

Eberhard Karls Universität Tübingen
Wilhelm Schickard Institut Tübingen

Fachbereich Informatik

Minimizing the edge length ratio of planar poly-line graph drawings

Arbeitsbereich Algorithmik

Forschungsprojekt SS21

Autor: Benjamin Ulvi Çoban
MatNr. 3526251

Version vom: September 8, 2021

Betreuer: Prof. Dr. Michael Kaufmann

Contents

1	Introduction	3
2	Preliminaries	4
3	The Problem	6
3.1	The Symposium Challenge	6
3.2	Formalization Of The Problem	6
3.2.1	The edge-length ratio	6
3.2.2	Upper and lower bounds of the ratio	7
3.2.3	A small example - a triangle	7
4	Maximal planar graphs	9
4.1	Initial situation	9
4.2	Refinement step	10
4.3	Extending G to a supergraph G'	10
4.4	Elongation step	11
4.5	Using n bends	13
4.6	Summary	18
4.7	Runtime	19
4.8	An Example	20
5	2-trees	24
5.1	Previous results	24
5.2	The drawing algorithm for a maximal series-parallel graph	25
5.3	The modification	28
6	Other Results	32
6.1	k -nary trees	32
6.2	Certain family of 3-trees	34
6.3	Outerplanar graphs	35
7	Future Work	37
7.1	Tweaking the postprocessing algorithm	37
7.2	Further minimizations	37
7.3	New drawing algorithm approach	37
7.4	Implementation	37

1 Introduction

The topic of visualization of information relationships occur in various areas of work. Examples of the fields include circuit design, architecture, web science, social sciences, biology, geography, information security and software engineering. Over the last decades, many different efficient algorithms were developed for graph drawings in the Euclidean plane.

Among others, one classic question is to test whether a given network can be visualized with straight lines and prescribed edge lengths. This study is also related to several other topics like rigidity theory, structural analysis of molecules and sensor networks [[1], Page 1]. For over 25 years, an international symposium of Graph Drawing and Network Visualization takes place annually. 28th International Symposium of Graph Drawing and Network Visualization will be held from September 15th to 17th in Tübingen.

Part of the symposium is a traditional Graph Drawing Contest. The contest consists of two parts - the *Creative Topics* and the *Live Challenge*. The main focus for the Creative Topics lies on the creation of drawings of two given graphs. Aspects to consider for the visualization are clarity, aesthetic appeal and readability.

On the other hand, the Live Challenge is held similar to a programming contest. Participants, usually teams, will get a theme and a set of graphs and will have one hour of processing. The results will be ranked and the team with the highest score wins the competition. The teams will be allowed to use any combination of software and human interaction systems in order to produce the best results. Usually, the challenge is derived from a theoretical optimization problem.

On the occasion of the Graph Drawing contest held this year in Tübingen, this report covers the topic of drawing various graph classes with connections of approximately equal length.

2 Preliminaries

As otherwise mentioned, a *graph* $G = (V_G, E_G)$ is a tuple consisting of two sets - the set of vertices and the set of edges. An *edge* $e = (v, w), v, w \in V_G$ is a tuple and describes a connectivity relation between two vertices. Unless otherwise mentioned, the graphs are *undirected*. It means that the edge (u, v) is identical to the edge $(v, u), u, v \in V_G$. A *face* is a maximal open region of the plane bounded by edges. The degree of the graph is the amount of edges attached to it. A graph G' is called a supergraph of G iff $V_G \subseteq V_{G'}$ and $E_G \subseteq E_{G'}$.

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. G is maximal planar iff any further edge insertion violates the planarity property. A $k \times k$ grid is an undirected graph consisting of k rows and k columns of vertices. A vertex in the i -th row and j -th column is denoted as (i, j) .

A straight line drawing on a grid of size $k \times k$ is a drawing where every vertex has its unique row and column value and every edge is drawn as a straight line. In a polyline drawing, each edge is represented by a non-empty sequence of line segments $(e = (e_1, e_2, \dots))$, where two consecutive line segments intersect in a unique point, a bend. Every bend lies, like the vertices, on a unique grid point.

To measure the length, the euclidian distance of a line segment is introduced as a metric. It is defined as the square root of the sum of the quadratic difference of row and column. The unit length, ULin short, is defined as the distance between two consecutive points on the grid with either the same row or column value.

A path is a sequence of edges. A cycle is a path, so that the starting vertex and the ending vertex are identical. A tree is an acyclic graph. A graph is called connected if there is a path from every vertex to every other vertex of G . A k -ary tree is a graph where either every vertex has exactly k children and one parent or is of degree 1. These are called leaves. The root of a tree is a vertex with no parent. The height of a tree is defined as the length of the longest path starting from the root. A tree is called complete, when all leaves have the same height and no further vertices can be inserted without increasing the maximum height.

A 2-terminal series-parallel graph with terminals s, t is a recursively defined graph with one of the following three rules:

1. An edge (s, t) is a 2-terminal series-parallel graph
2. If $G_i, i = 1, 2$, is a 2-terminal series-parallel graph with terminals s_i, t_i , then in the serial composition t_1 is identified with s_2 to obtain a 2-terminal series-parallel graph with s_1, t_2 as terminals
3. If $G_i, i = 1, \dots, k$, is a 2-terminal series-parallel graph with terminals s_i, t_i , then in a parallel composition we identify all s_i into one terminal s and all t_i into the other terminal t and the result is a 2-terminal series-parallel graph with terminals s, t .

A series-parallel graph, SP-graph in short, is a graph for which every biconnected component is a 2-terminal series-parallel graph. A SP-graph is maximal if no edge can

be added so while maintaining a SP-graph.

A 2-tree is a recursively defined graph with at least three vertices. If $n = 3$, then the 2-tree is the K_3 . If $n > 3$, then start with a K_3 and every vertex added is adjacent to exactly two adjacent neighbours, forming a 3-clique. The class of 2-trees correspond to the class of maximal SP-graphs. [[2], Preliminaries], [[6], Preliminaries], [[3]]

3 The Problem

3.1 The Symposium Challenge

There has been recent attention to the edge-length ratio of a planar drawing, which describes the ratio between the lengths of the longest edge and the shortest edge in a drawing.

This year, the main topic addresses an optimization problem, namely the minimization of the edge-length ratio of poly-line drawings of planar, undirected graphs on a fixed grid. For a poly-line edge, the edge-length is the sum of the line segment lengths.

The input consists of a JSON file with the following entries:

- nodes** Every node has an unique ID value between 0 and the amount of nodes - 1, a value for the x and y coordinate each, delimited by the width and height
- edges** Every edge has an ID for source and destination each and an optional list of bend points, specified in x and y coordinate
- width (optional)** The maximum x -coordinate of the grid. If unspecified, the width is set to 1,000,000.
- height (optional)** The maximum y coordinate of the grid. If unspecified, the height is set to 1,000,000.
- bends** The maximum number of bends allowed per edge

The results of the optimization are also JSON files. The planarity of the graph shall be preserved and the poly-line edge-length ratio minimized by relocation of the nodes.

For the teams participating with their own tools, an embedding might not be given with the input. For the participants working manually, an embedding is already given beforehand. [4]

3.2 Formalization Of The Problem

3.2.1 The edge-length ratio

Let Ω_G be a given planar poly-line drawing. The length of an edge of e is defined as the sum of $k + 1$ line segments, induced by k bends. l_{\max} is the length of the longest edge in Ω_G , l_{\min} is the length of the shortest edge in Ω_G . The edge-length ratio r is defined as:

$$r = \frac{l_{\max}}{l_{\min}} \quad (1)$$

It trivially holds, that $r \geq 1$. r is said to be optimal if $r = 1$. Then, all the edges in a drawing are of the same length.

3.2.2 Upper and lower bounds of the ratio

A straight-line drawing Γ_G is drawn with zero bends. There are multiple straight-line drawing algorithms which produce a drawing of area $\mathcal{O}(n) \times \mathcal{O}(n)$. The area consumption of a straight-line drawing directly induces the bounds for the ratio. Let $k \times k$ be the grid Γ_G is drawn on, $k \in \mathcal{O}(n)$. The maximal length of a straight line is bound by $\sqrt{2}k$, from one corner of the grid to the diagonal opposing one, while the minimal length a straight line can inherit values obviously 1. The ratio therefore values $\sqrt{2}k$ in the worst case.

This automatically gives an upper bound for any poly-line drawing Ω_G since a straight-line drawing can be seen as a poly-line drawing with 0 bends. Bends will be included in a straight-line drawing to lengthen the shorter edges while the longest edge will be untouched.

3.2.3 A small example - a triangle

This example will illustrate the issue of finding a drawing with an optimal edge-length ratio. Consider K_3 , the complete graph with three vertices. K_3 has got one outerface and one inner face, in shape of a triangle. In order to find a drawing of K_3 with an optimal ratio, we will see that one bend per edge is mandatory.

A triangle with straight-lines as edges with length l will have a height of $\frac{\sqrt{3}}{2}l$ in order to be optimal regarding the ratio. Unfortunately, $\sqrt{3}$ is not a rational number. There do not exist two integers in order to express $\sqrt{3}$ as a fraction. As a consequence, there does not exist any combination of coordinates on a grid in order to draw the K_3 without any bends and an optimal ratio.

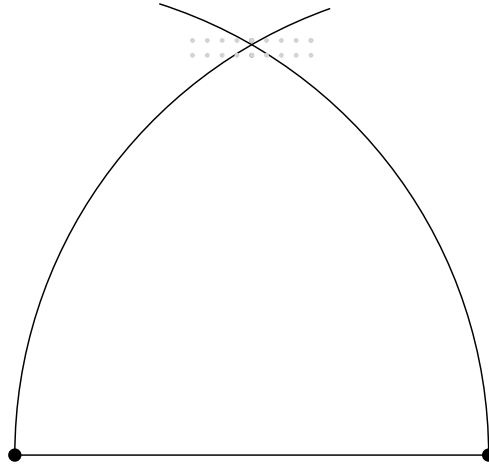


Figure 1: There is no grid point for an optimal drawing without any bends

But, when introducing one bend per edge, there exists a drawing with an optimal ratio.

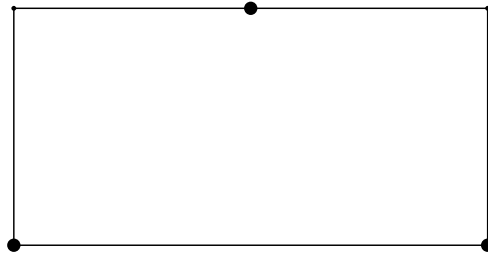


Figure 2: With one bend, K_3 admits an optimal drawing

Allowing bends will help minimizing the ratio of a drawing.

4 Maximal planar graphs

This section will demonstrate a postprocessing approach for an already given straight-line drawing of a maximal planar graph G , namely Γ_G , in order to minimize the edge-length ratio of planar graphs.

