Eberhard Karls Universität Tübingen
Wilhelm Schickard Institut Tübingen

Fachbereich Informatik

Smoothening of Kandinsky Drawings

Arbeitsbereich Algorithmik

zur Erlangung des akademischen Grades
Bachelor of Science

Autor: Benjamin Ulvi Çoban
MatNr. 3526251

Version vom: 30. Juni 2018

1. Betreuer: Prof. Dr. Michael Kaufmann
2. Betreuer: Henry Förster

Zusammenfassung

Graphenzeichnungen sind vielfältig in ihrer Anwendung - im Bezug auf VLSI Design oder Metroliniennetz sind *orthogonale* Zeichnungen besonders relevant. Dies bedeutet, dass jede Kante als Abfolge von horizontalen und vertikalen Liniensegmenten, welche rechtwinklig an sogenannten *Knicken* aneinander knüpfen, dargestellt wird. Ein weitverbreitetes Modell ist das sogenannte *Kandinsky Modell*. Die *Glättung* solcher Zeichnungen arbeitet zusätzlich mit Viertelkreissegmenten, sodass die Ecken abgerundet werden. Dabei ist das Verhalten der sogenannten *Komplexität* der Kanten - aus wievielen Segmenten solch eine Kante in der geglätteten Zeichnung nun besteht - zu untersuchen.

Im ersten Abschnitt der Ausarbeitung werden wir zeigen, dass die Glättung einer Kandinsky Zeichnung möglich ist. Ist die Eingangszeichnung kreuzungsfrei, so bleibt diese Eigenschaft erhalten. Die Ausrichtung der Knoten wird dabei nicht grundlegend verändert. Allerdings wird dabei die geglättete Zeichnung größer - der horizontale Platzverbrauch kann sich im schlimmsten Fall quadrieren. Die Komplexität der Kanten bleibt bei ausgehenden Kandinsky Zeichnungen überschaubar. Besteht eine Kante aus mehr als vier Segmenten, erhöht sich deren Komplexität um zwei.

Im nächsten Abschnitt beschäftigen wir uns mit verschiedenen Ansätzen, um an Platzverbrauch und Kantensegmenten der resultierenden Zeichung zu sparen. Wir zeigen einen Ansatz, um den horizontalen Platzverbrauch b im besten Fall auf $\sqrt{b}$ zu schrumpfen. Weiter wird eine Kombination aus Kreis- und Liniensegmenten vorgeschlagen, sodass der horizontale Platzverbrauch sich nur um den Wurzelwert $\sqrt{b}$ vervielfacht.

In weiterführender Arbeit wird schließlich durch eine Gadgetkonstruktion gezeigt, dass es $\mathcal{NP}$*-hart* ist zu entscheiden, ob ein Graph mit einer geglätteten Zeichnung ohne Knicke illustriert werden kann. *Achtelkreissegmente* werden auf ihre Eigenschaften untersucht, da sie relevant für weiterführende Projekte sein können.

Abstract

Graph drawings are diverse in their applications - considering VLSI or metro map design, *orthogonal* drawings are of interest. In an orthogonal drawing, every edge is illustrated as a sequence of axis-aligned line segments which intersect in a point, so-called *bends*. This thesis will examine one of the most common models of orthogonal drawings - the *Kandinsky model*. The *smoothening* of such Kandinsky drawings introduces circular quarter arc segments for smoothening the 90° angles. The behaviour of the *complexity* of such an edge - the amount of segments illustrating a smoothened edge relative to the orthogonal case - is to examine.

First, we will show the possibility of smoothening Kandinsky drawings. Several properties of the drawings are preserved. If the input drawing is crossing-free, the smoothened drawing will also be crossing-free and the orientation of the vertices is not mainly altered. However, the smoothened drawing may increase in size - the horizontal area consumption may be squared in the worst case. The complexity of input Kandinsky drawings may not significantly rise. If the input edge consists of at least four segments, then the complexity of the smoothened edge increases by two.

Next, several approaches in order to save segments and area consumption are established. A method is shown, how to shrink the horizontal area consumption by its square root in the best case. A combination of circular arc and line segments is introduced to reduce the area expansion in the smoothened drawing.

In the extensional work, a gadget construction is introduced to prove the $\mathcal{NP}$*-hardness* whether a given graph admits a *bendless* smoothened drawing. *Eighth circular arc segments* - or *octi arcs* - are examined for its properties as they may be relevant for further work.

Erklärung

Hiermit erkläre ich, dass ich diese schriftliche Abschlussarbeit selbstständig verfasst habe, keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe und alle wörtlich oder sinngemäß aus andern Werken übernommenen Aussagen als solche gekennzeichnet habe.

_____

Datum, Ort, Unterschrift

Contents

1 Introduction

Metro maps, circuits, networks, construction plans and many more - they all can be visualized with a corresponding *graph drawing*. Over the last decades, many different efficient algorithms were developed for graph drawings in the Euclidean plane. Especially, orthogonal graph drawings are of interest as they are applicable in various fields. In order to work with graph drawings efficiently, one has to consider the *quality* of a graph drawing. There is a huge variety of aspects to consider when we want to examine the quality of a drawing. The readability of the illustrated information, the size of the drawing - measured with the pair of vertices with the farthest distance in the drawing, and the *edge complexity* - the amount of consecutive line segments for an edge illustration are only a few aspects how to measure the drawing quality. Naturally, we try to create drawings as clearly as possible meaning to avoid drawings with a high edge complexity. If a given graph admits a *crossing-free*, or in other words *planar* drawing, we want to preserve this property in further processing approaches.
The American abstract artist *Mark Lombardi* gained approval for his aesthetic illustration of political-economic structures. The diagrams included *circular arcs* of different sizes and their even distribution around a vertex in order to vizualize connections adequately. It seems that the circular arcs emphasize the connection between components in sense of direction.

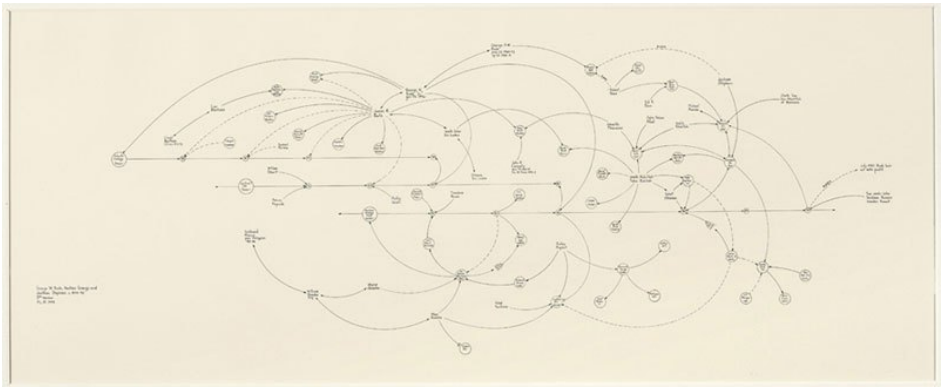


Figure 1: Work of Mark Lombardi [11]

Orthogonal drawings arise among others in VLSI design where quite many cables are following a similar path. The smallest angle between axis-aligned line segment is at most $\pi/2$ and their angular resolution is quite pleasing for the eye of the viewer. One fundamental, reliable model is the *Kandinsky model* which is based on a *grid embedding*. The vertices lie on a *coarse* grid while the edges lie on a *fine* grid extending the coarse grid. It may appear that an orthogonal drawing may convey some structural information, so *smoothening* those edges is of interest. In this thesis, we focus on the smoothening of Kandinsky drawings by introducing circular arcs, inspired by *Lombardi drawings* as illustrated in Figure 1[6][7].

(a) Orthogonal drawing (b) Smoothened drawing

Figure 2: Smoothening a drawing for aesthetic appeal

By postprocessing an input drawing like illustrated in Figure 2a and 2b, we also have to consider possible shape alternations. It is desirable that the orientation of the vertices is preserved, meaning that e.g. a metro map can still be read reasonably after the smoothening process[10].
However, it is a priori not guaranteed that there is a smoothening for every input Kandinsky drawing with a reasonable complexity increase. The introduction of circular arcs might arise some conflicts in sense of planarity. Dealing with postprocessing algorithms, we have to focus on new area bounds and the behaviour of the edge complexity in order to quantify the resulting quality of the smoothened drawing.

2 Preliminaries

2.1 Definitions

As otherwise mentioned, a *graph* $G = (V_G, E_G)$ is a tuple consisting of two sets - the set of vertices and the set of edges. An *edge* $e = (v, w), v, w \in V_G$ is a tuple and describes a connectivity relation between two vertices. Unless otherwise mentioned, the graphs are *undirected*. It means that the edge (u, v) is identical to the edge (v, u),$u, v \in V_G$. A *face* is a maximal open region of the plane bounded by edges. The *degree* of a vertex states the amount of edges incident to the vertex. The *degree* of a graph G is the maximum of the degree of its vertices. A *drawing* Γ of a graph G is a function, where each vertex is mapped on a unique point $\Gamma(v)$ in the plane and each edge is mapped on an open Jordan curve $\Gamma(e)$ ending in its vertices. A graph is *planar* if and only if there exists a crossing-free representation in the plane. An *embedding* of G is the collection of counter-clockwise circular orderings of edges around each vertex of V_G. [5, p.225]
In the following sections, G will be a planar undirected simple graph. k-planarity will refer to a planar graph with maximum degree k. We will now define the distinctive layouts of graphs.

Definition 1 (Line drawings). *A straight line drawing is a drawing where every edge is drawn as a straight line. In a polyline drawing, each edge is represented by a non-empty sequence of line segments ($e = (e_1, e_2, ...)$), where two consecutive line segments intersect in a unique point. The complexity of an edge is the length of its line segment sequence.*

Definition 2 (Bends). *A bend describes the orientation of two segments. A right bend is defined by a 90° angle in counter-clockwise direction, whereas a left bend is defined by a 270° angle respectively.*

Definition 3 (Polyedge). *A polyedge is a sequence of edge segments such that two consecutive segments meet in a single point in the drawing.*

Definition 4. *A polyedge is called uniform if and only if all bends are of the same direction. Similarily, a polyedge is alternating if and only if all bends are alternating (staircase).*

Definition 5 (Orthogonal drawings, [5, p. 225]). *An orthogonal drawing of a graph is a polyline drawing where every edge consists of polyline segments in alternating horizontal and vertical direction. It is clear that all bends are right or left bends, as defined before.*

Definition 6 (Ports). *A port of a vertex in an orthogonal drawing describes the position, where the edges are connected to. Due to the fact that every edge of an orthogonal drawing consists of horizontal and vertical segments, there are four ports per vertex in total for each of the cardinal directions.*

One of the most fundamental models is the so-called *Kandinsky model*. It is a well-established and widely used graph-drawing model. It contains a *grid embedding*, where a grid is used to draw the polyedges and vertices as boxes.

Definition 7 (Kandinsky, [9]). *A Kandinsky drawing $\Gamma(G)$ of a graph with arbitrary maximum degree is an orthogonal drawing of a graph G on a grid embedding. This model inherits*

- *a coarse grid for the vertices which is a subset of a fine grid for the edges. The granularity of the fine grid is determined by the maximum degree of G;*
- *every vertex is illustrated as a box with a uniform size centered on the coarse grid;*
- *every edge is drawn as a sequence of alternating horizontal and vertical line segments;*
- *there are as many edges per vertex side possible as the maximum degree of G specifies.*

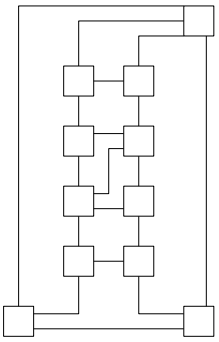
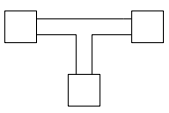
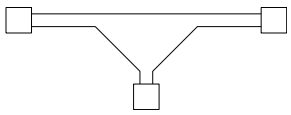


Figure 3: An example for a Kandinsky drawing

A face is *empty*, if for each vertex defining the face, the corresponding edges are connected to the same port. See Figure 4a for the empty T-face.



(a) Empty T-face



(b) Almost empty T-face

Figure 4: Introducing diagonal line segments, empty faces can be avoided in a drawing

Those empty faces are generally forbidden in the Kandinsky model, as they imply computational errors. In order to avoid empty faces, the model is slightly altered by introducing diagonal line segments resulting in drawings with so-called *almost empty* faces (See Figure 4b).

Definition 8 (Podevsaef drawings, [4])**.** *A Planar Orthogonal Drawing With Equal Vertex Size and Almost-Empty faces (Podevsaef drawing in short) is a drawing in which:*

- *The vertices are mapped on a point in the plane underlying a coarse grid and are illustrated with a uniform box,*
- *the polyedges are illustrated in the plane as a sequence of horizonal, diagonal and vertical line segments,*
- *arbitrary many edge segments can be connected to one port of the vertex box (usually, the degree of the graph as an upper bound),*
- *the diagonal segment is of length at most third of the length of the unit of the coarse grid and is never incident to a vertex box,*

- *the minimum of the angles formed by two consecutive segments of the edge is always 135°.*

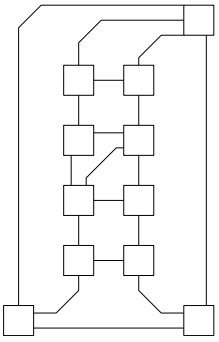


Figure 5: An example for a podevsaef drawing

2.2 Plane Sweep Algorithm

The plane sweep algorithm is a procedure to gain information or even modify the drawing of a graph. The main components are a *sweep line*, which iterates over a drawing in a desired direction and a data structure for so-called *events*. The sweep line recognizes predefined conditions as events and saves them in the *event holder*. The data structure used for storing the events is a balanced tree in standard practice in order to guarantee an appropriate runtime regarding deletion, insertion and update functions. It may be used to determine the number of crossing line segments or even alter the shape of a given drawing, as we will see with the stretching technique.

2.3 4M Algorithm [8]

Orthogonal drawing algorithms may output drawings with many bends and a large area. The 4M Algorithm addresses the quality improvement of orthogonal graph drawings. Mainly, there are four operations given by the algorithm - *Moving, Matching, Morphing* and *Merging*. The input drawing Γ is preprocessed to Γ'', before any of those algorithms work on them. At first, crossings and bends are substituted with a vertex rectangle, then horizontal stripes, creating not necessarily bounded rectangles, are introduced. The reason for this preprocessing is to avoid overlapping parts in the resulting drawing. For this thesis, the *Moving* algorithm may be of interest. *Matching*, *Morphing* and *Merging* operations may actively alter the size of the vertex boxes and are therefore not part of this thesis.

The Model

The drawing plane is subdivided by horizontal and vertical gridlines of a unit spacing λ. The vertices are represented by rectangles of size $\left(w\lambda - \frac{\lambda}{2}\right) \times \left(h\lambda - \frac{\lambda}{2}\right); w, h \in \mathbb{N}_+$. Those rectangles overlap the grid lines by $\frac{\lambda}{4}$.

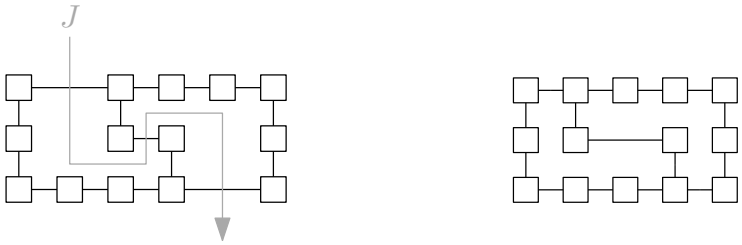
Moving

Finding a moving line is the first step for area savings. In order to save in horizontal area, the moving line is set vertically from top to bottom in the drawing. For vertical area savings, define the moving line similarly from left to right in the drawing.

