introduce a variable $a_e \in \{0, 1\}$, that indicates if the edge e is in the subgraph G' (or equivalently, if $c_V(u) \neq c_V(v)$) or not. For technical reasons, it is useful to define

$$a_e = \begin{cases} 0 & \text{if } e \in E' \\ 1 & \text{if } e \notin E' \end{cases}$$

Then, as an objective function, we want to minimize $\sum_{e\in E} a_e$.

It remains to translate the constraint, that the resulting graph $G' = (V, E')$ with $E' = \{e \in E : a_e = 0\}$ is bipartite. For this, recall the characterization from Remark 3.11. In order to obtain a subgraph G' from G that is bipartite, we need to remove from each odd cycle in G at least one edge. Let $\mathcal{C}_{\text{odd}}(G)$ be the set of odd cycles in G, i.e., the set of all subgraphs of G, for which there exists a $u \geq 3$ odd, such that they are isomorphic to C_u as defined in Remark 3.11. For some $C \in \mathcal{C}_{\text{odd}}(G)$, we denote by $V(C)$ the vertex set of C and by $E(C)$ the edge set of C. Then the linear program, that is equivalent to Problems 3.7 and 3.10 reads like

Problem 4.1 (Integer linear program formulation)**.** Given a graph $G = (V, E)$. Find $(a_e)_{e\in E} \in \{0, 1\}^E$ so that each odd cycle $C \in \mathcal{C}_{\text{odd}}(G)$ contains an edge $e_C \in E(C)$ such that $a_{e_C} = 1$ and the sum $\sum_{e\in E} a_e$ is minimized with this property.

Written as formula:

$$\min \sum_{e\in E} a_e \quad \text{s.t. } \forall C \in \mathcal{C}_{\text{odd}}(G) : \sum_{e\in E(C)} a_e \geq 1 \quad \text{and } a_e \in \{0, 1\}\ \forall\, e \in E$$

As Problems 3.7 and 3.10 are NP-hard, Problem 4.1 also is. But if we drop the condition that $a_e \in \{0, 1\}\ \forall\, e \in E$ and replace it by $x_e \in [0, 1]\ \forall\, e \in E$, we end up with a (continuous) linear program:

Problem 4.2 (LP-relaxation of Problem 4.1)**.** Given a graph $G = (V, E)$. Find $(x_e)_{e\in E} \in [0, 1]^E$ so that for each odd cycle $C \in \mathcal{C}_{\text{odd}}(G)$ the sum of edge weights $\sum_{e_C\in E(C)} x_{e_C}$ is at least one and the sum $\sum_{e\in E} x_e$ is minimized with this property.

Written as formula:

$$\min \sum_{e \in E} x_e \quad \text{s.t. } \forall C \in \mathcal{C}_{\text{odd}}(G): \sum_{e \in E(C)} x_e \geq 1 \quad \text{and } x_e \in [0,1]\ \forall\, e \in E$$

A downside of Problems 4.1 and 4.2 is that the number of constraints, which is $|\mathcal{C}_{\text{odd}}(G)|$ is in general exponential in the number of edges of G. M. Innerkofler [17] came up with a different formulation of Problems 3.7 and 3.10 as an integer linear program. As above, we introduce for each edge a variable $b_e \in \{0,1\}$ that will be 1 if e is monochromatic and 0 if e is bichromatic. So the objective function to be minimized is again $\sum_{e \in E} a_e$. Furthermore, we introduce a variable $c_v \in \{0,1\}$ for each vertex $v \in V$, that indicates the color. We now need to find constraints that guarantee that $b_e = 1$ if $e = \{u,v\}$ is a monochromatic edge. We observe, that e is bi-chromatic iff $c_u + c_v = 1$ and monochromatic iff if $c_u + c_v \in \{0,2\}$. So the constraint $c_u + c_v - 1 \leq b_e$ and $-c_u - c_v + 1 \leq b_e$ guarantee $b_e = 1$ for monochromatic edges. For bi-chromatic edges, we have $b_e = 0$ in an optimal solution.

So the problem reads as follows:

Problem 4.3 (Integer linear program formulation due to M. Innerkofler)**.**

$$\begin{aligned} \min \sum_{e \in E} b_e \quad \text{s.t. } & \forall\, e = \{u,v\} \in E: c_u + c_v - 1 \leq b_e \\ & \forall\, e = \{u,v\} \in E: -c_u - c_v + 1 \leq b_e \\ & \forall\, e \in E: b_e \in \{0,1\} \\ & \forall\, v \in V: c_v \in \{0,1\} \end{aligned}$$

This problem can be relaxed to a linear program as well:

Problem 4.4 (LP-relaxation of Problem 4.3)**.**

$$\begin{aligned} \min \sum_{e \in E} y_e \quad \text{s.t. } & \forall\, e = \{u,v\} \in E: z_u + z_v - 1 \leq y_e \\ & \forall\, e = \{u,v\} \in E: -z_u - z_v + 1 \leq y_e \\ & \forall\, e \in E: y_e \in [0,1] \\ & \forall\, v \in V: z_v \in [0,1] \end{aligned}$$

Observation 4.5 (Properties of the integer program and the LP-relaxation)**.**

1. Since $[0,1]^E$ is a bounded set, Problems 4.1 to 4.4 are bounded (integer) linear pro-

grams. Because $a_e = 1 \,\forall\, e \in E$ is a feasible solution for Problem 4.1, $x_e = 1 \,\forall\, e \in E$ is a feasible solution for Problem 4.2, $b_e = 1 \,\forall\, e \in E$, $c_v \in \{0, 1\}$ arbitrary $\forall\, v \in V$ is a feasible solution for Problem 4.3 and $y_e = 1 \,\forall\, e \in E$, $z_v \in [0, 1]$ arbitrary $\forall\, v \in V$ is a feasible solution for Problem 4.4, for all those problems there exists an optimal solution.

Let $G = (V, E)$ be a graph, let $(a_e)_{e \in E} \in \{0, 1\}^E$ be an optimal solution of Problem 4.1 with objective function value A, $(x_e)_{e \in E} \in [0, 1]^E$ an optimal solution of Problem 4.2 with objective function value X, $(b_e)_{e \in E} \in \{0, 1\}^E$ together with $(c_v)_{v \in V} \in \{0, 1\}^V$ an optimal solution of Problem 4.3 with objective function value B and $(y_e)_{e \in E} \in [0, 1]^E$ together with $(z_v)_{v \in V} \in [0, 1]^V$ an optimal solution of Problem 4.4 with objective function value Y. Then

2. The linear programs Problem 4.1 and Problem 4.3 are equivalent, hence $A = B$. But their relaxations Problem 4.2 and Problem 4.4 are not equivalent in general. In Problem 4.2 we allow that edges are "fractionally" removed from the graph to obtain a bipartite subgraph. In Problem 4.4, we allow vertices to be "fractionally" colored.

3. Since $\{0, 1\}^E \subseteq [0, 1]^E$ and $\{0, 1\}^V \subseteq [0, 1]^V$, we obtain $X \leq A$ and $Y \leq B$. So solving Problem 4.2 or Problem 4.4 gives a lower bound for the solution of Problems 4.1 and 4.3.

4. For Problem 4.4, $z_v = \frac{1}{2} \in \{0, 1\} \forall\, v \in V$ and $y_e = 0 \,\forall\, e \in E$ is a feasible solution with objective function value 0. Hence, this solution is also optimal and we have $Y = 0$. So the LP-relaxed Problem 4.4 gives only a trivial lower bound for the solution of Problem 4.3.

5. Since A is integral, i.e., $A \in \mathbb{N}_0$, also $\lceil X \rceil \leq A$ is a lower bound for the solution of Problem 4.1.

6. Since $(\lceil x_e \rceil)_{e \in E} \in \{0, 1\}^E$ is a feasible solution to Problem 4.1, $\sum_{e \in E} \lceil x_e \rceil \geq A$ is an upper bound for the solution of Problem 4.1. Moreover, by the equivalence of Problems 3.7, 3.10 and 4.1, we can deduce a proper 2-coloring of G from $(\lceil x_e \rceil)_{e \in E}$. However, this upper bound is not very strong in general.

7. We do not know any general approximation ratio by LP-relaxation. However, in praxis, this method works quite good, as we will discuss in Section 8.

Remark 4.6 (Techniques to solve an integer linear program)**.** There are several techniques to directly solve an integer linear program including cutting planes method, Lagrange relaxing and Branch and Bound method. You can look them up in [21], pages 129 ff. These techniques either lead to an exact approach with exponential running time or a heuristic without an approximation ratio in general. We will not discuss these techniques here.

5 Solving the linear program

As we have seen that the lower bound obtained form Problem 4.4 is trivial, we want to find a way to solve Problem 4.2 in polynomial time hoping that the obtained bound is better. The number of variables in Problems 4.1 and 4.2 is $|E|$, which is polynomial in the input. But the number of constraints in the above formulations is $|\mathcal{C}_{\text{odd}}(G)|$, which is in general exponential in the input. Hence, we cannot use a standard method like the Simplex Algorithm [9] or Karmarkar's Algorithm [20] to solve Problem 4.2 efficiently. But by the Ellipsoid Method, there is a way of solving a linear program with exponentially many constraints given a polynomial oracle that can find an unsatisfied constraint in an infeasible solution or decide that a solution is feasible. The Ellipsoid Method used to be the first polynomial algorithm to solve linear programs and was first described by N. Z. Shor and L. Khachiyan in 1972. In this thesis, we reference to a simplified an more readable version of the Ellipsoid Method, see [28].

For this, we refine the problem described in [28] by an ε-estimation before we describe how to apply this method to Problem 4.2:

Problem 5.1 (Polytope element problem)**.** Given a "precision parameter" $\varepsilon > 0$ and a set of $m \in \mathbb{N}$ liner inequalities in $n \in \mathbb{N}$ variables, described by $Ax \leq B$ where $A \in \mathbb{R}^{m \times n}$ is the *system matrix*, $B \in \mathbb{R}^m$ is the *right hand side vector* and the inequality is read index-wise, such that the *polytope* $S = \{x \in \mathbb{R}^n : Ax \leq B\}$ is either empty or bounded. Task: Find some $x \in S$ or decide $\operatorname{Vol}(S) < \varepsilon$, where $\operatorname{Vol}(S)$ is the *volume* of S, i.e., the n-dimensional Lebesgue measure of S.

Now we reduce Problem 5.1 to Problem 4.2. This step is dependent on the specific structure of Problem 4.2 and not described by [28].

Lemma 5.2. *If we can solve Problem 5.1 with a complexity that is polynomial in n and in $\log(\frac{1}{\varepsilon})$, then we can polynomially solve Problem 4.2.*

Proof. As in a linear program, the objective function $f : \mathbb{R}^n \to \mathbb{R}$ is linear, it can be described as $f(x) = f((x_1, \ldots, x_n)^\top) = \alpha^\top x = \alpha_1 x_1 + \ldots + \alpha_n x_n$ for some vector $\alpha \in \mathbb{R}^n$. So the objective function can be used as a constraint in a decision problem: For a number $c \in \mathbb{R}$, the linear program $\min f(x)$ s.t. $Ax \leq b$ has a feasible solution $\bar{x} \in \mathbb{R}^n$ with $f(\bar{x}) \leq c$ iff

$$\bar{x} \in S(c) := \left\{ x \in \mathbb{R}^n : \begin{pmatrix} A \\ \alpha^\top \end{pmatrix} x \leq \begin{pmatrix} B \\ c \end{pmatrix} \right\}$$

So the set $S(c)$, the set of all vectors x satisfying the constraints $Ax \leq b$ that have objective function value at least c, is a polytope.