The main idea is to elongate the shortest edge as long as possible while still maintaining planarity. Obviously, the longest edge of Γ_G will not be altered at all. The small length edges may be enclosed by small area faces which proves to be a challenge for a required elongation. In order to overcome this hurdle, the grid will be refined as much as is necessary. On the one hand, this increases the total area usage of the new drawing Γ'_G , but will provide sufficient area for any small length edge elongation on the other hand.

After the grid refinement, every inner face of G will be subdivided evenly. For every face, a new vertex will be inserted which is adjacent to the vertices defining the face. This will extend G to the supergraph G' . In the further process, the inserted edges and vertices serve as a guideline for the edge elongations.

Then, every straight line edge of G that is smaller than the longest edge will be elongated using a linear amount of bends, altering the straight-line drawing to a poly-line drawing. The line segments will stay in the respective adjacent faces defined by G' . The reader will observe that the amount of bends and the factor of grid refinement will play along and the edge-length ratio will become a constant.

4.1 Initial situation

At the beginning, a given straight-line drawing Γ_G of a planar graph G is given. We choose the drawing algorithm by Schnyder. The area consumption values $(n-1) \times (n-1)$ in the worst case [5]. When considering maximal planar graphs, the following fact describes the shape of every face of G .

Fact 1.

In a maximal planar graph G , every face is a triangle. Otherwise, G is not maximal planar.

Proof by contradiction. Let f be a face of G defined by $k > 3$ vertices. Then, it is possible to insert at least one more edge into f , contradicting the maximal planarity of G . \square

Regarding area consumption, the smallest possible face in Γ_G is a triangle with two edges of UL. If these smallest-area faces are adjacent to an edge, then there is no bend point on the grid available for further elongation. Each triangle consumes area of $\frac{1}{2}UL^2$ and the length of e cannot be increased.

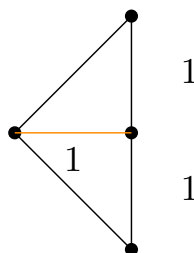


Figure 3: An edge of UL (marked in orange), enclosed by two smallest-area triangles

With a refinement in the first step, bend points are inserted into the adjacent triangles since their area consumption expands.

4.2 Refinement step

Fact 2.

Let f be a face with an area consumption of $\mathcal{O}(a(k))$, $a(k)$ monotone increasing function and k describing the granularity of the underlying grid. If the grid is refined by a factor of $c \cdot n$ for each dimension, then f consumes area of $\mathcal{O}(c^2 n^2 \cdot a(k))$.

So, by Fact 2, the smallest possible area of a face now equals $\frac{1}{2}(cn)^2 \text{UL}^2$. Since every edge of unit length now is at least $c \cdot n$ long, the amount of bend points inside a face f is bound by $\mathcal{O}((cn)^2)$.

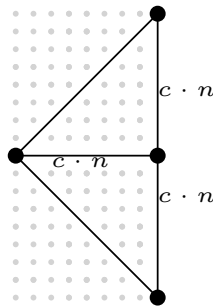


Figure 4: The worst case from Figure 3 after the refinement

In the following sections, a straight-line drawing Γ_G refined by a certain factor will be called Γ'_G , since the coordinates of the vertices changed. Although the shapes of the faces will stay the same, the area consumption is a different one.

4.3 Extending G to a supergraph G'

The supergraph is helpful for any transition of straight-line edge to a poly-line edge. It will describe where valid bend points lie. At first, the properties of maximal planar graphs is

Fact 3.

Let f be a face of a drawing Γ'_G with a constant amount of vertices defining it, inheriting area $\Theta(f(n))$. If G is extended in a way that a new vertex, connected to all the vertices defining f , is placed inside f so that f is divided evenly, the resulting faces of the supergraph have still area $\Theta(f(n))$.

Proof. Let a be the area consumption of a face f , $a \in \Theta(f(n))$. Since the face is divided evenly by a constant amount c , every new face consumes at most $\frac{a}{c} + \varepsilon$ area and at least $\frac{a}{c} - \varepsilon'$ area ($\varepsilon, \varepsilon' > 0$). $\varepsilon, \varepsilon'$ describe the rounding error derived from the grid.

$$a \in \Theta(f(n)) \Rightarrow \frac{a}{c} + \varepsilon, \frac{a}{c} - \varepsilon' \in \Theta(f(n))$$

□

So, if the grid is refined by $c \cdot n$, then, the face inheriting area $\frac{1}{2}(cn)^2 \mathbf{UL}^2$ can be subdivided into three faces of roughly area $\frac{1}{6}(cn)^2 \mathbf{UL}^2$. Since we are working on a grid with integers, there might be an approximation error which can be left out.

Fact 4.

Let G be a maximal planar graph and f a face of G . Extend G to G' by adding one vertex inside of f and connecting to every vertex defining f . Then, G' is maximal planar.

Proof. If f is an inner face, then it is a triangle with three vertices defining f . The vertex placed inside divides f into three triangles, since it is connected to every of the three vertices defining f . G' has now $v + 1$ vertices, three more edges and one face is substituted with three smaller faces. Therefore, the amount of faces is increased by two and Eulers Formula holds.

If f is the outerface, then there are three vertices on the outerface, as well. Otherwise, there would be edges addable to G , and G is not planar. The same holds for this case. \square

The intendation behind the vertex insertions is to evenly subdivide every face into three distinct new faces so every edge adjacent to two faces will have its own area reserved to be elongated in. This ensures the planarity of the output poly-line drawing naturally. The next Fact will answer the question where for each face, the vertices are inserted in practice.

Fact 5.

Let A, B, C be three points on the grid with their respective x and y coordinates, defining a triangle. Then, there exists a point M , called the centroid, which divides the triangle into three smaller triangles with roughly the same area usage, apart from a rounding error [7]. This point is defined as the arithmetic mean of A, B, C :

$$M \left(\frac{1}{3}(A_x + B_x + C_x), \frac{1}{3}(A_y + B_y + C_y) \right)$$

After the refinement and the graph extension, there are now sufficient bend points in reserved areas for every edge of G to be elongated in available.

4.4 Elongation step

The general goal is to level all edge lengths occurring in Γ'_G . It is a question of how many bends are used in practice. The more bends are available, the more line segments can be inserted inside of a face. In the following section, the factor of refinement is specified along with a lower bound of the worst case scenario - the unit length edge with the smallest-area triangles possible attached to it.

Fact 6 (Valid bend points).

A grid point b regarding a face f is *valid* iff it lies inside f and the minimal distance to an edge defining f lies in $(0, 1]$.

Fact 7 (Zig-zag elongation).

Let A be a bounding box with area consumption $\mathcal{O}(a(k))$, where k describes a granularity of the underlying grid. The height and the width of A is described as $\mathcal{O}(h(k)), \mathcal{O}(w(k))$, respectively.

A poly-line p can be drawn inside A with length $\mathcal{O}(a(k))$, using $\mathcal{O}(\min\{h(k), w(k)\})$ bends.

Proof. Let w.l.o.g. the height be larger than the width. The other case is valid analogously. The valid bend points are placed at the top and bottom row along the width at the bounding box. Start at one left corner of A and draw a line segment along the height. For the following line segments, alternate with bends points between the top and bottom row until a right corner of A is reached. The amount of bends are at most two times the width length. The length of the poly-line p is therefore at most:

$$\text{len}(p) \leq (2 \cdot w(k) + 2) \cdot h(k) \in \mathcal{O}(w(k)h(k))$$

□

Lemma 1.

When the grid is refined by a factor of $c \cdot n$ in each dimension, every edge of Γ'_G can be elongated up to a length of $\Omega((c \cdot n)^2)$, using $\mathcal{O}(c \cdot n)$ bends.

Proof. Consider an edge e of Γ_G with UL, enclosed by two smallest-area triangle faces. After the refinement and graph extension to $\Gamma'_{G'}$, the area consumption of the faces adjacent to e values $\mathcal{O}((cn)^2)$ [Fact 3 and 2]. Using all valid bend points in both faces adjacent to e , e can be redrawn with $c \cdot n - 5$ line segments.

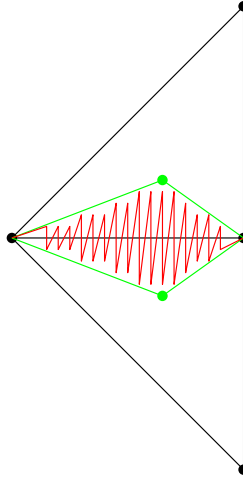


Figure 5: Using all valid bend points for line segment insertions

e is redrawn with a zig-zag elongation, using $2 \cdot (cn - 5)$ valid bend points. The new length of the sequence of line segments values at least:

$$\text{len}(e) \geq 3 \cdot \left(2 \cdot \sum_{i=1}^{\frac{cn-1}{3}-1} i + \sum_{i=1}^{\frac{cn-1}{3}-1} \sqrt{4i^2 + 1} \right) \quad (2)$$

$$> 3 \cdot \left(2 \cdot \sum_{i=1}^{\frac{cn-1}{3}-1} i + 2 \sum_{i=1}^{\frac{cn-1}{3}-1} i \right) \quad (3)$$

$$= 12 \cdot \sum_{i=1}^{\frac{cn-1}{3}-1} i \quad (4)$$

$$= \frac{2}{3}(cn)^2 - \frac{10}{3}cn - \frac{4}{3} \in \mathcal{O}((cn)^2) \quad (5)$$

□

With this minimum length of e , drawn with as many line segments as possible, the following result can be achieved.

Theorem 1. *Let Γ_G be given. If the drawing is refined by the factor of n^2 , then there is sufficient area in Γ'_G to elongate every edge so that the edge-length ratio is a constant.*

Proof. By equation 5, using $\mathcal{O}(n^2)$ bends, any edge can be elongated up to a length of $\mathcal{O}(n^4)$ while the length of the longest straight-line edge in an area of $\mathcal{O}(n^3) \times \mathcal{O}(n^3)$ lies in $\mathcal{O}(n^3)$. □

In practice, the longest edge of a straight-line drawing will not be altered in the transition from Γ_G to Γ'_G .

So, by a refinement of n^2 , it might be possible lengthen the shortest edge so that it is longer than the longest edge of Γ'_G . When the shortest edge can get longer than the longest one, then the same applies for every edge. But, this requires up to $2n^2$ bends per edge and is a bit over the top. In the next section, we will determine, that a linear amount of bends suffice so that the ratio is a constant.

4.5 Using n bends

In the last section, it was shown that a refinement by a factor of n^2 suffices to level the lengths of all edges. In this section, the number of bends used per edge is fixed to n . At first, it is shown that the edge-length ratio will still be a constant when using $n + 1$ line segments in order to draw an edge. Secondly, with help of the worst case example, a lower bound for the edge lengths is determined. Finally, the ratio will be investigated.

