# Drawing planar graphs with prescribed face areas

by

Lesvia Elena Ruiz Velázquez

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2010

© Lesvia Elena Ruiz Velázquez 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## Abstract

This thesis deals with planar drawings of planar graphs such that each interior face has a prescribed area.

Our work is divided into two main sections. The first one deals with straight-line drawings and the second one with orthogonal drawings.

For straight-line drawings, it was known that such drawings exist for all planar graphs with maximum degree 3. We show here that such drawings exist for all planar partial 3-trees, i.e., subgraphs of a triangulated planar graph obtained by repeatedly inserting a vertex in one triangle and connecting it to all vertices of the triangle. Moreover, vertices have rational coordinates if the face areas are rational, and we can bound the resolution.

For orthogonal drawings, we give an algorithm to draw triconnected planar graphs with maximum degree 3. This algorithm produces a drawing with at most 8 bends per face and 4 bends per edge, which improves the previous known result of 34 bends per face. Both vertices and bends have rational coordinates if the face areas are rational.

**Acknowledgements**

I would first like to thank my supervisor, Therese Biedl. Thank you for all your support, comments, ideas, for all the time dedicated and the interest shown in this project, for reading this thesis one hundred times, for all the enjoyable discussions and for guiding me and helping me to learn so many things during my stay at Waterloo. And most importantly, thanks for sharing with me your passion for research and for graph drawing.

I would also like to thank my committee members, Anna Lubiw and Timothy M. Chan, for taking the time to read this thesis and provide feedback.

Finally, thank you very much to all my family and friends for all your support and for all the fun during these months.

## Dedication

To my family: mom, dad and Karla, you are the best of my life.

# Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Graph drawing

Graph drawing algorithms are concerned with automatically producing graph drawings that are easy to read. Graphs can be used to represent objects or components and the relationships among them, and a graph drawing is intended to be helpful to visualize those relationships. For example, tools in software engineering use graphs to show relationships between modules in a large program; tools to design, emulate and troubleshoot networks use graphs to represent hosts and the interconnections among them.

A planar graph is a graph that can be drawn without crossings. Fáry, Stein and Wagner [20, 37, 42] proved independently that every planar graph has a planar drawing such that all edges are drawn as straight-line segments. This class of graphs is important because both edge crossings and a high number of bends can make a drawing difficult to understand as we can see in Figure 1.1. In this thesis, we will only consider planar drawings of planar graphs.

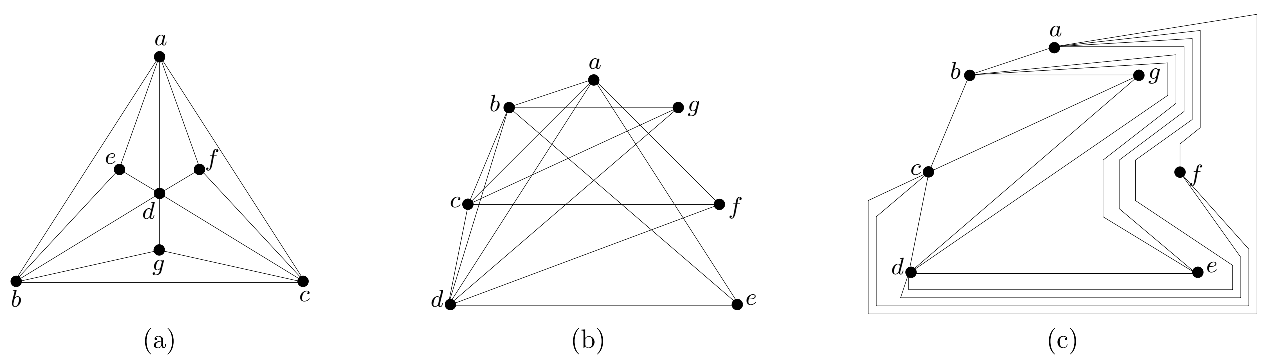



Figure 1.1: (a) A planar straight-line drawing of a graph $G$. (b) A non planar straight-line drawing of $G$. (c) A planar but not straight-line drawing of $G$.

Sometimes additional constraints are imposed on the drawings of planar graphs. The most common one is to have integer coordinates while keeping the area small; it was shown in 1990 that any $n$-vertex planar graph has a planar straight-line drawing with integer coordinates and $O(n^2)$ area [13, 36]. Another restriction might be to ask whether all edge lengths are integral; this exists if the graph is 3-regular [23], but is open in general. Also, one could try to maximize the smallest angle between two edges incident to the same vertex, or minimize the aspect ratio of the drawing, which is the ratio of the length of the longest to the shortest side of the smallest rectangle covering the drawing. The importance of all these constraints is noticeable when graphs are displayed on a computer screen. See [15] for the description of a variety of constraints that are commonly studied.

Some constraints conflict with each other. For example if small area is a major concern, other restrictions such as straight lines can be relaxed by drawing edges as polygonal chains, where both vertices and edge bends should have integer coordinates. Such drawings are called *polyline drawings*. For such drawings the goal is to keep the number of bends small, so that readability is not sacrificed. A special case of polyline drawings are orthogonal drawings, where edges are drawn as chains of vertical and horizontal segments. These drawings only exist if the vertices have at most 4 incident edges. Orthogonal drawings have applications in circuit and floor layouts, database, entity-relationship and data flow diagrams, and others. Many authors have worked on obtaining lower bounds on the number of bends necessary to make orthogonal drawings of planar graphs [2,4,5,29,38]. Rahman, Nishizeki and Naznin [32] studied conditions for a plane graph with maximum degree 3 to have an orthogonal drawing without bends. Ungar [40] proved that every plane, cubic cyclically 4-edge-connected graph has a rectangular representation.

## 1.2 Planar drawings with fixed face areas

In this thesis, we consider drawings with prescribed face areas, which means that the area of each interior face of the graph is given and the resulting drawing must satisfy this area requirement. We will give formal definitions of this in Section 1.4.

Ringel [34] raised the question of whether all planar graphs have straight-line drawings where all areas are prescribed and showed that such drawings do not exist for all planar graphs. See also Section 2.6. He also conjectured that they do exist for planar graphs with maximum degree 3. Thomassen [39] showed that Ringel's conjecture is true: Every planar graph with maximum degree three, for any given areas of interior faces, has a planar straight-line drawing such that the areas are respected.

Thomassen's work is the main motivation for our work. We asked whether the same kind of drawing is also possible while having rational coordinates, assuming all face areas are rationals.

**Conjecture 1** *Every planar graph with maximum degree 3, for any given rational areas of interior faces, has a planar straight-line drawing such that the areas are respected and the vertex coordinates are rational.*

This main question remains open, but we provide some partial results towards it in this thesis.

Thomassen's proof does not yield rational coordinates. His proof works by often contracting edges; he does not explain in detail how to "undo" such contractions in the final drawing without changing face areas. It is even less clear why rational coordinates could be obtained. Another obstacle is that even with contracted edges, the coordinates of some vertices are not rationals. See Section 2.6 for more details.

**Our results**

In Chapter 2, we will show that at least part of Thomassens's proof does yield rational coordinates. One of his main ingredients is his Lemma 4.1, which yields drawings for a subclass of graphs. We studied this subclass and realized that it is a subset of planar partial 3-trees (details will be given in Section 2.5.) Hence we studied how to draw planar partial 3-trees.

We show that every planar partial 3-tree, for any given set of areas for interior faces, admits a planar straight-line drawing that respects the face areas. It is quite easy to show that such drawings exist; our main contribution is that the coordinates are rational (presuming the face areas are). Furthermore, we can bound the resolution in terms of the number of vertices (albeit not polynomially). The time to find such a drawing is $O(n)$.

## 1.3 Cartograms

Ringel and Thomassen studied drawings with given interior face areas out of graph theoretic interest. However, such drawings are actually of high interest in the application area of cartograms, where faces (i.e., countries in a map) should be proportional to some property of the country, such as population, number of cases of a certain disease, amount of tons of carbon emissions, Internet users, etc. Cartograms are a useful tool to visualize such properties efficiently. Cartograms can also be used in combination with other tools to show two different properties. For example, one could use a cartogram where the area of each region corresponds to population, and where different tones of colors correspond to the number of cases of a disease [22]. This combination makes sense because the number of cases of a disease will probably be higher in more populated areas. Figures 1.2 and 1.3 are examples of cartograms that show the number of Internet users in 1990 and 2002.

There are a number of qualities to determine whether a cartogram is good. Two major ones are to preserve the *correct adjacencies* and to draw the *correct areas*.

Cartograms have rarely been done with straight-line drawings, but they sometimes use orthogonal drawings. Orthogonal drawings with small number of edges have the advantage that their area can be easily estimated by visual inspection. Figure 1.4 shows an orthogonal cartogram that show two variables at the same time, the area of each state represents its

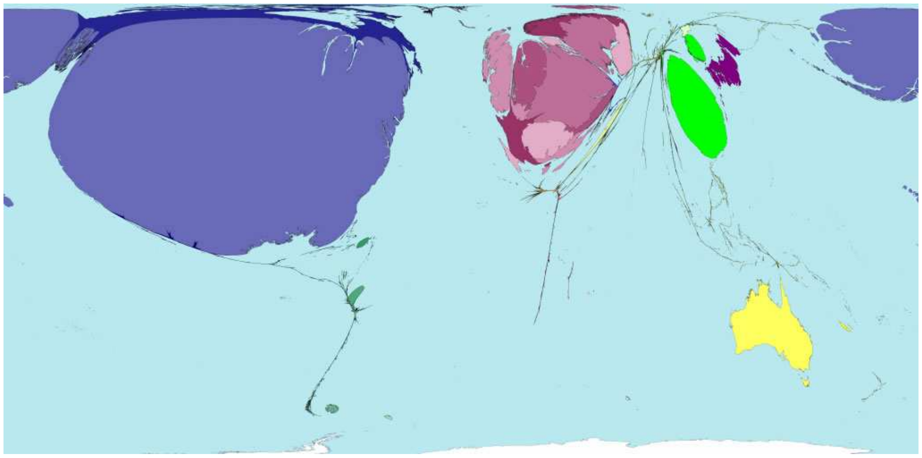


Figure 1.2: Internet users in 1990. ©Copyright 2006 SASI Group (University of Sheffield) and Mark Newman (University of Michigan).

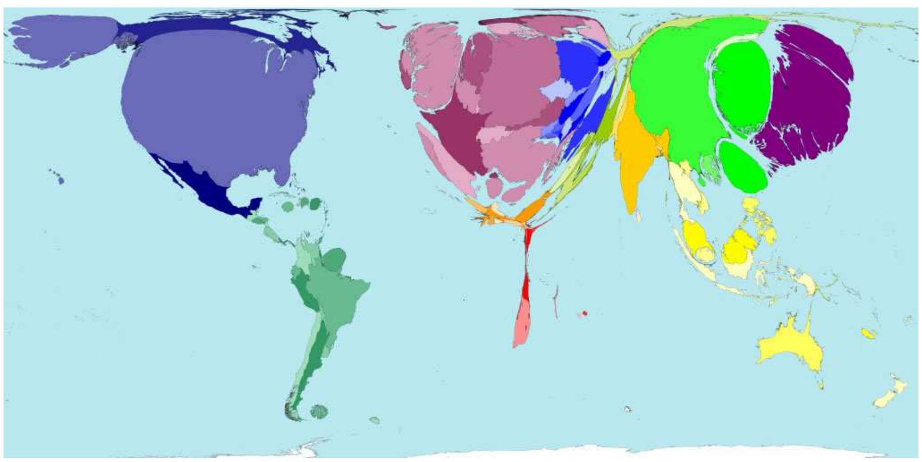


Figure 1.3: Internet users in 2002. ©Copyright 2006 SASI Group (University of Sheffield) and Mark Newman (University of Michigan).

number of electoral votes and the color represents the candidate who was more likely to win in each state.

Raisz introduced rectangular cartograms [33], which are rectangular drawings, i.e., every face (including the outerface) is a rectangle, with prescribed areas for interior faces. Not every graph admits a rectangular cartogram. Figure 1.5 shows an example where, in order to obtain the correct areas, some adjacencies would need to be changed. Even if face areas are ignored, not every plane triangulated graph admits a rectangular drawing. However, some classes of graphs do. Kant and He [27] and Ungar [40] proved independently that graphs with a 4-connected triangulated dual have a rectangular drawing. There are several ways to relax the requirements of rectangular cartograms, which include accepting error on the adjacencies or on the areas or allow more bends per face. In 2004, van Kreveld and Speckmann gave an algorithm to draw rectangular cartograms, where there can be a small error both on the

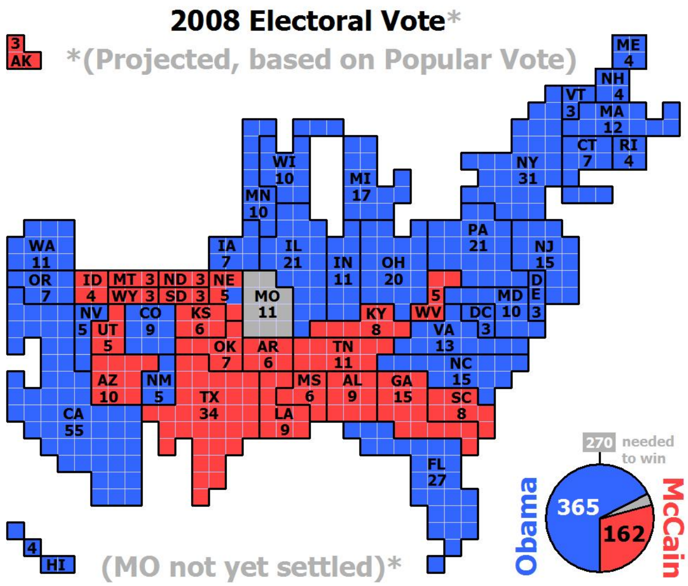



Figure 1.4: Cartogram of the projected 2008 Electoral Vote for US President (based on popular vote) with each square representing one electoral vote. ©Creative Commons.

adjacencies and on the areas [41]. In 2005, de Berg, Mumford, and Speckmann proved that any triconnected planar graph with maximum degree 3 and prescribed face areas can be drawn orthogonally with at most 60 bends per face [11]. The next year, the same authors improved their approach to make it easier to implement in polynomial time and included heuristics that reduce the number of bends in some cases [12]. In 2007, Kawaguchi and Nagamochi took the bound of 60 bends per face down to 34 [28]. In 2009, Rahman, Miura and Nishizeki, gave an algorithm to draw a special class of graphs called good slicing graphs with at most 8 bends per face [31]. Eppstein, Mumford, Speckmann and Verbeek found a necessary and sufficient condition for a rectangular layout (a partition of a rectangle into finitely many interior-disjoint rectangles) to be drawn with prescribed interior rectangle areas for any given areas [19].

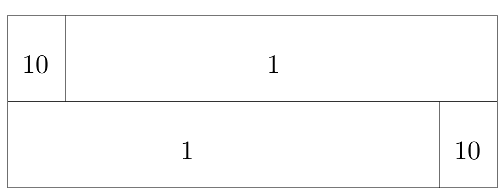



Figure 1.5: Example of a graph that doest not admit a rectangular cartogram.

### Our results

In Chapter 3 we improve the work by de Berg, Mumford, and Speckmann and Kawaguchi and Nagamochi. They showed that any cartogram with maximum degree 3 has an orthogonal drawing with at most 60 and 34 bends per face, respectively. We prove that the number of bends per face can in fact be reduced to 8. We also improve the work done by Rahman, Miura and Nishizeki, since they drew cartograms with at most 8 bends per face, but for a smaller class of graphs. We can also analyze other parameters of the drawing: every edge has at most 4 bends, and the coordinates can be made rational (albeit not polynomial.) Such drawings can be found in $O(n \log n)$ time.

Our approach is substantially different from the one in [11]. They obtained drawings by modifying the graph until it has a rectangular cartogram, and then "fixing up" adjacencies by adding thin connectors where needed. Our approach works throughout with the original graph and produces a drawing directly. We find this a more natural approach, and it also eases bounding the number of bends. Our algorithm is based on using Kant's canonical ordering [25], a useful tool for many graph drawing algorithms. However, it does not use the ordering directly, but instead uses it to split the graph"vertically" into two smaller graphs (or one graph and a face); this decomposition to our knowledge was not known before and may be of independent interest in the graph drawing community.

In the last chapter we conclude and give suggestions for future work.

## 1.4 Definitions

The following are general definitions in graph theory and will be helpful throughout this thesis.

A graph $G = (V, E)$ consists of a set $V$ of $n$ vertices and a set $E$ of $m$ edges, where an edge is an unordered pair of vertices.

Two vertices $u$ and $v$ connected by an edge $e$ are *adjacent* to each other, and *incident* to $e$. The *neighbors* of $v$ are its adjacent vertices. The *degree* of $v$ is the number of its neighbors and is denoted by $deg(v)$.

A *loop* is an edge that connects a vertex to itself. A graph has *multiple* edges if there is more than one edge between the same pair of vertices. A graph is *simple* if it has no loops or multiple edges. In this thesis, all graphs will be assumed to be simple.

A graph is *regular* if the degree of all its vertices is the same. In particular it is called *3-regular* or *cubic* if the degree of all its vertices is three.

A graph $G' = (V', E')$ is a *subgraph* of $G$ if $V'$ and $E'$ are subsets of $V$ and $E$, respectively, and $E' \subseteq V' \times V'$. An *induced* subgraph $G'$ of $G$ contains all the edges in $E$ connecting pairs of vertices in $V'$.

A graph is *complete* if any two vertices in $V$ are adjacent. A complete graph with $n$ vertices is denoted by $K_n$. In particular, the complete graph with four vertices is $K_4$.

A graph is *directed* (sometimes also called *oriented*) if the edges are ordered pairs of vertices. The *directed edge* $(u, v)$ is an *outgoing edge* of $u$ and an *incoming edge* of $v$ and it is drawn as an arrow from $u$ to $v$. The *indegree* (*outdegree*) of a vertex is the number of its incoming (outgoing) edges. A *partially oriented graph* has some directed, and some undirected edges.