In Problem 4.2, the matrix A consists of interval restrictions $x_i \leq 1$ and $-x_i \leq 0$ for $i \in \{1, \dots, n\}$ and odd cycle restrictions $-\sum_{x_j \in E(C)} x_j \leq -1$ for $C \in \mathcal{C}_{\text{odd}}(G)$. Without loss of generality, we can assume that the first $2n$ rows of A and B consist of the interval restrictions and all other rows are odd cycle restrictions. Then the top $n \times n$ sub-matrix of A is the identity matrix, and all other entries of A are $\in \{0, -1\}$. The first n entries in the vector B are 1, the next n entries are 0 and all others are -1. We have $\alpha_i = 1$ for all $i \in \{1, \dots, n\}$, so $f(x) = \sum_{i=1}^n x_i$.

Let $x^* \in S$ be an optimal solution and c^* be the optimal objective function value for Problem 4.2. We observe the following properties:

1. The n-dimensional vector that is one in every entry is feasible for Problem 4.2. So the set S is not empty.
2. For $c \in \mathbb{R}$, $S(c) = S \cap \{x \in \mathbb{R}^n : f(x) \leq c\}$
3. For $\bar{c} \leq \bar{d}$, we have $S(\bar{c}) \subseteq S(\bar{d})$
4. For $x \in S$, $y \in [0,1]^n$ with $y \geq x$ component-wise, we have $y \in S$.
5. $S = S(c)$ for $c \geq n$.
6. $S(c) = \emptyset$ for $c < c^*$.
7. $S(c) \neq \emptyset$ for $c \geq c^*$.
8. There exists an optimal solution $\tilde{x}^* \in \mathbb{Q}^n$ with denominator at most $n!$. Especially, c^* is rational and its denominator is bounded by $n!$. Let $I \subseteq \{1, \dots, n\}$ be the set of indices with $\tilde{x}_i^* = 1$ for all $i \in I$ and $J = \{1, \dots, n\} \setminus I$ the set of indices with $\tilde{x}_j^* < 1$ for all $j \in J$. Then $\max_{j \in J} x_j \leq 1 - \frac{1}{n!}$.
9. Let $x(\gamma) = \tilde{x}^* + \gamma y$, where y is the vector which is -1 on indices in I and 1 for indices in J. Then for every $0 < \gamma < \min(\frac{1}{3}, \frac{1}{n!})$, $x(\gamma) \in S(c^* + n\gamma)$.
10. $\text{Vol}(S(c^* + \varepsilon)) \geq \left(\frac{\varepsilon}{2n}\right)^n$ for all $0 < \varepsilon < \min(\frac{1}{3n}, \frac{1}{n \cdot n!})$

From Property 10 it follows that we can find the optimal objective function value by a binary search method in polynomial time since by property 8, $c^* \in Q = \{0 \leq \frac{p}{q} \leq n : p \in \mathbb{N}_0, q \in \mathbb{N}, q \leq n!\}$, and $\log(|Q|) \leq \log(n(n!)^2) \leq \log(n) + 2n \log(n)$ is polynomial in n. For each value $q \in Q$ we can determine by Problem 5.1 if $S(q)$ has positive volume (and hence $c^* < q$) or not (then $c^* \geq q$). Because distinct elements Q have distance at least $\frac{1}{(n!)^2}$, choosing the "precision parameter" $\varepsilon = \left(\frac{1}{2n(n!)^2}\right)^n$ suffices.

Now we want to prove those properties:

Properties 1 to 6 are clear by definition and construction.

Property 7: This is clear because $f(x^*) = c^*$.

Property 8: In a bounded linear program, there exists a corner where the optimal function value is taken. This corner is an intersection of n hyperplanes, where constrains are satisfied with equality. Hence, this corner is a solution of a linear equation system, where the system matrix A' is a sub-matrix of A, which is a rational matrix. As the entries of A are -1, 0, or 1, the determinant of A', which appears in the denominator by Carmer's rule, is at most $|\sum_{\sigma\in\mathcal{S}_n} \operatorname{sgn}(\sigma) \prod_{i=1}^n a'_{i,\sigma(i)}| \leq \sum_{\sigma\in\mathcal{S}_n} 1 \leq n!$.

Property 9: First we show $x(\gamma) \in [0,1]^n$. For $i \in I$, $0 < x(\gamma)_i = 1 - \gamma < 1$ by assumption. For $j \in J$, $0 \leq \tilde{x}_j^* < x(\gamma)_i = \tilde{x}_j^* + \gamma < 1 - \frac{1}{n!} + \frac{1}{n!} = 1$.

Now we have to show that the cycle bounds are satisfied for $x(\gamma)$. Let $C \in \mathcal{C}_{\text{odd}}(G)$ with edge indices $k_1, \dots, k_u$. If at least two indices are in I, then $\sum_{l=1}^u x(\gamma)_{k_l} \geq 2(1 - \frac{1}{3}) = \frac{4}{3} > 1$. If there is at most one index among $k_1, \dots, k_u$ in I, then $\sum_{l=1}^u x(\gamma)_{k_l} \geq \sum_{l=1}^u \tilde{x}_{k_l}^* + \gamma((n-1)-1) \geq 1$ provided $n \geq 2$, which is the only interesting case.

It remains to show $f(\gamma(x)) \leq c^* + n\gamma$. This is clear by definition: $f(\gamma(x)) = \sum_{i=1}^n \gamma(x)_i = \sum_{i\in I}(\tilde{x}_i^* - \gamma) + \sum_{j\in J}(\tilde{x}_j^* + \gamma) \leq \sum_{i=1}^n \tilde{x}_i^* + n\gamma$.

Property 10: By Property 9, $\bar{x} := x(\frac{\varepsilon}{2n}) \in S(c^* + \frac{\varepsilon}{2})$ and by construction $\bar{x}_i \in [\frac{\varepsilon}{2n}, 1 - \frac{\varepsilon}{2n}]$ for all $i \in \{1, \dots, n\}$. By Property 4, the set $X = \{\bar{x} + y : y \in [0, \frac{\varepsilon}{2n}]^n\}$ is a subset of S. We show that even $X \subseteq S(c^* + \varepsilon)$. Let $x \in X$. Then there exists a $y \in [0,1]^n$ with $y_i \leq \frac{\varepsilon}{2n}$ such that $x = \bar{x} + y$. Hence $f(x) = f(\bar{x} + y) = f(\bar{x}) + f(y) \leq c^* + \frac{\varepsilon}{2} + \sum_{i=1}^n y_i \leq c^* + \frac{\varepsilon}{2} + n\frac{\varepsilon}{2n} = c^* + \varepsilon$. Since X is a cube with side length $\frac{\varepsilon}{2n}$, $\operatorname{Vol}(S(c^* + \varepsilon)) \geq \operatorname{Vol}(X) = \left(\frac{\varepsilon}{2n}\right)^n$. □

Now we discuss the Ellipsoid Method, which is an algorithm to solve Problem 5.1, see also [28].

Algorithm 5.3 (Ellipsoid Method)**.** The Ellipsoid Method constructs a sequence of ellipsoids $(E_0, E_1, \dots, E_l)$ with center points $x_0, \dots, x_l$ with the following properties:

1. For every $k \in \{0, \dots, l\}$ the feasible set S is contained in E_k.
2. There exists a constant factor $c \in [0, 1[$ only depending on n such that $\operatorname{Vol}(E_k) \leq c \operatorname{Vol}(E_{k-1})$ for $k \in \{1, \dots, l\}$

A step that constructs from an ellipsoid E_{k-1} the next one E_k is visualized in Figure 8 on page 25.

For a sequence like this, we know that $\mathrm{Vol}(E_k) \leq c^k \mathrm{Vol}(E_0)$. So if $k > l := \frac{\log(\varepsilon)-\log(\mathrm{Vol}(E_0))}{\log(c)}$, we have $\mathrm{Vol}(E_k) < \varepsilon$. So after l steps, we can terminate the algorithm with the answer "$\mathrm{Vol}(S) < \varepsilon$". Now we have to describe how to construct $(E_0, E_1, \ldots, E_l)$.

First of all, we have to find an initial ellipsoid E_0. Let's assume we can find that in polynomial time with $\log(\mathrm{Vol}(E_0))$ is polynomial in n.

Next, we have to describe how to construct E_k from E_{k-1}. Assume without loss of generality, that E_{k-1} is the unit sphere. We can reach this by an affine linear transformation. If the midpoint x_k is in S, we finish returning x_k. Otherwise, we assume that we can find in time polynomial in n an unsatisfied constraint, i.e., an index $j \in \{1, \ldots, m\}$ such that $A_{j,\cdot}x > B_j$. Without loss of generality, assume this constraint states $x_1 > \delta$ for some $\delta > 0$. Since (transformed) S is convex, we know that (transformed) S is contained in the half-ellipsoid $H = \{x \in \mathbb{R}^n : x_1 > 0, \|x\| \leq 1\}$. By careful consideration, one finds out that the ellipsoid

$$E_k = \left\{ x \in \mathbb{R}^n : \frac{1}{(d-1)^2}(x_1 - d)^2 + \frac{1-2d}{(1-d)^2}\left(\sum_{i=2}^{n} x_i^2\right) \leq 1 \right\}$$

contains $\{x \in \mathbb{R}^n : x_1 > 0, \|x\| \leq 1\}$ for $d \in]0, \frac{1}{2}[$. Choosing $d = \frac{1}{n+1}$, we have

$$\frac{\mathrm{Vol}(E_k)}{\mathrm{Vol}(E_{k-1})} = \frac{n}{n+1}\left(\frac{n^2}{n^2-1}\right)^{\frac{n-1}{2}} < \exp\left(-\frac{1}{2(n+1)}\right) =: c$$

For more details, see [28]. In pseudocode, this algorithm reads as follows:

1: Choose the initial ellipsoid $E_0 \supseteq S$ with midpoint x_0.
2: Let $l = 2(n+1)\left(\log(\mathrm{Vol}(E_0)) - \log(\varepsilon)\right)$.
3: **for** $k \in \{1, \ldots, l\}$ **do**
4:     **if** $x_{k-1} \in S$ **then**
5:         **return** x_{k-1}
6:     **else**
7:         Find unsatisfied constraint $A_{j,\cdot}x > B_j$
8:         Construct E_k with mitpoint x_k containing $E_{k-1} \cap \{x \in \mathbb{R}^n : A_{j,\cdot}x > B_j\}$.
9: **return** "$\mathrm{Vol}(S) < \varepsilon$".

In fact, Ellipsoids do not have to be constructed explicitly. Every step takes $\mathcal{O}(n^3)$ time.