Lemma 2.

With a refinement by a factor of n^2 and using n bends and the supergraph extension as described above, the edge length ratio of the resulting drawing will lie in $\mathcal{O}(1)$.

Proof. With n valid bends, $n + 1$ line segments are inserted into reserved faces of at least $\mathcal{O}(n^2) \times \mathcal{O}(n^2)$ area. Since the bend points are valid ones, they lie at the boundary of the corresponding faces. When inserting a line segment on opposing bend points, its length will lie in $\mathcal{O}(n^2)$, since it spans over one dimension of the face. $n + 1$ line segments of such sort sum up to a total length of $\mathcal{O}(n^3)$. The length of the longest straight-line edge in a $\mathcal{O}(n^3) \times \mathcal{O}(n^3)$ drawing is also in $\mathcal{O}(n^3)$, therefore the ratio lies in $\mathcal{O}(1)$. \square

Next, a lower bound of edge length is determined for a specific kind of face of G - a triangle consisting of a horizontal line segment with length w - also the edge to be elongated - a vertical line segment of height h and the enclosing diagonal line segment.

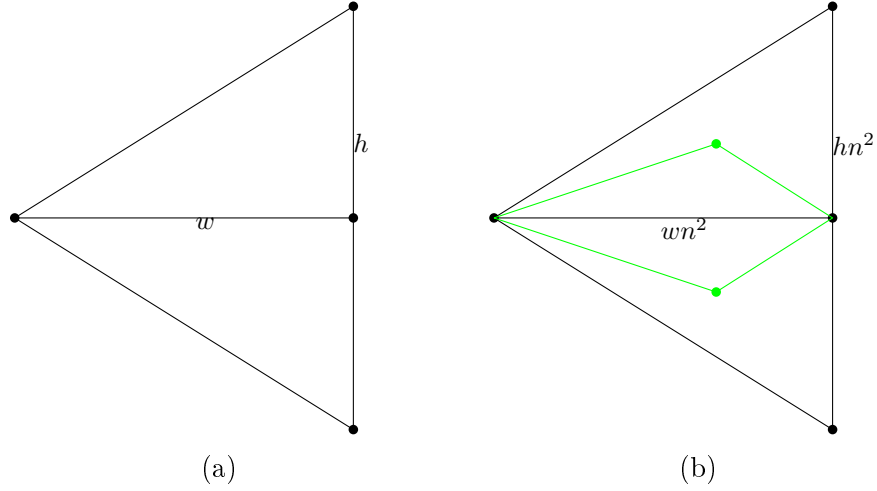


Figure 6: A general width w and height h , before (a) and after (b) grid refinement and graph extension

When recalling the position of the centroid [Fact 5], there are two cases of line segment insertions:

Case 1: $h > 3w$

In this case, the height of the face created by the supergraph extension is larger than the length of the edge of interest with length w .

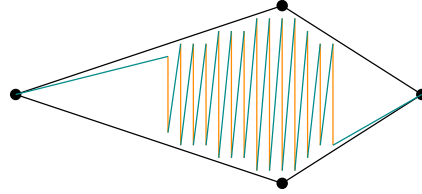


Figure 7: Inserting vertical line segments

The line segments of the new polyline are inserted vertically, using valid bend points so that all the inserted line segments are as long as possible.

Case 2: $h \leq 3w$

In this case, the length of the original edge e will at least as long as the new height to the inserted centroid M of G' . So, the line segments will be placed horizontally, parallel to e . In order to preserve planarity, a bend point must be placed near the inserted vertex M .

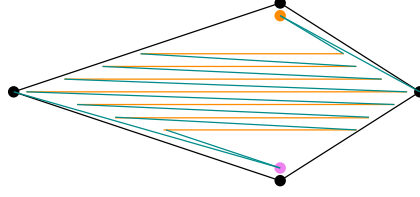


Figure 8: Inserting horizontal line segments. The orange dotted bend points preserve planarity

Lemma 3.

For two positive integers $w, h > 1$, it holds:

$$\frac{hn^2 - 1}{wn^2 - 1} < h$$

Proof. The two cases of line segment insertions are covered with the following two cases:

- Case 1: $h > w$

$$\frac{hn^2 - 1}{wn^2 - 1} < \frac{hn^2 - n^2}{wn^2 - n^2} = \frac{h - 1}{w - 1} < h$$

- Case 2: $h \leq w$

$$\frac{hn^2 - 1}{wn^2 - 1} \leq \frac{hn^2}{wn^2} = \frac{h}{w} < h$$

□

Lemma 4. When inserting up to $\frac{n}{2}$ line segments horizontally or vertically, depending on the height and the width of the face, the length of the resulting poly-line equals at least $n^3 - \mathcal{O}(n^2)$.

Proof. • Case 1: $h > 3w$

Let f be a face in Γ'_G defined by an horizontal edge $e = \overline{AB}$ with length $w' = w \cdot n^2$, w.l.o.g. a vertical line segment \overline{BC} of length $h' = h \cdot n^2$, and the enclosing diagonal line segment. When the graph is extended to G' , the centroid is placed in $\Gamma'_{G'}$ at:

$$M \left(\frac{1}{3}(A_x + B_x + C_x), \frac{1}{3}(A_y + B_y + C_y) \right)$$

Since the x-coordinate of B equals the one from C and the y-coordinate of A equals the one of B , it holds:

$$M \left(\frac{1}{3}A_x + \frac{2}{3}B_x, \frac{1}{3}C_y + \frac{2}{3}A_y \right)$$

For further length calculations, the point A is placed on the origin of the coordinate system. Then, the points are defined as follows:

$$\begin{aligned} A(0, 0) \\ B(w' - 1, 0) \\ C(w' - 1, h' - 1) \\ M\left(\frac{2}{3}(w' - 1), \frac{1}{3}(h' - 1)\right) \end{aligned}$$

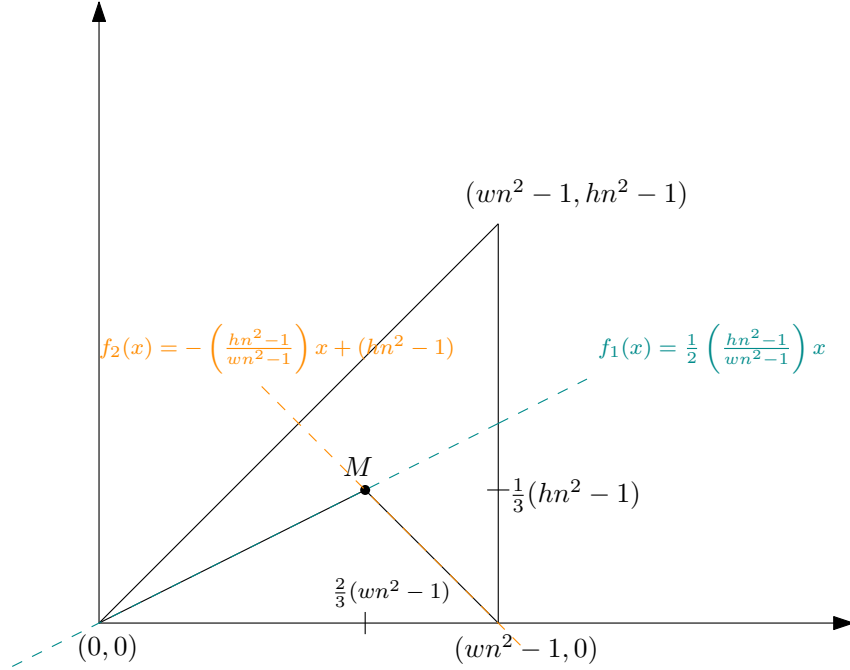


Figure 9: When A is placed on the origin, this figure shows the positions and linear function descriptions of the faces

When the y-coordinate of M is greater than w' , then vertically inserted line segments will be longer than horizontally inserted ones. The elongation with n bends and $\frac{n}{2}$ vertical line segments looks as follows:

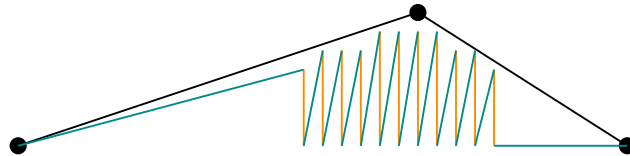


Figure 10: Inserting vertical line segments in one face only

And the lower bound of the total length can be estimated by the sum of the length of vertical line segments times two, since the diagonal line segments are at least as long as the vertical ones.

$$\text{len}(e') \geq \frac{1}{3}hn^2 \cdot \frac{n}{2} - \left(1 + \frac{1}{2} \left(\frac{h'-1}{w'-1}\right) \sum_{i=1}^{\left(\frac{n}{2}-1\right) \cdot \frac{2}{3}} i\right) - \left(\frac{h'-1}{w'-1} \sum_{i=1}^{\left(\frac{n}{2}-1\right) \cdot \frac{1}{3}} i\right) \quad (6)$$

$$+ \frac{1}{3}hn^2 \left(\frac{n}{2} - 1\right) - \frac{1}{2}h \frac{\left(\frac{n}{3} - \frac{2}{3}\right) \left(\frac{n}{3} + \frac{1}{3}\right)}{2} - h \frac{\left(\frac{n}{6} - \frac{1}{3}\right) \left(\frac{n}{6} + \frac{2}{3}\right)}{2} \quad (7)$$

$$= \frac{1}{3}hn^3 - \left(\frac{4}{9}h - \frac{1}{2}w\right)n^2 + \frac{1}{3}h \quad (8)$$

Since $h > 3w$ and $h, w \geq 1$, the length of the poly-line values at least $wn^3 - \mathcal{O}(n^2)$ and the statement holds.

- Case 2: $h \leq 3w$ In this case, the original straight line will be substituted with one horizontal line segment starting at A , ending in a bend point right before B . Furthermore, there will be a bend point right below C in order to preserve planarity. The other horizontal line segments will be placed above the bottom one between A and B . The line segments are shortened by w from the right side and $2w$ from the left side, in total $3w$ per insertion. Since we only consider one face, the number of bend points and line segments is halved.