Definition 9 (Moving line J)**.** *An area-saving moving line J is a line that fulfills the following conditions:*

- *J is directed and consists of horizontal and vertical segments*
- *J starts above the topmost object of Γ'' and ends below the bottommost object of Γ''*
- *J does not intersect any vertical edge of Γ''*
- *Every horizontal edge of Γ'' that is intersected by a piece of J which is directed downward has a finite length larger than or equal to two.*

Once J is found, we can reduce the length of the segments crossed downwards and lengthening the segments crossed upwards consecutively by one unit until one of the segments is reduced up to the unit length. Since J intersects the border of Γ'' in downward direction, the total area decreases.



(a) Illustration of the moving line J (b) After the moving operation

Figure 6: With help of the moving line, four units are saved horizontally in total

In Figure 6, an example is given for J crossing a drawing simultaneously upwards and downwards. Notice, that the number of bends is preserved with the moving operation. The reason lies in the fact that the horizontal line segments, which are crossed by J, have the length of at least two.

2.4 Topology Shape Metrics

The process from a graph as a mathematical tuple to a drawing requires an appropriate embedding in order to guarantee special properties such as planarity or a statement regarding the number of bends. Tamassia et al. presented an algorithm resulting in orthogonal drawings with a minimal number of bends for a given embedding. Surprisingly, computing the minimal number of bends of an orthogonal graph embedding is in P. The whole process is called *Topology Shape Metrics* [12]. This procedure is rather flexible in its applications as we will see onwardly.
The computation of an orthogonal drawing with a minimal number of bends of a simple graph basically divided into three phases - the *Topology phase*, the *Shape phase* and the *Metrics phase*.

- *Topology phase*
  This phase is initially used when there is no embedding specified in the first place. In this phase, a planar embedding on the plane is computed for the given graph.

- *Shape phase*
  This phase is also called the *orthogonalization phase*. The bends and its respective degrees of all the edges are computed along with the angles between the edges around a vertex. The result is an orthogonal representation.

- *Metrics phase*
  Also known as the *compaction phase*. Finally, the length of the edges and the position of the vertices are computed. The goal is to minimize the area the graph demands.

Every phase can be adjusted for special needs. As already mentioned, the Topology Shape Metrics approach serves as a standard practice in research due to its flexibility.[4]

2.5 Previous results

Our main goal is to smoothen the orthogonal drawings by introducing circular arcs. We will consider quarter and semi circular arcs in order to achieve a smoothened 90° bend.

Definition 10 (Smooth Orthogonal Layout - *SMOG* in short, [3]). *A Smooth Orthogonal Layout of a 4-planar graph G is a graph where*

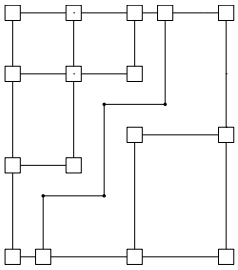
- *each vertex of G is drawn as a point on the plane;*
- *each edge of G is drawn as a sequence of axis aligned line segments and circular arc segments. The segments have to intersect in a point. The tangent at this point has to be horizontal or vertical, just as the segments itself;*
- *planarity is preserved;*
- *a port of a vertex is incident to at most one edge.*

A smooth orthogonal layout inherits a so-called *edge complexity* $k \geq 1$, if there does not exist any edge with complexity $k + 1$ or greater. The smooth orthogonal layout derives from an orthogonal drawing by postprocessing algorithms. The positions of vertices are altered in the horizontal direction. In fact, the smooth orthogonal layout representation of 4-planar graphs are already intensively studied regarding area bounds, complexity increase and the number of bends. Recall that the vertices of a 4-planar graph can have at most one edge per port - one each for North, South, West, East - and are illustrated as a point in the plane.

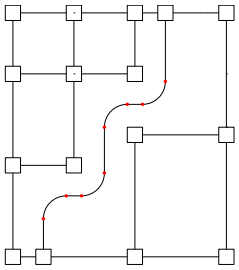
2.5.1 Fixed Layout Model

Considering an orthogonal drawing of a graph Γ_G in the Fixed Layout Model, the position of the vertices cannot be altered. This is a rather strict constraint. The implementation of circular arcs in a polyline lead to an increase of the edge complexity. Every bend is substituted with a quarter arc, resulting in two new bends in the worst case.

Theorem 1 ([3, p. 581, Figure 7]). *In the Fixed Layout Model, the edge complexity of a given orthogonal graph G might increase from k to $2k - 1$.*



(a) Staircase in Kandinsky



(b) SMOG representation

Figure 7: The complexity increase of 4-planar graphs with staircases

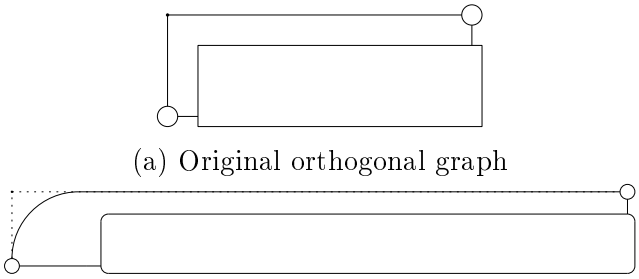
The reason for the edge complexity increase is the fixation of the vertices and following example regarding so-called "Staircase Edges". Regarding the significant rise of complexity and due to the fact, that circular arcs with a very small radius might be introduced, the Fixed Layout Model might not be suitable for smoothening an orthogonal drawing. Circular arcs with a small radius might not emphasize the ongoing direction of the edge, therefore for the viewers eye. A different approach is the Fixed Shape Model, where the orthogonal representation is preserved (the circular ordering of the polyedges around a vertex).

2.5.2 Fixed Shape Model

In the Fixed Shape Model, the circular ordering of the edges connected to a vertex is preserved. The vertices are of uniform size but can be repositioned on the coarse integer grid.

Definition 11 (Stretching technique, [3, p. 582])**.** *The stretching technique is a process where every horizontal edge of a given orthogonal drawing is elongated by the length of the longest vertical segment l of the drawing. The result is an orthogonal drawing of size up to $\mathcal{O}(n^2) \times \mathcal{O}(n)$, where n is the number of vertices.*

Due to the horizontal stretching technique by the factor of l, there is new space left and right from every vertical line segment. To be more precise, there is an empty box left and right from every vertical line with size $l' \times l'$, while l' is the length of the respective vertical line.[3, p. 583, Figure 5]
In practice, the SMOG Model is derived from the Kandinsky Model using basically two plane sweeps: The first plane sweep stretches the Kandinsky drawing horizontally by the factor of the longest vertical line segment. This plane sweep holds horizontal line segments as events. Every time a new horizontal line segment gets active, all the active segments are extended by l. This results in a stretching of every grid cell. The second plane sweep substitutes parts of the horizontal and vertical line segments with corresponding quarter circular arcs, making the drawing *smooth*. The resulting drawing is in $\mathcal{O}(n^2) \times \mathcal{O}(n)$ area due to the horizontal stretching. The worst case of the stretching technique results in a quadratic size of width. Consider the following example:



(a) Original orthogonal graph

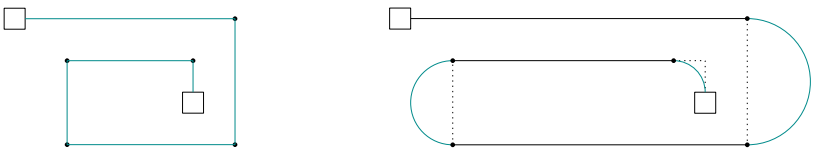
(b) After applying the stretching technique

Figure 8: Worst case illustration of the stretching technique

The box in Figure 8 illustrates a grid of vertices which is of size $\mathcal{O}(n) \times \mathcal{O}(n)$. There are therefore $\mathcal{O}(n)$ edges in the box horizontally and the polyedge of the outermost vertices contains a vertical segment of size $\mathcal{O}(n)$. By applying the stretching technique, the horizontal segment of the outer polyedge gets $\mathcal{O}(n)$ additions in length of size $\mathcal{O}(n)$, resulting in $\mathcal{O}(n^2) \times \mathcal{O}(n)$ area.

Theorem 2 ([3, p. 584])**.** *If the bends of a polyline are purely uniform (in the same direction), then there is a SMOG representation of that polyline without an increase of complexity. Similarily, if the polyline is purely alternating, the edge complexity raises from k to $\lceil \frac{3}{2}k \rceil$.*

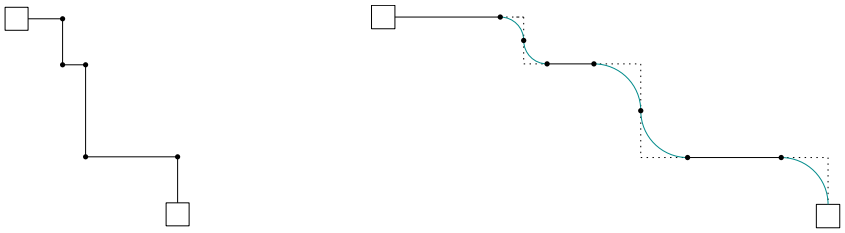
Proof (Sketch). If a polyline is purely uniform, consider the vertical segments as follows; If a vertical segment lies between two horizonal segments, substitute that vertical segment with a semicircle arc. If a vertical segment is connected to a vertex and a segment, substitute the vertical segment with a quadrant arc. If the polyedge consists of one vertical segment, then two vertices are at both ends. The edge complexity is not altered (Figure 9). The space around a vertical segment, necessary for the circle arc substitution, is guaranteed by stretching the entire drawing horizontally by a factor of the longest vertical segment.



(a) Uniform polyedge with 5 bends (b) No complexity increase in SMOG

Figure 9: The uniform case - No complexity increase under $\mathcal{O}(n^2) \times \mathcal{O}(n)$ area

Similarily, if the polyline is purely alternating, then due to the stretching technique the area of $|l'| \times |l'|$ left and right of a vertical segment l' is still guaranteed. We are able to substitute the staircase with same sized circular arcs with alternating turns, increasing the total number of bends from $k-1$ to $\lfloor \frac{3}{2}k \rfloor$ (Figure 10).



(a) Alternating polyedge with 5 bends (b) Complexity increase in SMOG

Figure 10: The alternating case - Complexity increase under $\mathcal{O}(n^2) \times \mathcal{O}(n)$ area

□

Theorem 3. *Let G be a 4-planar graph with an orthogonal drawing Γ_G. If any polyline is alternating at some point, it is possible that Γ_G can be minimized regarding the number of bends.*

Proof (Sketch). We show the theorem by flow minimization over the dual graph the following way: Let e be a polyedge seperating two faces f, g. For each convex bend in the face f, we send one unit of flow from f to g and vice versa. If there is an alternation in e, then there is a cycle of flow between f and g and the edge can be minimized. □

The flow minimization is an application of Tamassia [12]. It is possible that this minimization actively changes the shape in some point, arising a new model - the "Almost Fixed Shape" model [1].

Theorem 4. *In the Fixed Shape Model, an orthogonal graph G - with minimal number of bends and an edge complexity of k - can be transferred to a SMOG without an edge complexity increase under $\mathcal{O}(n^2) \times \mathcal{O}(n)$ area.*

Proof (Sketch). If G has got a minimal number of bends, then there is no alternation in any polyline by contraposition of Theorem 3. The polyedges are purely uniform and every vertical segment is replaced by either a quarter circle arc or a semicircle arc or it stays the same. As we already saw, uniform bends do not lead to an edge complexity increase. Planarity is preserved due to the stretching technique. □

2.6 Drawings with low complexity

The persistance of some bends lies in the orthogonality property of the drawing. If two vertices connected by a polyedge do not share the same x or y coordinate, then the polyedge has to overcome the differences in the regarding direction with horizontal and vertical segments, preserving the port constraint.

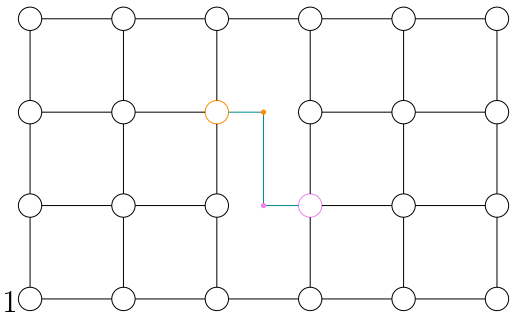


Figure 11: A drawing with a staircase of complexity 3

In Figure 11, we see an orthogonal drawing of a 4-planar graph a staircase edge of complexity 3. Notice that the cyan coloured polyedge is zig-zag shaped.

Theorem 5 ([3, Theorem 3, p. 583])**.** *Let G be a 4-planar graph with an orthogonal drawing Γ_G of complexity 3. Then, there is a complexity-4 smooth orthogonal layout in $\mathcal{O}(n^2) \times \mathcal{O}(n)$ area.*

Proof (Sketch). By the stretching technique, the space for arc substitution is guaranteed. For complexity-1 or complexity-2 edges, the complexity does not increase. Alternating complexity-3 edges will increase from 3 to 4. □

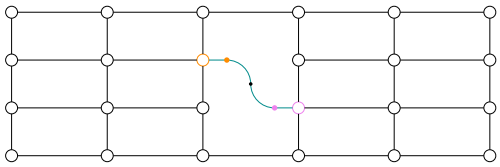


Figure 12: Smooth orthogonal layout of figure 11

In Figure 12, the drawing of figure 11 got stretched and circular arcs were substituted, resulting in a complexity-4 smooth orthogonal layout.

2.7 $\mathcal{NP}$-hardness

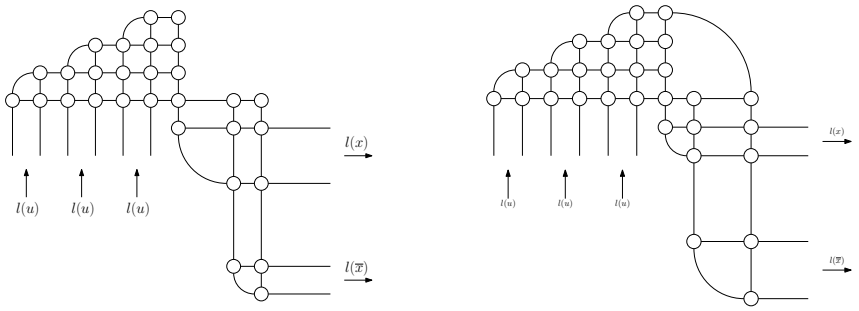
It is proved by Bekos et al. that it was $\mathcal{NP}$-hard to decide whether a 4-planar graph with a given representation admits a bendless SMOG. To be more precise:

Theorem 6. *Given a planar graph G of max-degree 4 and a SMOG representation $\mathcal{R}$, it is $\mathcal{NP}$-hard to decide whether G admits a bendless SMOG preserving $\mathcal{R}$. This is the implementation of the last step of the Topology Shape Metrics approach (Bend minimization by Tamassia, in P for orthogonal drawings).*

Proof (Sketch). The proof inherits a reduction from 3-$\mathcal{SAT}$ to a SMOG representation construction which is bendless if and only if a formula φ given in CNF is satisfiable. Γ_φ is constructed with so-called *auxiliary gadgets*, where the information flows along the faces encoded in their length. There are multiple gadgets, as shown below. The vertices are illustrated as circles.