For applying Algorithm 5.3 to Problem 4.2, we have to find an initial ellipsoid that contains

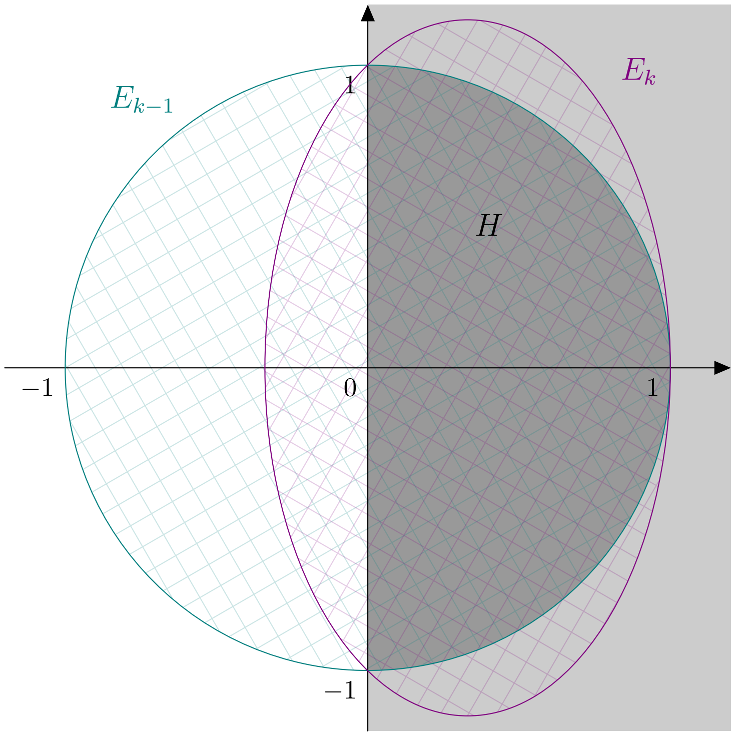


Figure 8: Step from E_{k-1} to E_k in two dimensions: Because we found the unsatisfied constraint $x_1 > 0$ that describes the gray shaded half-plane, the set S is contained in the dark gray shaded area H, which is the intersection of E_{k-1} and the gray shaded half-plane. The ellipsoid E_k contains H and its volume is smaller than the volume of E_{k-1} by a factor bounded by c.

all feasible solutions if feasible solutions exist. Since $x_e \in [0,1]$, we can choose the ball with midpoint $\frac{1}{2}$ in all coordinates and radius $\frac{\sqrt{|E|}}{2}$. Because the volume of an n-dimensional unit ball is $\frac{\sqrt{\pi}^n}{\Gamma(1+\frac{n}{2})}$ (see [10]), where Γ denotes the gamma function, its volume is bounded by $\left(\sqrt{|E|\pi}\right)^{|E|}$. So the logarithm of the volume is bounded by $\frac{|E|}{2}(\log(|E|)+\log(\pi)) \in \mathcal{O}(|E|\log(|E|))$.

Furthermore, we need to find an unsatisfied constraint (pseudocode line 7), which in this case is an odd cycle $C \in \mathcal{C}_{\text{odd}}(G)$ with $\sum_{e_C \in E(C)} x_{e_C} < 1$, or decide that all constraints are satisfied within polynomial time.

We first describe this problem in a more general notation:

Problem 5.4 (Finding an unsatisfied odd cycle in a graph)**.** Given a graph $G=(V,E)$ and a weight function $w : E \to [0,1]$. Task: Find an odd cycle

$$C = (\{v_1,\dots,v_u\},\{\{v_1,v_2\},\dots,\{v_{u-1},v_u\},\{v_u,v_1\}\}) \in \mathcal{C}_{\text{odd}}(G)$$

with sum of edge weights $w(C) := \sum_{e\in E(C)} w(e) = \sum_{i=1}^{u-1} w(\{v_i,v_{i+1}\}) + w(\{v_u,v_1\}) < 1$ or decide that such an odd cycle does not exist.

Now we are giving an algorithm to solve this problem:

Algorithm 5.5 (Solving Problem 5.4)**.** We can solve Problem 5.4 performing the following steps:

1: Evaluate $G^2 = (V,E^2)$ where

$$E^2 = \left\{ \{u,v\} \in \binom{V}{2} : \exists t \in V \ \{u,t\},\{t,v\} \in E \right\}$$

and

$$w^2 : \begin{array}{rcl} E^2 & \to & \mathbb{R} \\ \{u,v\} & \mapsto & \min\limits_{t\in V:\ \{u,t\},\{t,v\}\in E} w(\{u,t\}) + w(\{t,v\}) \end{array}$$

For $e^2 = \{u,v\} \in E^2$ let $m(e^2) = t$ for a vertex t with $w(\{u,t\})+w(\{t,v\}) = w^2(\{u,v\})$.

2: Evaluate the weighted all pairs shortest paths matrix $D \in \mathbb{R}^{V\times V}$ in the graph G^2 with weight function w^2. For $u,v\in V$ let $P(u,v)$ be the weighted shortest path between u and v in G^2. Let $P'(u,v)$ be the path in G obtained from $P(u,v)$ by replacing each edge $e=\{x,y\}$ by $\{x,m(e)\},\{m(e),y\}$.

3: **for** $e=\{u,v\}\in E$ **do**

4:     **if** $w(e) + D_{u,v} < 1$ **then**

5:         **return** "Unsatisfied odd cycle: $(V(P'(u,v)), E(P'(u,v)) \cup \{e\})$"

6: **return** "All odd cycles are satisfied".

The idea behind this method is that every odd cycle is a union of an edge $\{u, v\}$ and a path from u to v containing an even number of edges. Finding shortest paths in the *square graph* G^2 corresponds to finding shortest paths containing an even number of edges in G. As it suffices to find one unsatisfied cycle, it suffices to restrict to search for shortest paths.

Remark 5.6 (Including the binary search method in the Ellipsoid Method)**.** When we have found an ellipsoid E_k with midpoint x_k such that $f(x_k) \leq c$, we can start the Ellipsoid Method for $E(c')$ with $c' < c$ with E_k because by Property 3 in the proof of Lemma 5.2, we have $E(c') \subseteq E(c)$. However, if we chose $c' < c^*$, we have to run the full Algorithm 5.3 until $\mathrm{Vol}(E_k) < \varepsilon$ to decide this.

Concluding, we give a complexity analysis of the above method to solve Problem 4.2.

Lemma 5.7 (Complexity of solving Problem 4.2 with the Ellipsoid Method)**.** *With the above method, Problem 4.2 is solved within time complexity* $\mathcal{O}(|E|^4(|E|^3 + |V|^3)\log(|E|)^2)$.

Proof. For simpler notation, let $|V| = n$ and $|E| = m$.

As $c^* \in Q = \{0 \leq \frac{p}{q} \leq m : q \leq m!\}$, the binary search method on Q takes at most $\log_2(|Q|) \leq \log_2(m(m!)^2) \leq \log_2(m^{2m+1}) = (2m+1)\log_2(m) \in \mathcal{O}(m\log(m))$ steps.

In each step, we apply the Ellipsoid Method constructing not more than $l = (2m+1)(\log(\mathrm{Vol}(E_0)) - \log(\varepsilon))$ ellipsoids. We have $\log(\mathrm{Vol}(E_0)) \leq \log\left((\sqrt{m\pi})^m\right) = \frac{m}{2}(\log(m)+\log(\pi)) \in \mathcal{O}(m\log(m))$. Setting $\varepsilon = \left(\frac{1}{2m(m!)^2}\right)^m$ as in the proof of Lemma 5.2, we have $-\log(\varepsilon) = \log(\frac{1}{\varepsilon}) = m(\log(2) + \log(m) + 2\log(m!)) \leq m(\log(2) + \log(m) + 2\log(m^m)) \in \mathcal{O}(m^2\log(m))$. So in total, $l \in (2(m+1))\,\mathcal{O}(m\log(m)) + \mathcal{O}(m^2\log(m)) = \mathcal{O}(m^3\log(m))$.

Every step of the Ellipsoid Method takes $\mathcal{O}(m^3)$ effort to update the ellipsoid after finding an unsatisfied constraint. So it remains to analyze Algorithm 5.5:

Line 1 can be done in $\mathcal{O}(n^3)$ steps iterating over every triple $u, v, t \in V$.
Line 2 can be done by the Floyd-Warshall-Algorithmus (see [29], pages 177 ff) in $\mathcal{O}(n^3)$ time. We do not have to save all paths explicitly because parts of shortest paths are shortest paths as well. So it suffices to save for every pair $u, v \in V$ the predecessor of v in a shortest u-v-path.

The test in line 4 takes constant time and is repeated $\mathcal{O}(m)$ times.
Within the algorithm, line 5 can only be run once. In this case, we have to evaluate $P'(u, v)$. By repeatedly evaluating the predecessors, we get the path $P(u, v)$. Inserting $m(a, b)$ for a, b

all neighbors a and b in $P(u, v)$, we get $P'(u, v)$. Since the length of this path is bounded by the number of vertices, this can be done in $\mathcal{O}(n)$ time.

Summing up, the complexity of Algorithm 5.5 is $\mathcal{O}(n^3) + \mathcal{O}(n^3) + \mathcal{O}(m) + \mathcal{O}(n)$. Since in a graph $m \leq \binom{n}{2} \in \mathcal{O}(n^2)$, this is $\mathcal{O}(n^3)$ in total. This is in dense graphs less than the complexity of the ellipsoid updating step, which is $\mathcal{O}(m^3)$.

So for Problem 4.2, we get a complexity of $\mathcal{O}(m \log(m)) \cdot \mathcal{O}(m^3 \log(m)) \cdot (\mathcal{O}(m^3) + \mathcal{O}(n^3)) = \mathcal{O}(m^4(m^3 + n^3) \log(m)^2)$ □

> **Remark 5.8.** We have seen in Lemma 3.2, Property 5 that crossing graphs of rectilinear drawings of complete graphs are dense, i.e., $|E| \in \Theta(|V|^2)$. This not yet introduced Landau notation stands for a both-sided estimation, i.e., $|E| \in \mathcal{O}(|V|^2)$ and $|V|^2 \in \mathcal{O}(|E|)$. So the complexity of Problem 4.2 is $\mathcal{O}(|E|^7 \log(|E|)^2)$ in this case.

All in all, this method is more of theoretical interest, but hard to implement in praxis because the Ellipsoid method is numerically quite instable.

6 Upper bound by local optimization, gadget heuristics

In this chapter, we discuss how to find upper bounds for the monochromatic crossing number $\operatorname{cr}(G, \mathcal{D}, 2)$ for a graph $G = (V, E)$ and a drawing $\mathcal{D}$ of G. In contrast to Section 5, we are not only evaluating a bound b but will also give a 2-edge coloring $c_E : E \to \{0, 1\}$ that admits b, i.e., $|\operatorname{CrM}(G, \mathcal{D}, c_E)| = b$.

Because Problem 1.7 can be reduced to Problem 3.7, we can alternately discuss the problem on the crossing graph $G_\times$. In this way of looking at it, the 2-edge coloring c_E corresponds to a 2-vertex coloring of $G_\times$. So we can use properties of the crossing graph.

Observation 6.1 (Every coloring yields an upper bound)**.** Let $G = (V, E)$ be a graph, $\mathcal{D}$ a rectilinear drawing of G and $c_E : E \to \{0, 1\}$ a 2-edge coloring of G. Then, by definition, $|\operatorname{CrM}(G, \mathcal{D}, c_E)| \geq \operatorname{cr}(G, \mathcal{D}, 2)$. So any 2-edge coloring of G yields an upper bound to $\operatorname{cr}(G, \mathcal{D}, 2)$.

Observation 6.2 (Quality of the bound for complete graphs)**.** By Remark 2.8, we know that for any rectilinear drawing of K_{10} there exists an 3-crossing family. So setting $a = 2$ and $k = 10$ in Lemma 2.7, we get for $h \in \{10, 11, \ldots\}$:

$$\begin{aligned}\operatorname{cr}(K_h, 2) &\geq \frac{\binom{h}{10}}{\binom{h}{10} - \left(\sum_{j=0}^{3} 2^j \binom{3}{j}\binom{h-6}{10-j}\right) - 3\left(\sum_{j=0}^{2} 2^j \binom{2}{j}\binom{h-6}{8-j}\right)} \\ &= \frac{(h-1)(h-2)(h-3)(h-4)(h-5)}{15120\,(h-9)} =: A(h)\end{aligned}$$

For any 2-edge coloring c_E and any drawing $\mathcal{D}$ of K_h we have $|\operatorname{CrM}(K_h, \mathcal{D}, c_E)| \leq |\operatorname{Cr}(K_h, \mathcal{D})| \leq \binom{h}{4}$ due to Lemma 3.2, Property 5. So the approximation ratio is

$$\begin{aligned}\frac{|\operatorname{CrM}(K_h, \mathcal{D}, c_E)|}{\operatorname{cr}(K_h, 2)} \leq &\frac{\binom{h}{4}}{A(h)} = \frac{630\,(h-9)h}{(h-4)(h-5)} \\ =&630\frac{h^2 - 9h}{h^2 - 9h + 20} = 630\left(1 - \frac{20}{(h-4)(h-5)}\right) \\ \leq&630\end{aligned}$$

This implies, that coloring the edges of K_h arbitrarily is a 630-approximation of Problem 1.7 for every $h \in \{10, 11, \ldots\}$.