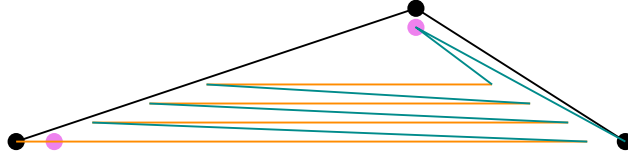


Figure 11: Inserting $\frac{n}{4}$ horizontal line segments. The orange dotted bend point preserves planarity

Then, the following lower bound for the length can be calculated:

$$\text{len}(e') \geq wn^2 \cdot \left(\frac{n-2}{4} - 3w \sum_{i=1}^{\frac{n-2}{4}} i\right) \quad (9)$$

$$+ wn^2 \cdot \left(\frac{n-2}{4} - 1\right) - 3w \sum_{i=1}^{\frac{n-2}{4}-1} i \quad (10)$$

$$= \frac{1}{2}wn^3 - \frac{35}{16}wn^2 + \frac{3}{4}wn - \frac{3}{16}w \quad (11)$$

This length is achieved by using the area of one adjacent face to e . However, this length is doubled when using both adjacent faces. The total length then values at least $wn^3 - \mathcal{O}(n^2)$ and since $w, h \geq 1$, the statement holds. \square

Lemma 5. *The calculated minimum length also holds for any edge of G , regardless of the orientation of the adjacent faces.*

Proof. Since every face is a triangle, the defining line segments can always be described with help of a linear function $f(x) = mx + b$, as already done in Figure 9. Consider a valid bend point K . Then, the next valid point lies either above or below by one row, or left or right by one column. The actual position is determined by the slope of the linear function, describing the line segment next to K . This slope is a constant. So, by inserting a new horizontal line segment above or below a preexisting one occupying K , or by inserting a new vertical line segment left or right a preexisting one occupying K , the difference of length between those inserted line segments is a constant one. Therefore, the resulting length of the zig-zag line is of form $n^3 - \mathcal{O}(n^2)$. \square

Theorem 2.

In $\Gamma'_{G'}$, every edge of G can be elongated to a poly-line with a length of at least $n^3 - \mathcal{O}(n^2)$, using n bends.

Proof. As a consequence of the refinement, every face f inherits at least area of $\mathcal{O}(n^2) \times \mathcal{O}(n^2)$. When considering bend points only valid iff they lie on the boundary of f and line segments must traverse the face, then there will be $n + 1$ line segments inserted with a length of at least n^2 , shortened by a linear amount in the worst case. This leads to n line segments with length at least $n^2 - \mathcal{O}(n)$, summing up to a total length of at least $n^3 - \mathcal{O}(n^2)$. \square

4.6 Summary

The previous results can be summarized to the following theorems:

Theorem 3.

Let G be a maximal planar graph, admitting a planar Schnyder straight-line grid drawing Γ_G with a maximal area consumption of $(n - 1) \times (n - 1)$. Let $f(n)$ be a monotone function which describes the amount of bends per edge allowed. If $f(n) = n$, and the grid is refined by a factor of n^2 , then Γ_G can be altered to a poly-line drawing Ω_G such that the edge-length ratio lies in $\mathcal{O}(1)$ in area $\mathcal{O}(n^3) \times \mathcal{O}(n^3)$.

Proof. Let Γ_G be a Schnyder straight-line drawing of a maximal planar graph with n vertices. Then, the edge-length ratio lies in $\mathcal{O}(n)$ since the length of the shortest edge values at least 1UL and the length of the longest edge values at most $\sqrt{2}(n - 1)$. Refine the grid by a factor of n^2 in each dimension to obtain Γ'_G in area $\mathcal{O}(n^3) \times \mathcal{O}(n^3)$. Then, by Fact 2, every inner face consumes at least area of $\mathcal{O}(n^4)$. For each inner face f , a vertex is inserted and connected to each of the three vertices defining f . Mark the new vertices and edges. These new vertices extend G to G' . G' has $3n - 6$ edges, there are 3 edges on the outerface and $3n - 9$ inner edges. Every edge of G is adjacent to two faces and enclosed by exactly four marked edges in G' . By Fact 3, the area enclosed by these marked edges is at least $\mathcal{O}(n^4)$ big and, by construction, this area is free in Γ'_G . By Fact ??, every edge of G drawn in Γ'_G can be elongated “zig-zag wise“ by a factor of n bends used within the marked enclosing edges. For an edge of G with length $\mathcal{O}(n^2)$ in Γ'_G , with n bends the zig-zag elongation results in a length of $\mathcal{O}(n^3)$, by Theorem 2 at least $n^3 - \mathcal{O}(n^2)$ long. By leveling the length of all edges of G to $\mathcal{O}(n^3)$ and deleting the edges and vertices of $G' \setminus G$ again, the result is a poly-line drawing Ω_G with an edge-length ratio of $\mathcal{O}(1)$ in area $n^2(n - 1) \times n^2(n - 1)$ and n bends per edge. \square

The following pseudocode sketches this drawing algorithm:

Algorithm 1: Modification of a straight-line drawing

Input: A Schnyder straight-line drawing Γ_G in area up to $n - 1 \times n - 1$
Output: A poly-line drawing Ω_G with a minimized edge-length ratio

```

1  $\Gamma'_G \leftarrow \Gamma_G.\text{RefineGrid}(n^2)$  ▷ Grid refinement
2  $G' \leftarrow G$  ▷ supergraph extension
3 for inner face  $f$  of  $G$  do
4    $V(G') \leftarrow V(G') \cup \{M_f\}$  ▷  $M_f$  subdivides  $f$ 
5    $E(G') \leftarrow E(G') \cup \{(M_f, A_f), (M_f, B_f), (M_f, C_f)\}$  ▷  $f$  defined by  $A_f, B_f, C_f$ 
6    $\text{Pos}(M_f) = \frac{1}{3}(\text{Pos}(A_f) + \text{Pos}(B_f) + \text{Pos}(C_f))$ 
7    $\text{DrawVertex}(M_f)$ 
8    $\text{MarkVertex}(M_f)$ 
9    $\text{DrawEdge}(\{(M_f, A_f), (M_f, B_f), (M_f, C_f)\})$ 
10   $\text{MarkEdge}(\{(M_f, A_f), (M_f, B_f), (M_f, C_f)\})$ 
11 float  $l_{\max} \leftarrow |\text{longestEdge}|$ 
12 for edge  $e$  in  $E(G)$  do
13    $\Gamma'_{G'}.\text{EvalValidBendPoints}(e)$  ▷ Zig-zag elongation
14   int  $\text{bends} \leftarrow 0$ 
15   while  $\text{bends} < n - 1 \wedge \text{len}(e) < l_{\max}$  do
16      $\text{bends} \leftarrow \text{bends} + 2$ 
17      $\text{InsertLineSegment}(e)$  ▷ as long as possible
18      $\text{DrawPolyLine}(e)$ 
19 for  $v \in V(G')$  do
20   if  $v.\text{marked}$  then
21      $\Gamma'_{G'}.\text{delete}(v)$ 
22 for  $e \in E(G')$  do
23   if  $e.\text{marked}$  then
24      $\Gamma'_{G'}.\text{delete}(e)$ 
25 return  $\Gamma'_{G'}$ 

```

4.7 Runtime

The above algorithm refines the grid by a factor of n^2 , runtime $\mathcal{O}(n^2)$. Then, the subdivision takes place for every face, these are constant many operations for a linear amount of faces, ergo the runtime is $\mathcal{O}(n)$. The for loop in line 10 to 16: The validation of bend points may take $\mathcal{O}(n^3)$ since this is the area upper bound for the largest possible faces in a $\mathcal{O}(n^3) \times \mathcal{O}(n^3)$ drawing. And since there are a linear amount of edges (planar graph), the for loop may run in $\mathcal{O}(n^4)$ in the worst case. The deletion of the marked vertices and edges runs in linear time. The total runtime is therefore $\mathcal{O}(n^4)$.

Theorem 4.

Every planar graph G admits a poly-line drawing with an edge-length ratio of $\mathcal{O}(1)$ in area $\mathcal{O}(n^3) \times \mathcal{O}(n^3)$, allowing up to n bends per edge.

Proof. At first, add edges to G so it is a maximal planar graph G_{\max} and mark them. Then, draw G_{\max} according to the proof of Theorem 3. The output is a poly-line grid

drawing in area $\mathcal{O}(n^2) \times \mathcal{O}(n^2)$ with an edge-length ratio of $\mathcal{O}(1)$ and up to $\mathcal{O}(n)$ bends per edge. Afterwards, remove all marked edges from G_{\max} to obtain Ω_G . \square

The following observation describes the behaviour of the ratio of large-scale graphs.

Observation 1.

Assuming, the base straight-line drawing Γ_G is a Schnyder drawing, then with a refinement of n^2 and a usage of n bends the edge-length ratio converges to $\sqrt{2}$ for $n \rightarrow \infty$.

Proof. In a Schnyder straight-line drawing Γ_G , the area consumed values $n - 1 \times n - 1$. The length of the longest edge values at most $\sqrt{2}(n - 1) < \sqrt{2}n$. After the refinement by a factor of n^2 , the longest edge values at most $\sqrt{2}n^3$. The ratio r_G can then be expressed as:

$$r_G = \frac{\sqrt{2}n^3}{n^3 - \mathcal{O}(n^2)} \xrightarrow[n \rightarrow \infty]{} \sqrt{2}$$

\square

It is intuitive that for large scale graphs, the ratio might be better than for small-scale graphs. A refinement by a larger factor enables more bend points and alternations, leading to a better approximation for the edge-length levelling.

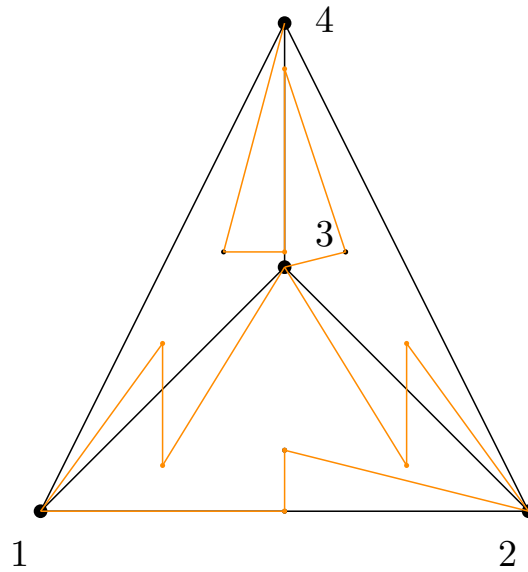
Observation 2.

Let Ω_G be a poly-line drawing of G with a ratio of r , using n bends. When the number of bends is increased to $r \cdot n$, then the ratio tends to 1 for large-scale graphs.

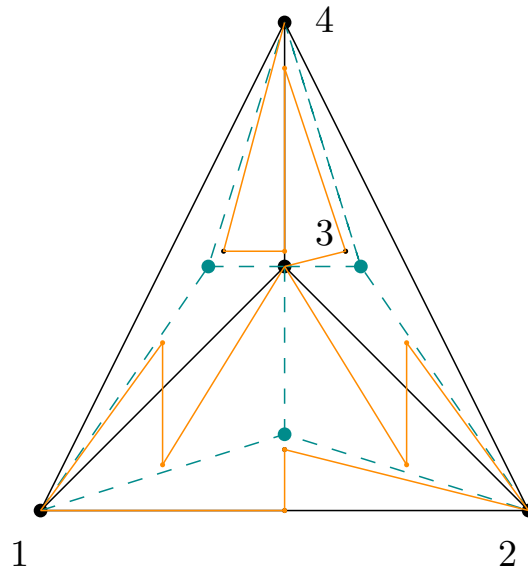
The reason for this to happen is that every line-segment of the polyline has length at least $n^2 - \mathcal{O}(n)$. Using r poly-lines, the length would be of form $r \cdot n^3 - \mathcal{O}(n^2)$. For large scale graphs, this might tend to 1.

