EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Karl Eberhard Universität Tübingen
Wilhelm Schickard Institut Tübingen

Fakultät für Informatik

Smoothening of Kandinsky Drawings

Arbeitsbereich Algorithmik

zur Erlangung des akademischen Grades
Bachelor of Science

Autor:	Benjamin Çoban MatNr. 3526251
Version vom:	30. Juni 2018
1. Betreuer:	Prof. Dr. Kaufmann
2. Betreuer:	Henry Förster

Contents

1 Introduction 4

2 Abstract 6

3 Preliminaries 7
- 3.1 Definitions 7
- 3.2 Plane Sweep Algorithm 8
- 3.3 Topology Shape Metrics 8
- 3.4 Previous results 9
  - 3.4.1 Fixed Layout Model 10
  - 3.4.2 Fixed Shape Model 10
- 3.5 Drawings with low complexity 12
- 3.6 $\mathcal{NP}$-hardness 13
  - 3.6.1 Correctness 14

4 The Kandinsky Model 15
- 4.1 Introduction 15
- 4.2 Area Investigation 15
  - 4.2.1 Between two segments - “Boxing“ 15
  - 4.2.2 Connected to a vertex 16
- 4.3 Edge Complexity Investigation 18
  - 4.3.1 Examining “good“ and “bad“ parts of a polyline 18
  - 4.3.2 Correctness of algorithm 1 20
  - 4.3.3 Correctness of algorithm 2 22
  - 4.3.4 Correctness of algorithm 4 24
- 4.4 Results 26

5 Saving measures 29
- 5.1 Introduction 29
- 5.2 Edge Complexity Bounds 29
  - 5.2.1 Podevsaef drawings 30
  - 5.2.2 Using the fragmentation 31
- 5.3 Area Bounds 32
  - 5.3.1 Plane sweep erasing 32
  - 5.3.2 Circular arc substitution 33
  - 5.3.3 Ellipses 34
  - 5.3.4 Readability 34
  - 5.3.5 Combination of circular arcs and vertical segments 35

6 Related Work 36
- 6.1 $\mathcal{NP}$ Hardness 36
- 6.2 Octi Arcs 38
  - 6.2.1 Examining the octi arcs 39
  - 6.2.2 Saving Space 39

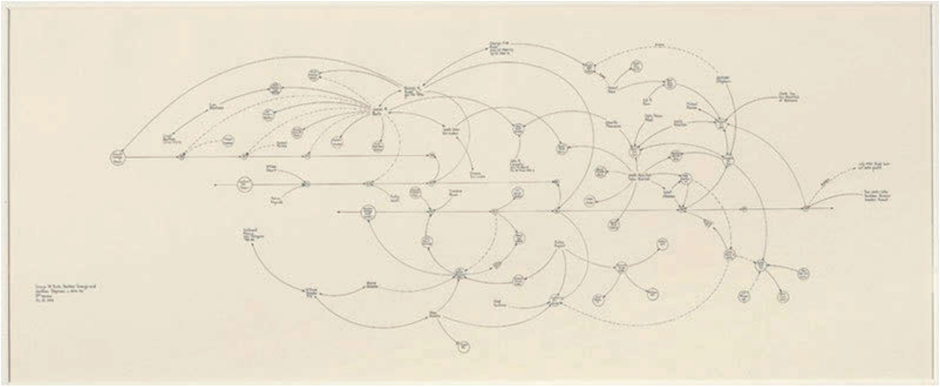
7 An Example 42

8 Future Work 44

9 Acknowledgements 45

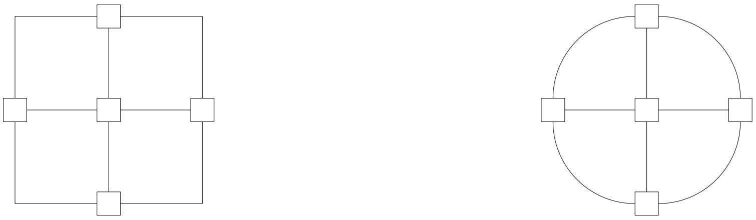
1 Introduction

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.



Work of Mark Lombardi [9]

Orthogonal drawings arise among others in VLSI design where quite many cables are following a similar path. The smallest angle between consecutive line segments is at most $\pi/2$ and their angular resolution is quite pleasing for the eye of the viewer. As a fundamental, reliable model lies 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*[6][7].



Orthogonal drawing

Smoothened drawing

Smoothening a drawing for aesthetic appeal

By postprocessing an input drawing, 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.[8]
However, it is a priori not guaranteed that there is a smoothened drawing for every input Kandinsky drawing. The introduction of circular arcs might rise 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 the quantify the resulting quality of the smoothened drawing.

2 Abstract

Firstly, we show that every Kandinsky drawing of arbitrary degree can be postprocessed to a smooth orthogonal drawing and planarity is preserved. As a base for postprocessing a Kandinsky drawing, we use the *Fixed Shape Model*, which preserves the orientation of the vertices. The area bounds of SMOGs are $\mathcal{O}(n^2) \times \mathcal{O}(n)$ in the worst case. The complexity of a polyedge *does not increase* if and only if the polyedge is *purely uniform* and rise to $\left\lfloor \frac{3}{2}k \right\rfloor$ if and only the the polyedge is *purely alternating*. In practice, it is possible that a polyedge contains several uniform and alternating parts. The *fragmentation* of a polyedge delivers a mathematical backbone in order to examine all kinds of situations. With help of the optimal fragmentation, we can guarantee that a rather *high complexity* of a polyedge in the original Kandinsky drawing *only increases to* $k+2$.

In the next section we figure out several approaches in order to *decrease* the complexity of edges and the area bounds of the smooth orthogonal drawings. A *modified plane sweep* can save up to $\mathcal{O}(n)$ area and is able to save some horizontal segments. This plane sweep suits as a handy approach to optimize given SMOG drawings. Further we show an alternative to circular arcs - a *combination* of a quarter arc and a vertical segment increases the complexity, but only needs $\sqrt{r}$ width relative to the original quarter arc.

Finally, we show a *gadget construction* as a *reduction* from the $\mathcal{NP}$-hard *SAT problem* to a bendless SMOG with arbitrary degree. *Octi arcs* may be relevant for further work - e.g. illustration of graphs with at most one crossing per edge - and are examined for its properties.

3 Preliminaries

3.1 Definitions

As otherwise mentioned, a *graph* $G = (V_G, E_G \subset V_G \times V_G)$ is a tuple consisting of two sets - the set of vertices and the set of edges. An *edge* $e = (v, w), v, w \in V$ 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. A face is *empty* if there exist two edges bounding this face with $0°$ bend difference in the drawing. 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. *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 4 (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 5 (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, reminding 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 6 (Kandinsky, [5, p. 225]). *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 centralized 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.*

Definition 7 (Podevsaef drawings, [4, p. 2])**.** *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°.*

3.2 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 an 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.