In order to find colorings with low monochromatic crossing number, we can do local optimization.

Heuristic 6.3 (Local optimization of the coloring)**.** For a graph $G = (V, E)$, a rectilinear drawing of $\mathcal{D}$ of G and some bound $r \in \mathbb{N}$, the canonical local optimization heuristic for finding a 2-edge coloring of G is given by:

1: Initialise an arbitary or random coloring $c_E : E \to \{0, 1\}$.
2: **for** $j \in \{1, \ldots, r\}$ **do**
3:     **for** $S \in \binom{E}{j}$ **do**
4:         Let $c'_E(e) = \begin{cases} 1 - c_E(e) & \text{if } e \in S \\ c_E(e) & \text{if } e \notin S \end{cases}$
5:         **if** $|\operatorname{CrM}(G, \mathcal{D}, c'_E)| < |\operatorname{CrM}(G, \mathcal{D}, c_E)|$ **then**
6:             Replace c_E by c'_E.
7:             Go to line 2
8: **return** c_E

Observation 6.4 (Properties of local optimization)**.**

1. The resulting coloring c_E is r-optimal, i.e, for every subset S of E with $|E| \leq r$, changing the coloring on S does not improve the monochromatic crossing number. Since c_E is 1-optimal, every edge $e \in E$ crosses at least as many differently colored edges than edges in the same color. Hence, it satisfies the Probabilistic bound from Lemma 2.5. Especially, we have $|\operatorname{CrM}(K_h, \mathcal{D}, c_E)| \leq \frac{1}{2}\binom{h}{4}$. Analogously to Observation 6.2, we find that Heuristic 6.3 is a 315-approximation on rectilinear drawings of complete graphs with $h \geq 10$.

2. The initialization in line 1 can be done in $\mathcal{O}(|E|)$ time. Whenever line 7 is reached, the coloring c_E was improved by at least one. Hence, we execute line 7 at most $|\binom{E}{2}| \in \mathcal{O}(|E|^2)$ times. In the **for** loops in lines 2 and 3, the set S is assigned $\sum_{j=1}^{r} \binom{|E|}{j} \in \mathcal{O}(|E|^r)$ many times. In a direct implementation, lines 4 and 5 can be done in $\mathcal{O}(\binom{|E|}{2}) = \mathcal{O}(|E|^2)$ time. This gives an overall running time for Heuristic 6.3 of $\mathcal{O}(|E|^2)\,\mathcal{O}(|E|^r)\,\mathcal{O}(|E|^2) = \mathcal{O}(|E|^{r+4})$, which is exponential in r.

3. If c^*_E is an optimal 2-edge coloring of $\mathcal{D}$, then $1 - c^*_E$, the coloring obtained from c^*_E by flipping all colors, also is. Let c_E, c'_E be 2-edge colorings. Denote by $\operatorname{diff}(c_E, c'_E) = \{e \in E : c_E(e) \neq c'_E(e)\}$. Since for any 2-edge coloring c_E we have $\operatorname{diff}(c_E, c^*_E) \,\dot\cup\, \operatorname{diff}(c_E, 1 - c^*_E) = E$, we obtain $\operatorname{diff}(c_E, c^*_E) \leq \left\lfloor \frac{|E|}{2} \right\rfloor$ or $\operatorname{diff}(c_E, 1 - c^*_E) \leq \left\lfloor \frac{|E|}{2} \right\rfloor$. So for $r \geq \left\lfloor \frac{|E|}{2} \right\rfloor$ the coloring c_E obtained by Heuristic 6.3 is optimal.

4. In praxis, we obtain better colorings performing Heuristic 6.3 if we do lines 6 and 7 not only for a strict inequality in line 5, but also with probability $\frac{1}{2}$ if the inequal-

ity is sharp. Instead of only terminating when reaching line 8, we also terminate after a previously defined number of iterations in which no improvement was made. Otherwise we do not know any bound on the running time for the modified heuristic.

We can improve Heuristic 6.3 a little bit. In the following, we will analyze some improvements on the crossing graph $G_\times$. All that changes in the formulation of Heuristic 6.3 is that edges will be replaced by vertices and crossings will be replaced by edges.

Remark 6.5 (Improving check)**.** The coloring c'_E in Heuristic 6.3, line 4 and the number of monochromatic crossings (edges) in line 5 do not have to be evaluated explicitly. We can instead evaluate lines 4 and 5 in the following way:

Let $G = (V, E)$ be a graph (e.g. a crossing graph), c_V a 2-vertex coloring and $G' = (V', E')$ an induced subgraph of G. For $e = \{v, w\} \in E$ we define:

$$\mathbb{I}_{c_V}(e) := \begin{cases} 1 & \text{if } c_V(v) \neq c_V(w) \\ -1 & \text{if } c_V(v) = c_V(w) \end{cases}$$

For $v \in V'$ we define:

$$\begin{aligned} g(G', c_V, v) :=& \sum_{w \in \Gamma_{G'}(v)} \mathbb{I}_{c_V}(\{v, w\}) \\ =& |\{w \in \Gamma_{G'}(V) : c_V(v) \neq c_V(w)\}| - |\{w \in \Gamma_{G'}(V) : c_V(v) = c_V(w)\}| \end{aligned}$$

Furthermore, we define

$$\begin{aligned} \mathcal{G}(G', c_V) :=& \sum_{v \in G'} g(G', c_V, v) = 2 \sum_{e \in E'} \mathbb{I}_{c_V}(e) \\ =& 2(|\{e \in E' : e \text{ bichromatic}\}| - |\{e \in E' : e \text{ monochromatic}\}|) \end{aligned}$$

So for two 2-vertex colorings c_V and c'_V of G, we have less monochromatic edges in G' with respect to c'_V than with respect to c_V iff $\mathcal{G}(G', c'_V) > \mathcal{G}(G', c_V)$. More precisely, the difference in the number of monochromatic edges is $\frac{1}{4}\left(\mathcal{G}(G', c'_V) - \mathcal{G}(G', c_V)\right)$.

Let G' be the subgraph of G induced by S. Since the coloring c'_V defined in line 4 differs from the coloring c_V exactly on the set S, we can evaluate if G has less monochromatic edges with respect to c'_V than to c_V by the following:

$$\begin{aligned} &\mathcal{G}(G, c'_V) > \mathcal{G}(G, c_V) && \Leftrightarrow \\ &\sum_{e \in E} \mathbb{I}_{c'_V}(e) > \sum_{e \in E} \mathbb{I}_{c_V}(e) && \Leftrightarrow \end{aligned}$$

$$0 > \left(\sum_{\substack{\{u,v\}\in E \\ u,v\in S}} +2 \sum_{\substack{\{u,v\}\in E \\ u\in S \\ v\in V\setminus S}} + \sum_{\substack{\{u,v\}\in E \\ u,v\in V\setminus S}} \right) \left(\mathbb{I}_{c_V}(\{u,v\}) - \mathbb{I}_{c'_V}(\{u,v\})\right) \quad \Leftrightarrow$$

$$0 > \sum_{\substack{\{u,v\}\in E \\ u\in S \\ v\in V\setminus S}} \mathbb{I}_{c_V}(\{u,v\}) \quad \Leftrightarrow \quad (1)$$

$$0 > \sum_{\substack{\{u,v\}\in E \\ u\in S \\ v\in V\setminus S}} \mathbb{I}_{c_V}(\{u,v\}) + 2\sum_{e\in E'} \mathbb{I}_{c_V}(e) - \mathcal{G}(G', c_V) \quad \Leftrightarrow$$

$$\mathcal{G}(G', c_V) > \sum_{v\in S} g(G, c_V, v) \quad (2)$$

To check line 5 of Heuristic 6.3, it suffices to check (2). For the initial coloring in line 1, the vector $g = (g(G, c_V, v))_{v\in V}$ can be evaluated in $\mathcal{O}(|E|) = \mathcal{O}(|V|^2)$ time since every edge contributes to exactly two vertices. Lines 4 and 5 then take $\mathcal{O}(r^2)$ time for each iteration. In line 6, we have to update g. This can again be done in $\mathcal{O}(|E|)$ time. Since an update is only done if the new coloring has fewer monochromatic edges, we perform line 6 only $\mathcal{O}(|E|)$ times. So we are left with an overall running time for Heuristic 6.3 of $\mathcal{O}(|V|^2)\,\mathcal{O}(|V|^r)\,\mathcal{O}(r^2) = \mathcal{O}(|V|^{r+2}r^2)$.

In line 5 we are searching for a set $S \in \binom{E}{j}$ if switching the colors in S improves the coloring given that every smaller set $S' \in \binom{E}{j'}$ with $j' < j$ does not have this property. We show that such a set S must have a specific structure:

Lemma 6.6 (Improving subsets are uniquely colored)**.** *Let $G = (V, E)$ be a graph and c_V a 2-vertex coloring. A set $S \in \binom{E}{j}$ as chosen in Heuristic 6.3, line 3 can only give a true statement in line 5 if the by S induced subgraph $G' = (S, E')$ has a unique minimal 2-coloring (up to switching all colors). This coloring is attained by $c_V|_S$.*

Proof. Let's assume S gives a true statement in Heuristic 6.3, line 5, i.e., it satisfies (1) and there exists a 2-coloring c^*_S of S with $c^*_S \neq c_V|_S$ and $c^*_S \neq 1 - c_V|_S$ that minimizes the number of monochromatic edges in G'. Let $M = \{v \in S : c_V(v) = c^*_S(v)\}$. Then $M \neq S$ and $M \neq \emptyset$. Because of the assumption that S is minimal, we have

$$\sum_{\substack{\{u,v\}\in E \\ u\in M \\ v\in V\setminus M}} \mathbb{I}_{c_V}(\{u,v\}) \geq 0 \quad \text{and} \quad \sum_{\substack{\{u,v\}\in E \\ u\in S\setminus M \\ v\in V\setminus(S\setminus M)}} \mathbb{I}_{c_V}(\{u,v\}) \geq 0$$

Since $V \setminus M = (V \setminus S)\,\dot\cup(S \setminus M)$, $V \setminus (S \setminus M) = (V \setminus S)\,\dot\cup\, M$ and $S = M\,\dot\cup(S \setminus M)$, adding those two inequalities and adding (1) gives

$$\sum_{\substack{\{u,v\}\in E \\ u\in M \\ v\in S\setminus M}} \mathbb{I}_{c_V}(\{u,v\}) > 0 \quad \Leftrightarrow \quad \mathcal{G}(G', c_V|_S) > \mathcal{G}(G', c_S^*(v))$$

This means that G' has less monochromatic edges with respect to c_V than with respect to $c_S^*(v)$. This contradicts the optimality of $c_S^*(v)$. So if S satisfies line 5, such a coloring $c_S^*(v)$ cannot exist. Hence, $c_V|_S$ is (up to switching all colors) the unique minimal coloring of S. □

However, the set of graphs with the property in Lemma 6.6 is exponential in the number of vertices, as it is obviously a superset of the set of connected bipartite graphs. Anyway, the subgraph isomorphism problem, i.e., the problem to decide if a graph G' is a subgraph of a graph G is NP-hard, see [7]. Hence, the problem to find all subgraphs of a graph isomorphic to a certain graph is hard as well.
There is no further restriction known that reduces the effort for Heuristic 6.3 significantly. So there is not a lot of hope to be able to transform it into a polynomial time algorithm.