4.8 An Example

In order to see the whole postprocessing, the ratio of a straight-line drawing of the complete graph K_4 is minimized. The input is a straight-line drawing Γ_{K_4} on a grid of size 3×3 .

Figure 12: Input drawing of K_4 on a 3×3 grid

The ratio r values $\sqrt{5}$, since the length of edge $(3, 4)$ values 1 and the length of the edges $(1, 4), (2, 4)$ value $\sqrt{5}$. Since $|V| = 4$, the grid is refined by $4 \cdot 4$, resulting in a grid of size 48×48 . The graph is extended, so that the area for edge elongations is defined.

Figure 13: Supergraph of K_4 on a 48×48 grid

Then, each edge is evaluated regarding its dashed cyan-colored bounding area and elongated with up to $n = 4$ bends.

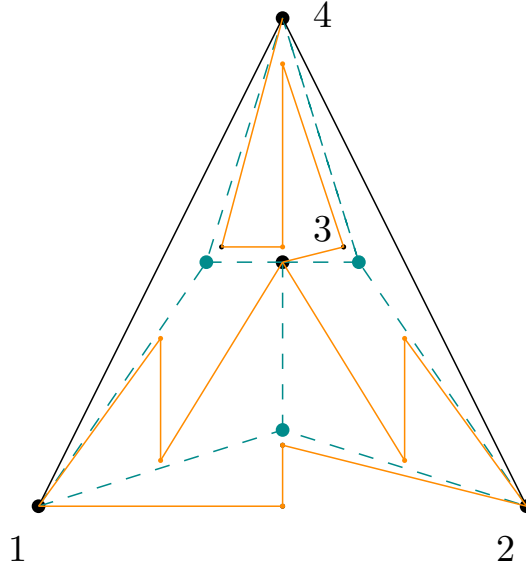
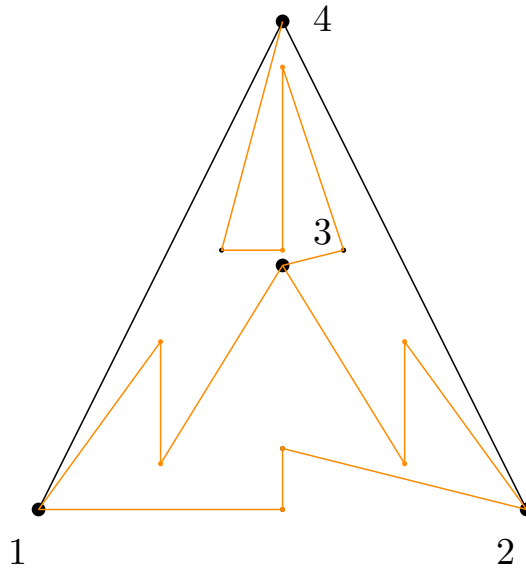


Figure 14: Inserting up to 4 bends per edge for ratio minimization

Figure 15: Ω_{K_4}

For the edges $(1,2)$, $(1,3)$, $(2,3)$, the elongation was softened to approximate the longest edge. In order to demonstrate, what happens with maximum elongation, the previously shortest edge was elongated as long as possible. The calculated lengths are:

$$|(1,4)| = |(2,4)| = 44, 72 \quad (12)$$

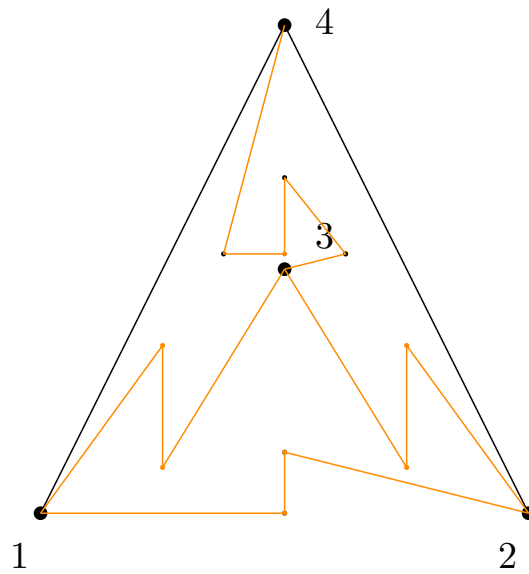
$$|(1,2)| = 44, 39 \quad (13)$$

$$|(3,4)| = 55, 45 \quad (14)$$

$$|(1,3)| = |(2,3)| = 43, 56 \quad (15)$$

$$\Rightarrow r = 1, 28 \quad (16)$$

The reason that r is not quite optimal lies in the fact that the previously shortest edge became the longest edge. So, it seems that the algorithm requires further adjusting regarding the resulting edge lengths. The following alternative drawing inherits a nearly optimal ratio:

Figure 16: Ω'_{K_4}

Here, $|(3, 4)| = 25, 24$ and $r' = 1, 04$. Nice.

5 2-trees

A new family of graphs is investigated for its edge-length ratio optimization on a fixed grid. A k -tree is an undirected graph which is incrementally created from a K_{k+1} in a way that each added vertex has exactly k neighbours such that those $k+1$ vertices form a clique. Following this definition, a 4-tree is not planar since it starts with a K_5 . Of interest is the 2-tree.

Since the class of 2-trees correspond to the class of maximal series-parallel graphs ([1], Page 2), a drawing algorithm by Biedl for SP-graphs is modified to minimize the edge-length ratio.

5.1 Previous results

There are several results for small planar drawings of outerplanar and series-parallel graphs. Among others, it was shown that:

- Every series-parallel graph has a visibility representation with $\mathcal{O}(n^{\frac{3}{2}})$ area
- Every series-parallel graph has a visibility representation with $\mathcal{O}(\Delta n \log n)$ area, where Δ is the maximum degree of the graph
- It was reproven that every outerplanar graph is series-parallel, therefore these results also apply for outerplanar graphs [[2]: Page 2,3]

Visibility representations and orthogonal drawings can be converted to poly-line drawings. The following relationships hold for drawing models:

Visibility Representation Vertices are boxes, edges are horizontal or vertical line segments. In a 1-directional visibility representation all edges are vertical line segments.

Orthogonal Box-Drawing Vertices are axis-aligned boxes (possibly degenerated to a line segment or point) and edges are sequences of contiguous horizontal or vertical line segments. A 1-directional visibility representation is also automatically an orthogonal box-drawing. In a flat orthogonal box drawing, every vertex is a vertical line segment.

Poly-line drawing Vertices are points, edges are sequences of contiguous straight-line segments. A transition point between two straight-line segments is called a bend. A poly-line drawing can be obtained by modifying an orthogonal box drawing in a way such that a vertex is reassigned to a point inside of the prior box and the incident edges are rerouted to this very point. Also, empty grid lines are added until every edge has length at least 2. The area consumption is asymptotically the same since the width and height is doubled at most. [[2]: Page 5,6]

Since visibility representations and orthogonal box drawings can be converted to poly-line drawings with asymptotically the same area, the upper bounds given for the visibility representation also hold for poly-line drawings.

5.2 The drawing algorithm for a maximal series-parallel graph

The drawing algorithm creates a visibility representation of a given maximal series-parallel graph and is also defined recursively based on 2-terminal SP graphs. The maximum height of the drawing is given by a recursive formula, depending of the heights of the subgraphs. [[2]: Page 6 to 11] Concerning the edge-length ratio, the longest edge might depend on the height of the drawing while the shortest edge might be of unit length. The upper bound for the height will be determined with help of a recursive function $h(m)$, where m describes the amount of edges.

Invariant Vertex s contains the upper right corner of the bounding box, while vertex t contains the lower right corner of the bounding box. The boxes for s and t can be stretched to the whole width of the drawing.

Base case the base case is the edge (s, t) . Simply put s over t and the invariant holds. Here, the recursive function for the height description stops at $h(1) = 2$.

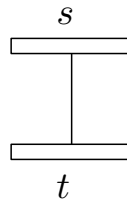


Figure 17: Basecase

Height The height of a drawing can be increased in order to preserve the invariant.

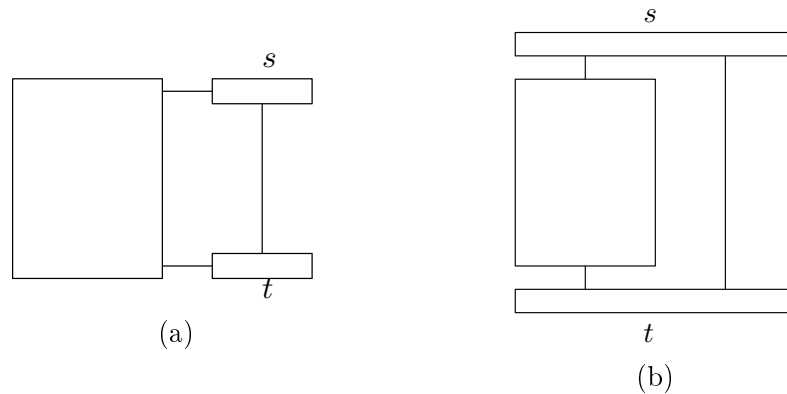


Figure 18: Terminal release so that s, t span the bounding box (b)

Inductive step This step is for $m > 1$ edges and holds two major cases: a parallel composition and a serial composition. If the composition is a parallel one, then the recursive drawings of the k parallel subgraphs will be of a serial composition and vice versa.

- During the parallel composition, $2 \leq k$ subgraphs will first be ordered in their number of edges ($m_i \leq m_{i-1}, i \in [1..k]$) and then recursively drawn. The subgraphs are drawn in serial since the original step is the parallel case. The heights of the subgraph drawings are adjusted and unified with a terminal s and t on top and bottom of the bounding box. The height is bound by $h_p(m) \leq \max\{h(m-1), h(\frac{m}{2}) + 2\}$

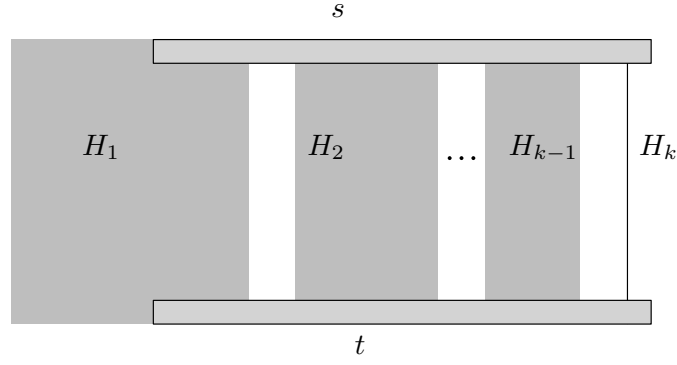


Figure 19: Case P

- For the serial composition, two subgraphs H_a, H_b with terminals s, x and x, t are recursively drawn (in parallel). In a trivial case, one of the subgraphs is an edge and $h(m) \leq h(m-1)$. Note, that x, t is an edge of H_b since the original graph is a maximal series-parallel one. There are two outcomes of serial placement. Either, the boxes for x and t share the same row on the bottom of the bounding box, being connected horizontally, or x lies in an interior row and is connected to t vertically. A precomputed index L determines, which case of serial composition holds ([2], Lemma 3.3, Page 11).