A *path* in a graph $G$ is a sequence $(v_1, v_2, ..., v_h)$ of distinct vertices of $G$, such that $(v_i, v_{i+1}) \in E$ for $1 \leq i \leq h-1$. A path is a *cycle* if $(v_h, v_1) \in E$. A *directed path* in a directed or partially oriented graph is a sequence $(v_1, v_2, ..., v_h)$ of distinct vertices of $G$, such that $(v_i, v_{i+1})$ is a directed edge. For a directed path, the vertices $v_1$ and $v_h$ are called *source* and *sink*, respectively. A directed path is a *directed cycle* if $(v_h, v_1)$ is a directed edge. A directed graph is acyclic if it has no directed cycles.

In this thesis, a given graph will always assumed to be undirected, but we will sometimes impose a direction onto some of the edges.

For this thesis, a *drawing* of a graph is an assignment of vertices to points and edges to curves in the 2D Euclidean plane. A drawing is *planar* if the curves of edges are disjoint except at the endpoints. A graph is planar if it has a planar drawing. In this thesis, we only study planar drawings of planar graphs, so assume for the remaining definitions that $G$ is a planar simple undirected graph. A planar drawing of $G$ splits the plane into connected pieces; the unbounded piece is called the *outerface*, all other pieces are called *interior faces*. Two faces are *adjacent* if they share an edge.

A *combinatorial embedding* of a graph is a clockwise ordering of edges around each vertex and choice of the outerface.

An undirected graph is *connected* if there is a path between each pair of vertices in $V$. A vertex is a *cut-vertex* if removing it and its adjacent edges makes $G$ disconnected. A graph is *biconnected* if it has no cut-vertex. A *cut-pair* is a pair of vertices whose removal makes $G$ disconnected. A graph is *triconnected* if it has no *cut-pair*. A triconnected graph has a single *combinatorial embedding* [43]. We assume that one combinatorial embedding (including the choice of the outerface) has been fixed for all graphs in this thesis.

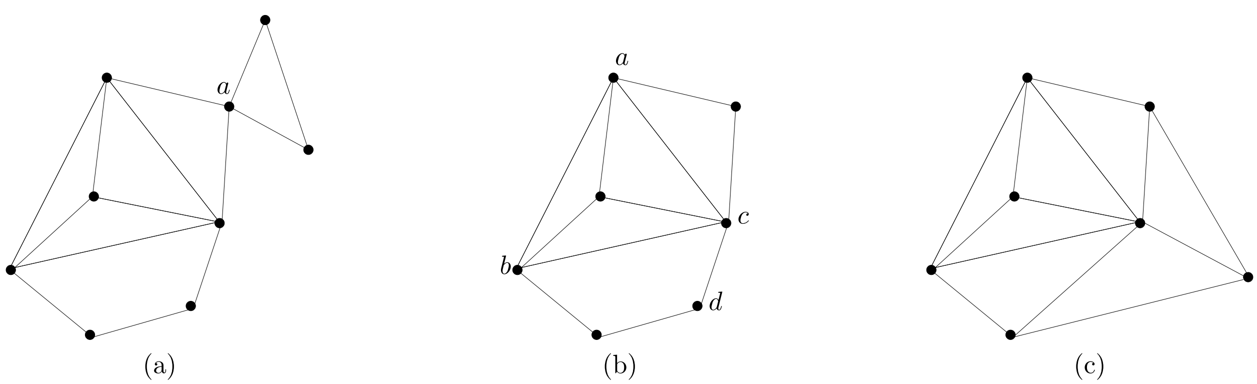



Figure 1.6: (a) A connected graph, where $a$ is a cut-vertex. (b) A biconnected graph where $(a, c), (b, c)$ and $(b, d)$ are cut-pairs. (c) A triconnected graph.

A planar graph is *triangulated* if it is not possible to add any edge without making the graph non-planar. Triangulated graphs are triconnected and have $2n - 4$ faces, which are all triangles, including the outerface. If the outerface is not a triangle, the graph is called a *near-triangulation*.

A drawing such that each edge is represented by a polygonal chain is a *polyline drawing*. A place where an edge switches direction is called a *bend*. A polyline drawing is a *grid drawing* if the vertices and the bends of the edges have integer coordinates. The width and the height of such a drawing are important because they determine the resolution in a display on a computer screen. A drawing that has vertices and bends with rational coordinates can be scaled by the least common denominator to obtain a grid drawing.

There are two common special cases of polyline drawings: straight-line and orthogonal drawings.

A *straight-line drawing* of $G$ is a drawing of $G$ where edges are straight-line segments. As always, we demand that this drawing is planar, i.e., edges are interior-disjoint.

In an *orthogonal drawing* of $G$, edges are paths of horizontal and vertical segments. Since vertices are represented by points, the maximum possible degree of a vertex is four. When making orthogonal drawings, it is important to keep the number of bends small, so that the drawings can be understood easily. See Figure 1.7 for an example of a straight-line drawing and two orthogonal drawings that illustrate how a smaller number of bends improves readability.

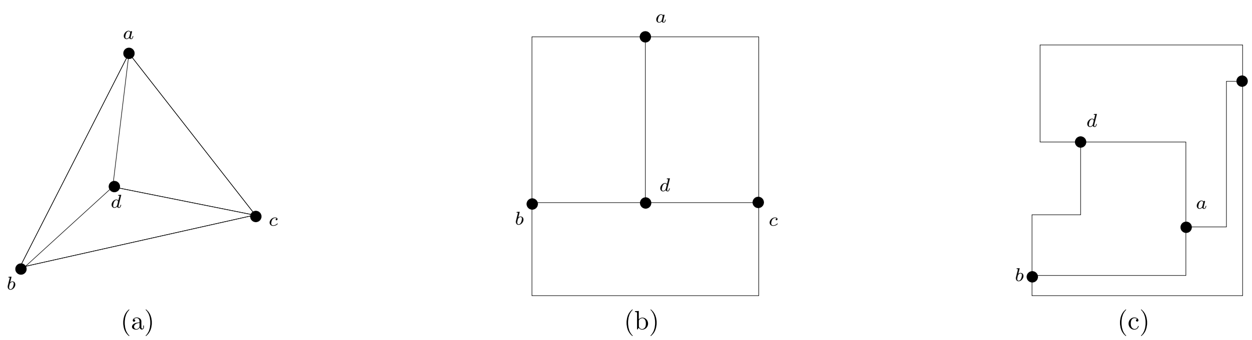



Figure 1.7: (a) A planar straight-line drawing of $K_4$. (b) A planar orthogonal drawing of $K_4$. (c) A planar orthogonal drawing of $K_4$ with many bends.

In this thesis we will study both straight-line and orthogonal drawings with an additional constraint: the areas of the interior faces of $G$ are given.

Let $A$ be a function that assigns non-negative[1] rationals to interior faces of $G$ (irrational face areas could be allowed, but would force irrational coordinates, so we will assume rational face areas throughout.) We say that a planar drawing of $G$ *respects the given face areas* if every interior face $f$ of $G$ is drawn with area $c \cdot A(f)$ for a constant $c$. A drawing of a graph

[1]Normally areas have to be positive, since otherwise edges must overlap. There will be one small exception to this in Section 2.6.

that respects given face areas is invariant under scaling. Thus, any value of the constant is acceptable, as long as it is the same for all faces. If $A \equiv 1$, then the drawing is called an *equifacial drawing.*

# Chapter 2

# Drawing planar partial 3-trees with given face areas

In this chapter, we consider the problem of creating straight-line drawings of graphs with prescribed interior face areas. It was shown by Ringel [34] that such drawings do not exist for all planar graphs, and by Thomassen [39] that they do exist for cubic planar graphs. Our focus is on another class of graphs: planar partial 3-trees. Thomassen proved in his Lemma 4.1, that a subclass of partial 3-tress can be drawn with straight-lines. We show here that every planar partial 3-tree has a straight-line drawing that respects given interior face areas. If the given areas are rational, the algorithm leads to rational coordinates for the vertices. Furthermore, we give bounds on the size of the grid. Most of the results in this chapter were presented at Graph Drawing 2009 [3]. We will start by defining planar partial 3-trees.

## 2.1 Planar partial 3-trees

A graph $G$ is a *$k$-tree* if it has an *elimination order*, i.e., a vertex order $v_1, \dots, v_n$ such that for $i > k$ vertex $v_i$ has exactly $k$ *predecessors*, i.e., earlier neighbours, and they form a clique.

A *partial $k$-tree* is a subgraph of a $k$-tree. Partial $k$-trees are the same as graphs of treewidth at most $k$ (which we will not define precisely here); such graphs have received huge attention in the last few years due to the ability to solve many NP-hard problems in polynomial time on graphs of constant treewidth [1, 6]. Some of these problems include Independent Set, Hamiltonian Circuits, Coloring and TSP. It is known how to determine whether a graph is a partial 3-tree in linear time [7, 35].

Let $G$ be a triangulated graph built up as follows: Start with a triangle $v_1, v_2, v_3$. For $i > 3$, to add vertex $v_i$, pick an interior face $f$ of the graph built so far, and make $v_i$ adjacent to all its vertices. Since $f$ is necessarily a triangle, $v_i$ then has three predecessors and this gives a planar 3-tree. One can show (mentioned for example in [24]) that all planar 3-trees can be built this way. See Figure 2.1 for an example of this construction.

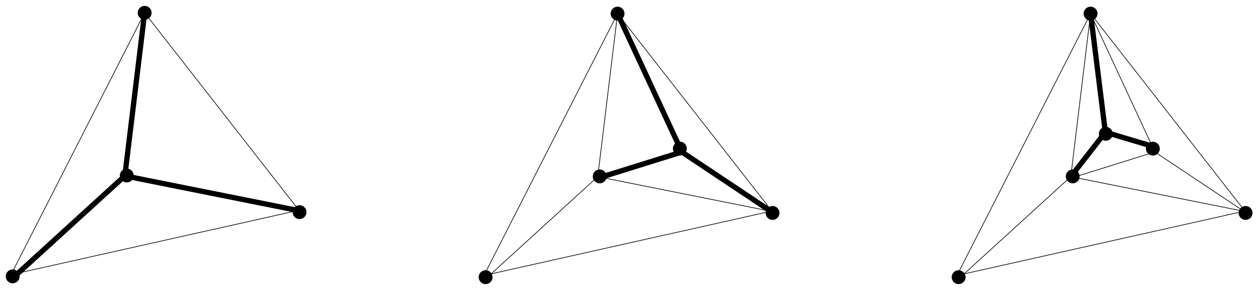


Figure 2.1: A planar 3-tree can be built up by picking an interior face and subdividing it into three triangles by inserting a new vertex in it.

A planar partial 3-tree is a graph $G'$ that is planar and is the subgraph of a 3-tree $G$ (see Figure 2.2.)

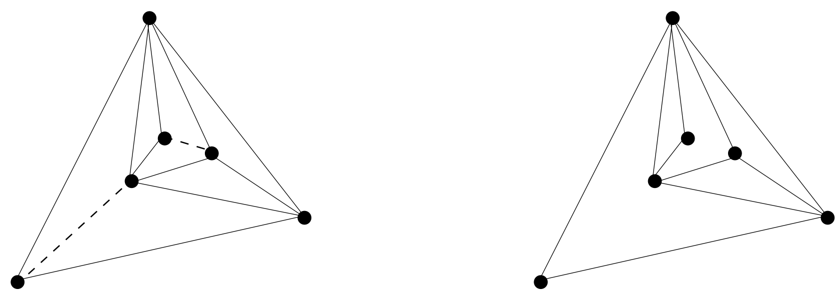
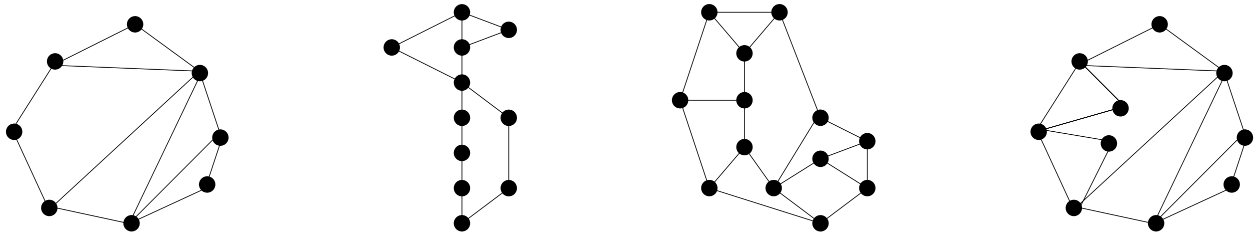


Figure 2.2: A planar partial 3-tree can be obtained from a planar 3-tree by deleting edges.

Planar partial 3-trees include, among others, outerplanar graphs (i.e., planar graphs where all vertices lie on the outerface), series-parallel graphs (graphs with a selected source and sink that can be obtained by joining two series-parallel graphs in series, by merging the sink of one graph to the source of the other, or in parallel, by merging the two sources and the two sinks together), Halin graphs [8] (graphs that can be obtained from a tree with no vertices of degree two, by connecting all the leaves with a cycle) and IO-graphs [18] (graphs that are either outerplanar or removing some independent set of its interior vertices leaves a 2-connected outerplanar graph.) See also Figure 2.3.

## 2.2 Drawing planar partial 3-trees

In this section, we show that every planar partial 3-tree can be drawn with given interior face areas. Since "drawing with prescribed interior face areas" is a property that is closed under taking subgraphs (see also Lemma 3), we mostly focus on drawing planar 3-trees. A vital ingredient is how to draw $K_4$ by placing one point inside a triangle.



(a) (b) (c) (d)

Figure 2.3: (a) An outerplanar graph. (b) A series-parallel graph. (c) A Halin graph. (d) An IO-graph.

**Lemma 1** *Let $T$ be a triangle with area $a$ and vertices $v_0, v_1, v_2$ in counterclockwise order. For any non-negative values $a_0 + a_1 + a_2 = a$, there exists a unique point $v^*$ inside $T$ such that the triangle $\{v_{i+1}, v_{i-1}, v^*\}$ has area $a_i$, for $i = 0, 1, 2$ and addition modulo 3. (See also Figure 2.4.) Moreover, if $a_0, a_1, a_2$ are rational and $v_0, v_1, v_2$ have rational coordinates, then $v^*$ has rational coordinates.*

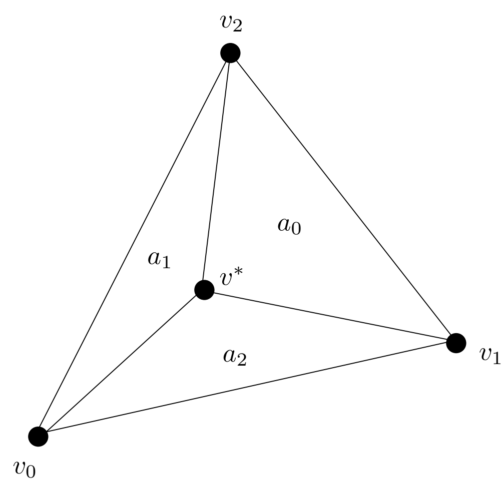



Figure 2.4: It is possible to draw $K_4$ with prescribed interior face areas.

**Proof:** The area of a triangle $\triangle pqr$, with $p, q, r$ in counterclockwise order can be calculated via determinants as follows:

$$2 \cdot area(\triangle pqr) \;=\; det \begin{vmatrix} p.x & p.y & 1 \\ q.x & q.y & 1 \\ r.x & r.y & 1 \end{vmatrix}$$

This formula is called the *signed area formula* because the determinant is positive if the vertices are counterclockwise around the triangle and negative otherwise.

Let $(x_0, y_0)$, $(x_1, y_1)$, $(x_2, y_2)$ be the coordinates of $v_0$, $v_1$, $v_2$, respectively, and use $(x^*, y^*)$ to denote the (unknown) coordinates of $v^*$. The area $a_i$ of the triangle $\{v_{i+1}, v_{i-1}, v^*\}$ must satisfy (for $i = 0, 1, 2$ and addition modulo 3):

$$\begin{aligned}
2 \cdot a_i &= \begin{vmatrix} x_{i+1} & y_{i+1} & 1 \\ x_{i-1} & y_{i-1} & 1 \\ x^* & y^* & 1 \end{vmatrix} \\
&= (x_{i-1} \cdot y^* - x^* \cdot y_{i-1}) - (x_{i-1} \cdot y_{i+1} - x_{i+1} \cdot y_{i-1}) + (x^* \cdot y_{i+1} - x_{i+1} \cdot y^*) \\
&= (x_{i+1} - x^*) \cdot (y_{i-1} - y^*) - (x_{i-1} - x^*) \cdot (y_{i+1} - y^*).
\end{aligned}$$

Multiplying by $(x_i - x^*)$ we obtain:

$$2 \cdot a_i \cdot (x_i - x^*) \;=\; (x_i - x^*) \cdot (x_{i+1} - x^*) \cdot (y_{i-1} - y^*) - (x_i - x^*) \cdot (x_{i-1} - x^*) \cdot (y_{i+1} - y^*).$$

Adding for $i = 0, 1, 2$ yields:

$$\begin{aligned}
\sum_{i=0}^{2} 2 \cdot a_i \cdot (x_i - x^*) &= (x_0 - x^*) \cdot (x_1 - x^*) \cdot (y_2 - y^*) - (x_0 - x^*) \cdot (x_2 - x^*) \cdot (y_1 - y^*) \\
&+ (x_1 - x^*) \cdot (x_2 - x^*) \cdot (y_0 - y^*) - (x_1 - x^*) \cdot (x_0 - x^*) \cdot (y_2 - y^*) \\
&+ (x_2 - x^*) \cdot (x_0 - x^*) \cdot (y_1 - y^*) - (x_2 - x^*) \cdot (x_1 - x^*) \cdot (y_0 - y^*) \\
&= \sum_{i=0}^{2} (y_i - y^*) \cdot [(x_{i+1} - x^*) \cdot (x_{i-1} - x^*) - (x_{i-1} - x^*) \cdot (x_{i+1} - x^*)] \\
&= 0.
\end{aligned}$$

Therefore,

$$\sum_{i=0}^{2} 2 \cdot a_i \cdot x_i \;=\; \sum_{i=0}^{2} 2 \cdot a_i \cdot x^*,$$

which yields the following value for $x^*$:

$$x^* = \frac{a_0 \cdot x_0 + a_1 \cdot x_1 + a_2 \cdot x_2}{a_0 + a_1 + a_2}. \tag{2.1}$$

Similarly for $y^*$, we obtain:

$$y^* = \frac{a_0 \cdot y_0 + a_1 \cdot y_1 + a_2 \cdot y_2}{a_0 + a_1 + a_2}. \tag{2.2}$$

Since $a_i$ is non-negative, the signed-area formula guarantees that $v^*$ lies to the left of the directed segments $v_0v_1$, $v_1v_2$, and $v_2v_0$, and hence inside $T$.