3.3 Topology Shape Metrics

The progress from a graph as a mathematical tuple to a drawing requires an appropiate embedding in order to guarantee special properties such as planarity or a statement regarding the number of bends. iia et al. presented an algorithm resulting in orthogonal drawings with a minimal number of bends for a given embedding. Surprisingly, computating the minimal number of bends of an orthogonal graph embedding is in P. The whole process is called *Topology Shape Metrics*. 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, the topology of an input graph is set by computing the maximal planar subgraph. If the input graph is not planar at all, then the solution inherits dummy vertices to planarize the graph. Those dummy vertices are removed again in the ongoing process.

- *Shape phase*
  This phase is also called the *orthogonalization phase.* The bends and its respective degrees of all the edges are computated along with the position of 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 is computed. The goal is to minimize the area the graph demands.[10]

Every phase can be utilized for special needs. As already mentioned, the Topology Shape Metrics approach serves as a standard practice in research due to its flexibility.[4, p. 3]

3.4 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 8 (Smooth Orthogonal Layout, [3, p. 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;*

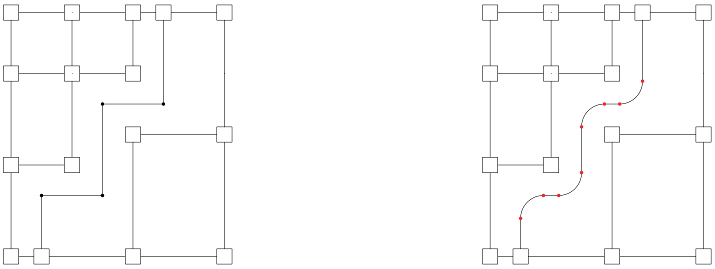
- *a port of a vertex is incident to at most one edge.*

A smooth orthogonal layout inherits a so-called *edge complexity* $k \geq 1$, if there does not exist any layout where the edge has 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 SMOG 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 side - one each for North, South, West, East - and are illustrated as a point in the plane.

3.4.1 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 4]). *In the Fixed Layout Model, the edge complexity of a given orthogonal graph G might increase from k to $2k - 1$.*



(a) Staircase in ortho (b) SMOG representation

Figure 3: The complexity increase of 4-planar graphs with staircases

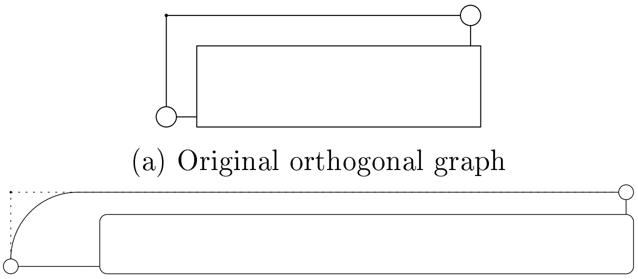
The reason for the edge complexity increase is the fixation of the vertices and following example regarding so-called "Staircase Edges“. It is already clear, that the Fixed Layout Model is rather too restrictive to process orthogonal graphs to SMOGs. A different approach is the Fixed Shape Model, where the orthogonal representation is preserved (the circular ordering of the polyedges around a vertex).

3.4.2 Fixed Shape Model

In the Fixed Shape Model, the orientation of the vertices (North, East, West, South) and the embedding (circular ordering of the edges connected to a vertex) is preserved. The vertices are of uniform size but can be repositioned on the course integer grid.

Definition 9 (Stretching technique, [3, p. 582]). *The stretching technique is a process where every horizontal edge of a given orthogonal drawing is lengthened by adding the length of the longest vertical segment l of the drawing. The result is an ortogonal 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 regarding 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 results in a stretching of every grid cell. The second plane sweep erases 90 degree bends by circular arc substitution, 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 following example:



(a) Original orthogonal graph

(b) After applying the stretching technique

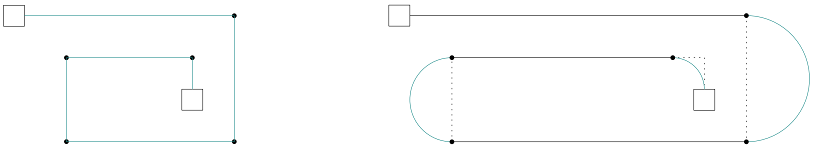
Figure 4: Worst case illustration of the stretching technique

The box in Figure 4 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; 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 is not altered. 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.

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 $\lfloor \frac{3}{2}k \rfloor$.



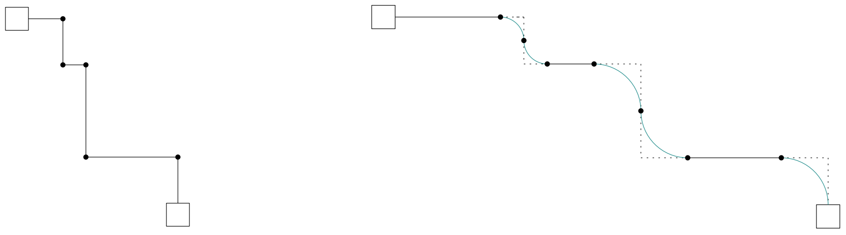
(a) Uniform polyedge with 5 bends (b) No complexity increase in SMOG

Figure 5: The uniform case - No 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



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

Figure 6: The alternating case - Complexity increase under $\mathcal{O}(n^2) \times \mathcal{O}(n)$ area

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 et al. 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. □

3.5 Drawings with low complexity

Definition 10. *There are so-called “Kandinsky bends“ which are not further reducable although the polyedge is zig-zag shaped.*

The persistance of these 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. There are up to two Kandinsky bends possible per edge, one for each vertex. In Figure 7, we see an orthogonal drawing of a 4-planar graph with a minimal total number of bends. Any shift of the vertices would result in a greater number of bends. The Kandinsky bends with its regarding vertices are coloured accordingly. Notice that the cyan coloured polyedge is zig-zag shaped.

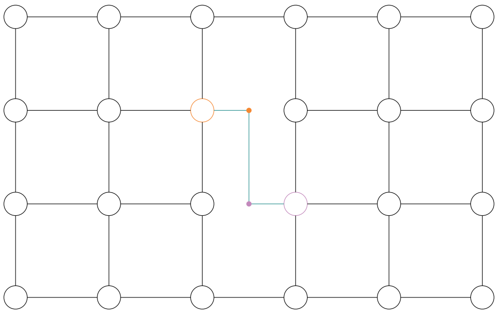


Figure 7: A planar drawing with a minimal number of bends

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. □

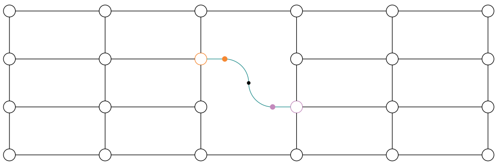


Figure 8: Smooth orthogonal layout of figure 7

In Figure 8, the drawing of figure 7 got stretched and circular arcs were substituted, resulting in a complexity-4 smooth orthogonal layout.

3.6 $\mathcal{NP}$-hardness

It is assumed by Bekos et al. that it was $\mathcal{NP}$-hard to decide whether a 4-planar graph in Kandinsky 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 graphs).*

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.

- *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:

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

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

- *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.

- *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`.

- *Auxiliary gadget*
  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.