7 Equivalence to Max-Cut

Another equivalent problem to Problems 1.7, 3.7 and 3.10 on the crossing graph is given by:

> **Problem 7.1** (Maximum cut problem)**.** Given a graph $G = (V, E)$. Task: Find a bipartition $V = A \dot\cup B$ such that the number of edges that connect vertices in A with vertices in B, i.e, the cardinality of the set $\{\{a, b\} \in E : a \in A, b \in B\}$ is maximized.

Proof of the equivalence. Let $V = A \dot\cup B$ be a bipartition. Then this bipartition induces a 2-vertex coloring:

$$c_V(v) = \begin{cases} 1 & \text{if } v \in A \\ 0 & \text{if } v \in B \end{cases}$$

On the other hand, any 2-vertex coloring $c_V : V \to \{0, 1\}$ defines a bipartition $A = \{v \in V : c_V(v) = 1\}$, $B = \{v \in V : c_V(v) = 0\}$ of the vertex set.

By a coloring or a bipartition of the vertex set, we partition the edge set into the set of monochromatic edges

$$E_1 = \{e \in E : e \in A^2 \vee e \in B^2\} = \{\{u, v\} \in E : c_V(u) = c_V(v)\}$$

and the set of bichromatic edges

$$E_2 = \{\{a, b\} \in E : a \in A, b \in B\} = \{\{u, v\} \in E : c_V(u) \neq c_V(v)\}$$

Since $E = E_1 \dot\cup E_2$, we have $|E| = |E_1| + |E_2|$. So maximizing $|E_2|$ is equivalent to minimizing $|E_1|$. $\square$

By equivalence to an NP-hard problem, Problem 7.1 is also NP-hard for general graphs. However, Problem 7.1 is solvable in polynomial time for planar graphs by translating into a maximum weighted matching problem, see [16].

One might wonder if geometric properties in the drawing carry over to the crossing graph such that we can adapt the algorithm solving Max-Cut in polynomial time to crossing graphs. However, in [6] it was shown that on crossing graphs of rectilinear drawings of non necessarily complete graphs, Problem 7.1 still is NP-hard.

Nevertheless, one can still apply algorithms that find a maximum cut to solve Problem 1.7. Some of these are given in [14, 15]. For general graphs, their worst case running time is exponential. It is not known, if this also holds for crossing graphs of rectilinear drawings of complete graphs.

8 Results for order types of small cardinality

Observation 8.1 (Comparison of bounds)**.** Let $G = (V, E)$ be a graph and $\mathcal{D}$ a rectilinear drawing of G. In Section 5, we have developed a method to find a lower bound $L(G, \mathcal{D})$ such that $L(G, \mathcal{D}) \leq |\operatorname{CrM}(G, \mathcal{D}, c_E)|$ for every 2-edge coloring c_E of G. If we find a 2-edge coloring c_E^* with $L(G, \mathcal{D}) = |\operatorname{CrM}(G, \mathcal{D}, c_E^*)|$, then c_E^* is optimal, i.e., it solves Problem 1.7.

By Observation 8.1, we get a sufficient criterion for a 2-edge coloring to solve Problem 1.7. This leads to the idea to solve Problem 1.7 by a two way algorithm: On one hand we try to obtain a coloring c_E with few monochromatic crossings by Local Search (see Section 6) and on the other hand evaluating lower bounds by LP-relaxation (see Section 5) we can check if c_E is already optimal. If the check is positive, we are done. If the check is negative, we have to try to improve either the coloring or the lower bound. In pseudocode, this reads as follows:

Algorithm 8.2 (Comparing bounds algorithm)**.**

Evaluate a lower bound $L(G, \mathcal{D})$ by the Ellipsoid method.
repeat
  repeat
    Evaluate a coloring c_E by Local Optimization.
  until $L(G, \mathcal{D}) = |\operatorname{CrM}(G, \mathcal{D}, c_E)|$ or a cretain bound on the number of computations or some timeout is reached.
  if $L(G, \mathcal{D}) = |\operatorname{CrM}(G, \mathcal{D}, c_E)|$ **then**
    return c_E
  Investigate more on the lower bound by methods mentioned in Remark 4.6 to obtain a new bound $L' \geq L$ that is at least as good as the old one.
  Replace L by L'.
  if $L(G, \mathcal{D}) = |\operatorname{CrM}(G, \mathcal{D}, c_E)|$ **then**
    return c_E
until some **return** statement is reached.

As neither Local Search nor the lower bound method in Remark 4.6 are polynomial in general, Algorithm 8.2 is not supposed to be polynomial either. But there is hope, that for many drawings $\mathcal{D}$ we do not have to apply an exact method with exponential running time to find out the 2-crossing number of $\mathcal{D}$, but finish by finding a coloring that attains the lower bound. In fact, it turns out that for most rectilinear drawings of small complete graphs, a slightly adapted version of this approach yields good results.

For investigations on the crossing number of rectilinear drawings, O. Aichholzer generated a database containing a representative for each class of essentially equal rectilinear drawings (see Definition 3.4) of the complete graph. A representative is given by an 8-bit or a 16-bit integer coordinate representation on the order type homepage, see [1].

The problem to obtain a representative of every class of essentially equal rectilinear drawings is not trivial. It was solved by transforming the line arrangement based on some point set by a duality approach into a pseudoline arrangement. It is shown that every rectilinear drawing corresponds to some extension of a smaller pseudoline arrangement. But on the other hand, it takes a lot of effort to decide if a certain pseudoline arrangement can be realized as an order type, i.e., a drawing. For more details see [4].

The number of essentially equal rectilinear drawing classes of the complete graph grows rapidly with the number of points, see Table 3.

n	3	4	5	6	7	8	9	10	11
$\lvert (K_n)_\sim \rvert$	1	2	3	16	135	3 315	158 817	14 309 574	2 334 512 907

Table 3: Number of order types for the complete graph up to 11 vertices.

For this thesis we analyze the rectilinear 2-edge coloring crossing number, i.e., solve Problem 1.7 for all rectilinear drawings $\mathcal{D}$ (called "order types" in [4]) of small complete graphs. For this thesis, we stop at ten vertices because we don't have enough computational power to apply the approach to eleven vertices as well.

Approach 8.3 (Applying Algorithm 8.2)**.** Due do practical and implementation reasons, we apply an adapted version of Algorithm 8.2:

1. First evaluate a coloring using Heuristic 6.3 with $r = 2$ in the variant given in Observation 6.4, property 4 several times and pick a best sample c_E.

2. Because of numerical instability and bad running time for practical purposes, we did not implement the Ellipsoid method for evaluating the lower bound, but used the GLPK solver (see [24]) that comes with the computer algebra system sage (see [27]). For this, we solved the linear program LP_i only containing constraints for odd cycles of length at most i for $i \in \{3, 5, 7\}$ and compared it with $|\,\mathrm{CrM}(K_k, \mathcal{D}, c_E)|$. If the bound coincides with the number of monochromatic crossings in the coloring, we are done. Otherwise, we iteratively add constraints for unsatisfied cycles like in Algorithm 5.5.

3. Continue on using Heuristic 6.3 with $r = 3$ and $r = 4$ several times. If we find a coloring c_E that attains the previously evaluated lower bound, we are done.

4. Solve the integer linear program ILP_i only containing constraints for odd cycles of length at most i for $i \in \{5, 7\}$ with GLPK. Check after each solved ILP if removing all edges $e \in E_\times$ with $a_e = 1$ in the solution of the ILP from the crossing graph $G_\times$ yields a bipartite graph. If so, reconstruct an optimal coloring. Otherwise add constraints for unsatisfied cycles like described in Algorithm 5.5.

Remark 8.4 (Implementation of the algorithm)**.** The computations were done via the Python based software package Sage, see also [27]. The source code of both evaluating the lower bound by the linear program and the upper bound by local optimization are published online as a sage worksheet, see [12].

Observation 8.5 (Behavior of Approach 8.3)**.**

1. The bound obtained from LP_3 is quite weak. For at least seven points, in most instances, this bound was not tight. This is not very surprising because five points in convex position generate a 5-cycle in the crossing graph but no 3-cycles.
2. Nevertheless, the bound obtained from LP_5 works fine for a lot of instances. Even for 12 vertices, in a test with 1000 randomly generated pointsets the LP_5 solution agrees with some coloring found by Heuristic 6.3 repeating it for 20 random initial colorings and $r = 3$ in all but two, which gives 99.8% of the cases. However, Local Search might get stuck in quite bad local optima. There exist 3-optimal solutions that have over 20 more monochromatic crossings than optimal ones. An example is given in the appendix in Section 11.2. For more details see [12]
3. Among all drawings of 10 points, there however exist instances where the LP_7 bound is strictly better than the LP_5 bound.
4. Properties of the LP-bound:
   (a) For all pointsets with at most 6 points, the LP-bound is tight.
   (b) For 7 points, there exists exactly one drawing where the bound is not tight. This is the case if they lie in convex position.
   (c) For 8 points there is also exactly one instance where the LP-bound is not tight. This is given for 7 points $v_1, \dots, v_7$ in convex positon, and one point is added where the straight line segments from v_8 to $v_1, \dots, v_7$ do not cross any other line semgent, see Figure 9.
   (d) For nine points, there are 11 drawings and for ten points, there are 965 drawings

where the LP bound is not tight.

5. There might be hope that according to the structure of the crossing graph of a rectilinear drawing of a complete graph, only certain subgraphs that satisfy the condition in Lemma 6.6 eventually appear using Heuristic 6.3. However, all of these subgraphs, that are also called "gadgets", for $r \leq 4$ and most of them for $r \leq 6$ were used during a test of Heuristic 6.3 on all graphs on 10 vertices where the LP-bound is not tight. So there is little hope that Heuristic 6.3 can be converted to a polynomial time approach.

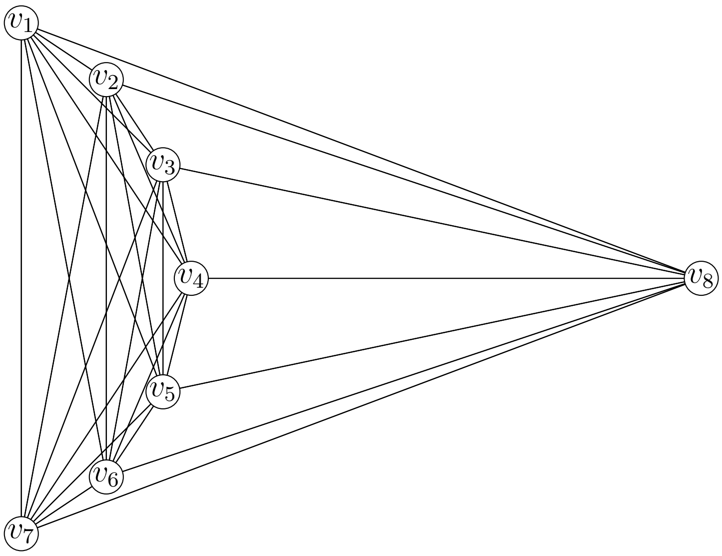


Figure 9: Rectilinear drawing of K_8 with LP-bound 8.4, which is not integral and hence cannot be tight.