These equations show immediately that if if $x_i, y_i, a_i$ for $i = 0, 1, 2$ are rational, $x^*$ and $y^*$ are also rational[1]. □

Recall that any planar 3-tree $G$ can be built up by repeatedly picking an interior face $f$ and subdividing it into three triangles by inserting a new vertex in $f$. In the following lemma, we will construct a drawing of a planar 3-tree with prescribed interior face areas, by applying Lemma 1 recursively to place each of the vertices of $G$ in its correct position inside $f$. Then, in Lemma 3, we will show how to add and delete some edges appropriately, so that the statement is also true for partial planar 3-trees.

**Lemma 2** *Every planar 3-tree can be drawn respecting prescribed interior face areas such that all coordinates are rational.*

**Proof:** Assume $v_1, \ldots, v_n$ is the vertex-order that defined the planar 3-tree $G$, where $v_1, v_2$ and $v_3$ are vertices on the outerface. We proceed by induction on $n$. The base case is $n = 3$, where this is obvious: place $v_1, v_2, v_3$ in a triangle of desired area (and with rational coordinates since the area is rational.)

If $n \geq 4$, then consider the $K_4$ formed by $v_n$ and its neighbours, $v_h$, $v_i$ and $v_j$. Let $f_1, f_2, f_3$ be the faces incident to $v_n$. In $G' = G - v_n$, the neighbours of $v_n$ form a triangle $T$ that is an interior face (it consists of the union of faces $f_1, f_2, f_3$.)

Define an area-function $A'$ for graph $G'$ as follows: For any interior face $f$ in $G'$, $A'(f) = A(f)$ if $f$ was a face in $G$ and $A'(f) = A(f_1) + A(f_2) + A(f_3)$ if $f = T$.

Observe that $G'$ is also a partial 3-tree, since we can define it with the induced order that defined $G$: $v_1, \ldots, v_{n-1}$. Thus, we can draw $G - v_n$ recursively, requiring area $A'(f)$ for each interior face in $G'$, obtaining rational coordinates for $v_h$, $v_i$ and $v_j$.

Then, by Lemma 1, $v_n$ can be added inside $T$ with rational coordinates, so that the area of $f_1, f_2$ and $f_3$ is correct. See also Figure 2.5. □

Since the equation for $x^*$ and $y^*$ in Lemma 1 is unique, the drawing of $G$ in Lemma 2 is unique (up to a linear transformation.)

**Lemma 3** *Every planar partial 3-tree can be drawn respecting prescribed interior face areas such that all coordinates are rational.*

[1]These equations for $x^*$ and $y^*$ can also be obtained using the asymmetric barycenter method, by setting the stiffness of the edges incident to $v^*$ to the area of the triangle that is not adjacent to them [36]. This was suggested to us by André Schulz at GD 2009.

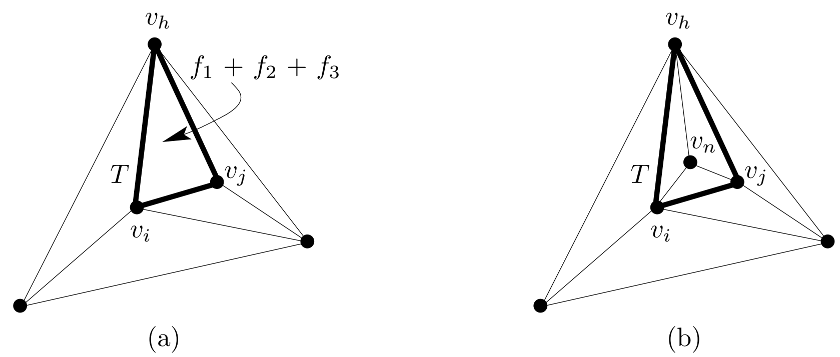



Figure 2.5: (a) $G'$ (b) $G$. Since $G'$ is a planar 3-tree, we can recursively draw it and then place $v_n$ in correct position to obtain the drawing of $G$.

**Proof:** It is known that a planar partial 3-tree $G'$ can be augmented into a 3-tree $G$ by adding edges. It is not obvious that $G$ can be assumed to be planar (for example, this is not true if we replace '3' by '4'), but one can show that this is indeed true (J. Kratochvíl, private communication; all crucial ingredients for this are in [17].) Thus, to draw a planar partial 3-tree $G'$ with prescribed interior face areas, add edges until it is converted into a planar 3-tree $G$. Then, apply Lemma 2 to draw $G$, and finally, remove all extra edges to obtain the drawing for $G'$.

It is necessary to assure that the areas of the interior faces of $G'$ can be respected in such a drawing. Each time an edge is added, it divides a face $f_i$ into two faces $f_i^1$ and $f_i^2$. Let $a_i$ be the prescribed area for $f_i$, then we choose area $a_i^1$ for face $f_i^1$ and $a_i^2$ for face $f_i^2$, such that $a_i^1 + a_i^2 = a_i$. Any such assignment is possible; if we use e.g. $a_i^1 = a_i^2 = \frac{a_i}{2}$, then the face areas will remain rational. When the extra edge is removed, $f_i$ will then have the correct area. See also Figure 2.6. □

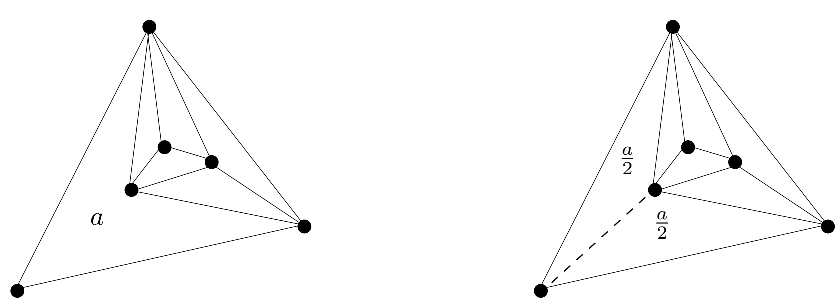



Figure 2.6: To draw a planar partial 3-tree, convert it into a planar 3-tree, draw it, and then remove all extra edges.

The results from Lemmas 2 to 3 are summarized in the following theorem:

**Theorem 1** *Let $G$ be a planar partial 3-tree and $A$ be an assignment of non-negative rationals to interior faces of $G$. Then $G$ has a straight-line drawing such that each interior face $f$ of $G$ has area $A(f)$ and all coordinates are rationals.*

## 2.3 How small are the coordinates?

Since our drawings have rational coordinates, we are now interested in the size of the required grid. In this section, we give bounds on the required resolution. This bounds are not polynomial. However, we provide an example where the coordinates of each vertex are obtained with this algorithm. In this example, the size of the grid grows exponentially, which suggests that even if tighter bounds were found, they would still be exponential. We showed in Lemma 2, that the drawing produced by this algorithm is unique (up to a linear transformation.) Therefore, the bounds found for this algorithm apply to any algorithm.

We need some notation. Let $v_1, \ldots, v_n$ be the vertex order in which $G$ was constructed with $v_1, v_2, v_3$ the outerface. The drawing is the one from Theorem 1. Recall that we can view the graph $G$ as being obtained from a smaller partial 3-tree by inserting a vertex $v_j$ into the triangle $T_j$ spanned by the three predecessors of $v_j$. Let $G[T_j]$ be the subgraph of $G$ induced by all vertices on or inside $T_j$. Let $v_{j_0}, v_{j_1}$ and $v_{j_2}$ be the predecessors of $v_j$. See Figure 2.7.

**Claim 1** *$G[T_j]$ has at most $n - j + 4$ vertices.*

**Proof:** Since $T_j$ was a face in the graph induced by $\{v_1, \ldots, v_{j-1}\}$ (see Figure 2.7), none of the interior vertices of $G[T_j]$ is in $\{v_1, \ldots, v_{j-1}\}$. Thus, all vertices in $G[T_j]$ are $v_j$, $v_{j_0}, v_{j_1}$ and $v_{j_2}$, in addition to some vertices in $\{v_{j+1}, \ldots, v_n\}$ and so, $G[T_j]$ has at most $n - j + 4$ vertices. □

**Claim 2** *$G[T_j]$ has at most $2n - 2j + 3$ interior faces.*

**Proof:** $G[T_j]$ is a triangulated graph with $n_j \leq n - j + 4$ vertices. Then, the number of interior faces of $G[T_j]$ is $2 \cdot n_j - 5 \leq 2(n - j + 4) - 5 = 2n - 2j + 3$. □

Recall that a drawing of a graph that respects given interior face areas is called equifacial if all faces have the same area. Since the coordinates of a newly placed vertex depend on the area of the faces around it, equifacial drawings are easier to analyze. Thus, we will first study bounds on equifacial drawings and then we will generalize this result to graphs with arbitrary face areas.

**Theorem 2** *Any planar 3-tree $G$ has an equifacial straight-line drawing with integer coordinates and width and height at most $\prod_{k=1}^{n}(2k + 1)$.*

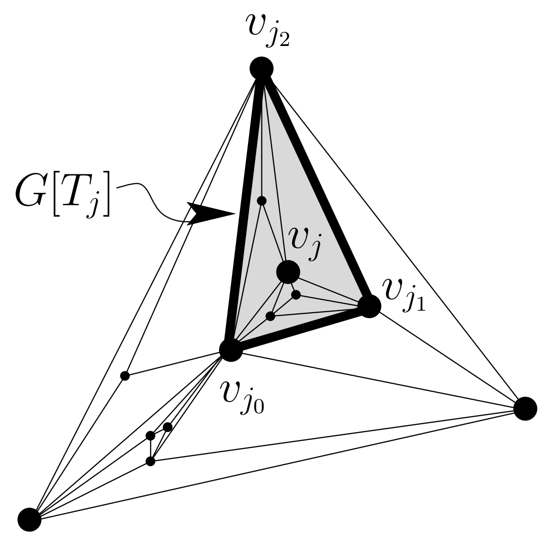



Figure 2.7: None of the interior vertices of $G[T_j]$ is in $\{v_1, \ldots, v_{j-1}\}$. $G[T_j]$ shaded in the figure.

**Proof:** We show that $G$ has an equifacial straight-line drawing with rational coordinates in $[0, 1]$ with common denominator at most $\prod_{k=1}^{n}(2k+1)$; the result then follows after scaling. We show the bound on the denominator only for $x$-coordinates; $y$-coordinates are proved similarly.

We assume that $v_1, v_2, v_3$ are at the triangle $T = \{(1, 0), (0, 1), (0, 0)\}$ (this can be enforced in the base case of Lemma 2.)

Let $f$ be the number of interior faces of $G$. $T$ has area $\frac{1}{2}$ and the drawing is equifacial, thus the area of each face is $\frac{1}{2f}$. Since $G$ is triangulated, it has $f = 2n - 5$ interior faces; so each interior face is drawn with area $a = \frac{1}{2\cdot(2n-5)} = \frac{1}{4n-10}$.

Let $f_j$ be the number of interior faces of $G[T_j]$. We will bound the denominator in terms of $f_j$ first.

**Claim 3** *Vertex $v_i$ has $x$-coordinate*

$$x_i = \frac{c_i}{\prod_{4\leq j\leq i} f_j} \tag{2.3}$$

*for some integer $c_i$*[2].

**Proof:** The proof is by induction on $i$. Nothing is to show for $i = 1, 2, 3$, since $x_i$ is an integer by choice of the points for the outerface triangle. For $i \geq 4$, let $v_{i_0}, v_{i_1}, v_{i_2}$ be the three predecessors of $v_i$.

[2] We will not compute the exact integer to keep notation simple.

By induction, Equation (2.3) holds for the predecessors of $v_i$, thus $x_{i_k}$ (the coordinate of $v_{i_k}$) satisfies

$$x_{i_k} = \frac{c_{i_k}}{\prod_{4 \le j \le i_k} f_j} \tag{2.4}$$

for $k = 0, 1, 2$ and some integer $c_{i_k}$.

Expanding both enumerator and denominator yields

$$x_{i_k} = \frac{c_{i_k}}{\prod_{4 \le j \le i_k} f_j} = \frac{c_{i_k} \cdot f_{i_k+1} \cdot \cdots \cdot f_{i-1}}{\prod_{4 \le j \le i-1} f_j} = \frac{c'_{i_k}}{\prod_{4 \le j \le i-1} f_j} \tag{2.5}$$

for some integer $c'_{i_k}$.

Equation (2.1) states that $x_i = \frac{a_0 x_{i_0} + a_1 x_{i_1} + a_2 x_{i_2}}{a_0 + a_1 + a_2}$, where $a_0, a_1, a_2$ are the areas of faces incident to $v_i$. For $k = 0, 1, 2$, each $a_k$ is the sum of the faces in some subgraph, and therefore an integer multiple of $\frac{1}{4n-10}$, say $a_k = \frac{b_k}{4n-10}$. Furthermore, $a_0 + a_1 + a_2$ is exactly the area of triangle $T_i$ spanned by $v_{i_1}, v_{i_2}, v_{i_3}$. Note that $T_i$ contains exactly the $f_i$ faces in $G_i$, and recall that all these faces have area $\frac{1}{4n-10}$, so $T_i$ has area $\frac{f_i}{4n-10}$.

Hence, as desired,

$$\begin{aligned} x_i &= \frac{a_0 x_{i_0} + a_1 x_{i_1} + a_2 x_{i_2}}{a_0 + a_1 + a_2} = \frac{\sum_{k=0}^{2} \frac{b_k}{4n-10} \frac{c'_{i_k}}{\prod_{4 \le j \le i-1} f_j}}{\frac{f_i}{4n-10}} \\ &= \frac{4n-10}{(4n-10) \cdot (\prod_{4 \le j \le i-1} f_j) \cdot f_i} \cdot \sum_{k=0}^{2} b_k \cdot c'_{i_k} = \frac{integer}{\prod_{4 \le j \le i} f_j}. \end{aligned}$$

□

Since $f_4, \ldots, f_n$ are integers, by Claim 3 all $x_i$'s have common denominator $\prod_{4 \le j \le n} f_j$.

By Claim 2, $f_j \le (2n - 2j + 3)$ and substituting $k = n - j + 1$ we get:

$$\prod_{4 \le j \le n} f_j \le \prod_{4 \le j \le n} (2n - 2j + 3) = \prod_{k=1}^{n-3} (2k + 1) \tag{2.6}$$

This proves Theorem 2. □

We can obtain similar (but uglier-looking) bounds for arbitrary integer interior face areas, by replacing '$f_j$' in Claim 3 by 'the sum of the face areas in $G[T_j]$', as shown in the following theorem.

**Theorem 3** *Any planar 3-tree $G$ has a straight-line drawing that respects given integer interior face areas, with integer coordinates and width and height at most $A_{max}^{n-3} \cdot \prod_{k=1}^{n-3}(2k+1)$, where $A_{max}$ is the largest area of $G$.*

Note that this theorem implies Theorem 2, since when the drawing is equifacial we have $A_{max} = 1$.

**Proof:**

**Claim 4** *Vertex $v_i$ has $x$-coordinate*

$$x_i = \frac{c_i}{\prod_{4 \le j \le i} \sum_{f \text{ face of } G[T_j]} A(f)} \tag{2.7}$$

*for some integer $c_i$.*

**Proof:** This proof is very similar to the one of Claim 3, and it is by induction on $i$. Nothing is to show for $i = 1, 2, 3$, since $x_i$ is an integer by choice of the points for the outerface triangle. For $i \ge 4$, let $v_{i_0}, v_{i_1}, v_{i_2}$ be the three predecessors of $v_i$.

By induction, Equation (2.7) holds for the predecessors of $v_i$, thus $x_{i_k}$ (the coordinate of $v_{i_k}$) satisfies

$$x_{i_k} = \frac{c_{i_k}}{\prod_{4 \le j \le i_k} \sum_{f \text{ face of } G[T_j]} A(f)} \tag{2.8}$$

for $k = 0, 1, 2$ and some integer $c_{i_k}$.

Expanding both enumerator and denominator yields

$$\begin{aligned}
x_{i_k} &= \frac{c_{i_k}}{\prod_{4 \le j \le i_k} \sum_{f \text{ face of } G[T_j]} A(f)} && (2.9)\\
&= \frac{c_{i_k} \cdot \sum_{f \text{ face of } G[T_{i_k+1}]} A(f) \cdot \ldots \cdot \sum_{f \text{ face of } G[T_{i-1}]} A(f)}{\prod_{4 \le j \le i-1} \sum_{f \text{ face of } G[T_j]} A(f)} && (2.10)\\
&= \frac{c'_{i_k}}{\prod_{4 \le j \le i-1} \sum_{f \text{ face of } G[T_j]} A(f)} && (2.11)
\end{aligned}$$

for some integer $c'_{i_k}$.

Equation (2.1) states that $x_i = \frac{a_0 x_{i_0} + a_1 x_{i_1} + a_2 x_{i_2}}{a_0 + a_1 + a_2}$, where $a_0, a_1, a_2$ are the areas of faces incident to $v_i$. For $k = 0, 1, 2$, each $a_k$ is the sum of the faces in some subgraph, and therefore an integer. Furthermore, $a_0 + a_1 + a_2$ is exactly the area of triangle $T_i$ spanned by $v_{i_1}, v_{i_2}, v_{i_3}$.