- *Variable gadget*
  For each variable x of φ, a variable gadget gets three edges of the same length $3 \cdot l(u)$ as input. The assignment is encoded in the following way:

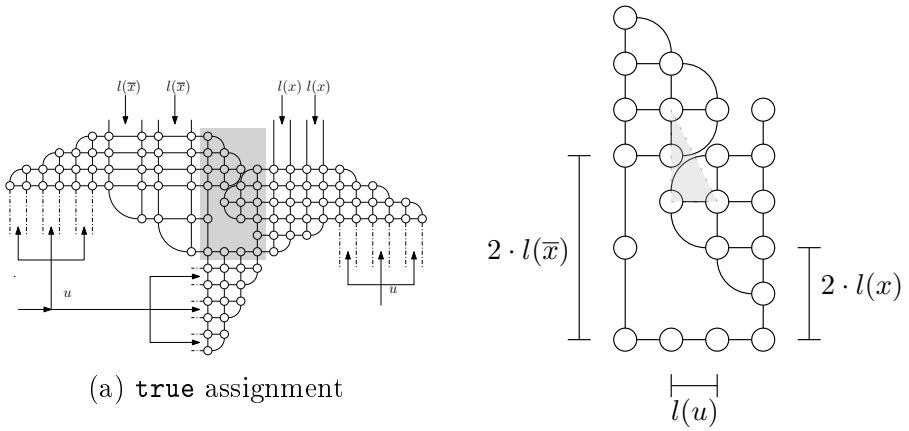
$$x = \texttt{True} \quad \Leftrightarrow \quad l(x) = 2 \cdot l(u), l(\overline{x}) = 1 \cdot l(u) \tag{1}$$
$$x = \texttt{False} \quad \Leftrightarrow \quad l(x) = 1 \cdot l(u), l(\overline{x}) = 2 \cdot l(u) \tag{2}$$



(a) `true` assignment (b) `false` assignment

Figure 13: Variable gadgets for the 4-planar case

- *Parity gadget*
  The parity gadget guarantees that a variable is defined as `true` or `false`. For instance, if $l(u) = 2$, then the variable gadget could set $l(x) = l(\overline{x}) = 3$ which is considered as an undefined variable.



(a) `true` assignment (b) `false` assignment

Figure 14: Parity gadget for the 4-planar case

Consider the triangle illustrated in Figure 14b. The center of the circular arcs shoujld be at distance greater than $2 \cdot l(u)$. Consider $\lambda = l(\overline{x}) - l(x)$ as dependence, then the length of this segment can be expressed as follows: $\sqrt{4\lambda^2 + l(u)^2}$. So, to avoid crossings, λ should value at least $\frac{\sqrt{3}}{2} \cdot l(u)$. It follows that $l(x), l(\overline{x}) \in (0, 1.067l(u)) \cup (1.933l(u), 3)$ in order to avoid crossings.

- *Clause gadget*
  For each clause of φ in CNF there is a gadget for the literals a, b and c. The length composition guarantees that at least one of those literals must be `true`.

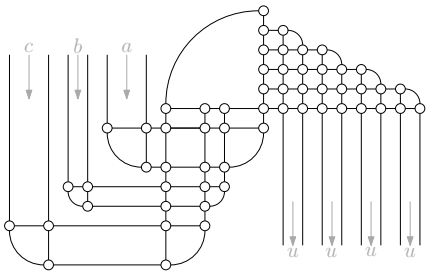
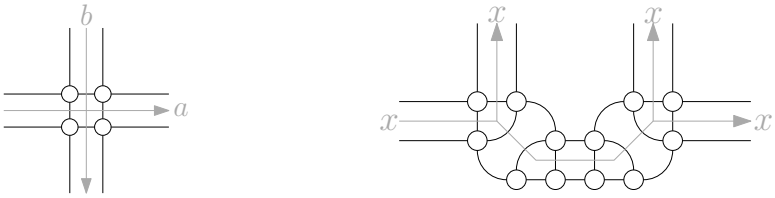


Figure 15: Clause gadget

- *Auxiliary gadget*



(a) Crossing gadget (b) copy gadget

Figure 16: Auxiliary gadgets

> The *crossing gadget* consists of a rectangle and lets the information flow "through it". The *copy gadget* is able to split the information in three copies. The *unit length gadget* serves as a measurement for the information encoding.

The construction of R_φ inherits a parity gadget for each variable. The i-th variable with its gadgets is placed below and on the right of the $(i-1)$-th variable. On the right lie the clause gadgets. On the bottom of Γ_φ lie several copy gadgets which "feed" the gadgets with the information given by a unit gadget.

Correctness

Recall that this is only a sketch of the reduction. For the whole proof, see [2].
Assume, that φ is satisfiable. Then we can set $l(x)$ and $l(\overline{x})$ for each variable x respectively dependent of the unit length $l(u)$. Arranging the variable and clause gadgets yields a bendless SMOG drawing Γ_φ. Now assume, that there is a bendless SMOG drawing Γ_φ. We are able to compute a truth assignment of φ by backtracking the clause gadgets of G_φ. By the composition of the truth assignments, at least one literal per clause has to be `true`. Therefore φ is satisfiable. □

3 The Kandinsky Model

3.1 Introduction

Previous results show that the Fixed Shape Model is flexible in its methods to create the smooth orthogonal drawings deriving from orthogonal planar drawings with maximum degree four. The Kandinsky Model is a well-established model to create an orthogonal drawing of a graph of arbitrary degree with a reasonable edge complexity. However, the Kandinsky Model differs in its properties. The vertices lie on a coarse grid and are illustrated with uniform boxes. If multiple edges are connected to the same port, then at most one is connected to another vertex.

Definition 12. *The bend or end property of an orthogonal drawing of a graph means that from every polyedge connected to a specific port of a vertex at most one is connected to another vertex with an edge complexity of one. The remaining polyedges are at least of complexity two and bend in some point.*

It is to examine if postprocessing Kandinsky drawings guarantees the area required for circular arc substitution and whether or not the planarity of a drawing is preserved in any case. Furthermore, we will examine the edge complexity of SMOGs derived from Kandinsky drawings. We will see, that the edge complexity of any polyedge does not increase excessively.

3.2 Area Investigation

The idea of the stretching technique is to create new space between edges and vertices. By doing so, the space for circular arcs will be guaranteed. As we already saw, the stretching technique is sufficient for 4-planar graph drawings. But what about planar drawings of arbitrary degree?
There are two major cases to distinguish - either a vertical line segment is between two other segments or it is connected to a vertex therefore on one of the ends of the polyedge.

Between two segments - "Boxing"

The stretching technique is a modification of an input graph fulfilling properties of the Fixed Shape Model. In this section we will examine the resulting space along vertical line segments. We will see that the new space created can be described as squares or *1* and we will see that the stretching technique will result in new space left and right from it.

Lemma 1. *Let l be the longest vertical line segment in a given orthogonal drawing of a graph G. Then, after applying the stretching technique by the factor of $|l|$, every vertical line segment l' which is between two segments got a new "box" of space left and right from it of size $|l'| \times |l| \leq |l|^2$.*

Proof. Let l' be a vertical line segment between two horizontal line segments. Then, by planarity of the orthogonal drawing, consider without loss of generality the minimum distance between l' and another vertical line segment l'' by $\delta(l', l'') \geq 1$ on the fine integer grid. Stretching every horizontal line segment by adding a segment of size $|l|$

results in a stretching of the distance of two vertical line segments. The distance of vertical line segments can be considered as a horizontal subtraction.

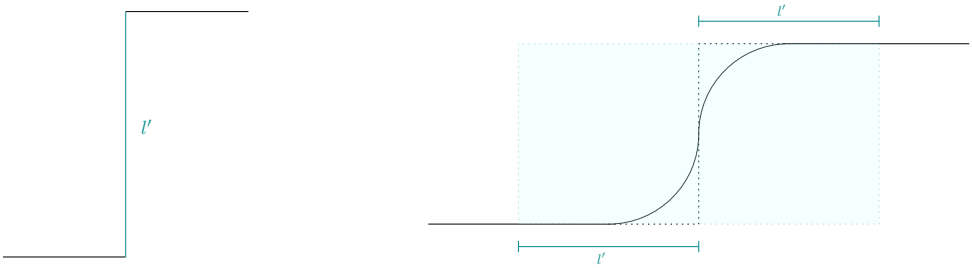
$$\delta(l', l'') = |l'_x - l''_x|$$

Stretching horizontally modifies the position of objects in x-coordinates only.

$$\delta'(l', l'') = |\,(|l| + |l'_x|) - (|l| + |l''_x|)\,| \tag{3}$$

$$\underbrace{=}_{|l|>0} |l| + (|l'_x - l''_x|) = |l| + \delta(l', l'') \tag{4}$$

So the distance between a vertical line and an object increased to minimum $|l|+1$. □



(a) Orthogonal input edge

(b) Two circular arcs result in +1 bend

Figure 17: The $l' \times l'$ box left and right from the original vertical line is empty after applying the stretching technique

This implies that there is new space to the left and right of a vertical line l' between two segments of size $|l'| \times |l'|$. We will consider the new space as a free "box" (Figure 17b). This means that the space for different situations of SMOG substitution is guaranteed in the intermediate case.

Lemma 2. *Let l' be the vertical line intermediate in a fragment. Then, take the box of the corresponding side and draw a half circle segment with radius $r = \frac{l'}{2}$. Furthermore, the complexity does increase if and only if the fragment is alternating.*

Proof. Let l' be the vertical line intermediate in a fragment after application of the stretching technique. Then, take the box of size $|l'| \times |l'|$ and center it around l'. Draw two quarter circle segment with the same radii $r = \frac{l'}{2}$ along the alternation. The area necessary is already guaranteed, all left to examine is the complexity increase in the alternating case which takes two bends away (get rid of the vertical line) and substitute with two quarter circle segments which result in three new bends. In the uniform case the complexity does not increase. □

Connected to a vertex

If a vertical segment is connected to a vertex, then the boxing argument does not apply, since the space between multiple ports on the same side of a vertex is not altered by the stretching technique. We will see that this will not lead to a problem.

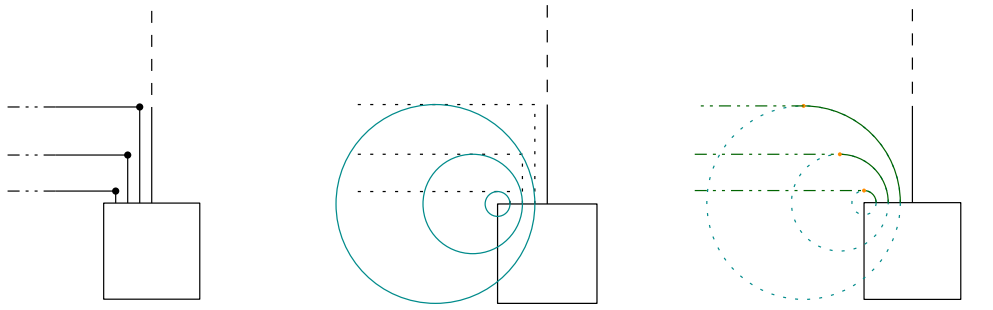
Lemma 3. *Let Γ_G be any orthogonal Kandinsky drawing and $v \in V_G$ a vertex with its ports. Then, there is at most one edge connected to each port of v with only one edge segment (complexity = 1). This describes the bend or end property of the polyedges regarding a single port.*

Proof. The vertices inherit a uniform size and are centered on a grid point. Either, a vertex is connected to another vertex with a single segment, then the center of both vertices share either the same x or the y coordinate, or two vertices are connected with a polyedge, then it has to bend at some point. The bend or end property of Kandinsky graphs is implied by the coarse grid on which the vertices are positioned. □

So, regarding a port of a vertex v, the edges either bend or end in a vertex. Now we want to examine the area regarding the circular arcs of the SMOG model.

Lemma 4. *Consider a vertex v and p edges connected to a port. Then, in SMOG, at most one of the p edges consists of a single segment ending in a vertex (as in Kandinsky). The other minimal $p-1$ edges start off with a circular arc, the original bend wanders along the horizontal segment and - most importantly - the complexity does not increase.*

Proof. The edge consisting of a single segment is not altered if it is vertical. If it is horizonal, the stretching technique adds the length of the longest segment. Therefore it does not change in its characteristics, only in length.
Consider the other minimal $p-1$ edges which are bending in some point (Figure 18a).



(a) Kandinsky situation (b) (c) SMOG representation

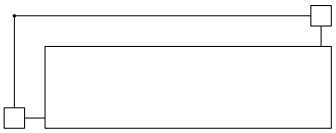
Figure 18: Smoothen multiple polyedges on the same port

The polyedges are connected equidistantly to a port. The increasing radii from outside-in (Figure 18b) and the original planarity in Kandinsky guarantee the planarity preservation of the resulting SMOG representation (Figure 18c). □

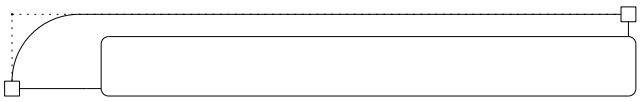
These results regarding graphs with arbitrary degree imply the functionality of the stretching technique even for the Kandinsky model. [3, p. 582, Section 4.1]

Corollary 1. *Let G be a graph with arbitrary degree and Γ_G a planar orthogonal drawing in the Kandinsky model. Then, the stretching technique from [3, p. 582, Section 4.1] is applicable guaranteeing the circular arc substitution. To be more precise, Γ_G can be postprocessed to a SMOG representation, taking $\mathcal{O}(n^2) \times \mathcal{O}(n)$ area.*

Proof. The *bend or end* property from Lemma 3 and the Kandinsky bends in SMOG from Lemma 4 guarantee the preservation of the planarity and the necessary space for the circular arc substitution is given, as we can see in Lemma 1. The worst case example drawing in the 4-planar situation regarding horizontal area bounds also takes effect in the k-planar situation.



(a) Original orthogonal graph



(b) After applying the stretching technique

Figure 19: Recall the worst case situation of Figure 8

□

3.3 Edge Complexity Investigation

As we saw in the previous section, the possibility of a SMOG representation is given even if the degree of the graph is arbitrary. All that remains is the examination regarding the number of bends of a polyedge. Previous results deliver an upper bound for 4-planar graphs, a polyedge with complexity k in the orthogonal drawing is bound by $\lfloor \frac{3}{2} \rfloor$ in the fixed shape model. We will see, that this upper bound still pertains.
Let G be a planar graph with arbitrary degree, an orthogonal drawing Γ_G is given and let k describe the edge complexity of a polyedge of Γ_G. The main goal of this section is to examine lower and upper bounds for k in SMOG.

Examining "good" and "bad" parts of a polyline

The general idea is to distinguish between two major cases: Line segments with alternating turns (staircase, zig-zags) and line segments with uniform turns (spirals, u-shapes). As we already saw, the SMOG representation of uniform turns does not increase the edge complexity. Therefore we examine polyedges regarding its properties of turns from one vertex to another. We will try to maximize the uniform part and simultaneously minimize the alternating part of a polyedge. We achieve this by so-called *fragmentation* of a polyedge.

Definition 13. *An edge fragment is a non-empty sequence of line segments. A fragmentation of a polyedge is a sequence of fragments.*

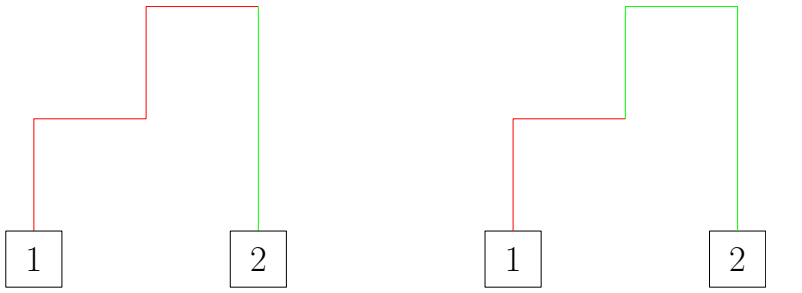
The main advantage of the fragmentation is that every fragment can be seen as an independent polyedge. The property of turns is crucial for the edge complexity thus the main criterion for the fragmentation.