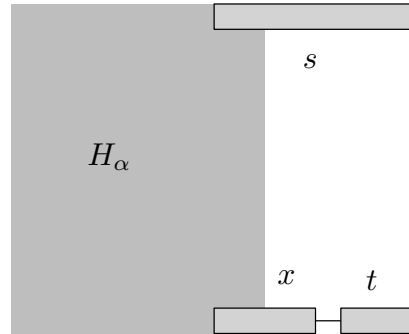


Figure 20: Case S1

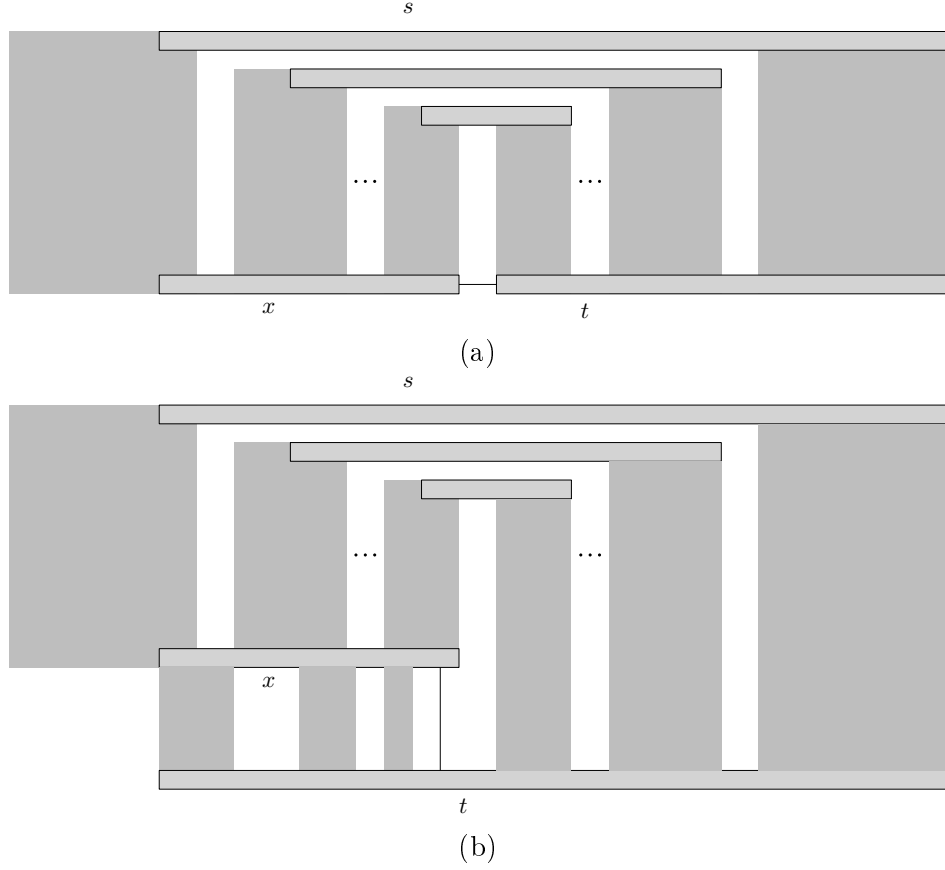


Figure 21: Cases S2(a) and S2(b)

The height is bound by:

$$h_{S2a}(m) \leq \max\{h(m-1), h\left(\frac{m}{2}\right) + L\} \quad (17)$$

$$h_{S2b}(m) \leq \max\{h(m_a) + 2, h\left(\frac{m_a}{2}\right) + L\} + h(m_L) + 1 \quad (18)$$

Output The resulting drawing is a flat orthogonal box drawing. There are no additional columns for the vertices, every vertex contains an incident vertical edge in the base case. Since no bend is created at any time, the box drawing is in fact a flat visibility representation. It was shown that the height of the drawing is bound by $\mathcal{O}(\sqrt{n})$ and there are maximal as many columns as there are edges. The resulting drawing is in area $\mathcal{O}(\frac{n}{\epsilon})$.

Runtime Since in this recursive algorithm every edge is touched exactly once, the total runtime lies in $\mathcal{O}(m + n)$. [2]: Page 10]

In the resulting flat orthogonal drawing without any bends / visibility representation with $\mathcal{O}(n)$ width and $\mathcal{O}(\sqrt{n})$ height, the longest edge lies in $\mathcal{O}(n)$ since edges are horizontal or vertical line segments. The shortest edge values 2 UL in the base case.

To transfer a flat orthogonal drawing to a polyline drawing, empty grid lines are inserted until every edge length is at least two. For a box of a vertex v , replaced the box by an arbitrary grid point and insert a bend for each edge connected to v for rerouting. For each vertex, a bend might be inserted, resulting in a polyline drawing with up to two bends per edge.

Lemma 6. *Every maximal series-parallel graph admits a polyline drawing with two bends per edge and an edge-length ratio of $\mathcal{O}(n + \sqrt{n}) = \mathcal{O}(n)$.*

Proof. In the transition, every box of vertex in the orthogonal box drawing is substituted with a dot on the grid, therefore two bends per edge. A vertex box has width at most $\mathcal{O}(n)$, like the total width. The longest vertical edge lies in $\mathcal{O}(\sqrt{n})$, is horizontally rerouted with a line segment in $\mathcal{O}(n)$. The shortest edge stays in $\mathcal{O}(1)$, therefore the total edge-length ratio lies in $\mathcal{O}(n)$. \square

In the next section, a modification of the three cases of the recursion will lead to a constant ratio while the area consumption is slightly increased.

5.3 The modification

The modification alters line segments for every case of recursion while maintaining the invariant. All line segments will be longer than a variable noted l_{\min} . The behaviour of the height for every recursion is of interest.

Invariant The invariant will still hold for every case of recursion.

Base case the base case is the edge (s, t) . Simply put s over t and the invariant holds. It must hold that $h(1) \leq l_{\min}$ in order to validate a small edge-length ratio.

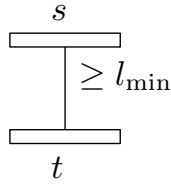


Figure 22: Case P

Height The height of a drawing can be increased in order to preserve the invariant. But, the height increase must value at least l_{\min} . Otherwise, an edge of length one could be placed below s or above t .

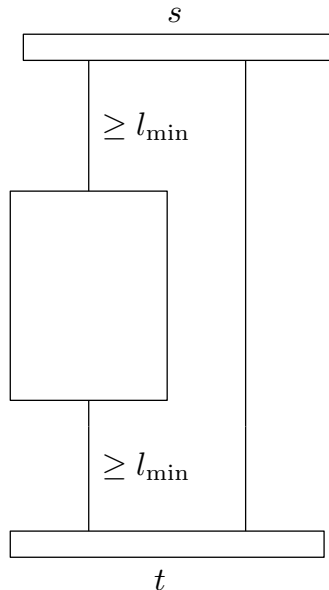


Figure 23: The terminal release must insert edges of length at least l_{\min}

Inductive step • During the parallel composition, $2 \leq k$ subgraphs will first be ordered in their number of edges ($m_i \leq m_{i-1}, i \in [1..k]$) and then recursively drawn. The height is bound by $h_p(m) \leq \max\{h(m-1), h(\frac{m}{2}) + 2 \cdot l_{\min} + 2\}$, since the terminal releases must still maintain a minimum length of l_{\min} .

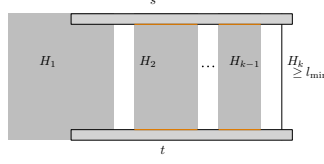


Figure 24: Case P in the modification. Where the orange line segments are placed, there are potential terminal releases which increase the height at least by l_{\min}

- For the serial composition, two subgraphs H_a, H_b with terminals s, x and x, t are recursively drawn (in parallel). In a trivial case, one of the subgraphs is an edge and $h(m) \leq h(m-1)$. Here, the vertical line segment gets a bend to be elongated up to $2 \cdot h_{S_1}(m)$. In the case S2a, the bend is inserted in the same way. Note, that possible terminal releases add the minimum length twice to $h_{S2a}(m)$. In the case S2b, no bend is introduced, but there are up to four terminal release induced height increases by l_{\min} .

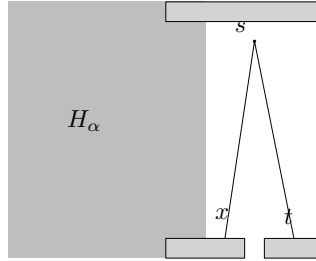


Figure 25: Case S1 gets a bend so the minimum length restriction still holds

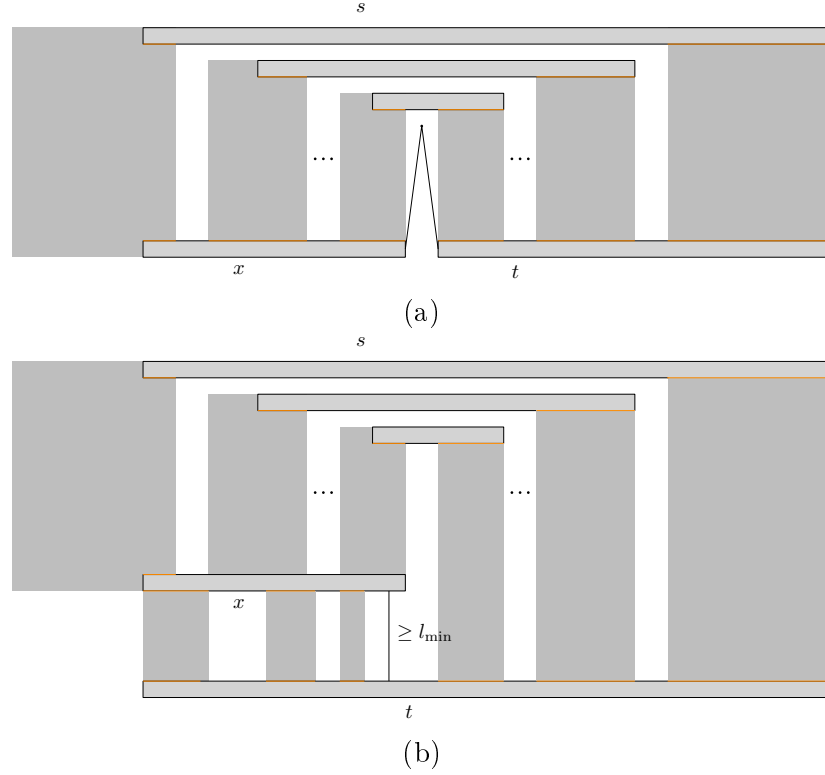


Figure 26: Cases S2a and b also get a bend where necessary and inherit potential terminal release height increases by l_{\min} , analogously to case P in Figure 24