Hence, as desired,

$$\begin{aligned} x_i &= \frac{a_0 x_{i_0} + a_1 x_{i_1} + a_2 x_{i_2}}{a_0 + a_1 + a_2} = \frac{\sum_{k=0}^{2} a_k \frac{c'_{i_k}}{\prod_{4 \le j \le i-1} \sum_{f \text{ face of } G[T_j]} A(f)}}{\sum_{f \text{ face of } G[T_i]} A(f)} \\ &= \frac{\sum_{k=0}^{2} a_k \cdot c'_{i_k}}{\sum_{f \text{ face of } G[T_i]} A(f)) \cdot \prod_{4 \le j \le i-1} \sum_{f \text{ face of } G[T_j]} A(f)} \\ &= \frac{integer}{\prod_{4 \le j \le i} \sum_{f \text{ face of } G[T_j]} A(f)} \end{aligned}$$

$\square$

The sum of the face areas in $G[T_j]$ can be upper bounded as follows:

$$\sum_{f \text{ face of } G[T_j]} A(f) \le \sum_{(2n-2j+3) \text{ largest faces } f \text{ in } G} A(f) \le (2n - 2j + 3) \cdot A_{max}$$

Therefore, the denominator is at most $\prod_{4 \le j \le n} ((2n - 2j + 3) \cdot A_{max})$.

By setting $k = n - j + 1$ on the denominator, we get

$$\prod_{4 \le j \le n} ((2n - 2j + 3) \cdot A_{max}) = A_{max}^{n-3} \cdot \prod_{k=1}^{n-3} (2k + 1)$$

which proves Theorem 3. $\square$

We also did experiments to see whether our bounds are tight. We computed (using Maple) the coordinates in Theorem 2 for the planar 3-tree $G$, with vertex order $v_1, \ldots, v_n$ where $v_j$ has predecessors $v_{j-1}, v_{j-2}, v_{j-3}$ for $j \ge 4$ (see also Figure 2.8). Since vertices $\{v_{j+1}, \ldots, v_n\}$ are all inside $G[T_j]$, this graph has $f_j = 2n - 2j + 3$ and hence is a good candidate to obtain the bound in Theorem 2.

Figure 2.9 shows the least common denominator for various values of $n$, both in $G$ and in the upper bound. The least common denominators are smaller in $G$ than those from the upper bound, but are growing in exponential fashion for the first values considered in

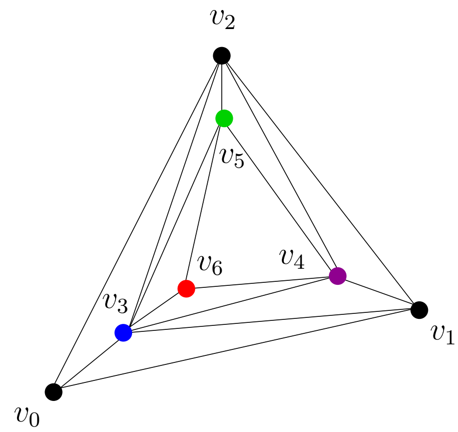



Figure 2.8: A planar 3-tree where $v_j$ has predecessors $v_{j-1}, v_{j-2}, v_{j-3}$ for $j \geq 4$.

| $n$ | LCD in drawing | upper bound |
|---|---|---|
| 10 | $5.0 \cdot 10^{3}$ | $2.0 \cdot 10^{6}$ |
| 50 | $3.1 \cdot 10^{34}$ | $2.8 \cdot 10^{75}$ |
| 100 | $1.0 \cdot 10^{82}$ | $1.7 \cdot 10^{183}$ |
| 500 | $1.0 \cdot 10^{427}$ | $2.0 \cdot 10^{1271}$ |
| 1000 | $2.8 \cdot 10^{852}$ | $4.8 \cdot 10^{2853}$ |

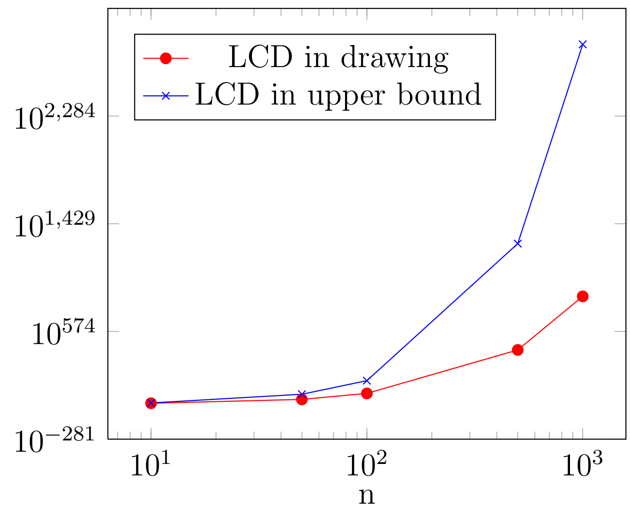



Figure 2.9: Lower and upper bounds on the resolution in the drawing.

our example. However, we have not been able to prove an exponential lower bound for this graph.

## 2.4 Time complexity

In this section, we show that our algorithm to draw planar 3-trees with prescribed interior face areas can be implemented to take $O(n)$ time. Most operations can clearly be done in $O(1)$ time per vertex. One non-trivial part is how to obtain the values $a_1, a_2, a_3$ for Equations 2.1 and 2.2; note that these are areas of faces of a whole subgraph, and hence not immediately available.

The algorithms by Bodlaender [7] and Sanders [35] determine whether a graph has treewidth at most $k$, and if so, return the elimination order of the vertices in the graph. We can use this elimination order to create a *splitting tree*: a tree of the triangles that are faces of a subgraph of $G$ during the algorithm (see Figure 2.10). The tree can be created in a top-down fashion and this will take $O(n)$ time since each vertex creates 3 children of a leaf node of the splitting tree.

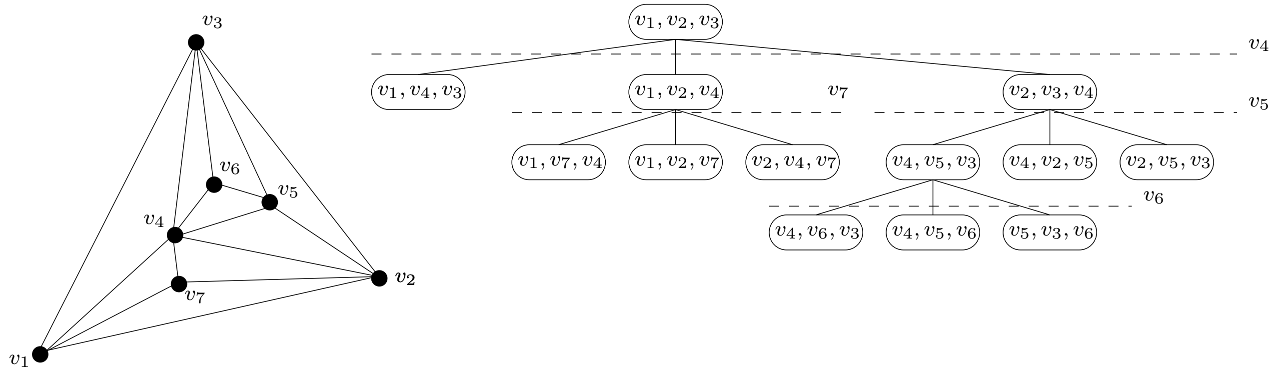



Figure 2.10: Planar 3-tree $G$, its elimination order, and its splitting tree.

The leaf nodes of the splitting tree represent single interior faces of $G$, and their areas are known. Thus, it is possible to traverse the tree bottom-up to calculate the area of each of the faces of subgraphs of $G$ that are needed by the algorithm. This will take $O(n)$ time, since the number of internal nodes of the tree is the same as the number of vertices that are not on the outerface and to calculate the area of each internal node takes constant time, because it is the sum of the area of its three children.

Finally, by traversing the splitting tree once more, but top-down, it is possible to obtain the coordinates of each of the vertices used to split $G$, using Equations 2.1 and 2.2. Once more, this operation takes $O(n)$ time, leading to a total running time of $O(n)$.

## 2.5 Partial 3-trees and Thomassen's proof

We mentioned in Section 1.2 that the work by Thomassen [39] was the main motivation for this thesis. In this section we show that the subclass of graphs studied by Thomassen in his Lemma 4.1 are partial 3-trees. We call these graphs a *PF-graph*. Thomassen showed that PF-graphs can be drawn respecting given interior face areas and with some restrictions on the placement of vertices that we will not review in detail here.

We first describe PF-graphs. Such a graph $G = (V, E)$ has no cut-vertex. Let $P$ be a path on the outerface of $G$. Each vertex not on $P$ has degree 2 or 3, and $V$ can be divided into sets such that $V = V_0 \cup V_1 \cup \cdots \cup V_q$, $V_0 = P$ and, for each $i$ ($1 \leq i \leq q$), each vertex of $V_i$ has at least two neighbours in $V_0 \cup V_1 \cup \cdots \cup V_{i-1}$. Vertices in the same set $V_i$ do not share any edges. See Figure 2.11a for an example.

**Claim 5** *The vertices in $V_1 \cup \cdots \cup V_q$ induce a forest.*

**Proof:** Every vertex in $V_1 \cup \cdots \cup V_q$ has at least two neighbours in $V_0 \cup V_1 \cup \cdots \cup V_{i-1}$, no neighbours in $V_i$, and degree 2 or 3, thus, every vertex has 0 or 1 edges connecting to $V_{i+1} \cup \cdots \cup V_q$. Therefore, it is a forest. □

We will show that $G$ is a minor of a Halin graph. Recall from Section 2.1 that Halin graphs are graphs that can be obtained from a tree with no vertices of degree two, by connecting all the leaves with a cycle $C$. Halin graphs are partial 3-trees [8].

PF-graphs are very similar to Halin graphs. One can see these graphs as being constructed by connecting the forest induced by $V_1 \cup \cdots \cup V_q$ to the path $P$ by adding edges.

There are 3 differences between a PF-graph $G$ and a Halin graph $G'$:

- The leaves in $G$ are connected by a path $P$. The leaves in $G'$ are connected by a cycle $C$.
- Vertices on $P$ in $G$ can have degree higher than 3, while all vertices on $C$ in $G'$ have degree exactly 3.
- Let $G_t$ be the graph obtained from $G$ by deleting the edges belonging to $P$ and $G'_t$ be the graph obtained from $G'$ by deleting the edges belonging to $C$. $G_t$ may be disconnected, while $G'_t$ is a tree, and thus connected.

To obtain a Halin graph $G'$ from any given PF-graph $G$, make the following changes to overcome each of the differences above:

1. Add the remaining edge to convert $P$ into a cycle. See Figure 2.11b.
2. For each vertex $v$ on $P$ with degree $d$ higher than 3, substitute $v$ by a path $P_v$ with $d-2$ vertices. Connect each of the edges incident to $v$ to a different vertex on $P_v$ according to the cyclic order around $v$ in the planar embedding. See Figure 2.11c.
3. Connect $G_t$ by adding edges. See Figure 2.11d.

$G'$ is a Halin graph, and therefore it has treewidth 3 (it is a partial 3-tree.) To obtain $G$ from $G'$ the only operations necessary are edge deletions and edge contractions, therefore $G$ is a minor of $G'$. Thus, the treewidth of $G$ is at most the treewidth of $G'$, and $G$ it is a partial 3-tree.

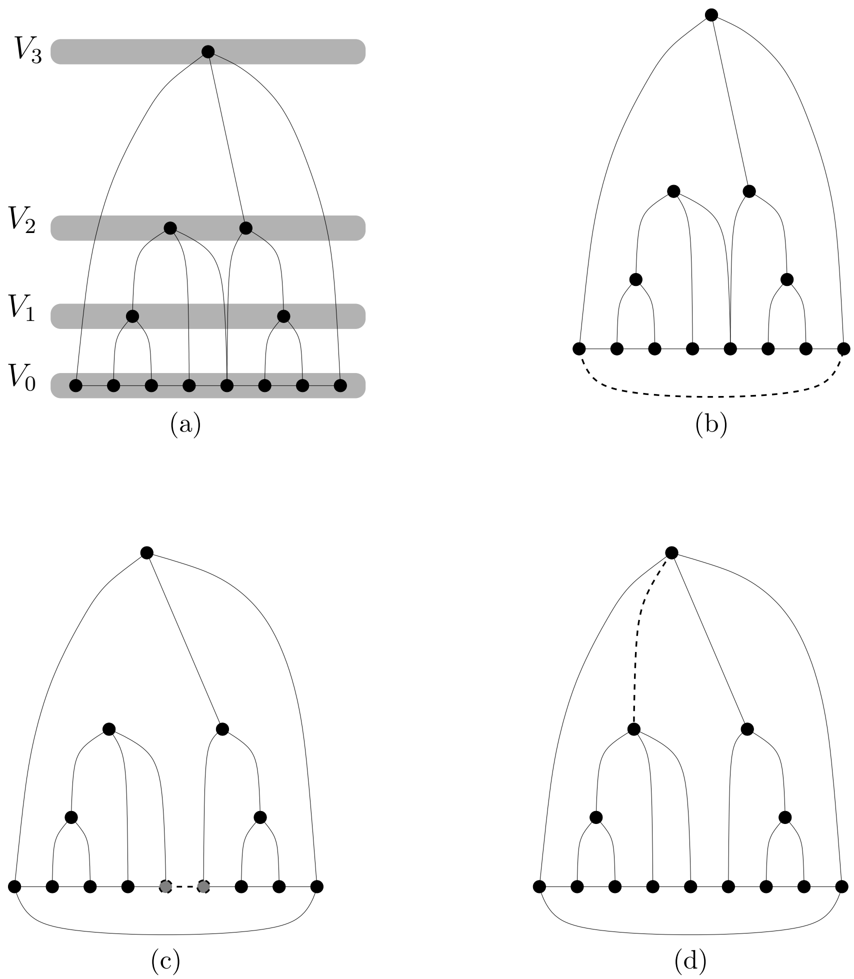



Figure 2.11: (a) Example of a PF-graph. Steps to convert a PF-graph into a Halin graph: (b) Step 1, convert $P$ into a cycle. (c) Step 2, split vertices with degree higher than 3 in $P$. (d) Step 4, make $G_t$ connected.

## 2.6 Beyond 3-trees

A natural question is whether the result from Theorem 1 can be generalized to planar partial 4-trees. In this section, we give some examples of planar partial 4-trees where no realization with rational coordinates is possible. Also, we give an example that shows that Thomassen's proof for cubic graphs does not lead directly to a drawing with rational coordinates.

The first example is the octahedron where all interior face areas are 1 except for two non-adjacent, non-opposite faces, which have area 3. As shown by Ringel [34], any drawing that respects these areas must have some complex coordinates. (Ringel's result was actually for the graph $G_1$ obtained from the octahedron by subdividing two triangles further; the resulting graph then has no equifacial drawing.) See Figure 2.12a for an illustration of this graph. Note that both the octahedron and $G_1$ are planar partial 4-trees, so not all partial 4-trees have equifacial drawings.

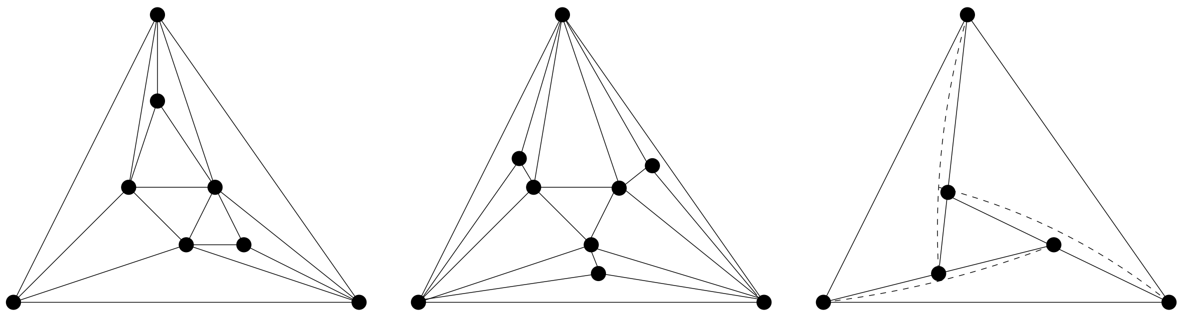



Figure 2.12: (a) Ringel's example of a graph that cannot be realized with prescribed interior face areas. (b) Graph that can be realized but not with rational coordinates. (c) Graph that part of Thomassen's proof is based on. This graph can be realized but not with rational coordinates.

The second example is the octahedron where all interior face areas are 1 except that the three faces adjacent to the outerface have area 3. (Alternatively, one could ask for an equifacial drawing of graph $G_2$ in Figure 2.12b.) Assume, after possible linear transformation, that the vertices in the outerface are at $(0,0),(0,13)$ and $(2,0)$. Computing the signed area of all the faces one can show that the vertices not on the outerface are at $(\frac{10}{3}+\frac{2\sqrt{3}}{13}, 5-\sqrt{3}), (\frac{10}{3}-\frac{2\sqrt{3}}{13}, 3)$ and $(\frac{6}{13}, 5+\sqrt{3})$. Thus even if a partial 4-tree has an equifacial drawing, it may not have one with rational coordinates.

The third example is again the octahedron, with three of the interior face areas prescribed to be 0, which forces some edges to be aligned as shown in Figure 2.12c. If all other interior faces have area $1/8$, and the outerface is at $(1,0),(0,1),(0,0)$, then similar computations show that some of the coordinates of the other three vertices are $(3\pm\sqrt{5})/8$. If the edges that appear dotted in the figure are removed, we obtain a graph that is a crucial ingredient in Thomassen's proof. Since this graph cannot be drawn with rational coordinates, then Thomassen's proof, as is, does not give rational coordinates.