Definition 14. *Like in Definition 4, a fragment is uniform if and only if all turns share the same direction. A fragment is alternating if and only if the turns are all alternating. A fragment is valid if and only if its turn property is either alternating or uniform.*

Lemma 5. *The fragmentation of a given polyedge e is not unique.*

Proof. Consider the polyedge $(1, 2)$ illustrated at Figure 20. Algorithm 1 takes the first three segments, considers its turn property and iterates along the polyedge. In the first picture, the segments starting from vertex 1 are obviously alternating, resulting

in an uniform fragment consisting of only one segment. Similarily, starting at vertex 2, the first three fragments are uniform, the rest along to vertex 1 alternating. Both fragmentations are correct.



(a) Fragmented from 1 to 2 (b) Fragmented from 2 to 1

Figure 20: Non-unique fragmentation

□

Definition 15. *Let e' and f' be two fragmentations. Then*

$$e' \sim_R f' \Leftrightarrow \Gamma_{e'} = \Gamma_{f'}$$

This means that two fragmentations are relative if and only if they describe the same polyedge in a given drawing.

Lemma 6. *The relation from Definition 15 is an equivalence relation.*

Proof. Describing the same image is trivially reflexive, symmetical and transitive. □

Definition 16. *Two fragments f and g are called incompatible if any segment transfer between f and g result in a turn property collision (Figure 21).*

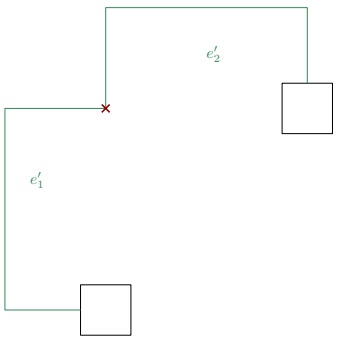


Figure 21: Incompatible uniform fragments e'_1, e'_2

What if the polyedge we want to fragment inherits an edge complexity of at most 2? Fragmentation is meant to partition large polyedges but the case for $k \leq 2$ still has to be considered. Note, that fragments of length up to two are simultaneously uniform and alternating. For a definite statement, those fragments lack in the third segment When we encounter a fragment of length two, we will interpret it as a uniform fragment first since uniformity does not necessarily increase the complexity. So now we are able to create a first - rather naive - algorithm in order to find a valid fragmentation.

Algorithm 1: `fragment_naively`$(e) \in \mathcal{O}(k)$

Input: Polyedge e with edge complexity $k, k \geq 2$
Result: e', a first fragmentation of e

```
1  if k = 2 then
2      e'_1 ← (e_1, e_2)
3      e'_1.uniform = true
4      e' ← (e'_1)
5  else
6      e'_1 ← (e_1, e_2, e_3)
7      e' ← ∅
8      i ← 1
9      if e'_1 uniform then
10         e'_1.uniform = True
11     else
12         e'_1.uniform = False
13     for j = 4 to k do
14         if e_j fits into the turn property of e'_i then
15             e'_i.append(e_j)
16         else
17             e'.append(e'_i)
18             i ← i + 1
19             e_i.uniform = ¬e_{i−1}.uniform
20             e_i ← (e_j)
21 e'.append(e'_i)
22 return e'
```

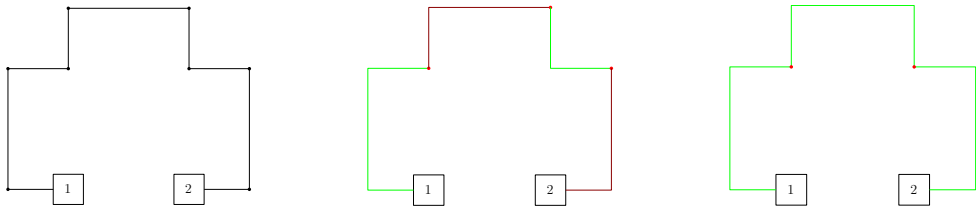
Correctness of Algorithm 1

This algorithm works for polyedges of length at least two. If the length is two, then the fragmentation will be unique since a fragment of length two is both uniform and alternating. If the complexity of the polyedge is greater than two, then lines 6 to 12 initialize the first fragment of length three and determines its turn property. For the remaining segments, this algorithm tests whether the next segment fits into the current fragment and appends it, if that is the case. If not, then the current fragment is appended to the returning fragmentation and a new fragment is created with the opposite turn property. By testing for every segment this algorithm returns a valid fragmentation and runs in $\mathcal{O}(k)$ by testing for every segment whether it would fit into the current fragment.

We will see in the following lemma, that this algorithm should be improved.

Lemma 7. *Algorithm 1 is not sufficient to determine a satisfying fragmentation.*

Proof. Consider following polyedge given in Figure 22. By applying algorithm 1, we achieve the following fragmentation. Green fragments illustrate uniform ones, red ones illustrate alternating ones. Red dots illustrate the breaking point in the fragment creation.



(a) original input polyedge (b) Result of algorithm 1 (c) more appreciating result

Figure 22: Bad example of a polyedge for algorithm 1

As we can see in Figure 22b, algorithm 1 delivers a fragmentation which consists of multiple fragments of length two. That is because algorithm 1 forces two consecutive fragments to differ in their turn property. As we already know, fragments of length two are not that conclusive in this case. The result given in Figure 22c consists of longer fragments and seems to be somehow more precise regarding the polyedge. The big difference lies in the abortion of the change constraint regarding consecutive fragments. □

How do we find the best way to describe a polyedge as a fragmentation? One way to approach it is to minimize the total number of fragments and the number of alternating fragments.

Lemma 8. *The amount of all possible fragmentations of a polyedge e is finite. To be more precise:*

$$|[e]_{\sim_R}| < 2^{k^2}$$

Proof. Given a polyedge e, the amount of possible fragments s is computed:

$$s \leq \sum_{i=1}^{k} \frac{k}{i} \cdot i = k^2$$

i denotes the length of the fragments starting from 1 to k. Consider e partitioned into $\frac{k}{i}$ fragments, then there are $\leq i$ ways to start the fragmentation. The offset lies in $[0, i-1]$. The cardinality of the power set containing all possible fragments values 2^{k^2}. □

As we already saw, there are multiple ways to describe a polyedge regarding a fragmentation. The next step is to determine, which fragmentation suits the best. Given a mathematical description to a given polyedge e, we want to describe the "best way" to determine the number of bends in SMOG. So we will pick the "best" fragmentation, which will be minimal in its number of alternating fragments and minimal in its total number of fragments. This leads to the following ordering relation:

Definition 17. *Let e', f' be two fragmentations of the same polyedge e ($e' \sim_R f'$). Then we define the so-called FragOrder relation:*

$$e' \leq f' \Leftrightarrow (\#_{altFrag}(e') \leq \#_{altFrag}(f')) \wedge (\#_{totalFrag}(e') \leq \#_{totalFrag}(f'))$$

Lemma 9. *FragOrder is sufficient to determine a minimum of $[e]_{\sim_R}$.*

Proof. Obviously, the relation $\leq$ is reflexive and transitive. Therefore, this relation is a preorder and we are able to determine a minimum of $[e]_{\sim_R}$ because in a finite set there is a minimum regarding a preorder. □

Comparing all possible fragmentations for the minimum will result in an exponential runtime. In order to fix this problem, we will take a slightly different approach - we will further get rid of alternating fragments by creating uniform-only fragments. This will result in longer fragmentations at first but we will see that it will be acceptable due to a new interpretation of this fragmentation.
We will be able to substitute alternating fragments as a sequence of uniform fragments.

Lemma 10. *A fragment f of length k' is alternating if and only if its purely uniform fragmentation f' of size $\lceil \frac{k'}{2} \rceil$ inherits uniform incompatible fragments of length 2 and $f \sim_R f'$ regarding the FragOrder relation. This enables us to distinguish uniform fragments from alternating ones in the output of Algorithm 2.*

Proof. If a fragment f of length k' is alternating, then every bend differs in its direction regarding the previous one. A fragment of length 2 inherits one bend. In an alternating fragment, the next one will cause a turn property collision, resulting in a new fragment. Again, fragments of lengths up to 2 are uniform by definition. Similarily, consider a sequence of fragments of length 2. If there were an optimization possible, then the sequence would not be completely incompatible as stated. This implies the staircase situation. □

We are willing to lengthen the fragmentation by coincidentally describing more precisely. This leads to a modification of algorithm 1.

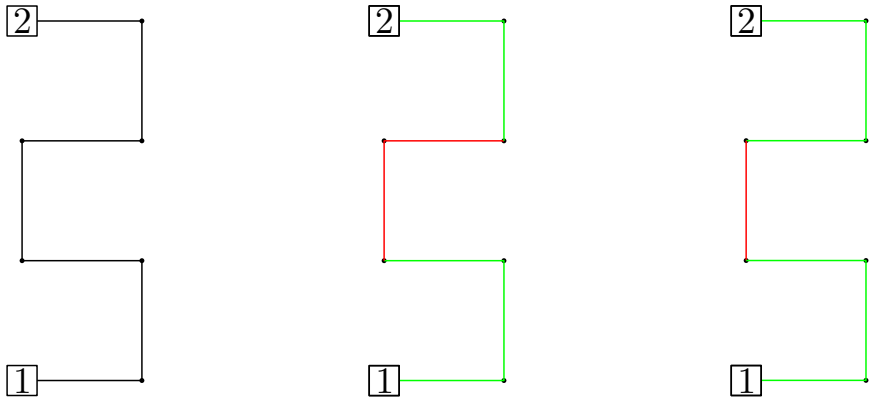
Algorithm 2: `fragment_uniform-only`$(e) \in \mathcal{O}(k)$

Input: Polyedge e with edge complexity k,$k \geq 2$
Output: Almost optimal fragmentation of e

```
1  e'_1 ← (e_1, e_2)
2  f ← ∅
3  i ← 1
4  for j = 3 to k do
5      if e'_i.append(e_j) is uniform then
6          e'_i.append(e_j)
7      else
8          f.append(e'_i)
9          i ← i + 1
10         e'_i ← (e_j)
11 f.append(e'_i)
12 Recheck(f)
13 return f
```

Correctness of algorithm 2

This algorithm is pretty similar to algorithm 1 with the big difference, that every fragment is now uniform. Initializing f as a list of fragments, this algorithm appends the current fragment if the next segment collides with its turn property. Unless there is only one segment left in the end, every fragment consists of at least two segments. On the other side, there are optimal fragmentations which inherit a fragment of length one. Consider algorithm 2 without line 12.



(a) original input polyedge (b) Result of algorithm 2 (c) more appreciating result

Figure 23: Bad example of a polyedge for the second uniform-only approach (red highlights the problematic fragment)

In Figure 23 the output of Algorithm 2 returns uniform-only fragments. The colour highlighting is used this time to illustrate the problem description. In Figure 23b, we see that the length of the fragments equals $(3, 2, 2)$, whereas in Figure 23c the individual length is $(3, 1, 3)$. We are still almost there, although there are situations, where the "bad" part of a fragmentation is of length one. Still, algorithm 2 functions the way we want to. We only have to *recheck* the fragmentation (Line 12 in algorithm 2), whether a fragment of length two can be further minimized by merging the last segment with the next fragment.

Algorithm 3: `recheck`$(f) \in \mathcal{O}(k)$

Input: A fragmentation f
Output: A fragmentation f rechecked

```
1 for i = 1 to f.length−1 do
2     if f_i.length = 2 then
3         v ← first segment of f_i
4         u ← second segment of f_i
5         if (u).append(f_{i+1}).uniform = true then
6             f_i ← (v)
7             f_{i+1} ← (u).append(f_{i+1})
8 return f
```

Obviously, this algorithms runtime is linear to the edge complexity of the input orthogonal polyedge.

Theorem 7. *Algorithm 2 is sufficient to determine the edge complexity of any polyedge in the smooth orthogonal drawing. Furthermore, the resulting fragmentation can be used to describe a polyedge adequately.*

Proof. We already know that uniform fragments do not increase in their complexity and the transition between two incompatible fragments increases the complexity by 1. This leads to a new computation regarding the edge complexity of any polyedge in the smoothened drawing:

$$ec(f) = \sum_{i=1}^{f.\texttt{length}} ec(e_i') + \underbrace{(f.\texttt{length} - 1)}_{\text{fragment transitions}} \tag{5}$$

The information of alternating fragments does not get lost. By substituting consecutive fragments of up to length two with one big alternating fragment, we are able to describe the behaviour of the polyedge adequately. □

Now we want to establish an algorithm which modifies the fragmentation resulted from algorithm 2 to describe the minimum.

Algorithm 4: Min computing of all the fragmentations regarding relation 17

Input: Fragmentation f computed by algorithm 2 and 3
Output: Minimal fragmentation $e_{\min}$ regarding the relation 17

```
1  e_min ← ∅
2  f_alt ← ∅
3  f_alt.uniform = false
4  for i = 1 to f.length do
5      if f_i.length ≥ 3 then
6          if f_alt ≠ ∅ then
7              e_min.append(f_alt)
8              f_alt ← ∅
9          f_i.uniform = true
10         e_min.append(f_i)
11     else
12         f_alt.append(f_i)
13 if f_alt ≠ ∅ then
14     e_min.append(f_alt)
15     f_alt ← ∅
16 return e_min
```

Correctness of algorithm 4

This algorithm gets the uniform-only fragmentation from algorithm 2 as input and determines whether a fragment is of length greater than two. If so, then this fragment is purely uniform and can be considered that way. However, if a fragment is of length one or two, these segments did not fit in any other uniform fragment due to collision reasons. They can be considered as alternating. By using Lemma 10, we know that consecutive fragments of length at most two are analogous to an alternating fragment. This algorithm merges those consecutive segments initialized in line 2-3 and appends them, if a consecutive fragment is of length at least three or there are no further fragments left at the end of the `for` loop. The longest possible fragmentation given by algorithm 2 is of length $\lceil \frac{k}{2} \rceil$, where k is the complexity of the original orthogonal polyedge and therefore this algorithm also terminates in $\mathcal{O}(k)$ runtime.

Theorem 8. *The fragmentation resulting from Algorithm 2 and 4 describes a minimal fragmentation regarding the FragOrder relation*

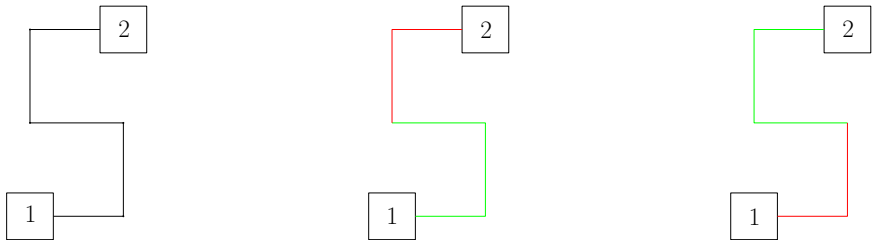
Proof by contradiction. Let $e_{\min}$ be the result of Algorithms 2 and 4 and $f_{\min}$ be the another valid fragmentation regarding the *FragOrder* relation. Assume that $f_{\min} < e_{\min}$. This would mean that the number of total fragments in f_{min} would be less and the number of alternating fragments are at most the same. This would mean there would be a fragment which can be split into the at most two consecutive fragments. Since Algorithm 2 merges uniform ones as long as possible and Algorithm 4 merges

alternating ones as long as possible, there is no possibility that this fragment could not have been saved beforehand. The same argument holds for the case that $f_{\min}$ would have less alternating fragments. So $f_{\min} \not\leq e_{\min}$. □

The only thing left to consider is the uniqueness of the computed minimum. We will see that the fragmentations can differ if we change the direction of fragmentation.

Lemma 11. *Let $e = (1, 2)$ be a polyedge. Then, the optimal fragmentation started at vertex 1 may differ from the fragmentation started at vertex 2.*

Proof. Consider the following example.