The height is bound by:

$$h_{S2a}(m) \leq \max\{h(m-1), h\left(\frac{m}{2}\right) + L + l_{\min} + 1\} \quad (19)$$

$$h_{S2b}(m) \leq \max\{h(m_a) + 2, h\left(\frac{m_a}{2}\right) + L\} + h(m_L) + 1 + 4 \cdot l_{\min} + 4 \quad (20)$$

Output

Runtime Since in this recursive algorithm every edge is touched exactly once, the total runtime lies in $\mathcal{O}(m + n)$. [2]: Page 10]

The resulting drawing is a box drawing with up to one bend per edge. It was shown that the height of the drawing is bound by $\mathcal{O}(\sqrt{n} + l_{\min})$. As before, in order to get a poly-line drawing, insert empty grid lines so that every edge is at least of length l_{\min} and substitute a box with a bend and a suiting line segment. The resulting poly-line drawing will inherit three bends per edge and will fit on an area of $\mathcal{O}(n) \times \mathcal{O}(\sqrt{n} + l_{\min})$.

Theorem 5. *Every 2-tree G admits a poly-line drawing on $\mathcal{O}(n^2)$ area with an edge-length ratio of $\mathcal{O}(1)$, using three bends per edge.*

Proof. Let $l_{\min} := n$. After applying the modified drawing algorithm by Biedl, the resulting box drawing inherits a height of $\mathcal{O}(\sqrt{n} + n) = \mathcal{O}(n)$. The width still remains at $\mathcal{O}(n)$. There is up to one bend per edge. After the transition to a poly-line drawing, boxes are substituted with line segments, bends and the corresponding vertex and the area bound still holds. To calculate an upper bound for the length of the longest possible edge, we recall that a 2-tree can have at most $2n - 3$ edges and the following bounds hold for the resulting drawing:

- Analogously to the original upper bound, after the modification the height is bound by:

$$h_{mod}(m) \leq 10 \cdot \sqrt{2n - 3} + 4n + 4 \quad (21)$$

- An edge corresponds to up to three columns after the modification since one bend is introduced for vertical aligned edges. The amount of the original width is therefore tripled and can be bound by $6n - 9$.

For the ratio, it holds:

$$r = \frac{l_{\max}}{l_{\min}} \leq \frac{h_{mod}(m) + 2 \cdot w_{mod}(m)}{n} \leq 26 - \frac{14}{n} \leq 26 \in \mathcal{O}(1) \quad (22)$$

□

6 Other Results

6.1 k -nary trees

Lemma 7. *The depth of a complete k -nary tree d values $\lfloor \log_k((k-1)n) \rfloor$*

Proof.

$$n = \sum_{i=0}^d k^i = \frac{k^{d+1} - 1}{k - 1} \quad (23)$$

$$\Leftrightarrow d = \log_k((k-1)n + 1) - 1 = \lfloor \log_k((k-1)n) \rfloor \quad (24)$$

□

Theorem 6. *Every k -nary tree admits a polyline drawing with a nearly optimal ratio, apart from a rounding error ε on area $\mathcal{O}(n^2 \log n)$, allowing one bend per edge.*

Proof. Generally, the resulting drawing will be constructed from top to bottom. The root vertex lies on top, the leaves lie on the bottom. Let $l := k^d + 1$. From the root vertex, place one vertex k^d UL to the left and one unit down. Place another child k^d UL to the right and one unit down. For the remaining $k - 2$ children, place bends inbetween those outer children equidistantly. Connect the two children and the bend points with the root with a line segment. There, the rounding error takes effect and is bound by 1. Place the remaining children below the respective bends so that the sum of the line segment equals l , apart from the rounding error. Between two children, there are $\frac{2k^d}{k-1} - 2$ free columns for the remaining drawing.

Further iterate over the depth. For depth i , place the outermost children $k^{d-(i-1)}$ aside and one unit down and the $k - 2$ bend points in between equidistantly. From the preceding depth, there are $\frac{2k^{d-(i-2)}}{k-1} - 2$ free columns between two children of depth i placed next to each other. Since $\frac{k^{d-(i-1)}}{k-1} \leq \frac{k^{d-(i-2)}}{k-1}$, the area for depth $i+1$ is guaranteed. The total width of the drawing is bound by $2 \cdot \sum_{i=0}^d k^i \in \mathcal{O}(n)$, and the total height is bound by $d \cdot l = d \cdot (k^d + 1) \in \mathcal{O}(n \log n)$. Every edge has length of approximately l , apart from a rounding error derived by the diagonals with one bend per edge and the total area consumption values $\mathcal{O}(n^2 \log n)$. □

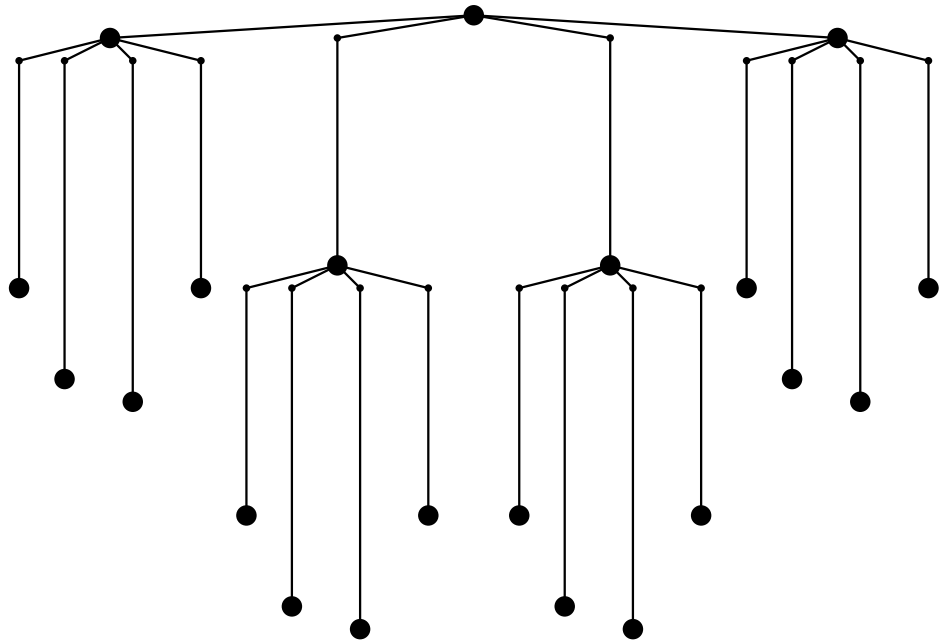


Figure 27: Complete 4-ary tree with depth 2

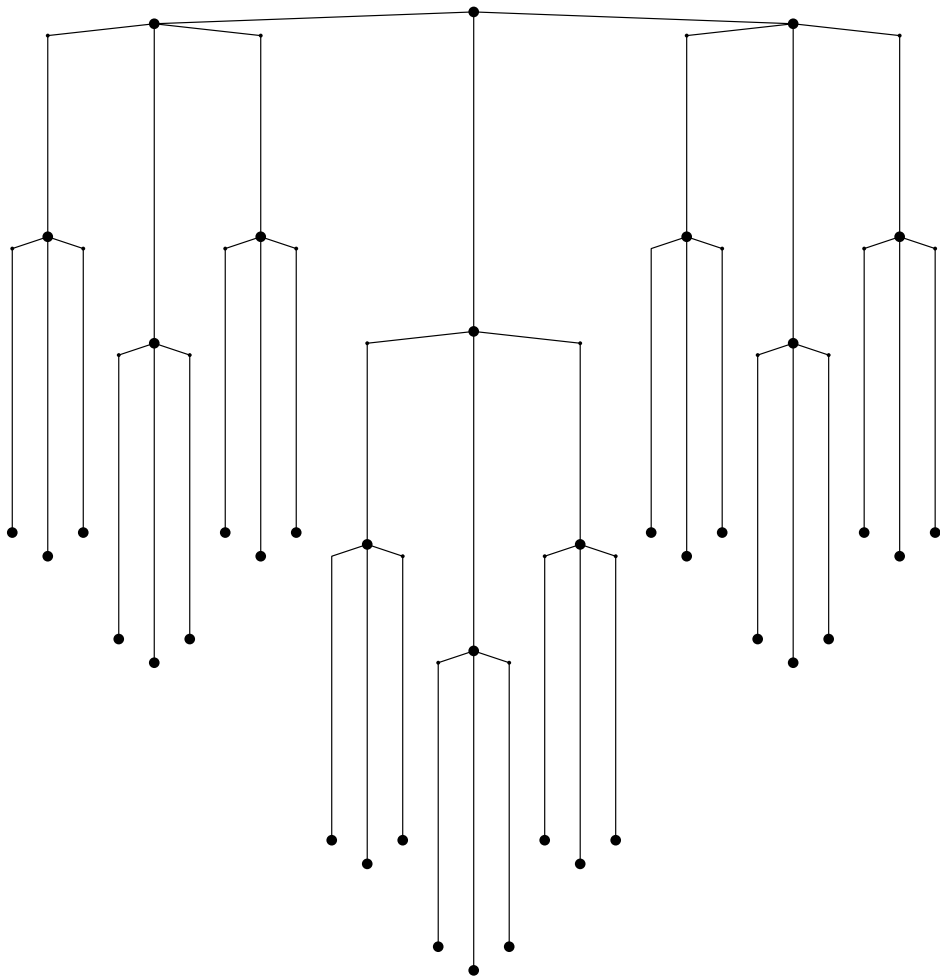


Figure 28: Complete 3-ary tree with depth 3

6.2 Certain family of 3-trees

Recall a k -tree, recursively defined. A 3-tree starts with a K_4 and every vertex inserted is connected to exactly three neighbours, creating a 4-clique.