# Chapter 3

# Orthogonal drawings with prescribed face areas

In this chapter, we give a recursive algorithm to create orthogonal drawings for 3-connected planar graphs with maximum degree 3 that respect given interior face areas. It was known already that such drawings always exist (see [11, 28].) However, the bounds on the number of bends were quite high. Our algorithm approaches the problem in a completely different way, leading to better bounds on the number of bends per edge and per face. We also show that the coordinates of all vertices and bends are rational. Throughout the chapter we will assume our graphs to be 3-connected and the given face areas to be positive integers. In particular, the area of any face is assumed to be at least one.

## 3.1 Canonical ordering and dart-shaped graphs

The first tool we use is the *canonical ordering*. This tool was introduced by Kant [25]; he proved in his Theorem 1.3 that every triconnected plane graph $G$ has a canonical ordering, and used it to create both straight-line and orthogonal drawings of planar graphs on the grid. We will use a slightly different (but equivalent) definition given in the book by Nishizeki and Rahman [30].

**Definition 1** *Let $G = (V, E)$ be a triconnected planar graph with an edge $(v_1, v_2)$ on the exterior face. Let $\pi = V_1, ..., V_K$ be an ordered partition of $V$, that is, $V_1 \cup ... \cup V_K = V$ and $V_i \cap V_j = \emptyset$ for $i \neq j$. Define $G_k$ to be the subgraph of $G$ induced by $V_1 \cup ... \cup V_k$, and denote by $C_k$ the outerface of $G_k$. We say that $\pi$ is a* canonical ordering *of $G$ if:*

- *$V_1$ is the set of all vertices on the interior face containing edge $(v_1, v_2)$.*
- *$V_K$ is a singleton $\{v_n\}$ where $v_n$ lies on the outerface, $(v_1, v_n) \in E$, and $v_n \neq v_2$.*

- *Each $C_k$ ($k > 1$) is a cycle containing $(v_1, v_2)$.*
- *Each $G_k$ is biconnected and internally triconnected, that is, removing two interior vertices of $G_k$ does not disconnect it.*
- *For each $k$ in 2,..., $K - 1$, one of the two following conditions holds:*

  *(a) $V_k$ is a singleton $\{z\}$, where $z$ belongs to $C_k$ and has at least one neighbor in $G - G_k$. See Figure 3.1a.*

  *(b) $V_k$ is a chain $\{z_1, ..., z_r\}$, where each $z_i$ has at least one neighbor in $G - G_k$, and where $z_1$ and $z_r$ each have one neighbor on $C_{k-1}$, and these are the only neighbors of $V_k$ in $G_{k-1}$. See Figure 3.1b.*

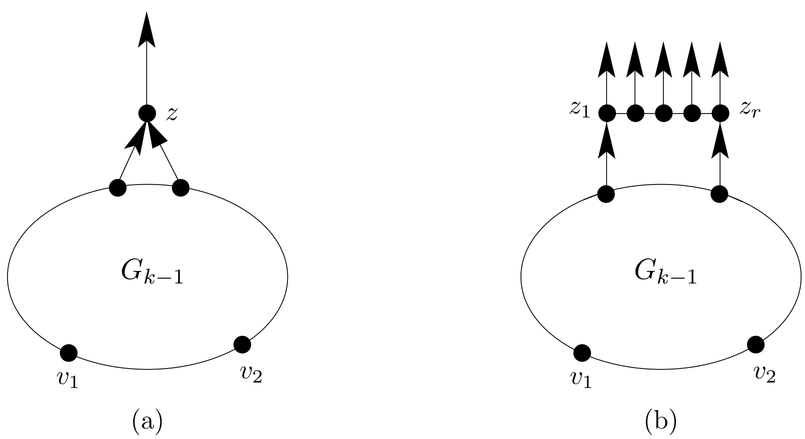



Figure 3.1: Canonical order construction: (a) $V_k$ is a singleton. (b) $V_k$ is a chain.

A *canonical orientation* of the edges of $G$ can be obtained from the canonical ordering. To do this, direct the edges from a vertex in $V_i$ to a vertex in $V_j$ if $i < j$. Edges between vertices in the same set remain undirected. Thus, a canonical orientation is a partial edge orientation of $G$, i.e., some edges are directed and some are not.

The canonical ordering and its variants have been used frequently for algorithms to draw planar graphs; see for example [9, 10, 16, 25]. Most of these algorithms proceed by adding $V_1, V_2, ...$ in this order to the drawing. Our approach is different (and has, to our knowledge, not been used before): We split the graph into two pieces based on directed paths in the canonical orientation and draw both pieces recursively. To explain this, we need a few more definitions:

In a partial orientation of the edges of $G$ we use the term *vertical path* for a simple directed path. Note that if the orientation is a canonical orientation, then a vertical path connects vertices in $V_{i_1}, V_{i_2}, ..., V_{i_k}$ with $i_1 < i_2 < \cdots < i_k$; hence the name. Let $v_s$ be the source and $v_t$ the sink of a vertical path $P$. Every vertex $v_m$ on $P$, except for $v_s$ and $v_t$, has two incident edges, $e_i$ and $e_o$, incoming and outgoing respectively, from the path. Any edge different from $e_i$ and $e_o$ is said to be to the *right* of $v_m$, with respect to $P$, if it appears after $e_o$ and before $e_i$, in clockwise order around $v_m$, and to the *left* of $v_m$, with respect to $P$, otherwise. See Figure 3.2 for an example.

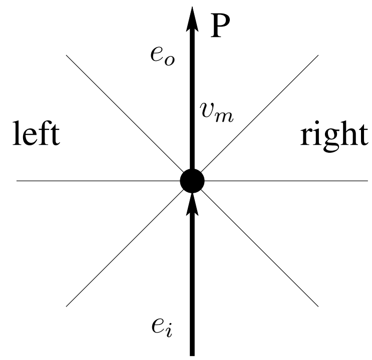



Figure 3.2: Edges to the left and right of a vertex $v_m$, with respect to a path $P$.

A *horizontal path* is a simple path where all edges are undirected. In a canonical orientation, all vertices in a horizontal path belong to the same set $V_i$; hence the name.

We now define a condition on subgraphs that will be crucial for splitting graphs in our drawing algorithm; see also Figure 3.3.

**Definition 2** *Let $G = (V, E)$ be a plane graph with a fixed partial edge orientation that has no directed cycle. $G$ is called* dart-shaped with respect to the orientation *if:*

*D1.* *Every interior face has an undirected edge $(u, v)$ and, walking from $u$ to $v$ in clockwise order around the face, there are a vertical path in the same direction as the walk, a horizontal path (possibly with no edges), and a vertical path in the opposite direction to the walk. See also Figure 3.3b.*

*D2.* *The outerface consists of a horizontal path $P_b$ (connecting a vertex $c_l$ to a vertex $c_r$), and two vertical paths, $P_l$ (from $c_l$ to a vertex $c_t$), and $P_r$ (from $c_r$ to $c_t$). These paths are interior vertex-disjoint. The vertices $c_l$, $c_r$, $c_t$ will be called* left, right *and* top corner, *and the paths $P_b$, $P_l$, $P_r$ will be called* bottom, left *and* right path *of $G$. See Figure 3.3a.*

*D3.* *$deg(c_l) = deg(c_r) = deg(c_t) = 2$. All other vertices have degree at most 3.*

*D4.* *Every vertex $\neq c_t$ has exactly one outgoing edge.*

Notice that D3. and D4. imply that no vertex on $P_b$ has an incoming edge.

From now on, whenever we speak of a dart-shaped graph $G$, we will use $c_l$, $c_r$, $c_t$, $P_b$, $P_l$, $P_r$ for its corners and paths without specifically recalling that notation. We now explain how to obtain a dart-shaped graph from a canonical order if the graph has maximum degree three.

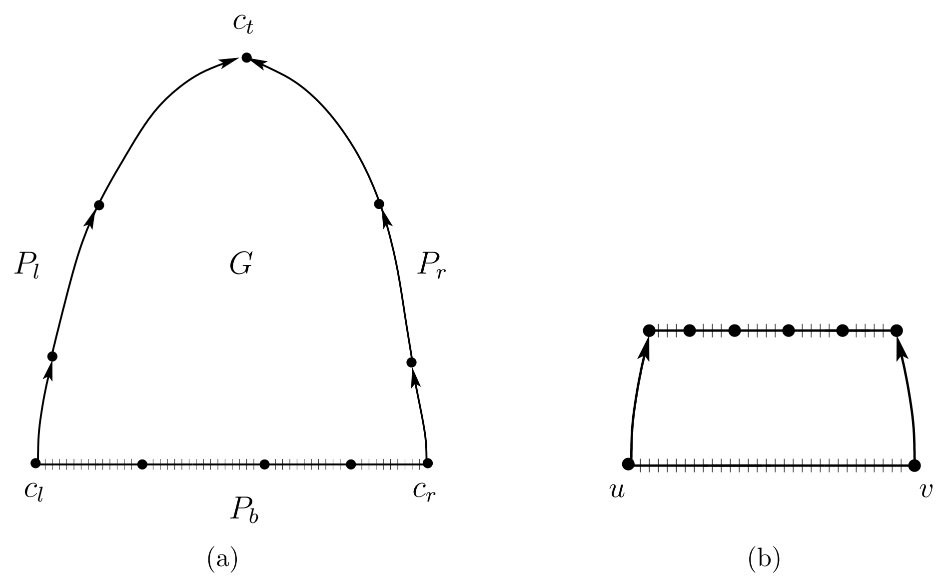



Figure 3.3: (a) Outerface of a dart-shaped graph. (b) An interior face of a dart-shaped graph. In both figures, hashed edges represent undirected edges.

**Lemma 4** *Let $G$ be a planar 3-connected 3-regular graph. Let $V_1 \cup \cdots \cup V_K$ be a canonical ordering of $G$, with $V_1 = \{v_1, \cdots, v_j\}$ and $V_K = v_n$. Assume $G$ has been partially oriented according to this canonical ordering. Let $G' = G - v_1$, with the induced partial orientation. Then $G'$ is dart-shaped and its corners are $c_l = v_j$, $c_r = v_2$ and $c_t = v_n$.*

**Proof:** By construction, the canonical ordering of $G$ is acyclic. We verify that $G'$ satisfies the conditions that a graph needs to be a dart-shaped graph (see Figure 3.4a for an example):

D1. Kant gives in [26] the main ingredients to prove that this is true for $G$, in his Lemmas 3 and 4. Since all the remaining faces in $G'$ were not modified, this condition holds.

D2. The outerface consists of the horizontal path $v_j, \ldots v_3, v_2$, between $v_j$ and $v_2$, and two vertical paths, one from $v_j$ to $v_n$ and the other from $v_2$ to $v_n$. The path between $v_j$ and $v_2$ is horizontal because all vertices in it belong to the same set in the canonical order, $V_1$. Condition (D1) holds for the interior face of $G$ that contains $v_1$, $v_j$ and $v_n$, thus the path from $v_j$ to $v_n$ is vertical. The path from $v_2$ to $v_n$ is vertical from the construction

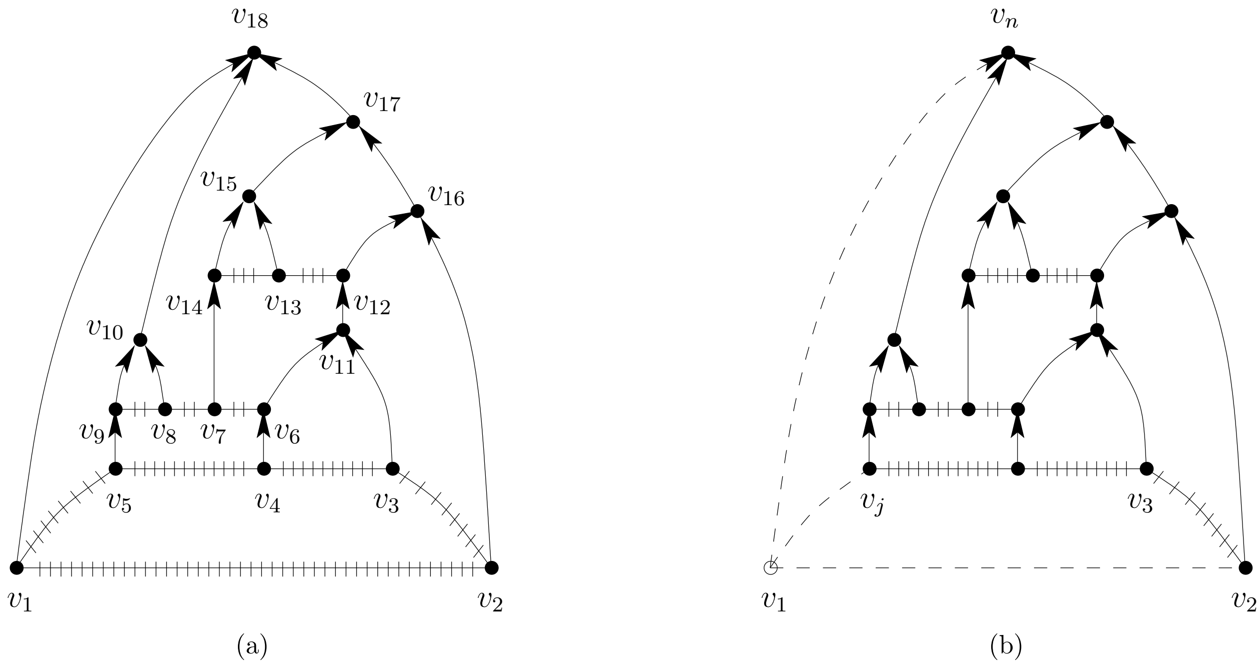



Figure 3.4: (a) Canonical orientation of $G$. (b) $G - v_1$ is dart-shaped.

of the canonical order: for every edge $(v_k, v_l)$ in the path, if $v_k$ was a singleton, it has indegree two, and therefore $(v_k, v_l)$ has to be vertical; if $v_k$ is the rightmost vertex of a chain, there will be a vertical edge from $G_{k-1}$ to $v_k$, and a horizontal edge to the left that belongs to the chain, thus $(v_k, v_l)$ has to be vertical.

D3. $G$ is 3-regular. To obtain $G'$, we deleted an edge incident to $v_j$, $v_2$ and $v_n$, so in $G'$ the corners $c_l$, $c_r$ and $c_t$ have degree 2 and all other vertices have degree 3.

D4. To see that each every vertex $\neq v_n$ has exactly one outgoing edge, we need to look at both the case when $V_k$ is a singleton and when it is a chain. When it is a singleton $z$, it has at least one neighbor in $G - G_k$, which means that it has at least one outgoing edge. Since $G_k$ it is 2-connected, $z$ has at least two neighbors in $G_{k-1}$, which define two incoming edges. Also, since $G$ is 3-regular, $z$ cannot have more than two incoming and one outgoing edge, so the claim holds for a singleton. When $V_k$ is a chain, $z_1, ..., z_r$, $z_1$ and $z_r$ are the only ones with one incoming edge and one undirected edge, the rest of them have two undirected edges. Also, all of them have at least one outgoing edge, but, since $G$ is 3-regular, they actually can only have one. Therefore, the claim holds for $G$. Vertex $v_1$ has no incoming edges, thus, deleting it does not remove any outgoing edge from vertices of $G'$, thus, the condition also holds for $G'$.

## 3.2 Decomposing a dart-shaped graph

In this section, we show how to split a dart-shaped graph into smaller dart-shaped graphs. This will allow us to divide recursively the graph into single faces. The partial edge orientation remains the same after this operation and throughout the chapter, and hence will not always be mentioned.

**Theorem 4** *Let $G$ be a dart-shaped graph. Then:*

*(a) $G$ can be decomposed into two dart-shaped graphs by splitting along a vertical path from the interior of $P_b$ to the interior of $P_l$ or $P_r$ (see Figure 3.5a), or*

*(b) $G$ can be decomposed into a dart-shaped graph and a face containing $c_l$ and $c_r$ by splitting along a horizontal path from the interior of $P_l$ to the interior of $P_r$ (see Figure 3.5b), or*

*(c) $G$ has only one interior face.*


Figure 3.5: (a) Vertical decomposition of $G$ into two dart-shaped graphs. (b) Horizontal decomposition of $G$ into a dart shaped graph and a face.

The proof of this theorem is divided into the following two lemmas, the first one addressing the case when $G$ can be split by a vertical path, and the second one the cases when $G$ can be split by a horizontal path or is a single face.

**Lemma 5** *Let $G$ be a dart-shaped graph for which $P_b$ has at least two edges. Then there exists a vertical path from a vertex $u \neq c_l, c_r$ in $P_b$ to a vertex $v \neq c_l, c_r, c_t$ in $P_l \cup P_r$ that breaks $G$ into two dart-shaped graphs.*

**Proof:** Pick any vertex $u \neq c_l, c_r$ in the interior of $P_b$. Follow the vertical path starting from the outgoing edge of $u$. This path is not a cycle because $G$ is a dart-shaped graph and thus acyclic, and it is unique since every vertex $\neq c_t$ has outdegree one. Also, the path must reach some vertex $\neq c_l, c_r, c_t$ on $P_l$ or $P_r$ since vertices on $P_b$ have indegree 0 and $c_t$ has indegree 2 and its incident edges are one on $P_l$ and the other on $P_r$. Let $v$ be the first vertex on $P_l \cup P_r$ that is on this path, and let $P_v$ be the path from $u$ to $v$. $P_v$ divides $G$ into two subgraphs, $G_l$ to the left of $P_v$, and $G_r$ to the right. See Figure 3.6 for an example.

We will prove that $G_l$ satisfies the conditions to be dart-shaped (the proof is quite similar for $G_r$):

D1. We neither created nor deleted any face. In $G$, every interior face satisfies this condition, therefore, the same will be true for $G_l$.