3.6.1 Correctness

Recall that this is only a sketch of the reduction. For the whole proof, see [2, p. 14]. 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, minimum one literal has to be `true`. Therefore φ is satisfiable. □

4 The Kandinsky Model

4.1 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 of 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 which result in the *bend or end* property of Kandinsky drawings. It is to examine if postprocessing Kandinsky drawings guarantee 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.

4.2 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.

4.2.1 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 “boxes“ 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.

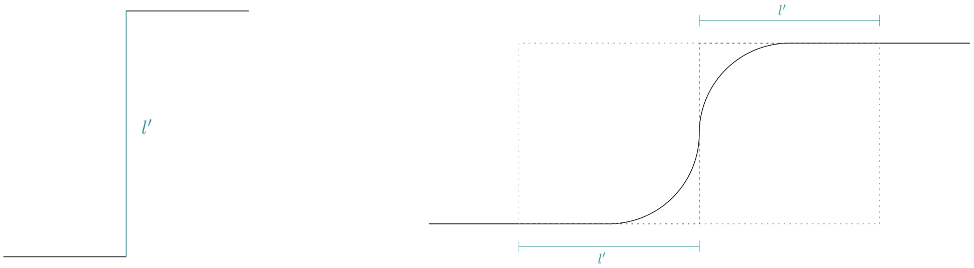
$$\delta(l', l'') = |l'_x - l''_x|$$

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 9: 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“. 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 $r = \frac{l'}{2}$ 0along the alternation. The space is already guaranteed, all left to demonstrate 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. □

4.2.2 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.

Definition 11. *The bend of 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.*

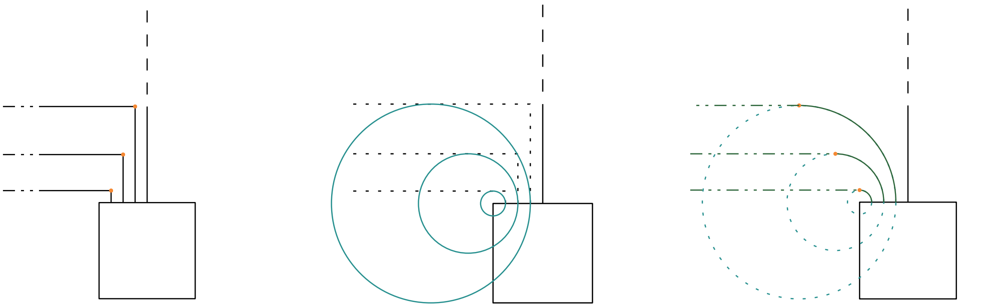
Lemma 3. *Let Γ_G be any orthogonal Kandinsky drawing and $v \in V_G$ a vertex with its ports. Then, there is at most one edge connected to each port of v with only one edge segment (complexity = 1). This describes the bend or end property of the polyedges regarding a single port.*

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 space 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.



(a) Kandinsky situation

(b)

(c) SMOG representation

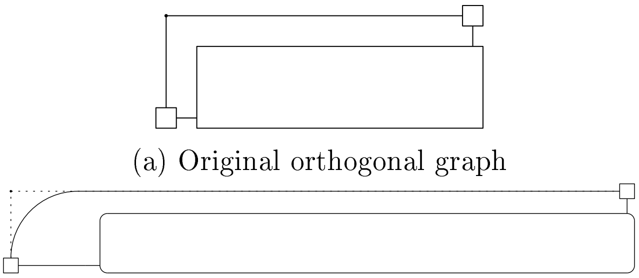
Figure 10: Smoothen multiple polyedges on the same port

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

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

Recall the worst case situation of Figure 4

□

4.3 Edge Complexity Investigation

As we saw in the previous section, the possibility of a SMOG representation is given in the k-planar situation. All that remains is the examination regarding the number of bends of a polyedge. Previous results deliver an upper bound of the 4-planar graphs by a cofactor of $\left\lfloor \frac{3}{2} \right\rfloor$ 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.

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

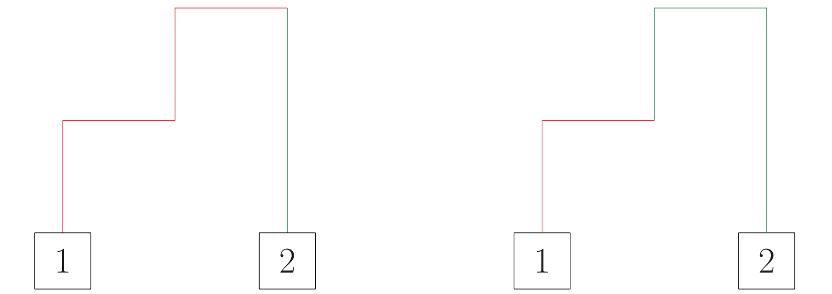
The main advantage of the fragmentation is that every fragment can be seen as an independent polyedge. The property of turns is crucial for the edge complexity thus the main criterion for the fragmentation.

Definition 13. *A fragment is pure in its turns; like in Definition 3, 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.*

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

Proof. Consider the polyedge $(1, 2)$ illustrated at Figure 12. The 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 12: Non-unique fragmentation

□

Definition 14. *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 14 is an equivalence relation.*

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

Definition 15. *Two fragments f and g are called incompatible if any segment transfer between f and g result in a turn property collision.*

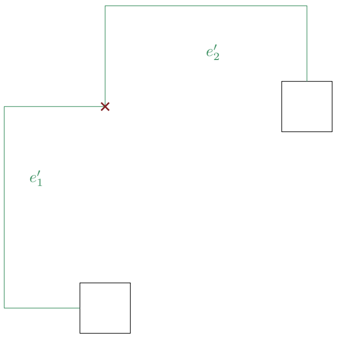


Figure 13: Incompatible uniform fragments e'_1, e'_2

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 small case still has to be considered.