(a) Input polyedge $(1, 2)$ (b) Fragmented from 1 to 2 (c) Fragmented from 2 to 1

Figure 24: Non-unique optimal fragmentation regarding direction

The fragmentation illustrated in Figure 24b from 1 to 2 starts with an uniform fragment of length three, which is not contained in the fragmentation the other way around (Figure 24c). □

We see, that the fragmentation is still pretty similar. The next lemma holds the uniqueness of the minimal fragmentation in sense of direction and that the choice of direction does not matter for the validity.

Lemma 12. *The fragmentation resulting from algorithm 2 and 4 is the unique minimum with respect to the direction of the fragmentation. The two minima regarding both fragmentation directions share the same property regarding the number of (alternating) fragments.*

Proof. Consider two fragmentations f_1 and f_2 regarding the same polyedge e, both computed reciprocally with Algorithm 2 and 4. Then, Theorem 8 holds and the number of alternating fragments and the total number of fragments are of the same size. However, as we saw in Figure 24, the fragmentation itself can differ. □

Now we achieved the best mathematical description of a polyedge in form of a fragmentation. This goes along pretty well with the already achieved results. The results of purely uniform or purely alternating polyedges also apply for purely uniform or purely alternating fragments because every fragment can be considered as a seperate polyedge simulated with dummy vertices. We will see that a fragment with edge complexity k' can be transferred to a SMOG fragment with edge complexity at most $\lfloor \frac{3}{2}k' \rfloor$. In fact, we will also see that every fragmentation with complexity k will result in a SMOG polyedge with complexity at most $\lfloor \frac{3}{2}k \rfloor$.

3.4 Results

In order to talk about the edge complexity of SMOGs drawings with arbitrary degree we have to consider the number of bends of the polyedges in SMOG. In this section we will look at the possible length of fragmentations, the merge of two fragmentations resulting in one extra bend and will summarize the results.

Lemma 13. *Let e be a polyedge with edge complexity k. Then the range of fragmentation lengths is bounded by $\lceil \frac{k}{2} \rceil$. To be more precise:*

$$|e'| \in \left\{1, ..., \left\lceil \frac{k}{2} \right\rceil \right\} \quad \forall e' \in [e]_{\sim_R}$$

Proof. If a polyedge with complexity k is purely alternating, then algorithm 1 will return a valid fragmentation consisting of a single alternating fragment whereas algorithm 2 will return a valid fragmentation of $\lceil \frac{k}{2} \rceil$ fragments. It is possible, that this last fragment is of length one.

□

Due to the stretching technique it is guaranteed that every fragment can be substituted with a SMOG fragment.

Lemma 14. *The complexity of an orthogonal fragment increases by a factor of $\frac{3}{2}$, iff the fragment is alternating. The complexity does not increase at all, if and only if the fragment is uniform in $\mathcal{O}(n^2) \times \mathcal{O}(n)$ area.*

This is a property given by the paper of Bekos et al. and as already proven it does not matter whether the fragments are between other fragments or connected to a vertex. [3, Figure 6, p. 584].
As seen in Figure 21, the situation arises that one converse bend relative to the others leads to incompatible fragments.

Lemma 15. *The transition of two incompatible fragments increases the edge complexity by one bend.*

Proof. If two fragments are incompatible then the degree change between 90° and 270° result in a circular arc substitution, analoguous to the staircase situation. This results in one more bend per incompatible fragment transition. □

Lemma 16. *Let Γ_G be a planar orthogonal drawing of a graph G with arbitrary degree. A purely alternating polyedge e with edge complexity k is the worst case situation regarding the number of bends after postprocessing the drawing to a smooth orthogonal drawing. This means that the edge complexity of every polyedge in SMOG is bounded by $\lfloor \frac{3}{2}k \rfloor$.*

Proof. Consider equation 5, computing the edge complexity of a postprocessed polyedge. At first, let e be a purely alternating polyedge. It follows for its optimal fragmentation:

$$\mathsf{ec}(f_e) = \sum_{i=1}^{\lceil \frac{k}{2} \rceil} \mathsf{ec}(e'_i) + \left\lceil \frac{k}{2} \right\rceil - 1$$

We have to consider two major cases - k being an even or an odd number.
k even
If k is even, then there exists a $n \in \mathbb{N}$ with $k = 2n$. It follows:

$$\sum_{i=1}^{n} 2 + n - 1 = 3n - 1$$
$$= \frac{3}{2}k - 1 < \frac{3}{2}k$$

k odd
If k is odd, then there exists a $n \in \mathbb{N}$ with $k = 2n - 1$.

$$\sum_{i=1}^{\lceil \frac{k}{2} \rceil} \mathtt{ec}(f_i) + \left\lceil \frac{k}{2} \right\rceil - 1$$

If the polyedge is of odd length and purely alternating, then there are $n-1$ fragments of length two and one fragment of length one.

$$\begin{aligned}
&\sum_{i=1}^{n-1} 2 + 1 + \left\lceil \frac{k}{2} \right\rceil - 1 \\
=&2n - 2 + 1 + \left\lceil \frac{2n-1}{2} \right\rceil - 1 \\
=&2n - 2 + \left\lceil n - \frac{1}{2} \right\rceil \\
=&3n - 2 = 3\left(\frac{k+1}{2}\right) - 2 = \frac{3}{2}k - \frac{1}{2} = \left\lfloor \frac{3k}{2} \right\rfloor
\end{aligned}$$

Now, let f_e be an optimal fragmentation of a non-trivial polyedge e computed by algorithm 2. Then, e is not purely alternating and by Theorem 7 at least one of the fragments is of length three or greater.

$$\Rightarrow f_e.\mathtt{length} < \left\lceil \frac{k}{2} \right\rceil$$

Longer uniform fragments shall not be a major problem, because their complexity does not increase. On the other hand, by a shorter fragmentation length, we save at least one transition bend mentioned in Lemma 15. Now, it immediately follows that the edge complexity does not exceed the staircase situation. □

Theorem 9. *Every orthogonal planar drawing of a graph with arbitrary maximum degree can be transferred into a SMOG with the complexity increase bounded by the factor* $\frac{3}{2}$ *in* $\mathcal{O}(n^2) \times \mathcal{O}(n)$ *area.*

Proof. Let e be a polyedge with edge complexity k. If e is purely uniform or alternating, Lemma 14 takes effect and the Theorem is true. If the polyedge is fragmentated non-trivially, we examine the composition with algorithm 2 and Lemma 16 takes effect that the number of bends are bound by the worst case staircase situation. In the previous section, we saw that the area is guaranteed due to the stretching technique and thus planarity is preserved. □

These results serve as an upper bound for the worst case of some Kandinsky drawings. In practice, the observations are quite different. Staircase situations only occur up to a certain complexity - for polyedges with higher complexity, it is guaranteed that the shape of the polyedge is mostly uniform. This is due to an optimization in the whole Kandinsky drawing algorithm. The Kandinsky drawings are of course not always computed with a minimal number of bends - as we know, this would be $\mathcal{NP}$-hard to decide, whether a given graph drawing inhibits the minimal number of bends - but the edge complexity is greedily decreased at a certain point of computation. This leads to a much better upper bound for the edge complexity increase of polyedges.

Lemma 17. *Let $k \geq 4$ be the edge complexity of a polyedge e from a Kandinsky drawing Γ_G with two Kandinsky bends. Then, the complexity increases to $k+2$ in the smooth orthogonal layout.*

Proof. e with edge complexity at least 4 has got at most two Kandinsky bends. What lies in between, is purely uniform. Algorithm 2 and 4 return therefore a fragmentation of length three. At the beginning and the end, there are two alternating fragments of length one, in between lies a purely uniform fragment of length $k-2$. In SMOG, the two Kandinsky bends raise the complexity by one bend. The uniform case does not increase the complexity at all, just as the alternating fragments of length one. Therefore, the edge complexity would increase to $k+2$.

$$\left\lfloor \frac{3}{2}k \right\rfloor = k+2 \Rightarrow k = 4$$

Therefore, the edge complexity rises by two if the polyedge inhibits at least four segments. □

4 Saving measures

4.1 Introduction

With our existing approaches we are able to translate any Kandinsky drawing with arbitrary degree to a smooth orthogonal layout with a complexity increase by 2 for larger polyedges in $\mathcal{O}(n^2) \times \mathcal{O}(n)$ area. The question arises whether it was possible to even save some bends or find a method to stretch with less area requirements in the worst case. In the following section, we examine possibilities to decrease the complexity of polyedges. Furthermore we try to lower the area upper bound in a possible tradeoff with some more bends.

4.2 Edge Complexity Bounds

Reconsider the drawing given in Figure 12. Due to the fact that the previous results considering 4-planar graphs are also holding for graphs of arbitrary degree, the results are applicable.

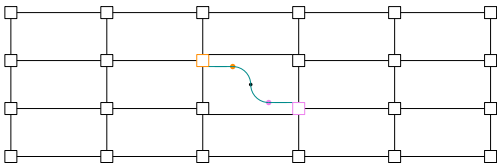


Figure 25: Recall Figure 12

Theorem 10. *Every Kandinsky drawing of a complexity-3 graph with arbitrary degree can be postprocessed to a complexity-4 smooth orthogonal layout in $\mathcal{O}(n^2) \times \mathcal{O}(n)$ area.*

Proof. Just as the proof of Theorem 5 holds, the Kandinsky bends are not erasable, leading to a staircase situation of complexity 3. The boxing brings along one more bend, resulting in a polyedge of complexity 4. □

The difference in our new situation is that we are able to reposition the polyedges on the ports. By having a 4-planar graph interpreted in Kandinsky, we are able to connect up to four edges on a single port. This gives us a new opportunity to optimize the drawings.

Remark 1. *The bend or end property complicates the port reassignment possibilities.*

If we recall Figure 12, we could try to alter the port the edge is connected to. Unfortunately, the uniformity of the vertex boxes lead to a possible collision illustrated in Figure 26a. By doing so, we would further increase the complexity of the edge.

(a) *bend or end* collision (b) Complexity increase

Figure 26: Bend or end property complicates matters

But on the other hand, the circular arcs are flexible enough to achieve a possible complexity decrease.

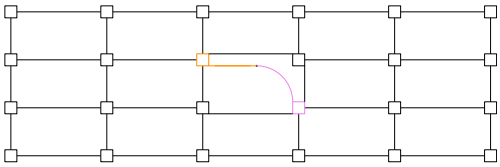
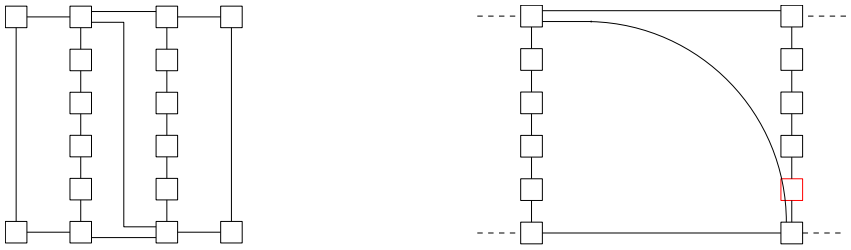


Figure 27: SMOG complexity decreases

However, this is not always possible, as we see in the following Figure:



(a) (b)

Figure 28: This might just be a candidate with negative result

An approach to find candidate edges for a port reassignment could lie in "half-bends" and diagonal segments, seen in the L shape and the T shape of the *Kandinsky drawings with almost-empty faces* (Podevsaef drawings in short). In our first approach, we mainly focus on complexity-3 zig zags in the original Kandinsky Model.

4.2.1 Podevsaef drawings

Recalling definition 8, Podevsaef drawings mainly differ from SMOGs in their diagonal segments and 135° bends. The approach to create a Podevsaef drawing is pretty similar to the smooth orthogonal case. At first, the planar Kandinsky drawing of the original graph G is computed which then gets modified by a modification of the *Topology Shape Metrics* algorithm ([4, p. 4]). The usage of diagonal segments enable us to illustrate triangular faces with two 135° bends rather than 90° bends. This solution for the bend or end property difficulty enables us to illustrate *almost empty faces*, still inheriting bounding edges with a 0° difference but not orthogonally, therefore *almost* empty. The following solutions were found for L shaped and T shaped triangles:

(a) (b)

Figure 29: L and T shapes with almost empty faces

Inspired by this illustration, we look for candidates for possible circular arc substitutions. Both examples from Figure 25 and Figure 28 might suit for a port reassignment.

Although the complexity decrease possibility differs, they both share the achievable planar diagonal half-bend substitution after the application of the stretching technique.

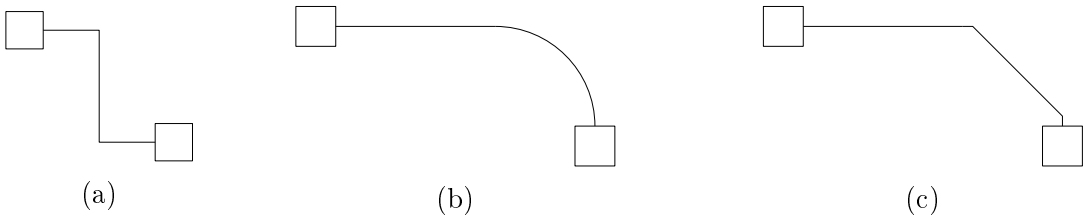


Figure 30: Complexity-3 zig zags examined

Lemma 18. *If a polyedge representation Γ_e is zig-zag shaped in Kandinsky (Figure 30a) and the port reassignment and circular arc substitution decrease the complexity in the resulting planar SMOG representation (Figure 30b), then there is a planar Podevsaef representation with two 135° bends (Figure 30c).*

So, according to lemma 18, if we look for the possibility of planar Podevsaef polyedge representation like illustrated in figure 30c in a polyedge, the port reassignment of one of the vertices might just decrease the complexity. But as we seen, not in every case.

4.2.2 Using the fragmentation

Another approach is to use the optimal fragmentation regarding a polyedge to determine whether it was possible to save some bends. The fragmentation itself does not consider the horizontal or vertical alignment of segments in the plane. The following example will motivate the next lemma:

Lemma 19. *If the optimal fragmentation of a polyedge contains a fragment of length one in between two other fragments and its line segment is vertical, then the complexity does not increase at this incompatible fragment.*

Proof. Consider the alternating fragment consisting of a single vertical segment in the optimal fragmentation (Figure 31b). Then, the fragments adjacent to it are uniform. Those fragments share the same turn direction because in the uniform-only fragmentation the fragments are alternating their direction of turns. Consider that the original fragment was of length two without the recheck. The next fragment is of length at least three because the fragmentation algorithms have been shifting a second segment. This particular situation enables us to substitute the vertical segment with a half-circular arc due to the same turns of the segments before (Figure 31c).

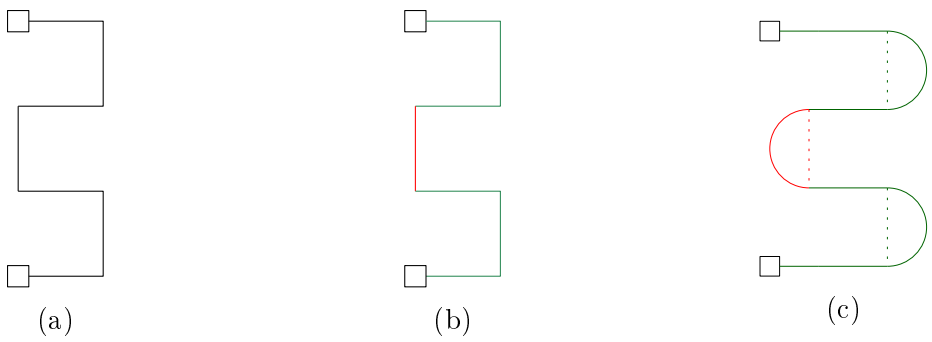


Figure 31: Illustration of the alternating fragment exception

□

4.3 Area Bounds

In this section, we will examine the possiblities of area bound optimization. Suppose that the original orthogonal drawing is very large and contains a large number of vertices, it is of interest to find a way to lower the upper bound. In the first approach, we will erase redundancies in a drawing with a plane sweep method. In the second approach, we will substitute the circular arcs with ellipses or specific segment combinations in order to lower the upper bound by $\mathcal{O}(\sqrt{n})$, which may increase the lower bound of the edge complexity on the other hand.