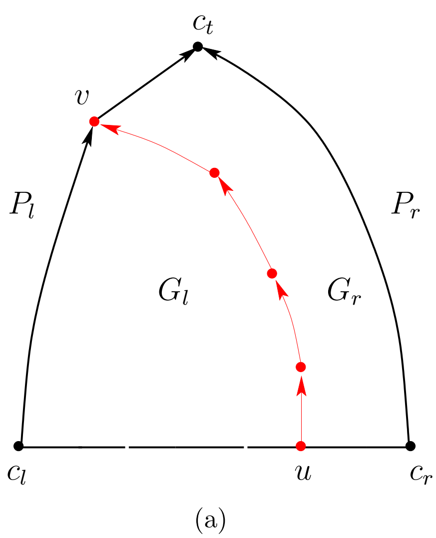
D2. There are two options for $c_t$: it may remain in $G_l$ or in $G_r$. If $c_t$ is in $G_l$, the outerface of $G_l$ will consist of the horizontal path from $c_l$ to $u$ and two vertical paths from $c_l$ to $c_t$ and from $u$ to $v$ to $c_t$. If $c_t$ is not in $G_l$, the outerface of $G_l$ will consist of the horizontal path from $c_l$ to $u$, and two vertical paths from $c_l$ to $v$ and from $u$ to $v$. In either case the claim holds, since the path from $u$ to $v$ did not contain vertices from $P_b$, $P_l$, $P_r$ in its interior.

D3. Since we did not add any edge, $deg(v) \leq 3$ holds for all $v \in V$. Vertex $c_l$ is the left corner of $G_l$ and $deg(c_l) = 2$. Vertex $u$ is the right corner for $G_l$. Since one incident edge of $u$ (on $P_b$) does not belong to $G_l$, and $deg_G(u) \leq 3$ we have $deg_{G_l}(u) \leq 2$. We know it is exactly two because $u$ has one directed edge on $P_v$ and one undirected edge on $P_b$ that does belong to $G_l$. Either $c_t$ or $v$ is the top corner in $G_l$. In the first case clearly $deg(c_t) = 2$. In the second case, one incident edge of $v$ (on $P_l$) does not belong to $G_l$ and $deg_G(v) = 3$, thus $deg_{G_l}(v) = 2$.

D4. Every vertex $\neq c_t$ had exactly one outgoing edge in $G$. To split $G$ into $G_l$ and $G_r$, we did not add any edges, and we only deleted edges adjacent to vertices on $P_v$. However, we did not delete any outgoing edge because these are in $P_v$. So, every vertex except the top corner in $G_l$ has exactly one outgoing edge. $\square$

**Lemma 6** *Let $G$ be a dart-shaped graph that has at least two interior faces. If $P_b$ is an edge, then there exists $c_l' \neq c_l, c_t \in P_l$, $c_r' \neq c_r, c_t \in P_r$ and a horizontal path from $c_l'$ to $c_r'$ that breaks $G$ into $G_b$ and $G_t$ such that $G_b$ is a single face containing $P_b$, and $G_t$ is dart-shaped.*

**Proof:** By condition (D2), $(c_l, c_r)$ is an undirected edge. Consider the interior face of $G$ adjacent to $(c_l, c_r)$. By (D1), it consists of the edge $(c_l, c_r)$, a vertical path $P_1 \subseteq P_l$ (since every vertex has only one outgoing edge) starting at $c_l$ and ending at some vertex $c_l'$, a vertical path $P_2 \subseteq P_r$ starting at $c_r$ and ending at some vertex $c_r'$, and (maybe) a horizontal



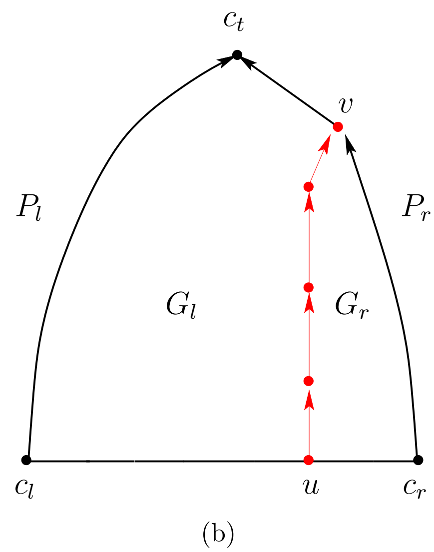




Figure 3.6: (a) Case when $v$ is in $P_l$. (b) Case when $v$ is in $P_r$.

path $P_h$ from $c'_l$ to $c'_r$. We know that $(c_l, c_r)$ cannot be this horizontal path because $c_l$ and $c_r$ are in $P_b$ and hence have no incoming edges. See Figure 3.6b.

If $P_h$ is empty, then $c'_l = c'_r$, which (since $P_l$ and $P_r$ are interior vertex-disjoint) implies $c'_l = c'_r = c_t$ and all of $G$ is one interior face and we are done.

So $P_h$ is non-empty. Let $G_t$ be the subgraph of $G$ that contains $P_h$ and $c_t$. We will now argue that $G_t$ is dart-shaped:

D1. We neither created nor deleted any face. In $G$, every interior face satisfies this condition, therefore, the same will be true for $G_t$.

D2. The outerface of $G_t$ consists of the horizontal path $P_h$ and two vertical paths, $P'_l$ from $c'_l$ to $c_t$, and $P'_r$ from $c'_r$ to $c_t$. We know that the latter two are vertical because they are subpaths of $P_l$ and $P_r$, respectively.

D3. We did not add any edges. Therefore the degree of each vertex in $G_t$ is at most its degree in $G$, which is $\leq 3$. $c'_l$ and $c'_r$ had degree at most three in $G$, but the incoming edge from $P_l$ or $P_r$, respectively, is not present in $G_t$, so their degree in $G_t$ is at most two. We know it is exactly two because each vertex has one outgoing edge on $P_l$ or $P_r$ and one undirected edge on $P_h$.

D4. Every vertex $\neq c_t$ in $G$ has outdegree one. Since we did not add edges, vertices in $G_t$ cannot have outdegree $> 1$. Since $G_b$ is a face in $G$, no interior vertex of $P_h$ can have outgoing edges to the face $G_b$, so they have an outgoing edge in $G_t$. In the case of $c'_l$ and $c'_r$, their outgoing edges belong to $P_l$ and $P_r$ and hence to $G_t$ as well.

Therefore, all vertices $\neq c_t$ have exactly one outgoing edge in $G_t$. □

This finishes the proof of Theorem 4: Every dart-shaped graph can be split into smaller graphs that are either dart-shaped or faces.

## 3.3 Overview of the algorithm

Now, we address the problem of how to create an orthogonal drawing of a graph $G$ that respects given interior face areas. Let's first take a look at the overview of the algorithm:

1. Compute the canonical order of $G$ and the partial edge orientation.
2. Let $G' = G - v_1$; $G'$ is dart-shaped.
3. Split $G'$ into two dart-shaped graphs $G_1$ and $G_2$.
4. Draw $G_1$ and $G_2$ separately (recursively): We will prescribe the shapes within which they are drawn.
5. Combine the drawings of $G_1$ and $G_2$ into a single drawing.

We have seen already that steps 1 - 3 can be done. The difficulty of this algorithm lies in how to combine the drawings of $G_1$ and $G_2$. These two graphs share vertices, so if we draw $G_1$ first, this forces some of the vertices to be at fixed locations in the drawing of $G_2$. Since our algorithm works recursively, we hence generally need to allow that the positions of some vertices on the outerface of the graph to be drawn are fixed. Since some of the vertices are fixed, it is not possible to split the drawing region into simple regions, such as rectangles. Figure 3.7 illustrates how having fixed vertices can increase the complexity of the regions in which $G'$ is split: if $v$ is fixed on the left and $u$ on the bottom, then no orthogonal drawing is possible where $G_2$ is drawn in a rectangle.

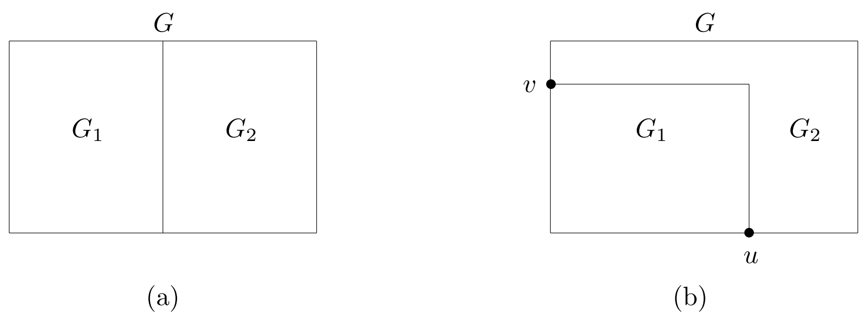



Figure 3.7: The drawing region of $G'$ is divided into two regions. (a) Without vertices in fixed position. (b) With vertex $v$ in fixed position.

Therefore, we must allow a drawing region for subgraphs that is more complex than a rectangle. We will use what we call a *T-staircase*, which is defined in Section 3.4. In order to draw subgraphs in it recursively, we need to break it apart into smaller T-staircases; we discuss this in Section 3.5. In Section 3.6 we describe exactly where on a T-staircase the vertices of $G'$ can be fixed, so that $G'$ can be drawn with correct face areas inside any T-staircase. We call this a *correct pinning*. We combine everything together and explain the choice of T-staircase in the outermost recursion in Section 3.7. Complete pseudocode will be given on pages 49 and 50.

## 3.4 T-staircases

In this section we define the shape called T-staircase. As we will see in the following sections, it is possible to draw any dart-shaped graph $G$ inside a T-staircase $T$ with area $A(G)$, respecting the given interior face areas of $G$.

**Definition 3** *A* T-staircase *is an x-monotone orthogonal polygon for which the upper chain consists of just one edge (the* top side*) and the lower chain consists of a descending staircase (the* left curve*), one horizontal edge (the* base*) and an ascending staircase (the* right curve*). Furthermore, all vertices of the polygon except the two bottommost ones are within distance* $\frac{1}{4t}$ *from the top, where $t$ is the length of the top side. See Figure 3.8 for an example.*

The *top $\varepsilon$-region* is the topmost region inside the T-staircase with height $\frac{1}{4t}$ (any value smaller than $\frac{1}{3t}$ would actually work.) The top $\varepsilon$-region has area at most $\frac{1}{4}$ since it is contained in a rectangle of width $t$ and height $\frac{1}{4t}$. All bends on the left and right curves lie inside the top $\varepsilon$-region. Thus, the segments of these curves outside the top $\varepsilon$-region are straight vertical lines. The *left* and *right stairs* are the portion of the left and right curves inside the top $\varepsilon$-region and the *left* and *right sides* are the segments of the left and right curves outside the top $\varepsilon$-region.

Let $h$ be the height of the left and right curves. The *left/right $\varepsilon$-region* is a $\frac{1}{4h} \times h$ rectangle adjacent to the left/right side, and inside the T-staircase.

Then, the sum of the area of the top, left and right $\varepsilon$-regions is at most $\frac{3}{4}$. We will later use the $\varepsilon$-regions to place all necessary bends. Since the area of these regions is smaller than the area of any face (recall that face areas are integers), it is possible to include any portion of the $\varepsilon$-regions in any face and still be able to draw it with correct area.

The *allowed segment* of the top side is the segment starting at the $x$-coordinate of the right side of the left $\varepsilon$-region and ending at the $x$-coordinate of the left side of the right $\varepsilon$-region. We will later see that vertices on the top side of a T-staircase will only be assigned to points along the allowed segment.

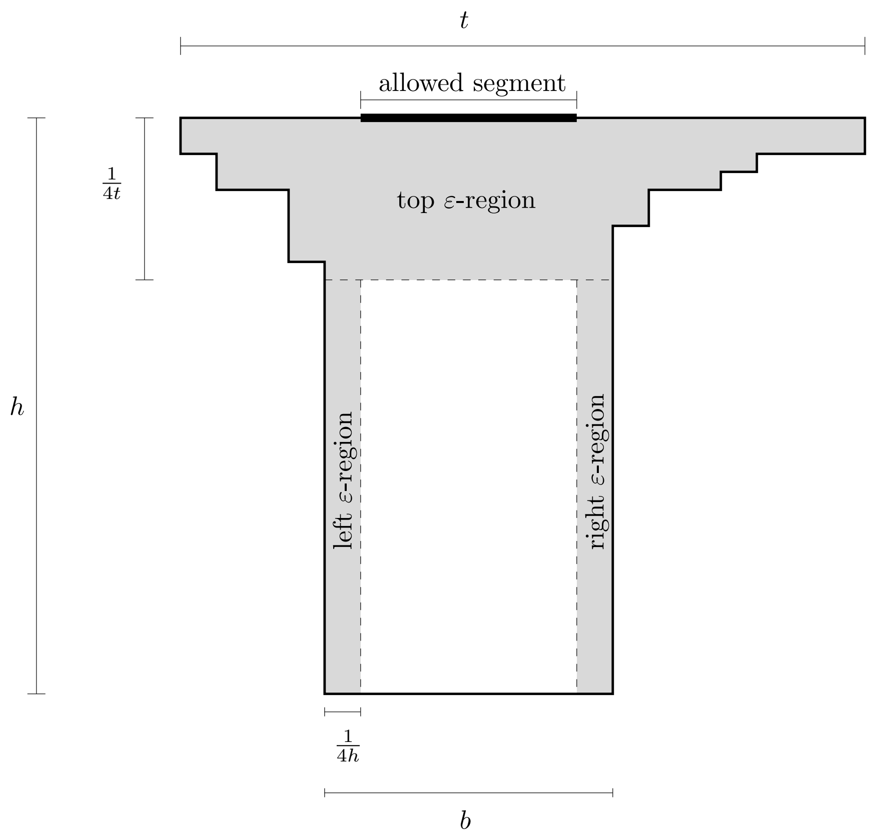



Figure 3.8: Example of T-staircase. Figure not to scale: The height of the $\varepsilon$-region is much smaller compared to the height of the T-staircase. A real example drawn to scale is shown in Figure 3.17 on page 52.

## 3.5 Decomposing T-staircases

The idea now is to break any dart-shaped graph $G$ into two pieces (as in Theorem 4), and place the pieces in a T-staircase $T$ recursively. To do so, we first must argue that $T$ can be divided into two T-staircases suitably.

**Lemma 7** *Let $T$ be a T-staircase and $A(T)$ its area. For any value $1 \leq A \leq A(T) - 1$, there exists a vertical line $l$ from a point $v$ on the allowed segment to a point $u$ on the base that divides $T$ into two T-staircases $T_l$ and $T_r$, of area $A$ and $A(T) - A$, respectively.*

**Proof:** To see that it is possible to choose the $x$-coordinate $X$ of $v$ and $u$ so that both sides of $l$ have correct area, imagine first that we choose $X$ to be the $x$-coordinate of the left end of the allowed segment, which is the same as the right side of the left $\varepsilon$-region. In this case, the area to the left of $l$ is all from the $\varepsilon$-regions and therefore smaller than 1, but $A$ is at least 1. If we now choose $X$ to be the $x$-coordinate of the right end of the allowed segment, the area to the right of $l$ will be smaller than 1, but $A(T) - A$ is at least 1. Therefore, since the area to each side of $l$ is continuous, by the mean value theorem, the correct value of $X$ must be between the left and right $\varepsilon$-regions. Both $T_l$ and $T_r$ are clearly T-staircases. See also Figure 3.9a. □

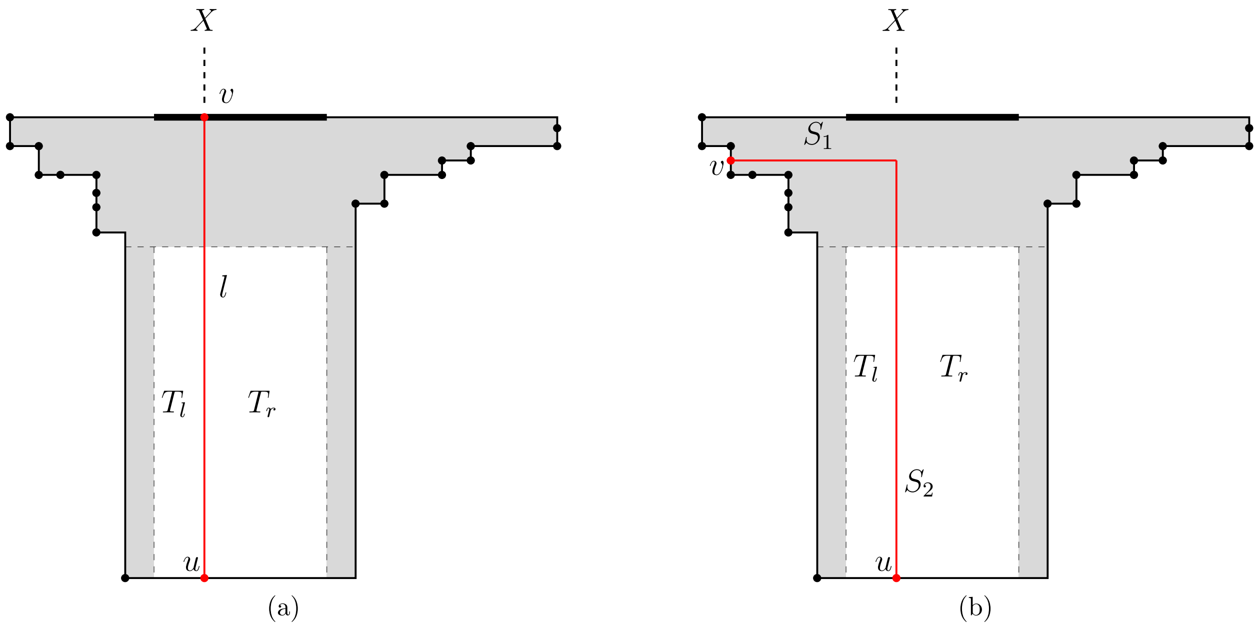



Figure 3.9: (a) Division of $T$, by a vertical line, into two T-staircases with prescribed areas, as in Lemma 7. (b) Division of $T$, by an orthogonal path, into two T-staircases with prescribed areas, as in Lemma 8.

**Lemma 8** *Let $T$ be a T-staircase. For any point $v$ on the interior of a vertical segment or a reflex corner of the left or right stairs and any value $1 \leq A \leq A(T) - 1$, there exists an orthogonal path $l$ from $v$ to a point $u$ on the base that divides $T$ into two T-staircases $T_l$ and $T_r$, of area $A$ and $A(T) - A$, respectively.*

**Proof:** We will only argue the case when $v$ is on the left stairs since the case when it is on the right is symmetric.