Lemma 7. *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     $e'_1 \leftarrow (e_1, e_2)$
 3     $e'_1$.uniform = true
 4     $e' \leftarrow (e'_1)$
 5 else
 6     $e'_1 \leftarrow (e_1, e_2, e_3)$
 7     $e' \leftarrow \emptyset$
 8     $i \leftarrow 1$
 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             $i \leftarrow i + 1$
19             $e_i$.uniform $= \neg e_{i-1}$.uniform
20             $e_i \leftarrow (e_j)$
21 $e'$.append($e'_i$)
22 return $e'$
```

4.3.2 Correctness of algorithm 1

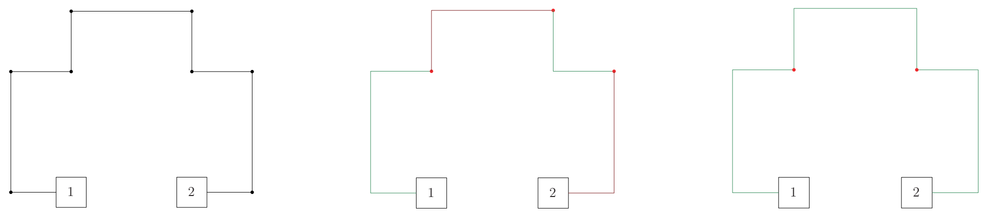
This algorithm works for polyedges with the length at least two. If the length is two, then the fragmentation will be unique since a fragment of length two is both uniform and alternating due to Lemma 7. If the complexity of the polyedge is greater than two, then lines 6 to 12 initialize the first fragment 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 8. *Algorithm 1 is not sufficient to determine a satisfying fragmentation.*

Proof. Consider following polyedge given in 14. 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.

As we can see in 14b, algorithm 1 delivers a fragmentation which consists of multiple



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

Figure 14: Bad example of a polyedge for the first approach

fragments of length two. That is because algorithm 1 forces two consecutive fragments to differ in their turn property. As we know from Lemma 7, fragments of length two are not that conclusive in this case. The result given in 14c 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. This leads to the following ordering relation:

Lemma 9. *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 illustrates 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 backbone 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.

Definition 16. *Let e', f' be two fragmentations of the same polyedge e ($e' \sim_R f'$). Then we define a relation:*

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

Lemma 10. *The relation from Defintion 16 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.
Using Lemma 7, we will be able to substitute alternating fragments as a sequence of uniform fragments.

Lemma 11. *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 definition 16. This enables us to distinguish uniform fragments from alternating ones in the output of algorithm 2.*

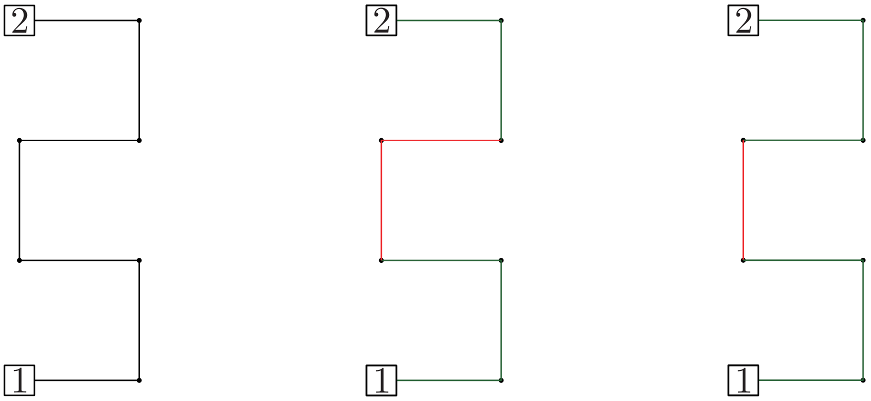
Proof. If a fragment f of length k' is alternating, then every bend differs in its direction regarding the previous one. A fragment of length 2 inherits one bend. In an alternating fragment, the next one will cause a turn property collision, resulting in a new fragment. Again, fragments of lengths up 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 $e'_1 \leftarrow (e_1, e_2)$
 2 $f \leftarrow \emptyset$
 3 $i \leftarrow 1$
 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         $i \leftarrow i + 1$
10         $e'_i \leftarrow (e_j)$
11 $f$.append($e'_i$)
12 Recheck($f$)
13 return $f$
```

4.3.3 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 15: Bad example of a polyedge for the second uniform-only approach (red highlights the problematic fragment)

In figure 15 the output of algorithm 2 returns uniform-only fragments. The colour highlighting is used this time to illustrate the problem description. In Figure 15b, we see that the length of the fragments equals $(3, 2, 2)$, whereas in Figure 15c 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: $\texttt{recheck}(f) \in \mathcal{O}(k)$

```
Input: A fragmentation $f$
Output: A fragmentation $f$ rechecked
for $i = 1$ to $f.\mathit{length}-1$ do
    if $f_i.\mathit{length} = 2$ then
        $v \leftarrow$ first segment of $f_i$
        $u \leftarrow$ second segment of $f_i$
        if $(u).\mathit{append}(f_{i+1}).\mathit{uniform} = \mathit{true}$ then
            $f_i \leftarrow (v)$
            $f_{i+1} \leftarrow (u).\texttt{append}(f_{i+1})$
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 16