Theorem 7. *There exists a family of 3-trees which obtain a polyline drawing with a constant edge-length ratio, allowing one bend per edge on area $\mathcal{O}(n^2)$.*

Proof. Let G be a graph consisting of $\frac{n}{3}$ so-called “nested” triangles, meaning that every triangle either encloses or is enclosed by another triangle. Let t_1 be the outermost triangle and t_{i+1} inside $t_i, i = 1, \dots, \frac{n}{3}$. The vertices defining t_i are called A_i, B_i, C_i . For the orientation, A_i will be on the bottom left corner, B_i on the bottom right corner and C_i on the top corner. The edges between the triangles are defined as:

$$(A_i, A_{i+1}), (A_i, B_{i+1}), (A_i, C_{i+1}) \quad (25)$$

$$(B_i, B_{i+1}) \quad (26)$$

$$(C_i, C_{i+1}), (C_i, B_{i+1}) \quad (27)$$

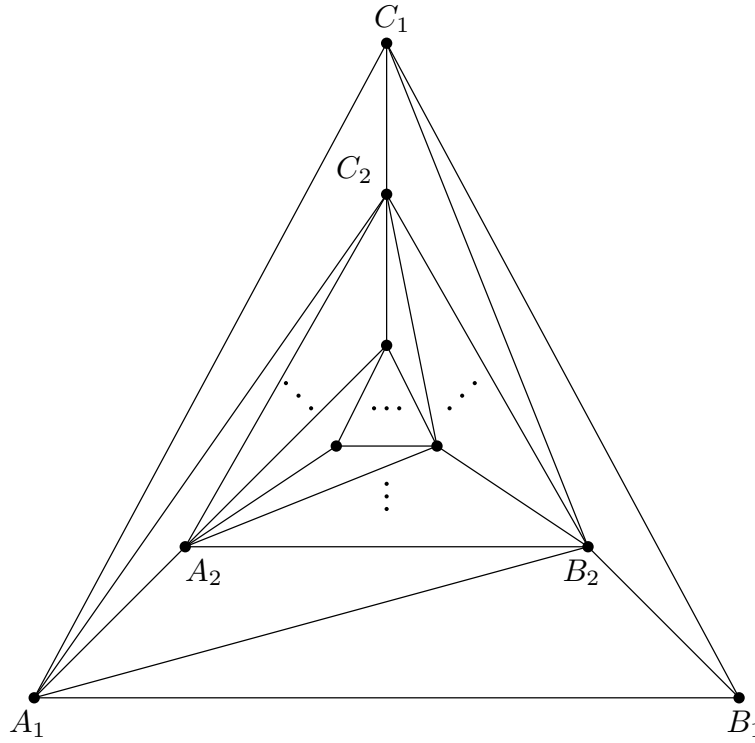


Figure 29: 3-tree with nested triangles

The connections between the triangles t_i, t_{i+1} fulfill the 3-tree property. ([6], Lemma 12, Page 23). At first, draw the triangles with a spacing of three between two consecutive triangles. This ensures two rows or columns of available bend points inbetween. Placing bends will lengthen the connections between two successive triangles. The bend points are placed as illustrated in the following figure:

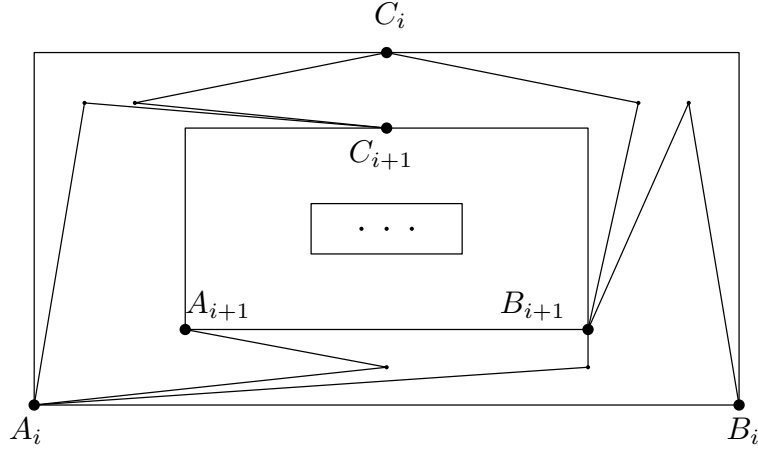


Figure 30: Polyline drawing scheme with one bend per edge

Then, the length of every connecting edge between t_i and t_{i+1} is greater than the shortest edge of t_{i+1} . The total shortest edge lies in $t_{\frac{n}{3}}$. The total longest edge lies in t_1 . Let the length of the edges of $t_{\frac{n}{3}}$ value n . Then, the length of the edges of t_1 value $n + 2 \cdot 3 \left(\frac{n}{3} - 1\right) = 3n - 6$. The ratio is bound by 3, the spacing value. The output is a polyline drawing with one bend per edge on a grid of area $(3n - 6) \times (3n - 6)$. \square

6.3 Outerplanar graphs

Outerplanar graphs are a family of series-parallel graphs. The result of 2-trees can be applied directly to outerplanar graphs. The following theorem describes the edge-length ratio in a poly-line drawing while maintaining that every vertex is placed on the outface.

Theorem 8. *Every outerplanar graph G obtains a poly-line drawing with an edge-length ratio of $\mathcal{O}(1)$ with up to four bends per edge in $\mathcal{O}(n^2)$ area.*

Proof. Since outerplanar graphs are series-parallel graphs, this proof is similar to the proof of Theorem 5. By Biedl, it was proven, that only the base case, the parallel case and the serial case S1 and S2a take place ([2], Page 15). All the cases guarantee that the vertices lie on the outer face.

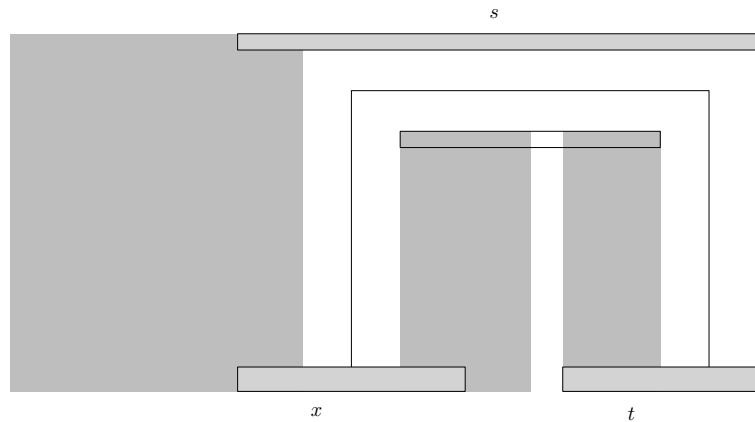


Figure 31: Case S2a is the only case for outerplanar graphs

With the modification used in Theorem 5, the height can be bound by:

$$h_{mod}(m) \leq 3 \log m + 2 + l_{\min} + 1 \leq 3(\log 2 + 1) + 3 \cdot \log n + n \in \mathcal{O}(n) \quad (28)$$

The longest edge can occur in case S2a, with two bends in the box drawing. In the polyline drawing, two additional bends are introduced for vertex v, t . Therefore, the longest edge is bound by:

$$l_{\max} \leq 2(w_{\text{mod}}(m) + h_{\text{mod}}(m)) = 6 \cdot \log 2 + 6 \log n + 14n - 2 \quad (29)$$

Then, it holds for the ratio:

$$r \leq \frac{20n}{n} = 20 \in \mathcal{O}(1) \quad (30)$$

□

7 Future Work

7.1 Tweaking the postprocessing algorithm

A refinement of n^2 and using $\frac{n}{2}$ line segments as long as possible successfully overcomes the hurdle of leveling the edge lengths. But, as seen in the example in Section 4.8, the lengths of the elongated edges might get longer than the original longest edge. The trend is that almost all edges will be longer than the longest edge. This is not desirable, since this worsens the ratio again by re-allocating the longest edge of the modified drawing. So, the question is how to adjust to the original longest edge without getting longer. Either, by not using the longest possible line segment insertions by default, or by using less bends in total. Maybe, a refinement by $o(n^2)$ suffices, for example a refinement by n or $n \log n$. It does prove that the area suffices for every edge to be elongated in and shows a first approach. But, considering the trend of getting the edges too long, the postprocessing algorithm presented in this report is not applicable in this state.

7.2 Further minimizations

We observed that a linear amount of bends is mandatory for the edge-length ratio to be bound by a constant. It would be of interest to further minimize either the number of bends without losing any upper bounds or the area the whole drawing takes place in. This would be achieved by further investigating and gauging the postprocessing algorithm.

7.3 New drawing algorithm approach

Take a look at the certain 3-tree family presented in Section 6.2. On one hand, the postprocessing algorithm proved that a refinement of at least n and zig-zag elongations result in a constant ratio. But, this family of 3-trees (maximal planar) are drawable with a constant ratio and still remain in area $\mathcal{O}(n^2)$. This exclusive nestedness of this graph family might give a hint that a certain property of graphs determine the upper bound for the area while guaranteeing a constant ratio. This might be something interesting to further investigate.

7.4 Implementation

When the postprocessing algorithm is remastered, an implementation of this algorithm will be of interest. The readability of a drawing will suffice with a high amount of zig-zag elongations. But, when the number of bends or the length of the inserted line segments is minimized further, the readability will be an aspect of interest when the implementation is there.

References

- [1] Carlos Alegria et al. “Planar Straight-line Realizations of 2-Trees with Prescribed Edge Lengths”. In: *CoRR* abs/2108.12628 (2021). arXiv: 2108.12628. URL: <https://arxiv.org/abs/2108.12628>.
- [2] Therese C. Biedl. “Small Drawings of Outerplanar Graphs, Series-Parallel Graphs, and Other Planar Graphs”. In: *Discret. Comput. Geom.* 45.1 (2011), pp. 141–160. DOI: 10.1007/s00454-010-9310-z. URL: <https://doi.org/10.1007/s00454-010-9310-z>.
- [3] Thomas H. Cormen et al. *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009. ISBN: 978-0-262-03384-8. URL: <http://mitpress.mit.edu/books/introduction-algorithms>.
- [4] 29th International Symposium on Graph Drawing and Network Visualization. *Planar polyline edge-length ratio on a fixed grid*. 2021. URL: <http://mozart.diei.unipg.it/gdcontest/contest2021/index.php?id=live-challenge>.
- [5] ICS. *Walter Schnyder’s Grid-Embedding Algorithm*. 2021. URL: <https://www.ics.uci.edu/~eppstein/gina/schnyder/>.
- [6] Debajyoti Mondal et al. “Minimum-Area Drawings of Plane 3-Trees”. In: *J. Graph Algorithms Appl.* 15.2 (2011), pp. 177–204. DOI: 10.7155/jgaa.00222. URL: <https://doi.org/10.7155/jgaa.00222>.
- [7] SureSolv. *How three medians meeting at centroid divides triangle area into 3 and 6 equal parts*. 2021. URL: <https://suresolv.com/basic-maths/basic-and-rich-geometry-concepts-part-6-proof-triangle-area-medians>.