Plane sweep erasing

The stretching technique does not increase the edge complexity excessively. On the other hand, it may appear that the horizontal area expansion of a drawing is unnecessarily big. In this section, a linear runtime plane sweep may be a stable solution regarding area and even edge complexity retrenchment. Consider the following example:

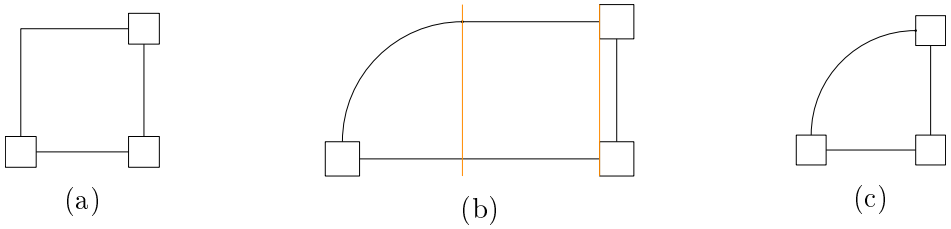


Figure 32: One plane sweep could eradicate redundant area and even segments without causing any damage

As you can see in Figure 32, an orthogonal Kandinsky drawing of a triangle gets transferred to a smooth orthogonal drawing in the Fixed Shape Model. The vertical orange-colored lines show the possible saving of area. Even one horizontal segment is eliminated in the process, reducing the complexity of the drawing to one.

Definition 18. *There is a plane sweep algorithm which eliminates unnecessary space and may even save some segments in linear runtime regarding the size of the drawing. The vertical segments do not have to be considered by the plane sweep.*

This plane sweep algorithm identifies the presence of *vertices* and *circular arcs* as an event. Obviously, it is undesirable to interfere with circular arcs while cutting the drawing. First we show, that considering vertices, circular arcs and horizontal line segments is sufficient.
The *vertical line segments* are not to be considered because either they end in one or two vertices. This means that vertices are sufficient in this case to be seen. If a vertical line segment is not connected to any vertex, then, by definition of the SMOG Model, they are connected to circular arcs which are also considered by the sweep line.
The horizontal segments are part of the events in order to determine which segments can be cut between two events. The plane sweep iterates from the left side to the right and is able to delete unnecessary horizontal redundancy with one sweep. Therefore, this plane sweep will run in linear runtime regarding the size of the drawing.

Lemma 20. *This area saving plane sweep is able to save up to $\mathcal{O}(\sqrt{w})$ area, where w is the width of the input drawing and it potentially lowers the overall complexity of a drawing.*

Proof. Consider following orthogonal drawing of the graph:

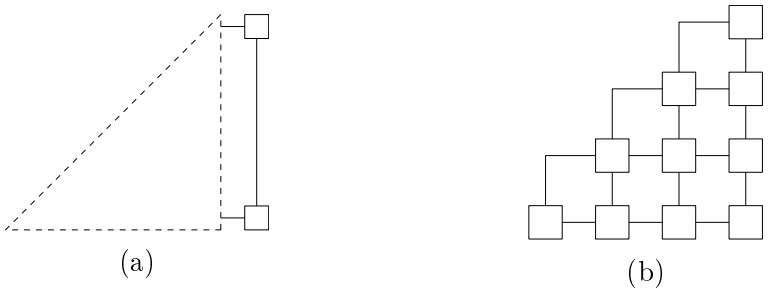


Figure 33: Orthogonal drawing with maximal saving possibilites

The dotted triangle of figure 33a consists of $\mathcal{O}(n) \times \mathcal{O}(n)$ vertices as illustrated in figure 33b. Applying the stretching technique and circular arc substitution, the resulting SMOG drawing is of size $\mathcal{O}(n^2) \times \mathcal{O}(n)$.

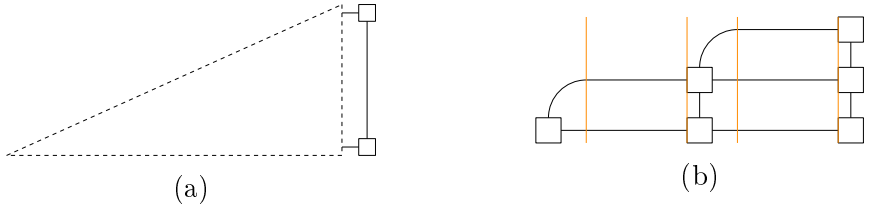


Figure 34: SMOG drawing with maximal saving possibilites

In figure 34b, the orange dotted lines indicate where the area saving plane sweep triggers an event. Notice that there are $\mathcal{O}(n)$ events.

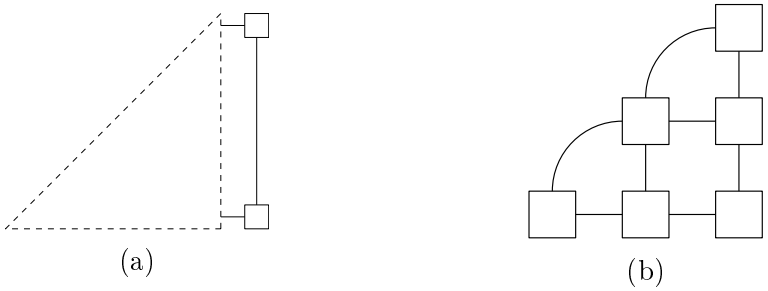


Figure 35: SMOG drawing after applying the plane sweep

After cutting area redundancies, we decrease the complexity of the graph to one and save $\mathcal{O}(n)$ area horizontally. □

However, the plane sweep will not work properly in all scenarios, as the following example will show:

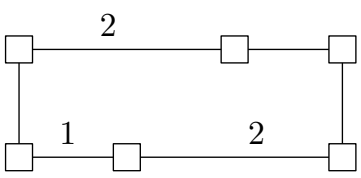


Figure 36: The plane sweep will not reduce anything

Consider the drawing in Figure 36 with a unit length of 1. While iterating from left to right, there will be a vertex as a new event with unit length difference to the last event. However, this drawing can be optimized regarding horizontal area as shown in figure 37.

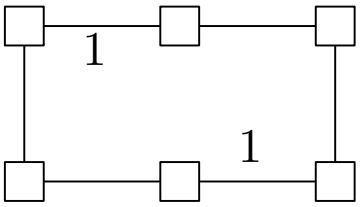


Figure 37: Optimized drawing derived from figure 36

In order to find a solution for this situation, a new approach based on the *4M* algorithm is formulated.

4.3.1 Modified 4M - Moving

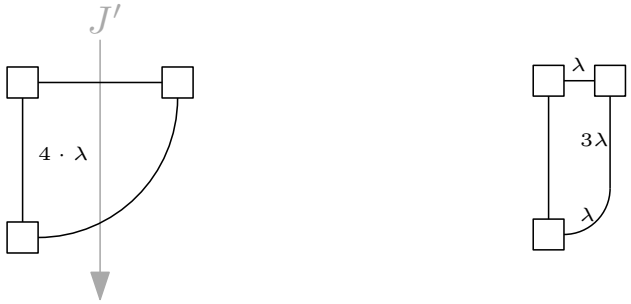
Recalling section 2.3, the 4M algorithm consists of multiple operations. In this section, we will consider the *moving* operation.

Model consistency

It is to examine whether the *4M* operations were suitable for drawings which underly the Kandinsky model. In the Kandinsky model, the vertices are represented with boxes of uniform size. In the orthogonal 4M model, the boxes have to overlap the grid lines by $\frac{\lambda}{4}$, and the rectangles were of size $\left(w\lambda - \frac{\lambda}{2}\right) \times \left(h\lambda - \frac{\lambda}{2}\right); w, h \in \mathbb{N}$. By setting $w = h = 1$, we achieve a consistency with the Kandinsky model and therefore can continue examining a 4M algorithm modification.

The modification

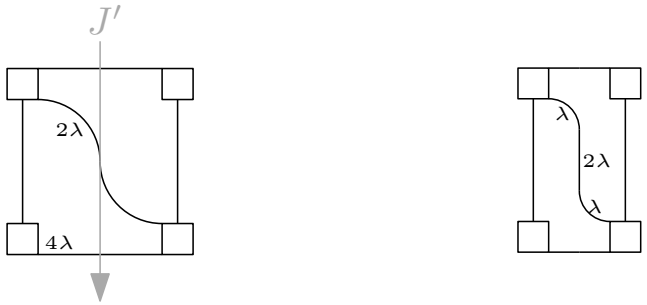
The focus lies in the horizontal area saving. The moving line J will still be directed and starts above to topmost object and ends below the bottommost object of the drawing. Considering orthogonal graphs, J only crosses horizontal lines from top to bottom. In the smoothened Kandinsky case, quarter circular arcs traverse in height and width - so they also have to be crossed for total area savings.



(a) J crossing the drawing (b) Circular arc substitution

Figure 38: A first, easy example

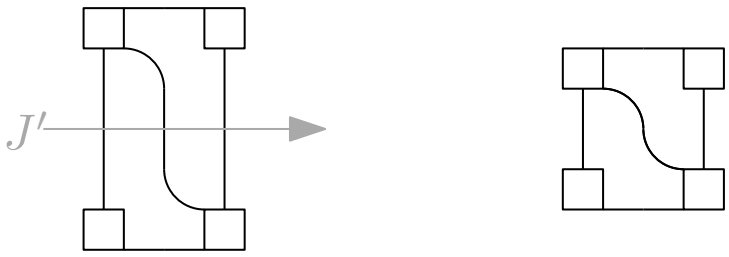
As you can see in figure 38, a quarter circular arc with radius r can be reduced in width by substituting with a circular arc with radius $r - c \cdot \lambda$ and a vertical segment of length $c \cdot \lambda$. λ describes the unit length, c describes the amount of unit reducings possible along J. By this method, the complexity may increase by one per crossed quarter circular arc.



(a) J' crossing the drawing (b) Quarter circular arc substitution

Figure 39: A second example

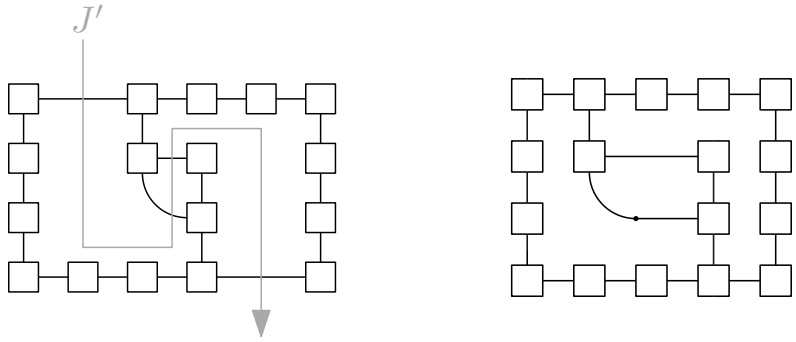
In this example, J' crosses two consecutive quarter circular arcs in a drawing of size 6×6. The result is a drawing of size 4×6, with an edge complexity of three. A verticel line segment of length one is introduced per substituted arc, merged to a vertical line segment of length two.
To further decrease the area of the drawing, a horizontal path from the leftmost object to the rightmost one would reduce and even eliminate the vertical line segments crossed by J'.



(a) J' crossing the drawing (b) Circular arc substitution

Figure 40: Vertical area saving

The area bounds of the drawing from Figure 39a can further be reduced by a horizontal moving line J', resulting in a 4×4 drawing. In order to do so, one has to *reduce the line segment between quarter circular arcs first.* This results in a possible complexity decrease, in this case a drawing of complexity two.
Recall the elongation of horizontal edges crossed by the original moving line J with an upward direction. What, if a quarter circular arc is crossed with an upward piece of J'? Then, the idea is to add a horizontal line to the quarter circular arc correspondingly with unit length. Consider the following example:



(a) J' crossing the drawing (b) Additional horizontal segments

Figure 41: The additional horizontal line segment increases the complexity of the drawing to two

This idea implies the modified moving approach.

Definition 19 (modified moving line J')**.** *An area-saving moving line J' for a smoothened drawing Γ'' is a line that fulfills the following conditions:*

- *J' is directed and consists of horizontal and vertical segments*
- *J' starts above the topmost object of Γ'' and ends below the bottommost object of Γ''*
- *J' does not intersect any vertical edge of Γ''*
- *In a polyedge e, either quarter circular arc segments or horizontal line segments might be crossed by J'*
- *Every horizontal edge of Γ'' that is intersected by a piece of J' which is directed downward has a finite length larger than or equal to two. If this horizontal line segment is connected to a circular arc segment, then it has a finite length larger or equal to one.*
- *Every circular arc segment of Γ'' that is intersected by a piece of J' which is directed downward has a finite radius larger than or equal to two.*

Since Γ'' is planar, its dual graph is well-defined and planar. J' can easily be found with *depth first search* in the planar dual graph. Then, after finding J', a smaller drawing can be computed the following way:

- The length of every edge of Γ'' that is crossed by J' in downward direction is decreased by one unit
- Simultaneously, the length of every edge crossed by J' in upward direction is increased by one unit
- If a quarter circular arc is crossed by J' in downward direction, then the arc is substituted by a quarter circular arc with a smaller radius by one unit with a vertical segment of unit length (refer to Figure 38)
- If a quarter circular arc is crossed by J' in upward direction, then a horizontal line segment of unit length is appended to the quarter circular arc correspondingly

Note, that this modified moving algorithm separates the drawing in two parts according to J' and then moving one part closer to the other one. The direction of area saving can be altered by switching the roles of vertical and horizontal line segments and finding a horizontal directed path J' instead of a vertical one.

Correctness

Consider a quarter circular arc c with radius r crossed by J' in a smoothened drawing. By substituting c with a corresponding horizontal and vertical line segment, h_c and v_c with length r, the original moving algorithm preserves the planarity of the drawing.
If the original moving line J crosses h_c downwards, then h_c will be shorter in length than v_c. The substitution with a smaller circular arc with radius $|h_c|$ will increase the total edge complexity since $|v_c| > |h_c|$.
If the original moving line J crosses h_c upwards, then h_c will be larger in length than v_c. The substitution with the input circular arc with radius $|v_c|$ increases the complexity since $|v_c| < |h_c|$.
The area necessary for circular arc substitution is already proven in the previous section, preserving the planarity.
All we need to prove is the choice of segment to cross, when a horizontal line segment h is appended to a circular arc c. We will be able to reduce h successively until h may be completely erased because c is still unaltered and travels the width and height necessary for the drawing. Thus, we will choose h for being crossed by J', possibly achieving an edge complexity reduction.

4.3.2 Circular arc substitution

The circular arcs used in smooth orthogonal drawings have a height and width of r and are the main reason for the quadratic total width in the worst case. In this section, we examine the possibilites of saving some space by substituting the circular arcs used with different segments. In our first approach, we will use ellipses to guarantee a width of $\sqrt{r}$, saving at least $\sqrt{n}$ area requirements. However, the aesthetics may suffer for large values. In our second approach, we will substitute the circular arc with a combination of a smaller circular arc and a vertical segment, also demanding $\sqrt{r}$ width. The aesthetics may be preserved but this definitely will increase the edge complexity of any drawing.