```
   Input: Fragmentation $f$ computed by algorithm 2 and 3
   Output: Minimal fragmentation $e_{\min}$ regarding the relation 16
1  $e_{\min} \leftarrow \emptyset$
2  $f_{\text{alt}} \leftarrow \emptyset$
3  $f_{\text{alt}}$.uniform = false
4  for $i = 1$ to $f$.length do
5      if $f_i$.length $\geq 3$ then
6          if $f_{alt} \neq \emptyset$ then
7              $e_{\min}$.append($f_{\text{alt}}$)
8              $f_{\text{alt}} \leftarrow \emptyset$
9          $f_i$.uniform = true
10         $e_{\min}$.append($f_i$)
11     else
12         $f_{\text{alt}}$.append($f_i$)
13 if $f_{alt} \neq \emptyset$ then
14     $e_{\min}$.append($f_{\text{alt}}$)
15     $f_{\text{alt}} \leftarrow \emptyset$
16 return $e_{\min}$
```

4.3.4 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 11, 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.

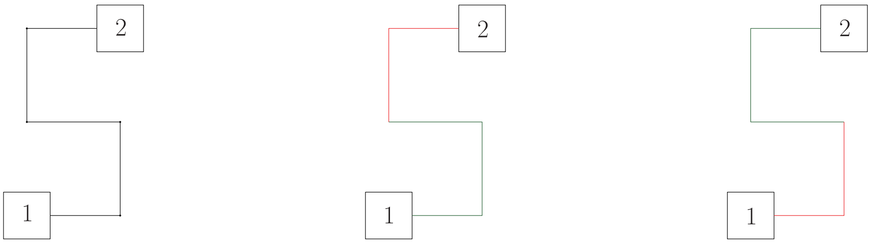
Theorem 8. *The fragmentation resulting from algorithm 2 and 4 describes a minimal fragmentation regarding the relation 16.*

Proof by contradiction. Let $e_{\min}$ be the result of algorithms 2 and 4 and $f_{\min}$ be the another valid fragmentation regarding the relation 16. 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 values for the case that $f_{\min}$ would have less alternating fragments. So $f_{\min} \not< e_{\min}$. □

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 12. *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 16: Non-unique optimal fragmentation regarding direction

The fragmentation from 1 to 2 starts with an uniform fragment of length three, which is not contained in the fragmentation the other way around. □

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 13. *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 16, 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 $\left\lfloor \frac{3}{2}k' \right\rfloor$. In fact, we will also see that every fragmentation with complexity k will result in a SMOG polyedge with complexity at most $\left\lfloor \frac{3}{2}k \right\rfloor$.

4.4 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 14. *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 15. *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 13, the situation arises that one converse bend relative to the others leads to incompatible fragments.

Lemma 16. *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 17. *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.

<u>k even</u>

If k is even, then there exists a $n \in \mathbb{N}$ with $k = 2n$. It follows:

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

<u>k odd</u>

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

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

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, thus e it is not purely alternating, by Theorem 7 at least one of the fragments is of length three or greater.

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

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 16. 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 15 takes effect and the Theorem is true. If the polyedge is fragmentated non-trivially, we examine the composition with algorithm 2 and Lemma 17 takes effect that the number of bends are bound by the worst case staircase situation. In the previous section, we saw that the space 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 18. *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. □

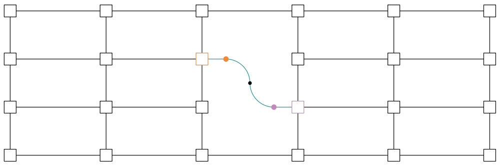
5 Saving measures

5.1 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 of $k+2$ 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.

5.2 Edge Complexity Bounds

Reconsider the drawing given in Figure 8. Due to the fact that the previous results considering 4-planar graphs are also holding for graphs of arbitrary degree, the results are applicable.