To find the monochromatic crossing number $\mathrm{cr}(K_{10}, \mathcal{D}, 2)$ for every drawing $\mathcal{D}$ of K_{10}, it took more than 3000 CPU hours with the Sage implementation in [12] on a standard desktop computer. Although the approach is polynomial on each instance, the computational effort grows rapidly with the number of points because the number of order types, i.e., the number of essentially equal rectilinear drawing classes, is exponential in the number of points. So within the available computational power, it was not possible to continue with $\mathrm{cr}(K_{11}, \mathcal{D}, 2)$ for all drawings $\mathcal{D}$ of K_{11}, because evaluating the lower bound solving LP_5 requires a lot of computational effort.

However, it was possible to determine $\mathrm{cr}(K_{11}, 2) = 10$ using Observation 2.4: It suffices to evaluate a lower bound ≥ 10 for all those rectilinear drawings where every subset of ten points has a lower bound of at most 9 to show that the the minimal number of monochromatic crossings over all drawings is 10. Analyzing the database, O. Aichholzer found out, that only 24410 rectilinear drawings have this property. See [12] for the check that all these drawings $\mathcal{D}$ have $\mathrm{cr}(K_{11}, \mathcal{D}, 2) \geq 10$. For a summary of the monochromatic crossing numbers, see Table 4.

k	≤ 8	9	10	11
$\lvert\operatorname{cr}(K_k, \mathcal{D}, 2)$	0	2	5	10

Table 4: 2-colored monochromatic crossing number for the complete graph up to 11 vertices.

9 Generalizations and related problems

9.1 More colors

One obvious generalization of Problem 1.7 is to allow more than two colors. Then the problem reads as follows:

> **Problem 9.1.** Given a rectilinear drawing $\mathcal{D}$ of a complete graph K_k for some $k \in \mathbb{N}$ and a number $a \in \mathbb{N}$. Find an a-edge coloring that minimizes the number of monochromatic crossings.
>
> In other words, find an a-edge coloring c_E with $|\operatorname{CrM}(\mathcal{D}, c_E)| = \operatorname{cr}(\mathcal{D}, a)$.

For Problem 9.1, Observations 2.1 to 2.4 and 2.6 and Lemmas 2.5 and 2.7 still apply, since they do not depend on the restriction that the number of available colors is two.

Problem 9.1 can be still translated into a problem on the crossing graph. Analogously to Problem 3.7, we see that Problem 9.1 is equivalent to the following problem:

> **Problem 9.2** (Minimum vertex coloring problem on the crossing graph)**.** Given $\times(\mathcal{D}) = (V_\times, E_\times)$ the crossing graph of a rectilinear drawing of a complete graph. Find an a-vertex coloring that minimizes the number of monochromatic edges.
>
> $$\min_{c_{E_\times} \text{ is an } a\text{-coloring of } E_\times} \left|\{\{e, f\} \in E_\times : c_{E_\times}(e) = c_{E_\times}(f)\}\right|$$

But when it comes to Section 4, the above approach cannot be translated for this problem, because here we used the characterization that a graph is bipartite iff it does not contain an odd cycle, see [23] and [21], Proposition 2.27 on page 38. Such a characterization does not exist for a-partite graphs. Moreover, the decision problem if a given graph is a-partite is NP-hard, see [29], page 84. Hence, Sections 5 and 8 also do not apply for this problem. An adapted version of Heuristic 6.3 can be used. Instead of just changing the color, one has to check for all possibilities of different colors if an improvement is possible. When it comes to Section 6, Observation 6.1 still holds, but we lose Observation 6.2. Lemma 6.6 does not generalize on more than two colors.

9.2 Non-complete graphs

Another quite obvious generalization is to allow general graphs instead of complete graphs only. It is shown that in the general setting, the problem is NP-complete, see Section 7 and [6].

In Observations 2.6, 6.2 and 8.5, Lemma 2.7, and Corollary 2.9, as well as in Lemma 3.2, property 5, we use the property that the given graph is complete. All other approaches directly carry over to the non-complete case.

9.3 Minimizing over drawings

Given a graph G, it might also be interesting not only to minimize over all colorings c_E of a given drawing $\mathcal{D}$ of a graph, but also minimizing over all drawings $\{\mathcal{D}$ drawing of $G\}$. This leads to the following problem:

> **Problem 9.3.** Given a graph G. The task is to find a drawing $\mathcal{D}$ and an 2-edge coloring that minimizes the number of monochromatic crossings.
>
> In other words, find a 2-edge coloring c_E and a drawing $\mathcal{D}$ with $\mathrm{cr}(G, 2) = |\operatorname{CrM}(G, \mathcal{D}, c_E)|$.

For $G = K_k$ with $k \in \{1, \ldots, 11\}$, this problem is solved, see Table 4. In general, a valid, but exponential time approach that solves this problem is to evaluate $\mathrm{cr}(G, \mathcal{D}, 2)$ for every rectilinear drawing $\mathcal{D}$ of G as we did in Section 8. This is very hard as it is even hard to determine all drawings for a given $k \in \mathbb{N}$. Moreover, we can use Observation 2.4 if information on the crossing numbers of subgraphs is available.

One can also apply a local search heuristic on the set of dawings of G by e.g. vertex flips, i.e., shifting vertices across lines in the complete line arrangement induced by the points. However, we do not know anything about the performance of this heuristic. We even do not know if minimal colorings of drawings that only differ by one vertex that is shifted to the other side of a line has any structure in common with the unshifted one.

In general, there is little hope to find an efficient algorithm that solves Problem 9.3.

9.4 Non-rectilinear drawings

One more assumption we can omit is that we draw the graph in the plane rectilinearly. For this, we must generalize the term "drawing" to the non-rectilinear case.

Definition 9.4 (Drawing of a graph)**.** Let $G = (V, E)$ be a graph. A *drawing of a graph* into a topological space T is a pair of a map $\iota : V \to T$ and a family of continuous maps $(\iota_e : [0, 1] \to T)_{e\in E}$ with the property that $\iota_e(\{0, 1\}) = \iota(e)$ and $\iota_e(]0, 1[) \cap \iota(V) = \emptyset$ for all $e \in E$.

It is reasonable to assume that ι is an injective map.

The rectilinear case is covered by this definition choosing $T = \mathbb{R}^2$ and any ι_e for $e = \{u, v\}$ is the line segment between the drawn end points $\iota(u)$ and $\iota(v)$, i.e., $\iota_e(\alpha) = \alpha\iota(u) + (1 - \alpha)\iota(v)$ for $\alpha \in [0, 1]$, given that the vertices u, v are sorted in some (e.g. lexicographical) order.

In this more general case we have to declare what crossing means. This is not as simple as in the rectilinear case, because a lot of special cases, e.g. tangenital arcs, self-intersections or singular points can occur. In the literature, there are several non-equivalent definitions of crossings for general drawings. As an example, we will consider the following definition of crossings.

Definition 9.5 (Crossings of a drawing of a graph)**.** Let $G = (V, E)$ be a graph and $I = (\iota, (\iota_e)_{e\in E})$ a drawing of G into a topological space T. A *crossing* of I (or loosely speaking of G if the applied drawing is clear) is an intersection of drawn edges, i.e., a crossing is a two-element set of the form $C = \{(\iota_e, t_e), (\iota_f, t_f)\}$ where $e, f \in E$ such that $\iota_e(t_e) = \iota_f(t_f)$ with $t_e, t_f \in]0, 1[$. In this case we say that the edges e and f *cross*, C is called a *crossing of e and f* and the point $P = \iota_e(t_e) \in T$ is called a *crossing point* of the edges e and f.

The crossing graph of such a general drawing is not necessarily a simple graph, if we take multiple crossings of pairs of edges into account. For this, we define: A *multi-graph* is a graph where edges have a multiplicity, i.e., there is a function $\mathrm{mlt} : E \to \mathbb{N} \cup \{\infty\}$ where $\mathrm{mlt}(e)$ is called the *multiplicity* of the edge e.

A coloring of edges carries over to a coloring of the drawing and to the corresponding coloring of the crossing graph in a canonical way.

Definition 9.6 (coloring of embeddings and crossings)**.** Let $G = (V, E)$ be a graph and $c_E : E \to \{1, \ldots, a\}$ an a-edge coloring for some $a \in \mathbb{N}$. $I = (\iota, (\iota_e)_{e\in E})$ be a drawing of G into a topological space T. Then we transmit the edge-coloring of G to a coloring of $\{\iota_e : e \in E\}$ by $c_E(\iota_e) = c_E(e)$. A crossing $C = \{(\iota_e, t_e), (\iota_f, t_f)\}$ is called *monochromatic* if $c_E(\iota_e) = c_E(\iota_f)$. In this case, we define the *color of the crossing* C by $c_E(C) = c_E(\iota_e)$. We denote by $\mathrm{Cr}(I)$ the set of all crossings of I, by $\mathrm{CrM}(I, c_E)$ the set of all monochromatic crossings of I for the coloring c_E.

We generalize the terms of crossing numbers:

Definition 9.7 (crossing number)**.** Let $G = (V, E)$ be a graph and $I = (\iota, (\iota_e)_{e\in E})$ be an embedding of G into a topological space T. Then the *a-crossing number of I*, denoted $\mathrm{cr}(I, a)$, is defined by the minimum number of monochromatic crossings of I among all a-edge colorings of G, i.e.,

$$\mathrm{cr}(I, a) = \min_{c_E : E \to \{1,\dots,a\}} |\mathrm{CrM}(I, c_E)|$$

The *a-crossing number of G*, denoted $\mathrm{cr}(G, a)$, (with respect to a topology T) is defined by the minimum number of monochromatic crossings of any embedding J into T.

$$\mathrm{cr}(G, a) = \mathrm{cr}(G, a, T) = \min_{J \text{ embedding of } G \text{ into } T} \mathrm{cr}(J, a)$$

In the general formulation, it is hard to draw any relevant consequences. But there are some special cases which still can be interesting. Remaining with $T = \mathbb{R}^2$ and restricting to *simple drawings* will give an interesting problem that is quite related.

Definition 9.8 (simple drawings)**.** Let $G = (V, E)$ be a graph. A drawing $I = (\iota, (\iota_e)_{e\in E})$ of G into $\mathbb{R}^2$ where $(\iota_e)_{e\in E}$ are Jordan curves, i.e., regular and continuously differentiable curves, is called *simple* if it satisfies the conditions from Observation 2.1 and every crossing is non-degenerated, i.e., the crossing curves are not tangential.

One might wonder, if among all drawings, monochromatic crossing minimizing drawings are simple. However, this is not evident, and still an open problem. But one can show a much weaker statement: Let $G = (V, E)$ be a graph, $I = (\iota, (\iota_e)_{e\in E})$ a drawing of G in $\mathbb{R}^2$ and c_E an a-edge coloring of G such that $\mathrm{cr}(G, a) = |\mathrm{CrM}(I, c_E)|$. Then the induced sub-drawings $(\iota, (\iota_e)_{e\in E_j})$ for $E_j = \{e \in E : c_E(e) = j\}$ for $j \in \{1, \dots, a\}$ are simple drawings. The indirect proof of this can be done for each color class analogously to the proof for the uncolored problem, which you can find in [2].

An interesting matter of fact is that simple drawings with less than nine vertices behave like rectilinear drawings, i.e., every crossing graph of a simple drawing of at most eight points appears also as a crossing graph of a rectilinear drawing. When we consider graphs with nine or more points, this is not true any more because of Pappus's hexagon theorem, named after the ancient Greek mathematician Pappus of Alexandria [8].

Hence, Properties 1, 2 and 3 in Remark 2.8 carry over to simple drawings.

But the minimal monochromatic crossing number for simple drawings of K_9 is strictly smaller than the rectilinear one, see Figure 10.