Connect $v$ and $u$ with a path $l$ as follows: Draw a horizontal line segment $S_1$ from $v$ to $u$'s $x$-coordinate $X$ (we will discuss how to obtain $X$ appropriately soon) and a vertical line segment $S_2$ from there to the base.

To see that we can select $X$ so that the area on both sides of $l$ is correct, imagine first that $X$ is the $x$-coordinate of the left end of the allowed segment, which is the same as the right side of the left $\varepsilon$-region. In this case the area to the left of $l$ is all contained in the $\varepsilon$-regions, thus smaller than 1, but $A$ is at least 1. Now, imagine that $X$ is the $x$-coordinate of the right end of the allowed segment. In this case the area to the right of $l$ is all contained in the $\varepsilon$-regions, thus smaller than 1, but $A(T) - A$ is at least 1. Therefore, by the mean value theorem, $X$ must lie somewhere between the left and right $\varepsilon$-regions.

Let $T_l$ be the shape to the left of $l$ and $T_r$ the shape to the right of $l$. Both $T_l$ and $T_r$ are clearly T-staircases, since the new bend on $l$ lies in the top $\varepsilon$-region. Since the length of the top side of $T_l$ and $T_r$ is at most the length of the top side of $T$, the height of the top $\varepsilon$-region of $T_l$ and $T_r$ is at least the height of the top $\varepsilon$-region of $T$. Thus, the stairs of $T$ lie inside the top $\varepsilon$-regions of $T_l$ and $T_r$. See Figure 3.9b for an example. □

**Lemma 9** *Let $T$ be a T-staircase. Let $u$ be a point on a vertical segment of the left stairs that is not on a convex corner, or let $u$ be the left endpoint of the allowed segment. Let $v$ be a point on a vertical segment of the right stairs that is not on a convex corner, or let $v$ be the right endpoint of the allowed segment. For any value $1 \leq A \leq A(T) - 1$, there exists an orthogonal path $l$ with at most 4 bends, from $u$ to $v$ that divides $T$ into a T-staircase $T'$ and a shape $B$, of area $A$ and $A(T) - A$, respectively.*

**Proof:** Depending on the position of $u$ and $v$, the following four cases apply:

i. The point $u$ is on a vertical segment of the left stairs and $v$ is on a vertical segment of the right stairs of $T$. We usually cannot draw a straight line from $u$ to $v$ because their positions are fixed, and the line between them may not be horizontal and/or may not create an area of appropriate size. Thus, we connect $u$ to $v$ with a path $l$ with four bends, as follows:

   - Draw a horizontal line segment $S_1$ from $u$ to the right until the boundary of the left $\varepsilon$-region.
   - Similarly, draw a horizontal line segment $S_5$ from $v$ to the left until the boundary of the right $\varepsilon$-region.
   - Segments $S_2[S_4]$ attach to $S_1[S_5]$ and go downward until some $y$-coordinate $Y$ (we will discuss $Y$ soon).
   - Segment $S_3$ is a horizontal segment at $y$-coordinate $Y$ that connects $S_2$ and $S_4$.

   See Figure 3.10a for an example.

ii. The point $u$ is on a vertical segment of the left stairs and $v$ is the right endpoint of the allowed segment of $T$. We connect $u$ to $v$ similarly as we did in case i. except that segment $S_5$ does not exist. Instead, $S_4$ connects $v_t$ with $S_3$, which is at $y$-coordinate $Y$. See Figure 3.10b for an example.

iii. The point $u$ is on the allowed segment of $T$ and $v$ on a vertical segment of the right stairs. This case is symmetric to case ii.

iv. Both $u$ and $v$ are points on the allowed segment of the top of $T$. In this case neither $S_1$ nor $S_5$ exist, and $S_2/S_4$ connects $u/v$ with $S_3$, which is at $y$-coordinate $Y$.

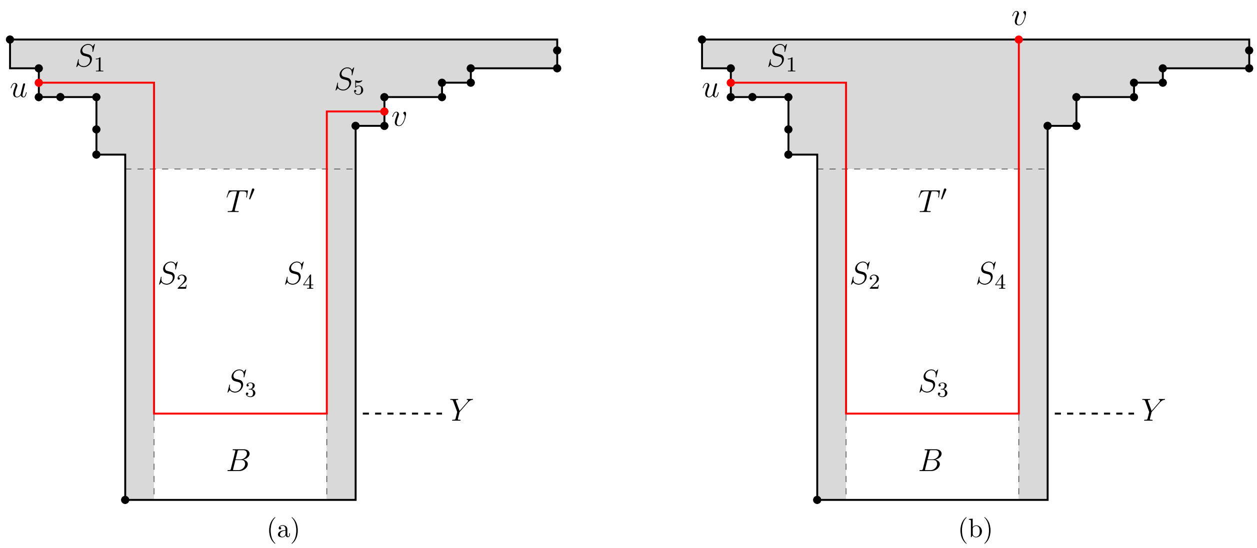



Figure 3.10: (a) Example of $B$ and $T'$ when $u$ is on the left stairs and $v$ on the right stairs. (b) Example of $B$ and $T'$ when $u$ is on the left stairs and $v$ on the top.

In any of the cases above, we can see that $T'$ is a T-staircase since the bends, if any, are in the top $\varepsilon$-region.

To see that we can draw $l$ so that the area of $T'$ is correct, imagine we use $Y = 0$. Then the region below $l$ is contained inside the $\varepsilon$-regions and has area smaller than 1, but $A(T) - A$ is at least 1. So $Y = 0$ is too small. Imagine on the other hand we used $Y = \min\{y(u), y(v)\}$. Since $u$ and $v$ are inside the top $\varepsilon$-region, all of $l$ would be inside the top $\varepsilon$-region and the region above has area smaller than 1, but $A$ is at least 1. Therefore, by the mean value theorem, there exists some $0 < Y < \min\{y(u), y(v)\}$ such that the area of the region above $l$ is exactly $A$. □

## 3.6 Pinning dart-shaped graphs to T-staircases

Now, we will discuss under which constraints it is possible to guarantee that a dart-shaped graph with prescribed interior face areas can be drawn inside a T-staircase:

**Definition 4** *Let $G$ be a dart-shaped graph and $T$ be a T-staircase that has area $A(G)$. A partial assignment of outerface vertices of $G$ to points on the boundary of $T$ is called a* correct pinning *of $G$ to $T$ if the following constraints are satisfied:*

*C1.* *A vertex $v \neq c_l, c_t$ on $P_l$ may be assigned a point that lies on the left stairs. If $deg(v) = 3$, then it has not been assigned to a point that lies on the interior of a horizontal segment of the stairs or a convex corner. See also Figure 3.11.*

*A vertex $v \neq c_r, c_t$ on $P_r$ may be assigned a point that lies on the right stairs. If $deg(v) = 3$, then it has not been assigned to a point that lies on the interior of a horizontal segment of the stairs or a convex corner.*

*C2. The order of points assigned to vertices along the boundary of $T$ corresponds to the order of vertices along the outerface of $G$. Any unassigned vertex $v \neq c_l, c_r, c_t$ with $deg(v) = 3$ could be assigned to a point on the allowed segment or the interior of the base such that the order is respected.*

*C3. There are at most 6 bends that are on $T$, i.e., all except at most 6 corners of $T$ have a vertex assigned to them. Moreover, the bends can only be at: the endpoints of the top side, the endpoints of the base, and the bottommost corner of the left and right stairs. See Figure 3.12 for an example.*

*C4. If there is a bend at the bottommost corner of the left or right stairs, then there are no vertices assigned to the vertical segment below it, inside the top $\varepsilon$-region.*

We now come to a crucial lemma: Our conditions on pinning were chosen such that the T-staircase can be split suitably and the subgraph is pinned correctly. Therefore, a recursive algorithm is possible.

**Lemma 10** *Let $G$ be a dart-shaped graph with a correct pinning to a T-staircase $T$.*

1. *If $G$ can be split by a vertical path into two dart-shaped graphs $G_l$ and $G_r$ (Theorem 4(a)), then $T$ can be split into two T-staircases $T_l$ and $T_r$ with area $A(G_l)$ and $A(G_r)$, respectively (Lemmas 7 and 8), such that $G_l$ can be pinned correctly to $T_l$ and $G_r$ to $T_r$.*
2. *If $G$ can be split by a horizontal path into a dart-shaped graph $G_t$ and a face $G_b$ (Theorem 4(b)), then $T$ can be split into a T-staircase $T'$ and a shape $B$ with area $A(G_t)$ and $A(G_b)$, respectively (Lemma 9), such that $G_t$ can be pinned correctly to $T'$.*

**Proof:** We have already indicated all crucial theorems and results. It remains to verify that the vertices that belong to both subgraphs can be pinned correctly on the path that divides the T-staircase. This requires a fairly tedious case analysis (depending on where the endpoints of the dividing path are pinned), as well as going through all the conditions of a correct pinning. The reader is encouraged to skip these details on first read, and go to Section 3.7 on page 45, where we put everything together in one algorithm.

1. Assume $G$ can be divided into two dart-shaped graphs $G_l$ and $G_r$ by a vertical path from a vertex $u \neq c_l, c_r$ in $P_b$ to a vertex $v \neq c_l, c_r, c_t$ in $P_l \cup P_r$ (Lemma 5). Assume that $v \in P_l$; the other case is similar.

   Since $G$ is pinned correctly to $T$, vertex $v$ could have been assigned to points on the left stairs. If $v$ has not been assigned to a point yet, choose a suitable place for it as

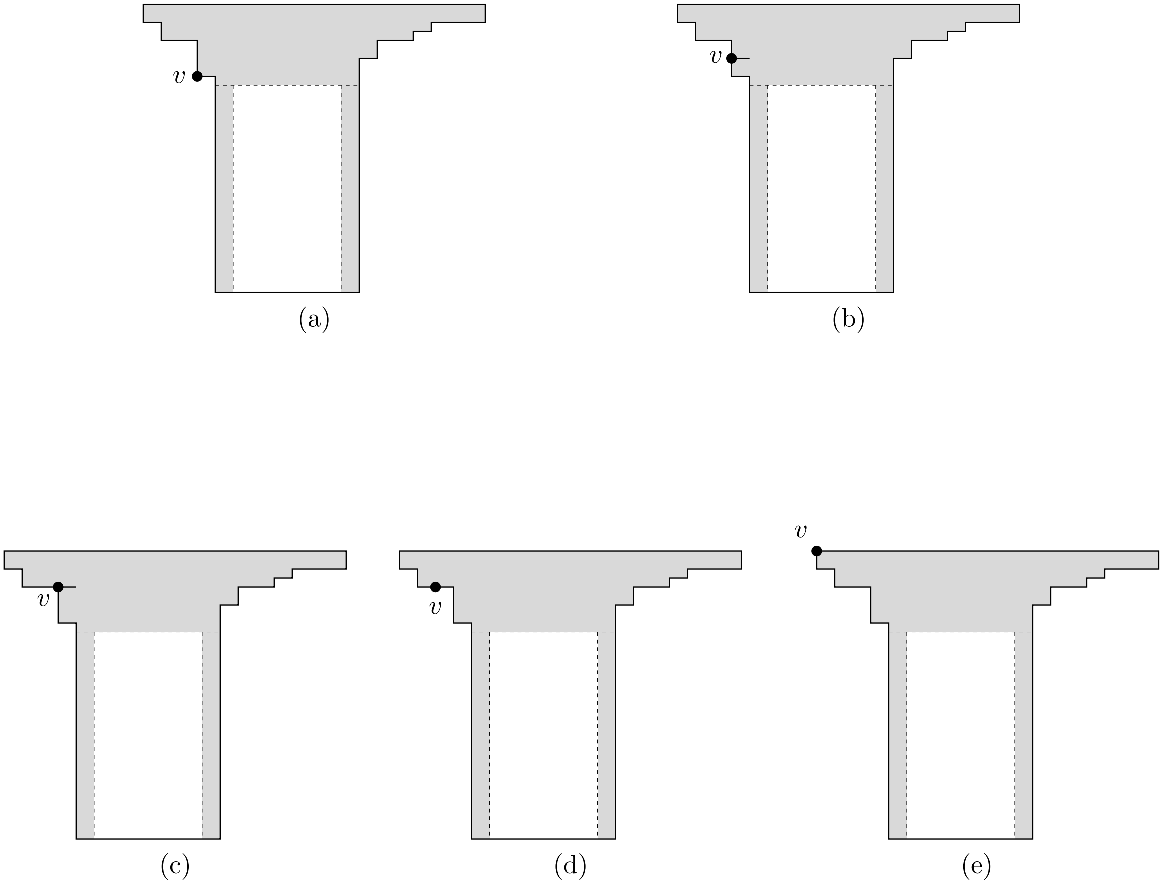



Figure 3.11: Possible correct positions for $v$ in $T$. The positions on figures (a), (d) and (e) are only allowed if deg($v$)=2.

follows: traverse the outerface forward and backwards from $v$, until finding a vertex in each direction that has already been pinned to $T$. Then place $v$ anywhere between those two vertices. If possible, assign $v$ to the allowed segment, since that will reduce the number of bends, but do not fix its $x$-coordinate yet. If it is not possible to assign it to the allowed segment, fix its position on a vertical segment of the stairs.

Then, there are three cases:

i. $v$ has been assigned to the allowed segment of $T$.

By Lemma 7, it is possible to find a value for the $x$-coordinate of $v$ in $T$, such that a vertical line $l$ from $v$ to the base of $T$ divides it into two T-staircases, $T_l$ to the left of $l$ with area $A(G_l)$, and $T_r$ to the right of $l$, with area $A(G_r) = A - A(G_l)$. Fix the position of all vertices in $P_v$ along $l$, in order, and inside the top $\varepsilon$-region. See Figure 3.13a.

Now verify that $G_l$ is pinned correctly to $T_l$ (the case of $G_r$ is similar):

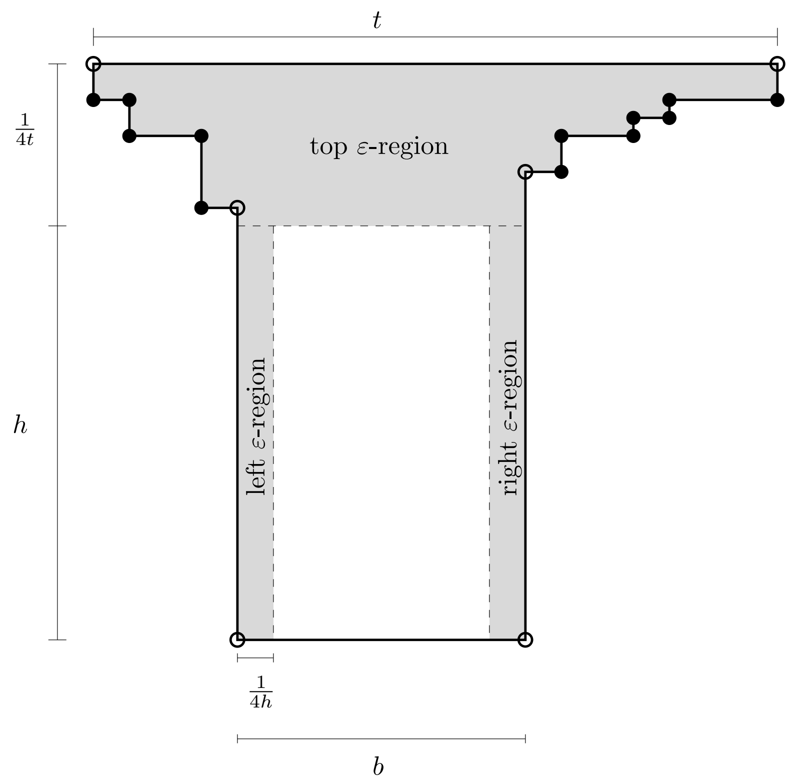



Figure 3.12: Possible bends in a T-staircase if a dart-shaped graph is pinned correctly to it. Filled points represent corners where a vertex must have been assigned.

C1. All vertices on $P_v$, except for $u$ have been fixed along $l$ (which is the right curve for $T_l$), inside the top $\varepsilon$-region, on a vertical segment. Vertex $u$ has been assigned to the base. This is correct since $u$ is the right corner of $T_l$. Thus, this condition holds.

C2. All vertices on $P_v$ were assigned in order. Hence the only unpinned vertices on the outer-face are those that were unpinned in $G$, and hence they can be pinned to the allowed segment or the base while maintaining the order around the outer-face.

C3. No bends were created.

C4. The left stairs are the same as the left stairs of $T$ where no new vertices were assigned, and the right stairs are a segment of $l$, which has no bends.

ii. $v$ has been assigned to a point on the left stairs of the T-staircase.