Recall Figure 8

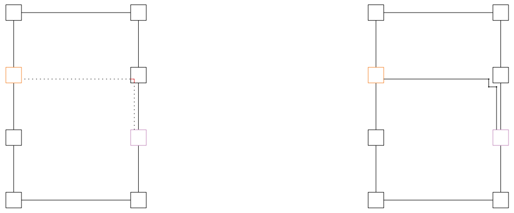
Theorem 10. *Every Kandinsky drawing of a complexity-3 graph with arbitrary degree can be illustrated as a complexity-4 smooth orthogonal layout in $\mathcal{O}(n^2) \times \mathcal{O}(n)$ area.*

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 8, 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 18a. By doing so, we would further increase the complexity of the edge.



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

Figure 18: Bend or end property complicates matters

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

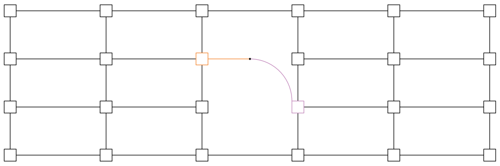
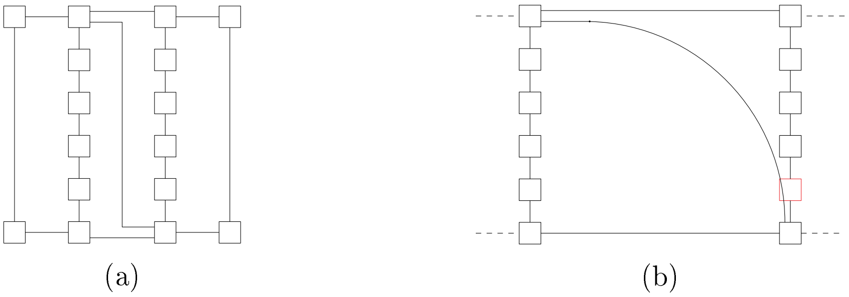


Figure 19: SMOG complexity decreases

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



(a)

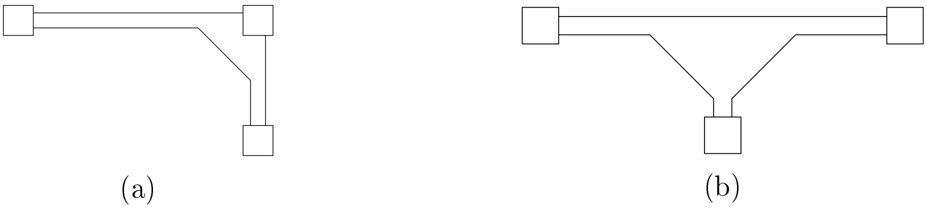
(b)

Figure 20: 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.

5.2.1 Podevsaef drawings

Recalling definition 7, 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 computated 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. Following solutions were found for L shaped and T shaped triangles:



(a)

(b)

Figure 21: 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 7 and figure 20 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.

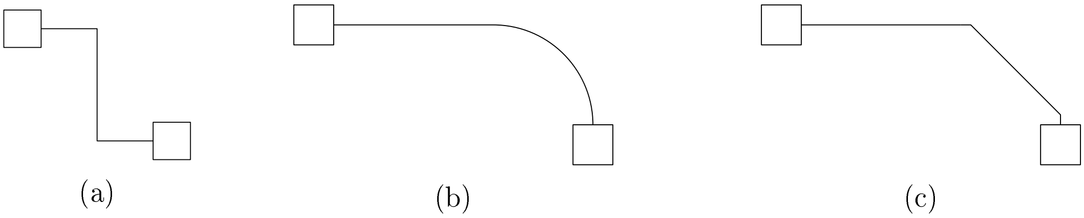


Figure 22: Complexity-3 zig zags examined

Lemma 19. *If a polyedge representation Γ_e is zig-zag shaped in Kandinsky (Figure 22a) and the port reassignment and circular arc substitution decrease the complexity in the resulting planar SMOG representation (Figure 22b), then there is a planar Podevsaef representation with two 135° bends (Figure 22c).*

So, according to lemma 19, if we look for the possibility of planar Podevsaef polyedge representation like illustrated in figure 22c in a polyedge, the port reassignment of one of the vertices might just decrease the complexity. But as we seen, not in every case.

5.2.2 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 20. *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 23b). 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 23c).

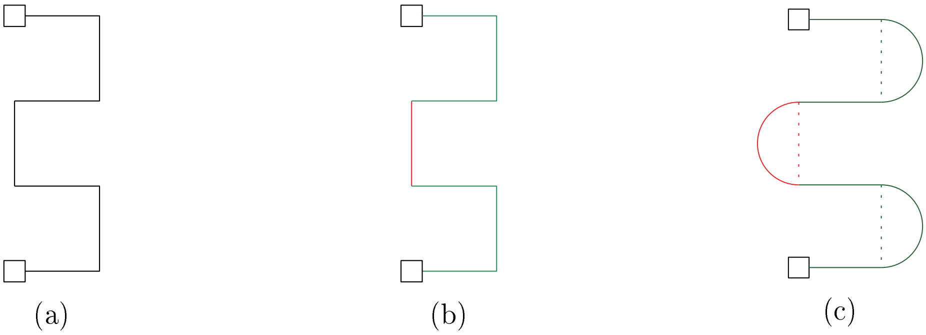


Figure 23: Illustration of the alternating fragment exception

□

5.3 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 rise the lower bound of the edge complexity on the other hand.

5.3.1 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 following example:

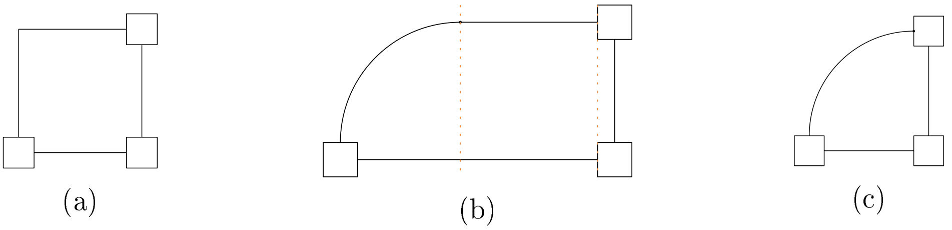


Figure 24: One plane sweep could eradicate redundant area and even segments without causing any damage

As you can see in Figure 24, an orthogonal Kandinsky drawing of a triangle gets transferred to a smooth orthogonal drawing in the Fixed Shape Model. The vertical orange-dotted 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 17. *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 21. *This area saving plane sweep is able to save up to $\mathcal{O}(n)$ area and potentially lowers the overall complexity of a drawing, resulting in a SMOG drawing of size $\mathcal{O}(n) \times \mathcal{O}(n)$.*

Proof. Consider following orthogonal drawing of the graph:

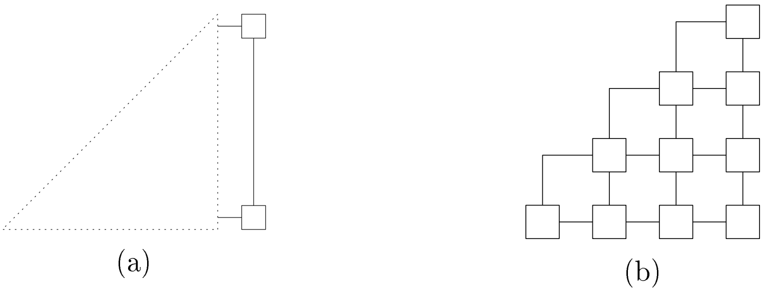


Figure 25: Orthogonal drawing with maximal saving possibilites

The dotted triangle of figure 25a consists of $\mathcal{O}(n) \times \mathcal{O}(n)$ vertices as illustrated in figure 25b. Applying the stretching technique and circular arc substitution, the resulting SMOG drawing is of size $\mathcal{O}(n^2) \times \mathcal{O}(n)$.

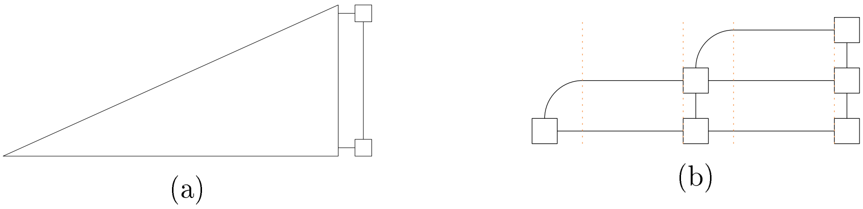


Figure 26: SMOG drawing with maximal saving possibilites

In figure 26b, the orange dotted lines indicate where the area saving plane sweep triggers an event. Notice that there are $\mathcal{O}(n)$ events.