Ellipses

Using a quarter of an ellipse, we could achieve a guaranteed width of $\sqrt{r}$, resulting in drawings of size $\mathcal{O}(n \cdot \sqrt{n}) \times \mathcal{O}(n)$. We will take a look at following example equation given for an ellipse:

$$\frac{x^2}{5} + \frac{y^2}{25} = 1 \qquad\qquad x, -y \in \mathbb{R}_+ \tag{6}$$

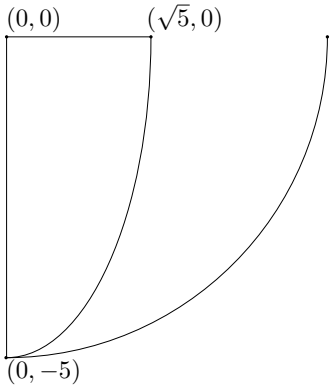


Figure 42: Illustration of equation 6

For the orientation of the ellipse arcs, we will pick the values of x and y adequately. The extreme values of equation 6 are $(0, -5)$ and $(\sqrt{5}, 0)$, it would fit right in instead of a circular quarter arc with radius 5. If 5 was the longest vertical segment, it would be sufficient to stretch the drawing by $\sqrt{5}$. This implies - given a height l, serving as radius for the original circular arcs - following equation:

$$\frac{x^2}{l} + \frac{y^2}{l^2} = 1 \qquad x, -y \in \mathbb{R}_+ \tag{7}$$

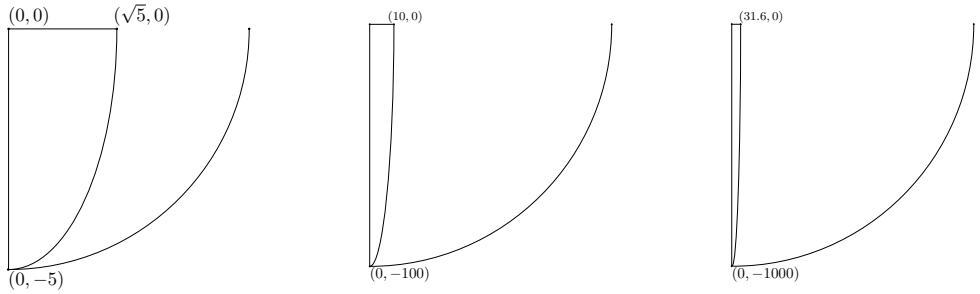
With following extreme values: $(0, -l)$ and $(\sqrt{l}, 0)$. For every vertical segment of length l', we could compute the ellipse given by equation 7 and by gauging our values for x and y we could pick the right orientation of the arc from its appropriate quadrant.

Correctness

The argument is very similar to the "Boxing". Instead of a square box, we will deal with rectangles of size $w \times h$. Let l be the longest vertical segment in a given orthogonal drawing Γ_G. By stretching the drawing by the factor of $\sqrt{|l|}$, every vertical line segment v now has a free rectangle area of size $\sqrt{|l|} \times |v|$ left and right from it. Therefore, v has a free rectangle area of size $\sqrt{|v|} \times |v|$, since $|v| \leq |l| \Rightarrow |\sqrt{v}| \leq |\sqrt{l}|$, guaranteeing the free area for the ellipse arc substitution and a drawing of size $\mathcal{O}(n \cdot \sqrt{n}) \times \mathcal{O}(n)$ in the worst case.

Readability

Using arcs from ellipses seems to be a good idea at first since we can actually save space, even in the worst case. But using those arcs decrease the readability of a drawing, the bigger the longest vertical segment gets.

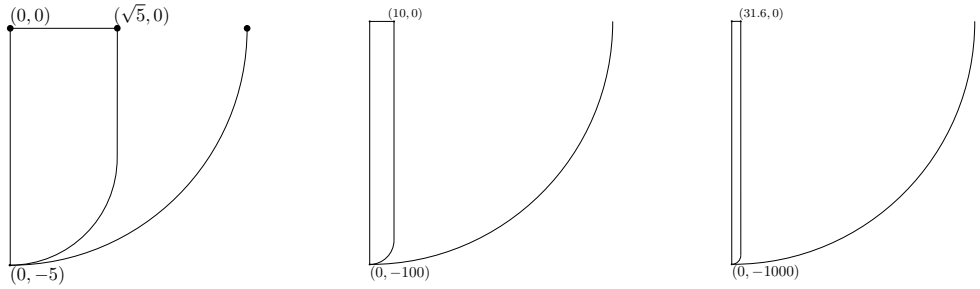


(a) $|l'| = 5$ (b) $|l'| = 100$ (c) $|l'| = 1000$

Figure 43: Illustration of increasing values for the vertical segment result in very steep ellipse arcs

4.3.3 Combination of circular arcs and vertical segments

Maybe a slightly different approach would be to set the width traveled by $\sqrt{l'}$, whereas l' describes the length of the horizontal segment. A combination of a circular arc with radius $\sqrt{l'}$ and a vertical segment of length $l' - \sqrt{l'}$ would illustrate the outgoing edge more precisely.



(a) $|l'| = 5$ (b) $|l'| = 100$ (c) $|l'| = 1000$

Figure 44: Illustration of increasing values for the vertical segment with the combination of a circular arc and a vertical segment

On one hand, we would still take $\mathcal{O}(n \cdot \sqrt{n}) \times \mathcal{O}(n)$ area in the worst case but on the other hand the complexity in this altered smooth orthogonal layout would significantly increase.

5 Extensional Work

5.1 $\mathcal{NP}$ Hardness

Bekos et al. proposed a reduction from the $\mathcal{NP}$-hard problem `3-CNF-SAT` to a bendless SMOG of maximum degree four by constructing gadgets, which encode a given propositional formula φ into an information "flow" through the gadgets. φ is satisfiable if and only if the respective bendless gadget construction preserves planarity.
We will extend the proof to graphs with arbitrary degree by creating a tunnel for every edge where all possible realizations with minimum edge complexity are drawn inside.

Assumption 1. *Based on the unit length $l(u)$ between two vertices on the coarse grid, we assume that the size of the vertex boxes are at most $\frac{1}{2}l(u) \times \frac{1}{2}l(u)$.*

The box size approximation ensure that a straight-line edge connecting two neighboured vertices on the coarse grid can be drawn. The boxes reach up to $\frac{1}{4}l(u)$ into the edge, ensuring a drawn length of the edge by length minimal $\frac{1}{2}l(u)$. As we already know, the maximal degree m of the graph G yields the granularity of the fine grid causing possible m edges per port regarding a vertex.
Regarding the proof for the $\mathcal{NP}$-hardness of the bendless problem, the main difference between the 4-planar case and the m-planar case is the port position of the edge. In the 4-planar case, the position of the straight-line / quarter arc edges is fixed on the center of the port. However, in the m-planar case, there are m possible positions on a single port. The main similarity between both cases is the position of the vertices. Because the vertices lie on the coarse grid, the difference in x-direction equals the difference in y-direction. Furthermore, the m possible quarter arcs inherit the same center of the circles. As depicted in Figure 45, there are multiple quarter arc connections possible

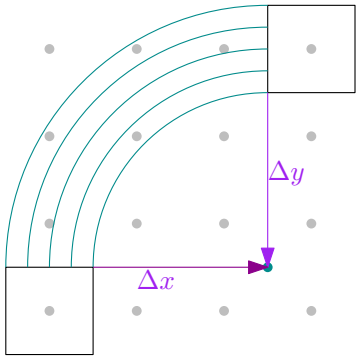


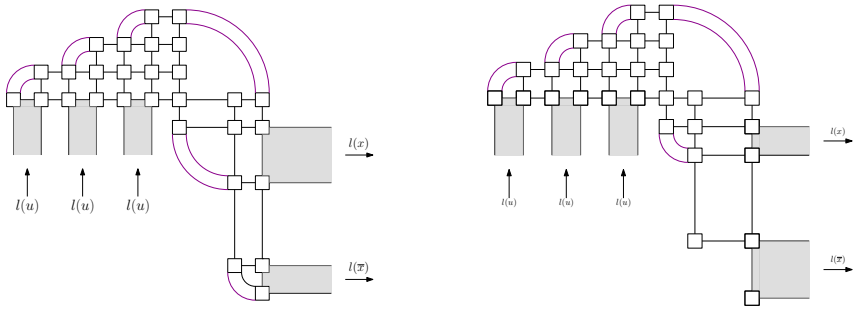
Figure 45: Multiple possible bendless connections

from one vertex to another. They all share the same center (the cyan dot). This also holds for straight line segments between two vertices. This leads to a gauge of the gadgets introduced in the $\mathcal{NP}$-hardness proof of Theorem 6. We will only illustrate the tunnel for the circular arcs for clear visualization. The straight edges between two vertices are interpreted as tunnels.

Theorem 11. *Given a planar orthogonal drawing of a graph G of arbitrary degree and a SMOG representation $\mathcal{R}$, it is $\mathcal{NP}$-hard to decide if it can be smoothened without any bends, preserving $\mathcal{R}$. The $\mathcal{NP}$-hardness still holds even if $\mathcal{R}$ requires all edges to be drawn as straight-line segments or quarter circular arcs. This is a slight modification of the gadget construction by Theorem 6.*

Proof. asically, for the $\mathcal{NP}$-hardness, the reduction from $\ni -\mathcal{CNF}-\mathcal{SAT}$ is very similar to the construction of $\mathcal{R}_\varphi$ of Theorem 6. However, the gadgets vary.

- *Variable gadget*
  The variable gadgets inherit the multiple arcs from Figure 45 illustrated by the

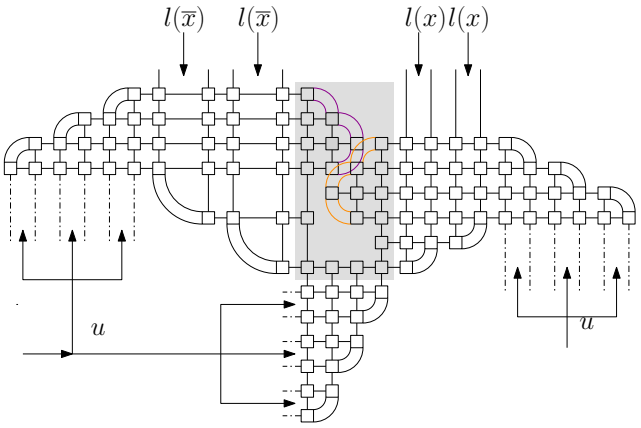


(a) `true` assignment (b) `false` assignment

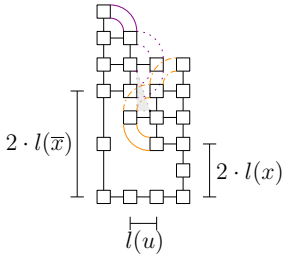
Figure 46: new variable gadgets

magenta edges. The maximum radius still values $l(u)$ like in the 4-planar case, but the center is shifted to the regarding corner of the vertex box. This is not a problem for the variable gadgets, however the offset of the center implies possible crossings in the parity gadget.

- *Parity gadget*
  Looking at the parity gadgets, one can see the possible crossings given by the new possible position of the edges.



(a) `false` assignment



(b) In detail

Figure 47: Parity gadget with possible collision

We know, that the maximum radius is still $l(u)$, the offset shortens the height and width of the triangle given in 47b. The diagonal has to be at least $2 \cdot l(u)$. Like in the 4-planar case (see Figure 14), the triangle delivers a `true` or `false` assignment. Due to Assumption 1 and Figure 45 the center of those quarter circular arcs differ in height and width by $\frac{3}{2}\lambda$ and $\frac{1}{2}l(u)$ respectively, which imply the size of the triangle. The outermost edges regarding a port are assumed for the following calculation.

$$\begin{aligned} 2 \cdot l(u) &< \sqrt{\frac{9}{4}\lambda^2 + \frac{1}{4}l(u)^2} \\ \Leftrightarrow 4 \cdot l(u)^2 &< \frac{9}{4}\lambda^2 + \frac{1}{4}l(u)^2 \\ \Leftrightarrow \frac{15}{4}l(u)^2 &< \frac{9}{4}\lambda^2 \\ \Leftrightarrow \frac{5}{3}l(u)^2 &< \lambda^2 \\ \Leftrightarrow \lambda &> \sqrt{\frac{5}{3}}l(u) \approx 1,291 \cdot l(u) \end{aligned}$$

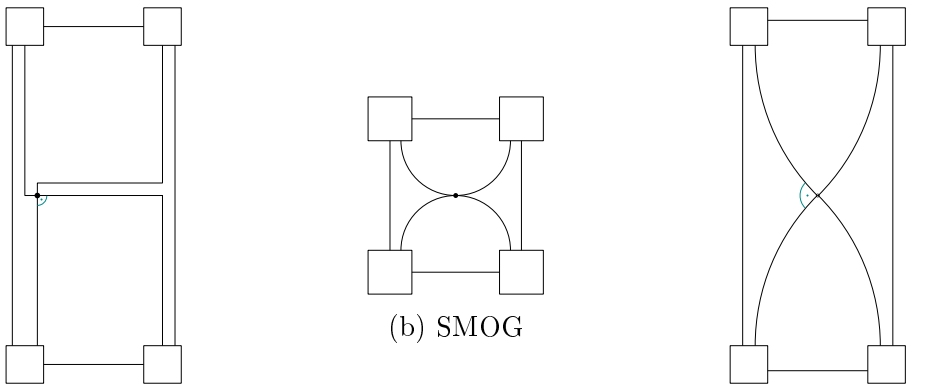
So, in order to avoid crossings, following property must hold for $l(x), l(\overline{x})$:

$$l(x), l(\overline{x}) \in (0, 0.8545 \cdot l(u)) \cup (2.1455 \cdot l(u), 3)$$

The other gadgets (clause gadget, auxiliary gadgets) stay unaltered. The reduction from a given 3-$\mathcal{SAT}$ formula in CNF to a drawing Γ_φ is given and can be calculated in polynomial time. As in Theorem 6, if φ is satisfiable, the clauses can still be implemented and connected in bendless gadgets. Similarily, if G_φ is bendless, there is a `true` assigment for at least one of the literals in a clause. Therefore, φ is satisfiable and this completes the proof. □

5.2 Octi Arcs

Given a non-planar drawing, it is desirable that the eye of the viewer can easily distinguish crossings from vertices. Further, the crossings shall be illustrated in a way that the ongoing direction of the respective edge is clear. We want a degree constraint of 90° on the crossings, which illustrate the crossing and the direction of the edges very accurately.



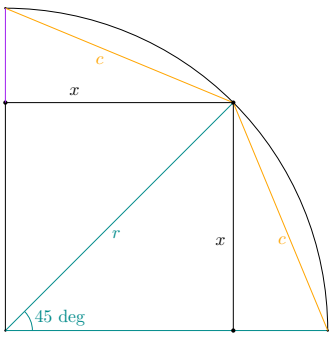
(a) Orthogonal (b) SMOG (c) Octi arc solution

Figure 48: Various illustrations of the hourglass drawing

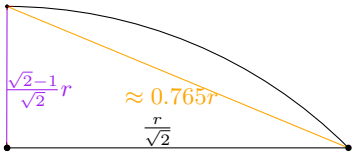
The orthogonal drawing illustrated in Figure 48a inherits Kandinsky bends resulting in an edge complexity of 3. The degree of the graph values 3, so there are three connections per port possible (granularity of the fine grid). Therefore, this drawing is consistent with the Kandinsky model.
The SMOG representation in Figure 48b inherits a decrease in the edge complexity by 1 but the crossing is not illustrated in a visibly clear way. Introducing eighth circular arcs, or *octi arcs*, we could find a compromise between the edge complexity and the visible clear crossing (Figure 48c). Note that orthogonality of the crossing is illustrated with a 45° turn compared to the crossing of Figure 48a.

5.2.1 Examining the octi arcs

Using the octi arcs seems to be rather flexible. An octi arc implies a 45° turn in the drawing which is crucial for a smooth representation of diagonally shaped crossings in a drawing.