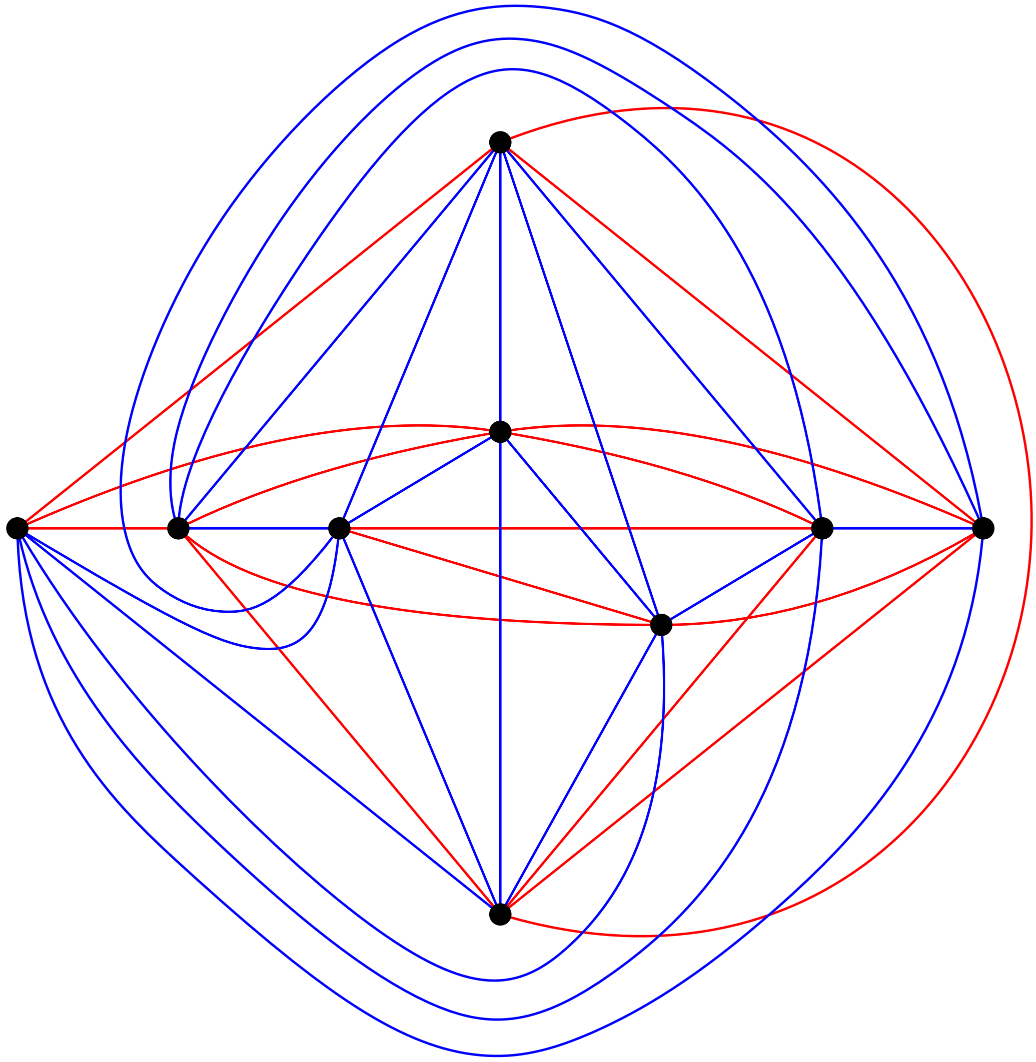


Figure 10: Within simple drawings, K_9 can be drawn with only one monochromatic crossing. (Figure taken from [5])

Of course, one can also investigate what happens for different topologies than $\mathbb{R}^2$. In this case, the set of possible drawings strongly depends on the topology. For example, for some topologies (e.g. $\mathbb{R}^3$) every graph can be drawn crossing-free. The less restrictions the topology gives on the crossing behavior of the graph, the harder the problem will get in general. By NP-hardness of the maximal bipartite subgraph problem, this problem is expected to be NP-hard in those topologies.

10 Conclusion

Within this thesis, Problem 1.7 (see page 4) could neither be classified to be polynomially solvable nor to be NP-hard. However, we reformulated the problem as a special case of the maximum bipartite subgraph problem on the crossing graph, see Problem 3.10 on page 15, and as a special case of the Max-Cut problem, see Problem 7.1 on page 34. Furthermore, we gave two formulations of Problem 1.7 as an integer linear program, see Problems 4.1 and 4.3 on pages 17 and 18. The first formulation gave us a useful LP-relaxation, see Problem 4.2 on page 17. The issue that the problem has exponentially many constraints in general can be overcome by the use of the Ellipsoid Method, see Algorithm 5.3 on page 23.

From an approximation point of view, we showed that already a trivial algorithm provides a 630-approximation for instances with at least 10 vertices, see Observation 6.1 on page 29, so we classified the problem into the complexity class of combinatorial optimization problems with a constant approximation factor, usually denoted as APX. As the local optimization heuristics given in Heuristic 6.3 on page 30 behaves much better in practical approaches, we conjecture that there is room for improvement on the approximation factor.

Finally, we compared the local optimization heuristics with the bound obtained from LP-relaxation on small instances. With the above methods, we managed to evaluate the exact solution of Problem 1.7 on all essentially different drawings with up to ten points, see Observation 8.5 on page 37. In the majority of the cases, the upper and the lower bound agree proving the exactness of the solution. However, in some few cases, an exact solution approach was necessary. Moreover, we found out that any drawing of eleven points needs at least ten monochromatic crossings.

For further approach, it would be interesting to classify Problem 1.7, either to be polynomially solvable or to be NP-hard. If this is not successful, or if Problem 1.7 turns out to be NP-hard, it might be interesting to find better approximation algorithms or prove better bounds on the local optimization approach.

List of Tables

List of Figures

References

[1] O. Aichholzer. Enumerating Order Types for Small Point Sets with Applications, 2006.
URL `http://www.ist.tugraz.at/staff/aichholzer/research/rp/triangulations/ordertypes/`
→ page 36

[2] O. Aichholzer. Lecture notes on discrete and computational geomtery, 2015. Institute for Software Technology, TU Graz.
→ pages 5, 10, 42

[3] O. Aichholzer. On the rectilinear crossing number, 2015.
URL `http://www.ist.tugraz.at/staff/aichholzer/research/rp/triangulations/crossing/`
→ pages ii, 9, 10

[4] O. Aichholzer, F. Aurenhammer, H. Krasser. Enumerating Order Types for Small Point Sets with Applications. *Proc.* 17^{th} *Ann. ACM Symp. Computational Geometry*, 11–18. Medford, Massachusetts, USA, 2001.
→ page 36

[5] O. Aichholzer, R. F. Monroy, A. Fuchs, C. H. Toscano, I. Parada, B. Vogtenhuber, F. Zaragoza. On the 2-colored crossing number. *Proc. European Workshop on Computational Geometry EuroCG '19*, 56:1–56:7. Utrecht, Netherlands, 2019.
→ pages 10, 11, 43, 45

[6] O. Aichholzer, W. Mulzer, P. Schnider, B. Vogtenhuber. NP-Completeness of Max-Cut for Segment Intersection Graphs. *Proc.* 34^{th} *European Workshop on Computational Geometry EuroCG '18*, 32:1–32:6. Berlin, Germany, 2018.
→ pages 34, 40

[7] S. A. Cook. The complexity of theorem-proving procedures. *Proc. 3rd ACM Symposium on Theory of Computing*, 151–158, 1971.
→ page 33

[8] H. S. M. Coxeter. *Introduction to Geometry*. John Wiley & Sons, 1969.
→ page 42

[9] G. B. Dantzig. Origins of the simplex method. , Department of Operations Research Stanford University Stanford, 1987.
URL `https://apps.dtic.mil/dtic/tr/fulltext/u2/a182708.pdf`
→ page 21

[10] M. M. Day. *Normed Linear Spaces*, 27–52. Springer, 1958.
→ page 26

[11] P. Erdős. Graph theory and probability. *Canadian Journal Of Mathematics*, 11: 34–38, 1959.
→ page 7

[12] A. Fuchs. Sage worksheets on the monochromatic rectilinear crossing number, 2019.
URL `http://www.crossingsnumbers.org/projects/monochromatic/tools/index.html`
→ pages 37, 38

[13] M. R. Garey, D. S. Johnson. Computers and intractability: A guide to the theory of NP-completeness, 1979.
→ page 15

[14] F. Hadlock. The minimal 2-coloration problem. *Proc. 3^{rd} Southeastern Conference on Comboinatorics, Graph Theory and Computing*, 221–241. Utilitas Mathematica, F. Hoffman and R. B. Levow and R. S. D. Thomas, Winnipeg, Canada, 1972.
→ page 34

[15] F. Hadlock. Optimal graph partitions. *Proc. 4^{th} Southeastern Conference on Comboinatorics, Graph Theory and Computing*, 309–328. Utilitas Mathematica, F. Hoffman and R. B. Levow and R. S. D. Thomas, Winnipeg, Canada, 1973.
→ page 34

[16] F. Hadlock. Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal on Computing*, 6:1, 86–87. 1975.
URL `https://epubs.siam.org/doi/abs/10.1137/0204019`
→ page 34

[17] M. Innerkofler. Personal communication, 2019.
→ page 18

[18] J. L. W. V. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30 (1): 175–193, 1906.
→ page 8

[19] P. János. Intuitive geometry, in memoriam László Fejes Tóth, 2008.
URL `https://www.renyi.hu/conferences/intuitiv_geometry/`
→ page ii

[20] N. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4: 373–395, 1984.
→ page 21

[21] B. Korte, J. Vygen. *Kombinatorische Optimierung*, 2 *Spektrum*. Springer, 2012.
→ pages ii, 19, 39

[22] K. Kuratowski. Sur le problème des courbes gauches en topologie. *Fundamenta Mathematicae*, 15: 271–283, 1930.
→ page 10

[23] D. Kőnig. Gráfok és alkalmazásuk a determinánsok és a halmazok elméletére. *Matematikai és Természettudományi Értesítő*, 34: 104–119, 1916.
→ pages 15, 39

[24] A. Makhorin. Glpk (gnu linear programming kit), 2012.
URL `https://www.gnu.org/software/glpk/`
→ page 36

[25] B. McKay. graph formats, 2019.
URL `http://users.cecs.anu.edu.au/~bdm/data/formats.html`
→ page 49

[26] J. J. O'Connor, E. F. Robertson. Harold Scott MacDonald Coxeter, 2019.
URL `http://www-history.mcs.st-andrews.ac.uk/Biographies/Coxeter.html`
→ page ii

[27] sage.
URL `http://www.sagemath.org/`
→ pages 36, 37

[28] S. Saxena. Ellipsoid Method for Linear Programming made simple, 2017.
URL `http://arxiv.org/abs/1712.04637v1;http://arxiv.org/pdf/1712.04637v1`
→ pages 21, 23, 24

[29] A. Steger. *Diskrete Strukturen*, Band 1, Kombinatorik – Graphentheorie – Algebra *Springer-Lehrbuch*. Springer, Institut für Theoretische Informatik, ETH Zürich, Universitätsstraße 6, 8092 Zürich, Schweiz, 2. auflage , 2007.
→ pages 2, 15, 27, 39

[30] K. Wagner. Über eine eigenschaft der ebenen komplexe. *Mathematische Annalen*, 114: 570–590, 1937.
→ page 10

11 Appendix

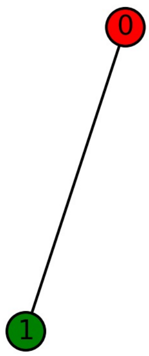
11.1 Gadgets up to five vertices

In the following, we give a list of all gadgets with up to five vertices, i.e., of all graphs that are up to inverting all colors uniquely 2-vertex-colorable such that the number of monochromatic edges is minimized. Although we are interested in unlabeled graphs, we will, for better describability, give an arbitrary labeling of the graph. The graphs are given on one hand by a figure (vertices drawn as circles with color red or green) and edges as line segments between the circles. On the other hand, the graphs are given in G6-Notation, i.e. a printable code of ASCII symbols where the first symbol characterizes the number of vertices in the graph and the following characters describe six bits of the upper triangular part of the adjacent matrix each. For more details on the G6-Notation see [25].