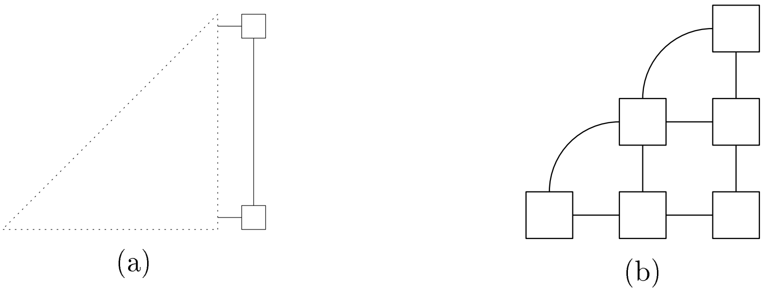


Figure 27: SMOG drawing after applying the plane sweep

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

5.3.2 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 rise the edge complexity of any drawing.

5.3.3 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:

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

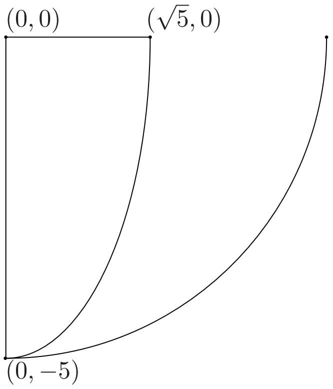


Figure 28: Illustration of equation 6

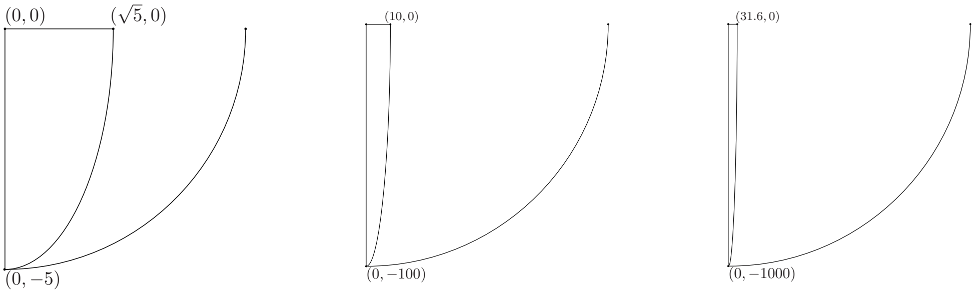
For the orientation of the ellipse arcs, we will pick the values of x and y adequately. The extreme values of equation 6 are $(0, -5)$ and $(\sqrt{5}, 0)$, it would fit right in instead of a circular quarter arc with radius 5. If 5 was the longest vertical segment, it would be sufficient to stretch the 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. Utilizing the strict monotony of the square root function, we would now be able to stretch the original Kandinsky drawing by the square root of the longest vertical segment $\sqrt{l}$, guaranteeing $\mathcal{O}(n \cdot \sqrt{n}) \times \mathcal{O}(n)$ area.

5.3.4 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.

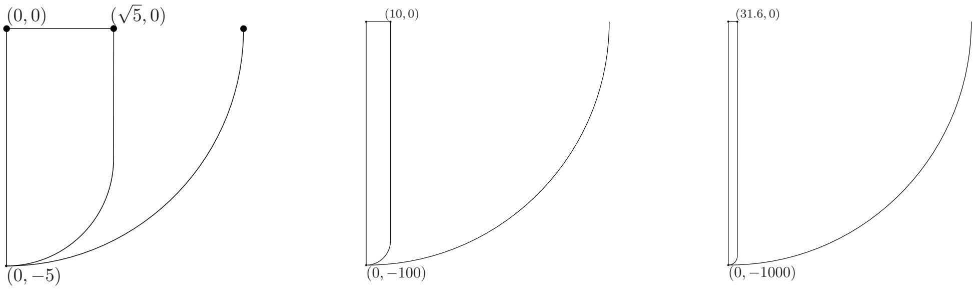


(a) $|l'| = 5$ (b) $|l'| = 100$ (c) $|l'| = 1000$

Figure 29: Illustration of increasing values for the vertical segment result in very steep ellipse arcs

5.3.5 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 30: 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 orthognal layout would significantly rise.

6 Related Work

6.1 $\mathcal{NP}$ Hardness

Bekos et al. proposed a reduction from the $\mathcal{NP}$-hard problem `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.

Theorem 11. *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 31, there are multiple quarter arc connections possible 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.

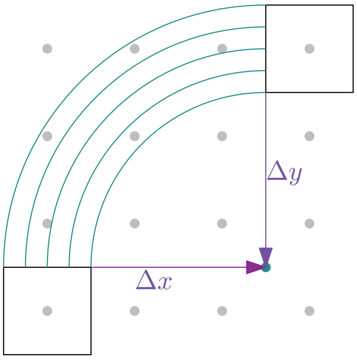
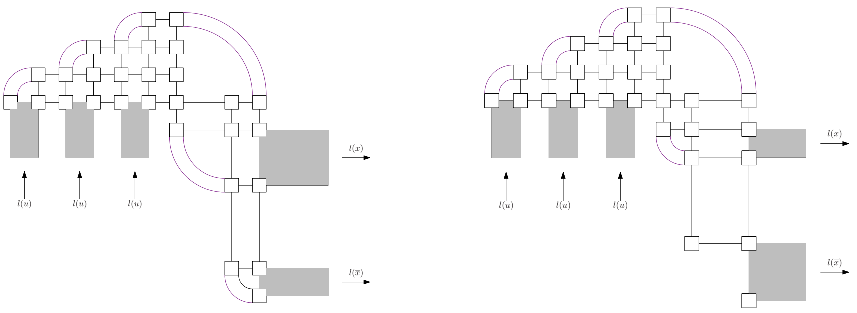


Figure 31: Multiple possible bendless connections

Theorem 12. *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 whether G admits a bendless SMOG 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. Basically, for the $\mathcal{NP}$-hardness, the reduction from 3-$\mathcal{SAT}$ 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 31 illustrated by the 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.

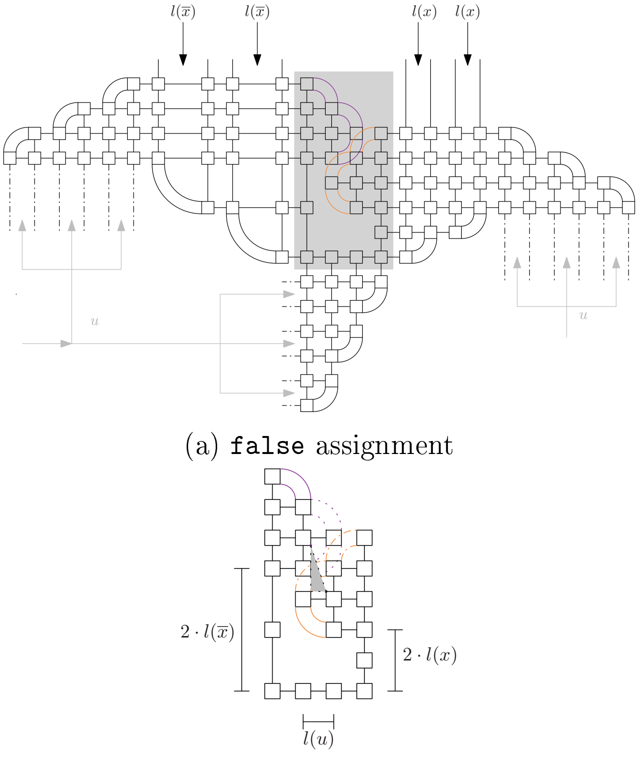


(a) `true` assignment

(b) `false` assignment

Figure 32: new variable gadgets

- *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 33: 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 33b. The diagonal has to be at least $2 \cdot l(u)$.

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

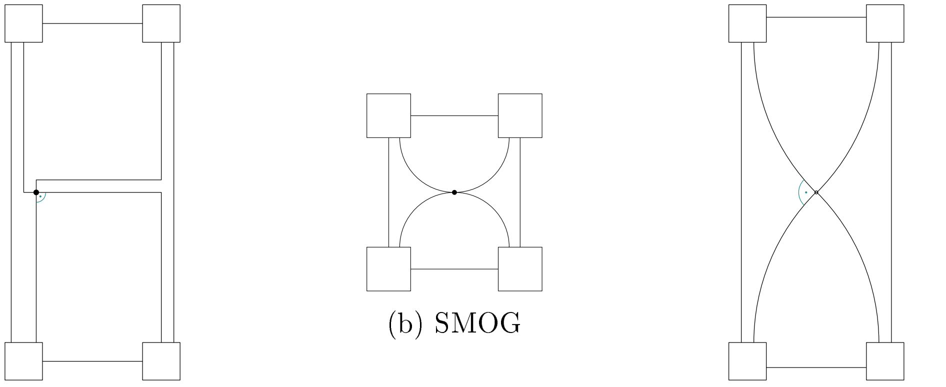
So, in order to avoid crossings, following property must hold for $l(x), l(\overline{x})$:

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

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. □

6.2 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

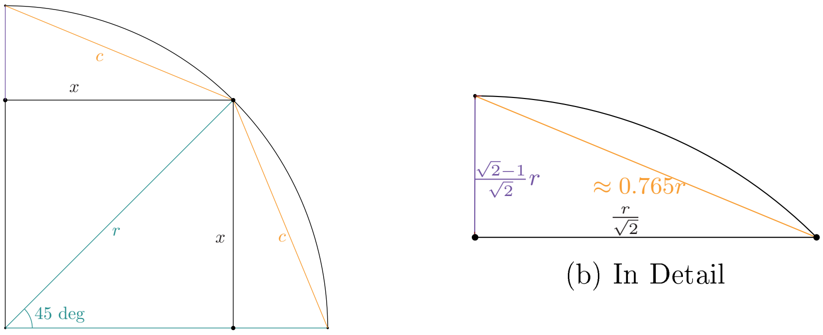
Figure 34: Various illustrations of the hourglass drawing

The orthogonal drawing illustrated in Figure 34a 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 34b inherits a decrease in the edge complexity by 1 but the crossing is not illustrated in a visibly clear way. Introducing *octi arcs*, we could find a compromise between the edge complexity and the visible clear crossing. Note that orthogonality of the crossing is illustrated with a 45° turn compared to the crossing of Figure 34a.

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



(a) Construction

(b) In Detail

Figure 35: Examination of octi arcs

$$2x^2 = r^2 \tag{8}$$