(a) Construction



(b) In Detail

Figure 49: Examination of octi arcs

Due to trigonometry, the length of several edges can be calculated in dependence of r:

$$2x^2 = r^2 \tag{8}$$

$$\Leftrightarrow x = \frac{r}{\sqrt{2}} \tag{9}$$

Furthermore, for the distance between the endpoints of an octi arc:

$$c^2 = (r-x)^2 + x^2 \tag{10}$$

$$= r^2 - 2xr + 2x^2 \tag{11}$$

$$= r^2 - 2\frac{r}{\sqrt{2}} \cdot r + r^2 \tag{12}$$

$$= 2r^2 - \sqrt{2}r^2 \tag{13}$$

$$\Leftrightarrow c = \left(\sqrt{2-\sqrt{2}}\right) r \tag{14}$$

$$\approx 0.7654 \cdot r \tag{15}$$

5.2.2 Saving Space

In the previous section, we saw some approaches to effectively save space, either by postprocessing a SMOG or by substituting circular arcs with an adequate alternative. In

this section, we will examine whether octi arcs are suitable as a substitution alternative. The smooth 90° bend is achieved by using two octi arcs with different radii and a diagonal segment for flexibility. As we already saw, in order to reach a height of the radius r, an octi arc without any further ado reaches a width of $(\sqrt{2}-1)r$. So the sole octi arc is still taking $\mathcal{O}(r)$ width. What will happen, if we add another octi arc with a diagonal segment in order to maintain a smooth drawing? The question arises whether we were able to save space. The answer will be - a little bit, but not quite enough in the long run.

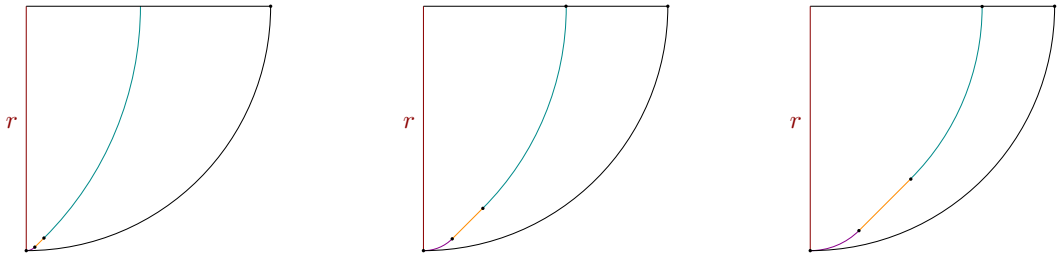


Figure 50: Various ratios of the combination of octi arcs and diagonal segments

Introducting a small octi arc to begin with seems to be consistent with the idea of smooth drawings as the resulting bend will be 45° on both sides, resulting in differentiable curves. However, we will see that the width of the suggested combination of segments gets greater as the small first octi arc gets introduced. One one hand, we will save width because the second, big octi arc has less height to overcome, but on the other hand we gain width with the small octi arc getting wider. Let r be the original height to manage, r' the height, the octi arc has to manage together with a small octi arc at the beginning.

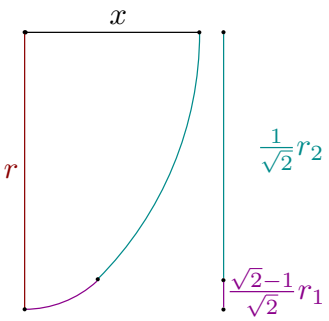


Figure 51: Illustration of the combination of octi arcs

Without loss of generality, we will examine the properties without a diagonal segment. In figure 51, r describes the height and the original radius of the circular ars. x describes the width gained by two octi arcs.

$$r = \frac{\sqrt{2}-1}{\sqrt{2}}r_1 + \frac{1}{\sqrt{2}}r_2 \qquad r_1, r_2 \in (0, r) \tag{16}$$

$$x = \frac{1}{\sqrt{2}}r_1 + \frac{\sqrt{2}-1}{\sqrt{2}}r_2 \tag{17}$$

From equation 16 we get the following dependency, which is used for equation 17:

$$\begin{aligned} & & r_1 &= \sqrt{2}r - \left(\sqrt{2}-1\right) r_2 & \\ \Rightarrow & & x &= \left(\sqrt{2}-1\right) r + 2r_2 & \in \Theta(r) \end{aligned}$$

Actually we could save some space. With the following radii for the octi arcs we can save half the length of the original radius.

$$r_1 := \frac{\sqrt{2}+1}{6} r, r_2 := \frac{1}{6} r \qquad \Rightarrow x = \frac{1}{2} r \tag{18}$$

Unforunately, substituting the circular arcs with the mentioned combination in the orthogonal drawing will still take $\mathcal{O}(n^2) \times \mathcal{O}(n)$ area in the worst case.

6 An Example

Consider the following input graph drawing Γ_G:

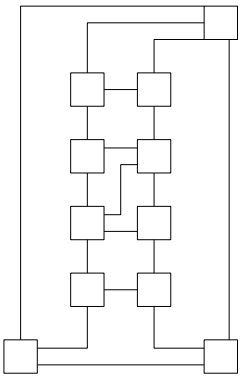


Figure 52: Example input drawing Γ_G

For measurement, the unit of length lies in the width of the illustrated boxes. The longest vertical segment lies on the left of the drawing and is of length 10. The complexity of the drawing equals three. The drawing is of size 7×11.

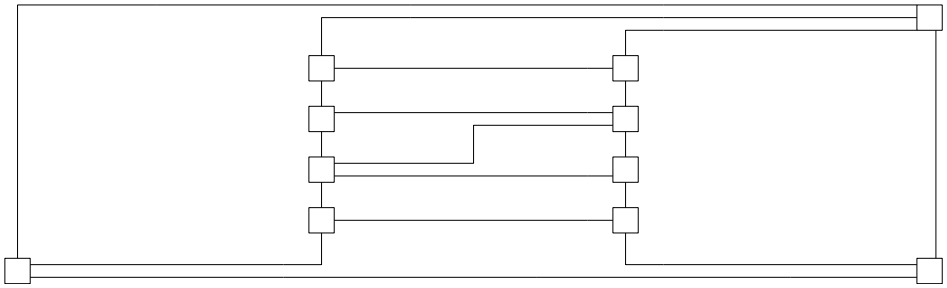


Figure 53: Γ_G after the stretching technique application

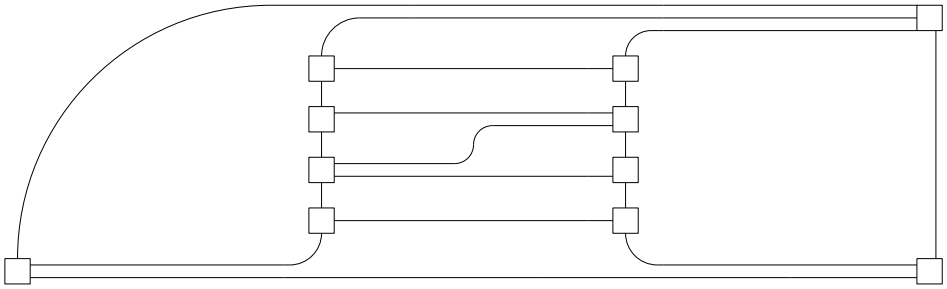


Figure 54: Γ_G after the circular arc substitution

After stretching and smoothening the complexity of the resulting SMOG drawing rises from 3 to $\left\lfloor \frac{3}{2} \cdot 3 \right\rfloor = 4$. The size of the drawing equals 37×11. The drawing inherits some redundancy which we can get rid of thanks to the saving measures.

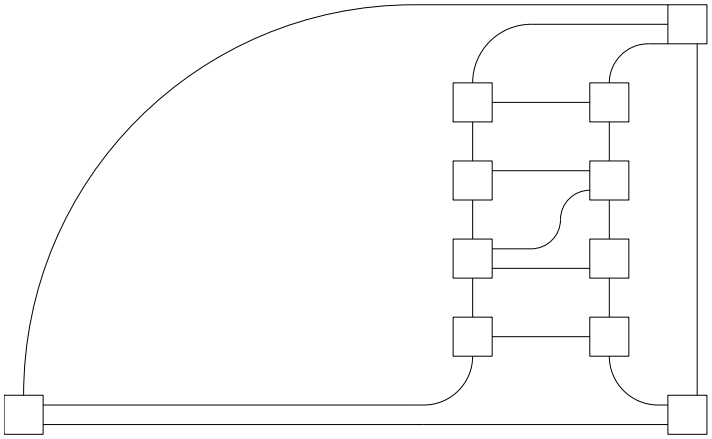


Figure 55: Γ_G after the saving plane sweep application

The drawing lost 51,4% in horizontal area requirements resulting in a drawing with 18×11 area bounds. Also, the complexity decreased from 4 to 3.

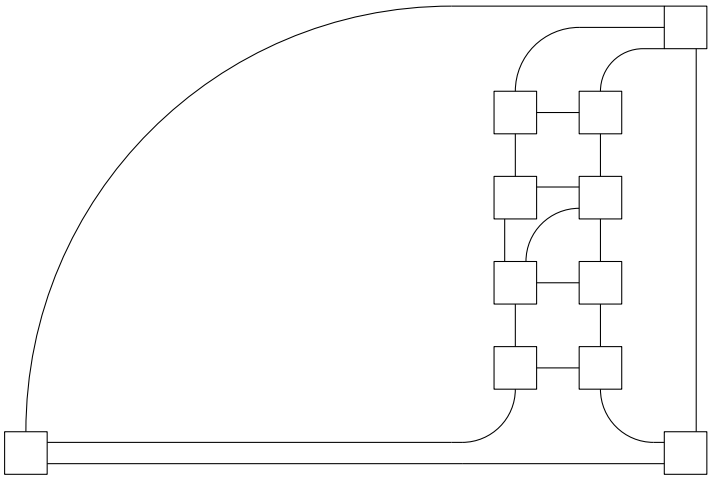


Figure 56: Γ_G after the port reassignment

The port reassignment results in a smoothened drawing of complexity 2 and the area consumption can further be reduced, resulting in a drawing of size 16×11.

7 Future Work

Further use of the fragmentation

The main purpose of the fragmentation is having a subdivision of complex polyedges. In the orthogonal case, we are able to distinguish uniform parts from alternating parts in a large polyedge. It is possible to extend the work of field with similar results to other classes of drawings.

Implementation

Now that we achieved the guarantee of Kandinsky drawings being able to be postprocessed to a SMOG, an *implementation* - for example with the `yFiles`-bibliography in *Java* - would be of interest. If we got the implementation, then we would be able to examine a set of orthogonal drawings and their smoothened results - the appearance of Kandinsky drawings and SMOGs. Further, we would be able to make a statement whether an area saving measure like the circular arc substitution would result in visibly clear drawings.

Graphs with crossings

The results in this work consider only graphs with planar drawings. It would be desirable to consider graph with *non-planar drawings*. The illustration of crossings shall be visibly distinguishable from vertices and polyedges. Our first approach of introducing octi arcs did result in a non-consistent model since octi arcs with a 45° bend introduce new slopes.

Further saving approaches

Certain properties of a polyedge fragmentation, circular arc substitution, the saving plane sweep and the port reassignment enables us to save some *bends* and *area consumption* of the drawing. Regarding the port reassignment, we considered alternating polyedges of complexity three. Naturally, it would be desireable to consider the port reassignment in a more general case. Also a point of interest would be saving measures by cutting horizontally and vertically. So the choice of cut alignment may result in an even better drawing. Cutting through a circular arc could mean a substitution with a smaller circular arc and a line segment accordingly.

8 Acknowledgements

I would like to thank Prof. Michael Kaufmann and Henry Förster for instructing this final thesis. Further I appreciate helpful discussions with Dr. Michalis Bekos. I want to thank especially my parents for their patience and encouragement despite hardest family issues. A special thanks goes to Thomas Stüber, who made me appreciate the theoretical field of computer science.

References

[1] Md. Jawaherul Alam, Michael B. Dillencourt, and Michael T. Goodrich. "Capturing Lombardi Flow in Orthogonal Drawings by Minimizing the Number of Segments". In: *CoRR* abs/1608.03943 (2016). arXiv: 1608.03943. URL: http://arxiv.org/abs/1608.03943.

[2] Michael A. Bekos, Henry Förster, and Michael Kaufmann. "On Smooth Orthogonal and Octilinear Drawings: Relations, Complexity and Kandinsky Drawings". In: *Graph Drawing and Network Visualization - 25th International Symposium, GD 2017, Boston, MA, USA, September 25-27, 2017, Revised Selected Papers*. 2017, pp. 169–183. DOI: 10.1007/978-3-319-73915-1_15. URL: https://doi.org/10.1007/978-3-319-73915-1_15.

[3] Michael A. Bekos et al. "Smooth Orthogonal Layouts". In: *J. Graph Algorithms Appl.* 17.5 (2013), pp. 575–595. DOI: 10.7155/jgaa.00305. URL: https://doi.org/10.7155/jgaa.00305.

[4] Michael A. Bekos et al. "The Effect of Almost-Empty Faces on Planar Kandinsky Drawings". In: *Experimental Algorithms - 14th International Symposium, SEA 2015, Paris, France, June 29 - July 1, 2015, Proceedings*. 2015, pp. 352–364. DOI: 10.1007/978-3-319-20086-6_27. URL: https://doi.org/10.1007/978-3-319-20086-6_27.

[5] Christian A. Duncan and Michael T. Goodrich. "Planar Orthogonal and Polyline Drawing Algorithms". In: *Handbook on Graph Drawing and Visualization*. 2013, pp. 223–246.

[6] Christian A. Duncan et al. "Lombardi Drawings of Graphs". In: *J. Graph Algorithms Appl.* 16.1 (2012), pp. 85–108. URL: http://jgaa.info/accepted/2012/Duncan+2012.16.1.pdf.

[7] David Eppstein. "Planar Lombardi Drawings for Subcubic Graphs". In: *Graph Drawing - 20th International Symposium, GD 2012, Redmond, WA, USA, September 19-21, 2012, Revised Selected Papers*. 2012, pp. 126–137. DOI: 10.1007/978-3-642-36763-2_12. URL: https://doi.org/10.1007/978-3-642-36763-2_12.

[8] Ulrich Fößmeier, Carsten Heß, and Michael Kaufmann. "On Improving Orthogonal Drawings: The 4M-Algorithm". In: *Graph Drawing*. Ed. by Sue H. Whitesides. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 125–137. ISBN: 978-3-540-37623-1.

[9] Ulrich Fößmeier and Michael Kaufmann. "Drawing High Degree Graphs with Low Bend Numbers". In: *Graph Drawing, Symposium on Graph Drawing, GD '95, Passau, Germany, September 20-22, 1995, Proceedings*. 1995, pp. 254–266. DOI: 10.1007/BFb0021809. URL: https://doi.org/10.1007/BFb0021809.

[10] Seok-Hee Hong, Damian Merrick, and Hugo A. D. do Nascimento. "Automatic visualisation of metro maps". In: *J. Vis. Lang. Comput.* 17.3 (2006), pp. 203–224. DOI: 10.1016/j.jvlc.2005.09.001. URL: https://doi.org/10.1016/j.jvlc.2005.09.001.

[11] Mark Lombardi. *Work of Mark Lombardi*. June 2018. URL: https://i1.wp.com/www.pierogi2000.com/wp/wp-content/uploads/LombardiGeorgeHarkenEnergy.jpg.

[12] R. Tamassia. “On Embedding a Graph in the Grid with the Minimum Number of Bends”. In: *SIAM Journal on Computing* 16.3 (1987), pp. 421–444. DOI: 10.1137/0216030. eprint: https://doi.org/10.1137/0216030. URL: https://doi.org/10.1137/0216030.