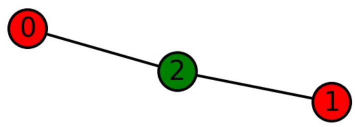
These graphs have been generated by enumerating all labeled connected graphs, picking a representative in every isomorphism class and choosing only those that satisfy the property in Lemma 6.6. The graphs are sorted lexicographically by their G6-Notation.

`@`

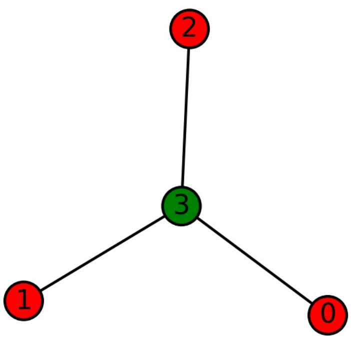
`A_`



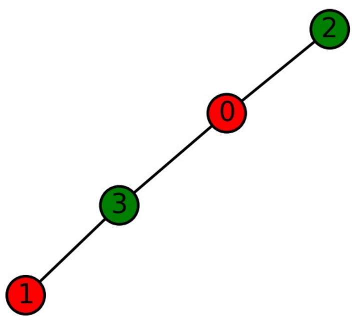
`BW`



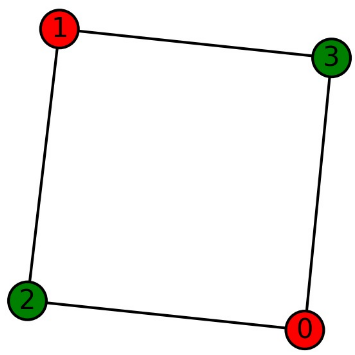
`CF`



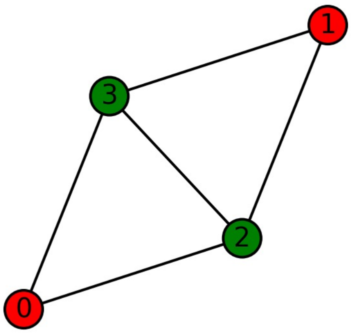
`CU`



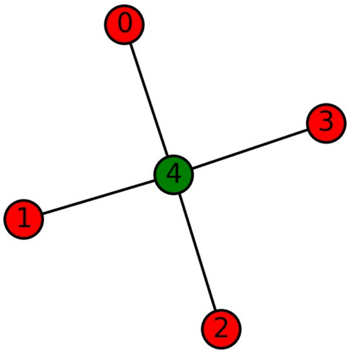
`C]`



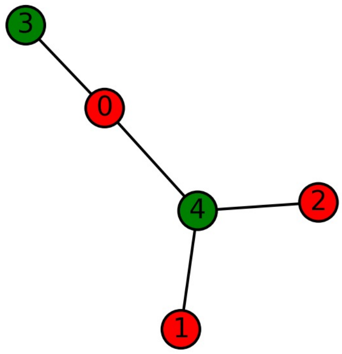
C^



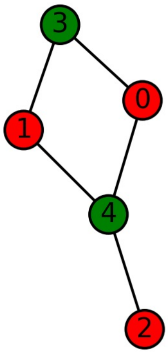
D?{



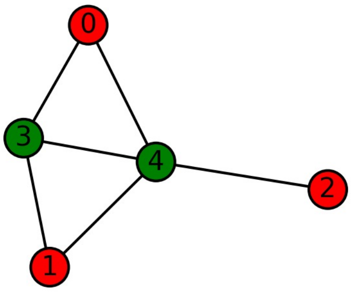
DCw



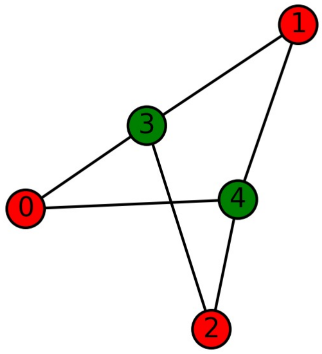
DEw



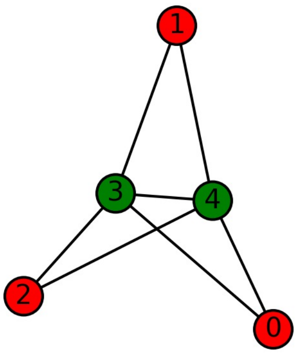
DE{



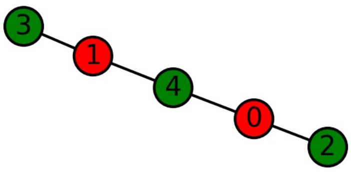
DFw



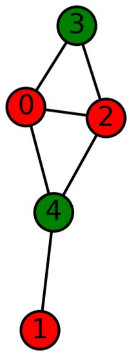
DF{



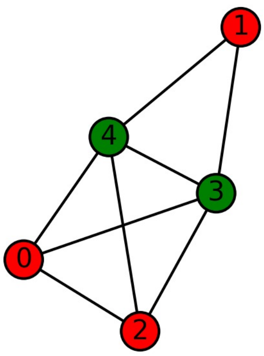
DQo



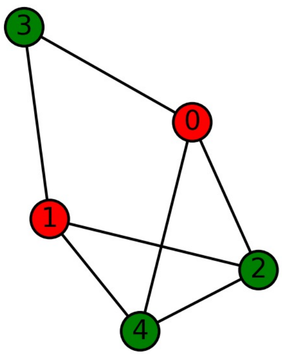
DTw



DV{



D]w



11.2 Example for a pointset with bad local optima in the gadgets heuristics

We consider the rectilinear drawing $\mathcal{D}$ of K_{12} given by the coordinates in Table 5. An illustration of the drawing is given in Figure 11. The edges that have at most one crossing and are removed from the crossing graph for our considerations, as they can always be colored such that they do not induce a monochromatic crossing. The adjacency matrix of the crossing graph $G_\times$ of $\mathcal{D}$, i.e. its 2-cores, is illustrated in Figure 12: The vertices are sorted by ascending degree. A dot in a square "●" symbolizes an edge (a "1" in the matrix) while a square without a dot "■" symbolizes a non-edge (a "0" in the matrix). The colors of the squares and dots in the illustration correspond to the colors of the vertices of the crossing graph: If both vertices are in the same color, the square and the dot are in that color, i.e., "●" and "■" if both are red, and "●" and "■" if both are green. If the two vertices are in different colors, the square and dot is gray, "●" and "■". In the left side of the figure, the colors are according to an optimal 2-coloring where the optimality, 72 monochromatic crossings, is proven by the LP-5 lower bound. The right side of Figure 12 shows the same adjacency matrix with a 3-optimal coloring with 97 monochromatic crossings, i.e., a local optimum that has 25 more monochromatic crossings than the globally optimal one.

x-coordinate	y-coordinate
0.293621540367584	-0.654331384542488
-0.369389604540473	0.484346230374869
0.0201655518276823	-0.0288584754094638
0.0314579750123125	-0.275797325975042
0.567905091127356	-0.206021530902484
-0.627667065973825	0.630323880804552
0.615036514366687	0.162729939245451
0.372665583257629	0.310953969342043
0.0735417752851854	-0.332551613198758
-0.288337988592218	-0.320819491010896
0.00654308455385613	0.0844978120806193
-0.325958310747484	-0.150097634076213

Table 5: Coordinates of an example set of 12 points with a bad local optimum in the gadgets heuristics

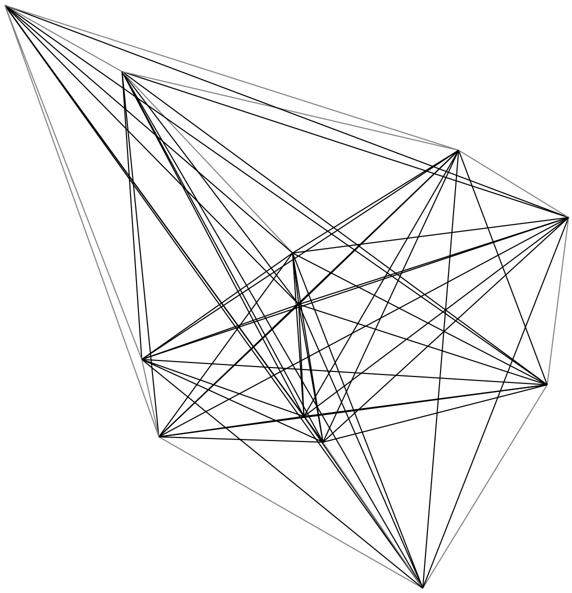


Figure 11: Illustration of the example drawing of K_{12} with coordinates given as in Table 5

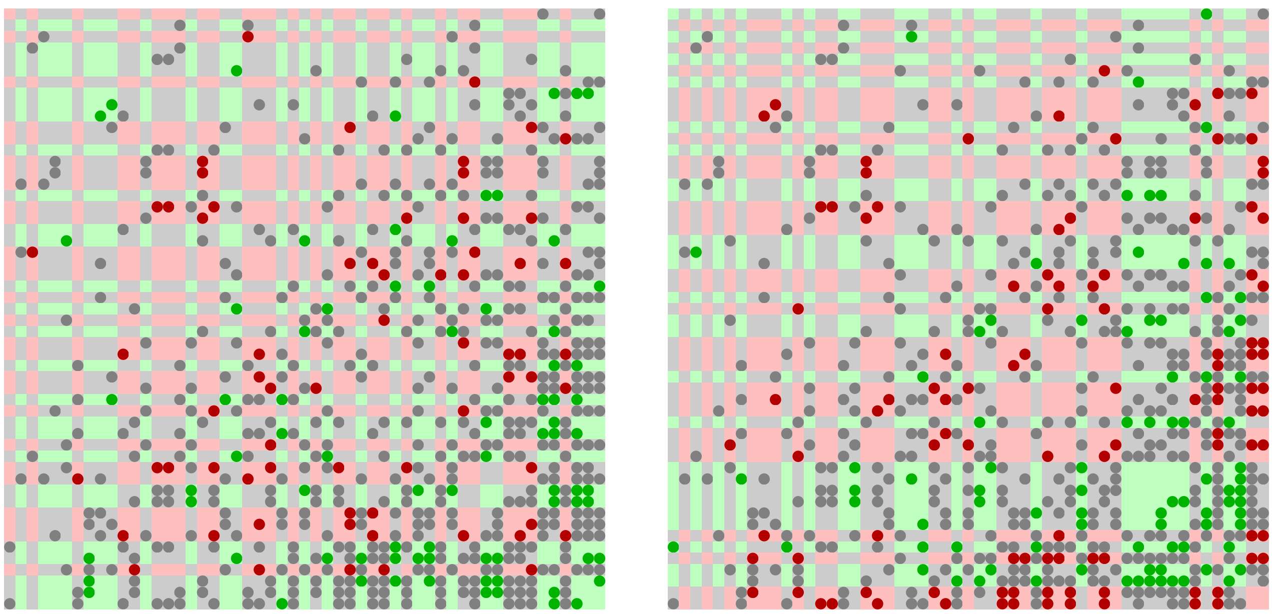


Figure 12: Adjacency matrix of the crossing graph as dot plot