$$\Leftrightarrow x = \frac{r}{\sqrt{2}} \tag{9}$$

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

$$\begin{aligned} c^2 &= (r-x)^2 + x^2 && (10)\\ &= r^2 - 2xr + 2x^2 && (11)\\ &= r^2 - 2\frac{r}{\sqrt{2}} \cdot r + r^2 && (12)\\ &= 2r^2 - \sqrt{2}r^2 && (13)\\ \Leftrightarrow c &= \left(\sqrt{2-\sqrt{2}}\right) r && (14)\\ &\approx 0.7654 \cdot r && (15) \end{aligned}$$

6.2.2 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.

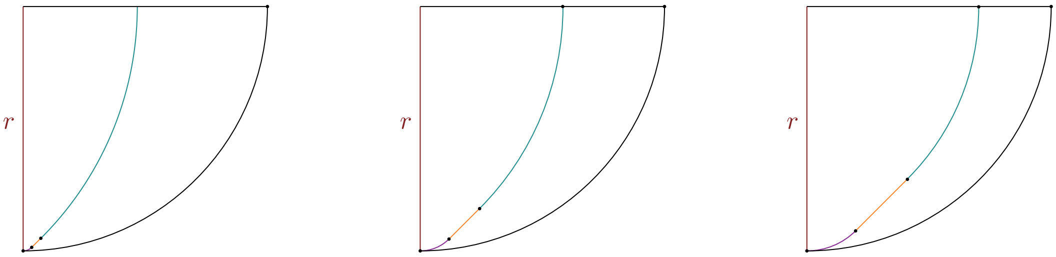


Figure 36: Various ratios of the combination of octi arcs and diagonal segments

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. Without loss of generality, we will examine the properties without

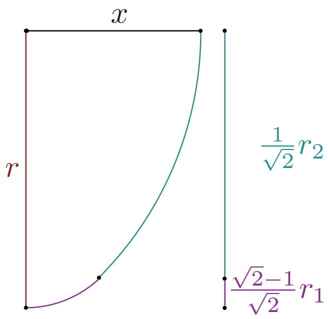


Figure 37: Illustration of the combination of octi arcs

a diagonal segment. In figure 37, 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:

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

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

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

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.

7 An Example

Consider the following input graph drawing Γ_G:

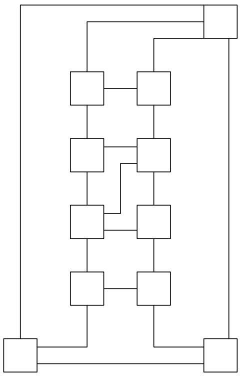


Figure 38: Example input drawing Γ_G

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.

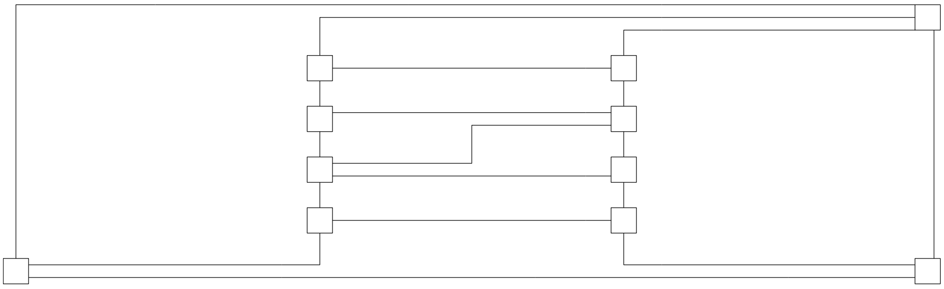


Figure 39: Γ_G after the stretching technique application

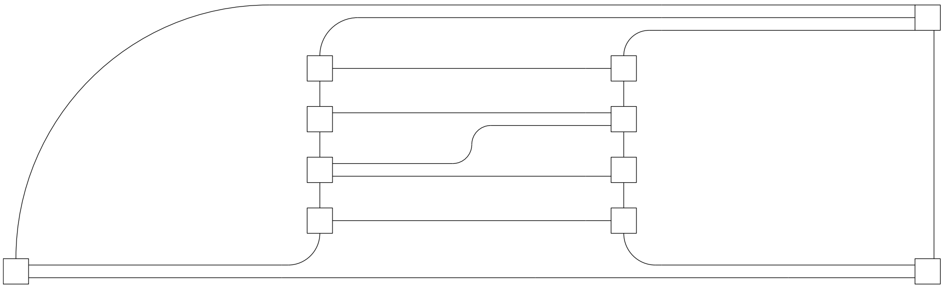


Figure 40: Γ_G after the circular arc substitution

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 drawing inherits some redundancy which we can get rid of thanks to the saving plane sweep. Now, the size of the drawing equals 37 × 11.

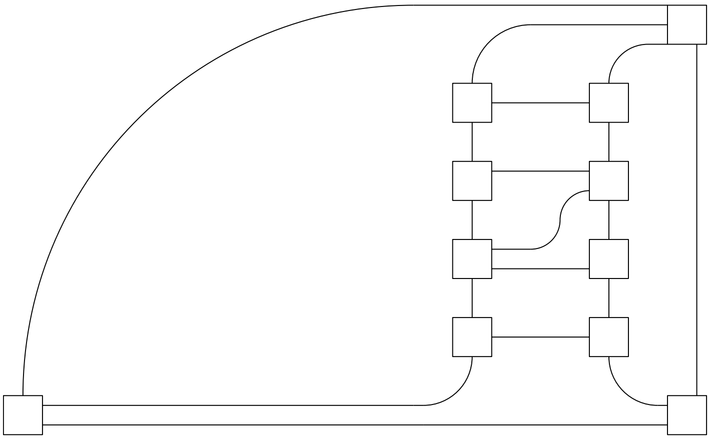


Figure 41: Γ_G after the saving plane sweep application

The drawing lost 52,7% in horizontal area requirements resulting in a drawing with 19.5×11 area bounds. Also, the complexity decreased from 4 to 3.

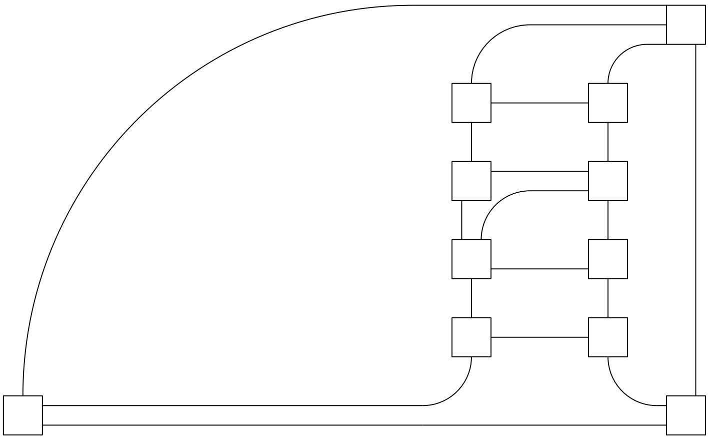


Figure 42: Γ_G after the port reassignment

The port reassignment results in a smoothened drawing of complexity 2.

8 Future Work

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.

9 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 Alam, Michael Dillencourt, and Michael Goodrich. "Capturing Lombardi Flow in Orthogonal Drawings by Minimizing the Number of Segments". In: (Aug. 2016).

[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*. Ed. by Fabrizio Frati and Kwan-Liu Ma. Cham: Springer International Publishing, 2018, pp. 169–183. ISBN: 978-3-319-73915-1.

[3] Michael A. Bekos et al. "Smooth Orthogonal Layouts". In: *Journal of Graph Algorithms and Applications* 17.5 (2013), pp. 575–595. DOI: 10.7155/jgaa.00305.

[4] Michael A. Bekos et al. "The Effect of Almost-Empty Faces on Planar Kandinsky Drawings". In: *Experimental Algorithms*. Ed. by Evripidis Bampis. Cham: Springer International Publishing, 2015, pp. 352–364. ISBN: 978-3-319-20086-6.

[5] Christian A. Duncan and Michael T. Goodrich. *Planar Orthogonal and Polyline Drawing Algorithms*. 2013.

[6] Christian A. Duncan et al. "Lombardi Drawings of Graphs". In: *CoRR* abs/1009.0579 (2010). arXiv: 1009.0579. URL: http://arxiv.org/abs/1009.0579.

[7] David Eppstein. "Planar Lombardi Drawings for Subcubic Graphs". In: *CoRR* abs/1206.6142 (2012). arXiv: 1206.6142. URL: http://arxiv.org/abs/1206.6142.

[8] Seok-Hee Hong, Damian Merrick, and Hugo A. D. do Nascimento. "Automatic Visualisation of Metro Maps". In: *J. Vis. Lang. Comput.* 17.3 (June 2006), pp. 203–224. ISSN: 1045-926X. DOI: 10.1016/j.jvlc.2005.09.001. URL: http://dx.doi.org/10.1016/j.jvlc.2005.09.001.

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

[10] Stroh. *Constraints In Orthogonal Graph Drawing*. March 2018. URL: http://slideplayer.org/slide/660000/.