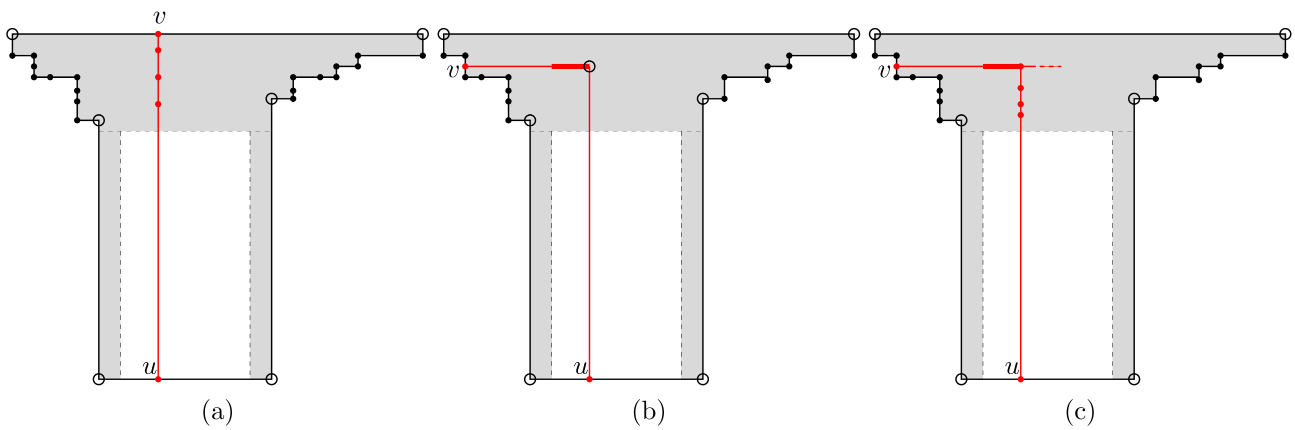



Figure 3.13: (a) Pinning of vertices to $T_l$ and $T_r$ in case i. (b) Pinning of vertices to $T_l$ and $T_r$ in case ii if there is no vertex on $l$ with an edge to the right. (c) Pinning of vertices to $T_l$ and $T_r$ in case ii if there is a vertex on $l$ with an edge to the right. In all three cases circled points represent possible bends and thick line segments represent possible points where vertices that were not fixed yet can be pinned.

By condition (C1), $v$ cannot be on the interior of a horizontal segment or on a convex corner, since $deg(v) = 3$.

By Lemma 8, it is possible to draw an orthogonal path $l$, with one bend, from $v$ to a point on the base that divides $T$ into two T-staircases, $T_l$ to the left of $l$ with area $A(G_l)$, and $T_r$ to the right of $l$, with area $A(G_r) = A - A(G_l)$.

Pin some of the vertices on $P_v$ as follows: Traverse the path from $v$ backwards to $u$ to find a vertex with an edge to the right of $P_v$ (i.e., in $G_r$ but not in $G_l$). There are two cases:

(a) There are no vertices between on $P_v$ with an edge to the right.
In this case, all vertices $\neq u, v$ on $l$ remain unassigned. They will later be drawn on the interior of the horizontal segment of $l$, but we do not know their $x$-coordinates yet.

(b) There is at least one vertex on $P_v$ with an edge to the right.
Let $w$ be the first (closest to $v$) such vertex. All vertices between $v$ and $w$ on $P_v$ remain unassigned, $w$ is assigned to the bend on $l$, and all vertices from $w$ to $u$ are pinned, in order, on the vertical segment inside the top $\varepsilon$-region. Note that $w$ has degree 2 in $G_l$ since it has degree 3 in $G$, two edges in $P_v$, and one to the right.

Now, we analyze why $G_l/G_r$ are pinned correctly to $T_l/T_r$ (see Figures 3.13b and 3.13c for an illustration):

C1. All vertices except for $u$ assigned in this step are in the vertical segment of $l$, inside the top $\varepsilon$-region. This segment represents the right stairs in $T_l$ and part of the left stairs in $T_r$. The only vertex at a convex corner is $w$ in $T_l$ (in case (b)), but $w$ has degree 2 in $G_l$. Vertex $u$ has been assigned to the base. This is correct since $u$ is the right corner of $T_l$ and the left corner of $T_r$. Thus, this condition holds.

C2. All vertices pinned in this step were assigned in order. The only unassigned vertices were unassigned in $G$, or are on the path between $w$ and $v$ (if $w$ exists). The latter can be pinned to the allowed segment for $T_l$ (hence the conditions are satisfied for $G_l$), and they have degree 2 in $G_r$ and hence need not satisfy any condition for $T_r$.

C3. In $T_l$, the left stairs are the lower part of left stairs of $T$, thus all the bends except (maybe) for the bottommost one have a vertex assigned. If the bottommost corner of the left stairs of $T$ is a bend, there are no vertices assigned below it, so $v$ must be above it. Thus, it is not possible to create a new bend below it. The right stairs is a segment of $l$, thus there are no bends.

In $T_r$, the left stairs are the upper part of the left stairs of $T$ in addition to part of the orthogonal path $l$. As discussed for $T_l$, there is no bend on this part of the left stairs of $T$. Thus, the only possible bend is the one of $l$. The right stairs are the same as the right stairs of $T$, thus it can only have a bend at the bottommost step.

C4. In $T_l$, the left stairs are the lower part of left stairs of $T$, thus there are no vertices assigned below the bend (if any). The right stairs have no bends.

In $T_r$, the left stairs are the upper part of the left stairs of $T$ in addition to part of the orthogonal path $l$. There are two options: If there was a vertex $w$ with an edge to the right, it was fixed at the bend of $l$, thus there is nothing to prove. On the other hand, if there was no such vertex, all vertices on $l \neq u, v$ remain unassigned, so the condition holds for the left stairs. The right stairs are the same as the right stairs of $T$, thus there are no vertices assigned below the bend (if any).

2. Assume $G$ can be divided into a dart-shaped graph $G_t$ and a single face $G_b$ by a horizontal path $P_h$ from a vertex $c'_l \neq c_l, c_t \in P_l$ to a vertex $c'_r \neq c_r, c_t \in P_r$ (Lemma 6.)

   Since $G$ is pinned correctly to $T$, $c'_l$ may have been pinned on the left stairs and $c'_r$ may have been pinned on the right stairs. If they have not been assigned to a point yet, we can place them either on the top side or on a vertical segment of the stairs, as long as they appear in the same order as they do in $G$, and we pick points for them as explained in case 1.

   There are four cases for the positions of $c'_l$ and $c'_r$. For any of them, by Lemma 9 it is possible to draw an orthogonal path $l$, from $c'_l$ to $c'_r$, that divides $T$ into a T-staircase

$T'$ and a shape $B$ with prescribed face areas. We will not assign any vertices $\neq c_l', c_r'$ to positions in $T'$.

We will only argue that $G_t$ is pinned correctly to $T'$ in the case where $c_l'$ is assigned to a point $v_l$ on a vertical segment of the left stairs and $c_r'$ to a point $v_r$ on a vertical segment of the right stairs of $T$; the other cases are similar (and even easier). See Figure 3.14a for an example.

C1. Since $P_l'$ and $P_r'$ are subpaths of $P_l$ and $P_r$, respectively and the top side and stairs of $T'$ are part of the top side and stairs of $T$, this condition holds for all vertices placed before this step. The only vertices that might have been fixed in this step are $c_l'$ and $c_r'$. These vertices become the left and right corners in $T'$. Thus this condition holds.

C2. Vertices other than $c_l'$ and $c_r'$ on $P_h$ (if any), have not been assigned any point on $T$, but could be pinned to the bottom edge. All other vertices on the outerface of $G_t$ were also on the outerface of $G$ and the assignment of points is unchanged, so the condition holds.

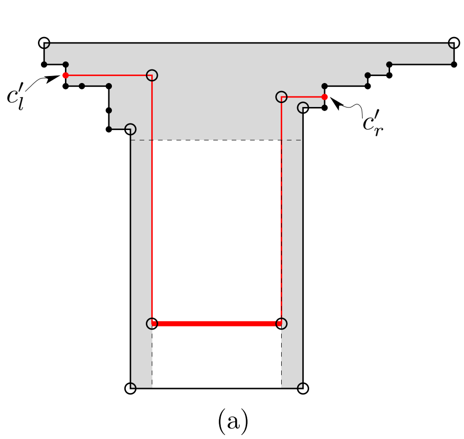
C3. Since (C3) and (C4) hold for $T$, $c_l'$ and $c_r'$ can only be below the bottommost corner if it is not a bend. The left and right stairs of $T'$ are part of the left and right stairs of $T$ in addition to the segments of $l$, thus they can only have bends at the corners on the top or along $l$. The segments of $l$ form the base and the bottommost steps of both stairs; the bends in $l$ hence are allowed.

C4. Vertices other than $c_l'$ and $c_r'$ on $P_h$ (if any), had not been assigned any point on $T$. Thus, there was no vertex pinned below the bottommost corners, inside the $\varepsilon$-region. □

## 3.7 Putting it all together

Finally, we have enough tools to prove that any 3-connected planar graph with maximum degree three has an orthogonal drawing with given interior face areas and at most 4 (resp. 8) bends per edge (resp. face). We will first show in Lemma 11 how to draw a dart-shaped graph inside a T-staircase using Lemmas 5 to 10 recursively. By Lemma 4, any 3-connected planar graph $G$ with maximum degree three is a dart-shaped graph. Then, in Theorem 5 we will find a T-staircase to which $G$ can be pinned correctly.

**Lemma 11** *Let $G$ be a dart-shaped graph that is pinned correctly to a T-staircase $T$ of area $A(G)$. $G$ has a drawing inside $T$ that respects the pinned vertices and has the following properties:*



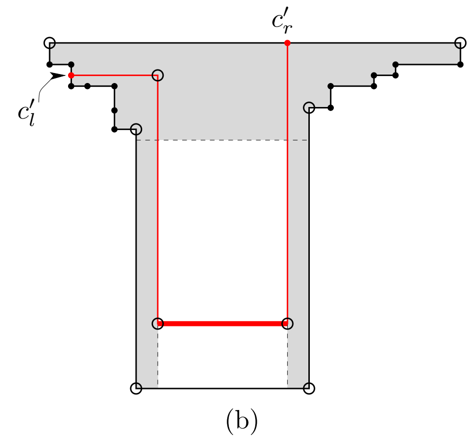




Figure 3.14: (a) Pinning of vertices to $T'$ where $c_l'$ is pinned to the left stairs and $c_r'$ to the right stairs. (b) Pinning of vertices to $T'$ where $c_l'$ is pinned to the left stairs and $c_r'$ to the top side. In both cases, circled points represent possible bends and thick line segments represent possible points where vertices that were not fixed yet can be pinned.

*P1. Every interior face $f$ of $T$ has area $A(f)$.*

*P2. Every edge has at most 4 bends.*

*P3. Every face has at most 8 bends.*

**Proof:** We will recursively draw $G$ inside $T$. We have two cases:

(a) There is at least one vertex $u$ on $P_b$ between $c_l$ and $c_r$. In this case, by Lemma 5 we can divide $G$ into two dart-shaped subgraphs $G_l$ and $G_r$; by Lemmas 7 and 8 we can divide $T$ into two T-staircases: $T_l$ with area $A(G_l)$, and $T_r$ with area $A(G_r)$. Furthermore, by Lemma 10, $G_l$ can be pinned correctly to $T_l$ and $G_r$ to $T_r$. Then, we can recursively apply Lemma 11 on both $G_r$ and $G_l$.

(b) There are no vertices on $P_b$ between $c_l$ and $c_r$. In this case, $G$ could have a single interior face (which then has correct area). Otherwise, by Lemma 6 we can divide $G$ into a face $G_b$ and a dart-shaped subgraph $G_t$; by Lemma 9 we can divide $T$ into a region $B$ of area $A(G_b)$ and a T-staircase $T'$ with area $A(G_t)$; and by Lemma 10, $G_t$ can be pinned correctly to $T'$. Then, we can recursively apply Lemma 11 on $G_t$. We do not recurse on $G_b$ since it is already a face and has correct area.

This proves property P1. We now need to argue that each edge has at most 4 bends. Note that any edge is contained within a path created during the algorithm. The worst number of bends for a path occurs when connecting a horizontal path from a vertex on the left side to a vertex on the right side of the T-staircase (Lemma 9 case i.). This path has 4 bends.

Now we argue that each face has at most 8 bends. Since $G$ is pinned correctly to $T$, according to the definition $T$ has at most 6 bends. Moreover, by Lemma 10, it is possible to recursively divide $T$ into T-staircases that have at most 6 bends. In the base case, these T-staircases are single faces and in the worst case they have 6 bends. The only face not covered by this argument is the face $B$ below $l$ that we create in Lemma 9. Going through the cases in the proof of this lemma, the maximum number of bends is 8. Specifically, in case i., there are 4 bends on $l$, and at most 4 bends of $T$ belong to $B$, since the top corners of $T$ are not in $B$. For all other cases, we may have more top corners of $T$ in $B$, but in exchange the number of bends on $l$ decreases, leading to a total of at most 8 bends in all cases. See Figure 3.14 for an example.

□

Lemma 11 gives the main ingredients to prove our theorem:

**Theorem 5** *Any 3-regular 3-connected planar graph $G = (V, E)$ can be drawn orthogonally with given interior face areas, at most 4 bends per edge, and at most 8 bends per face.*

**Proof:** Recall that the vertices of $G$ can be labeled $v_1, v_2, \ldots, v_n$, according to a canonical order of $G$. Let $G'$ be $G - v_1$ and orient $G'$ so that it is dart shaped (as in Lemma 4.) Note that the corners of $G'$ are $c_l = v_j$, $c_r = v_2$ and $c_t = v_n$.

Let $T$ be any rectangle with area $A = A(G')$. Clearly $T$ is a T-staircase. Assign some of the vertices of $G'$ to points in $T$ as follows:

Assign $v_j$ to the bottom left corner, $v_2$ to the bottom right corner, and $v_n$ to the left top corner of $T$ (see Figure 3.15a.) Since $v_j$, $v_2$ and $v_n$ are adjacent to $v_1$ in $G$, each of them has degree 2 in $G' = G - v_1$. Pin all vertices on the vertical path from $v_j$ to $v_n$ to the left side of $T$, in order, and inside the $\varepsilon$-region. It is straightforward to verify that all vertices have been pinned correctly to $T$.

Then, by Lemma 11, we can recursively draw $G'$ inside $T$, respecting given interior face areas, and with at most 4 bends per edge and 8 bends per face.

Finally, we need to add the vertex $v_1$ and its three incident edges. Doing this adds two interior faces. Let $f_1$ be the face formed by $(v_1, v_j), (v_1, v_n)$ and the path from $v_j$ to $v_n$, and $f_2$ be the face formed by the cycle $v_1, v_2, \ldots, v_j$.

Draw $v_1$ at $v_j$'s $y$-coordinate so that the edge $(v_1, v_j)$ is horizontal. Then, choose $v_1$'s $x$-coordinate so that the area of $f_1$ is correct if the edge from $v_1$ to $v_n$ is drawn as a path with one bend. Finally, draw the edge $(v_1, v_2)$ as an orthogonal path with two bends, so that the area of $f_2$ is correct. See Figure 3.15b. □

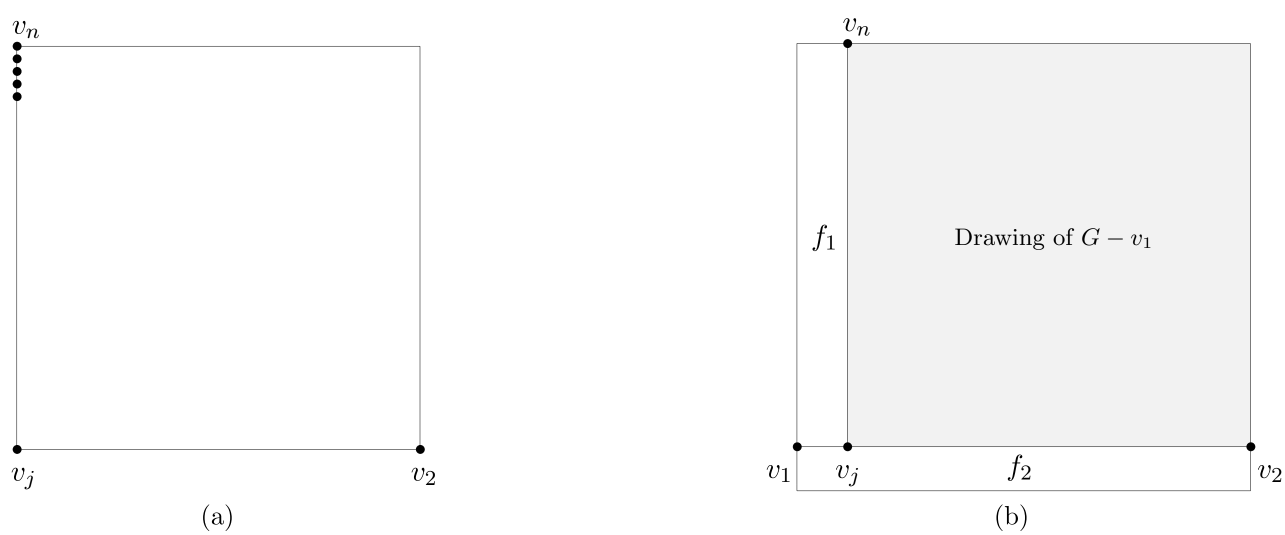



Figure 3.15: (a) Pinning of $v_2$, $v_j$, $v_n$ and all vertices on path from $v_j$ to $v_n$ to $T$. (b) Adding $v_1$ to $T$.

The following pseudocode for Draw($G$) and RecursivelyDraw($G$, $T$) summarizes our algorithm:

**Algorithm 1** Draw($G$).

Draw ($G$) {

1. Compute canonical ordering of $G$, $v_1$, $v_2$, $\ldots$, $v_n$.
2. Let $G' = G - v_1$.
3. Let $T =$ rectangle with area $A(G')$.
4. Pin $v_j$, $v_2$, $v_n$ to bottom-left, bottom-right, top-left corner of $T$ and all vertices on $P_l$, in order, on the left side of $T$.
5. RecursivelyDraw($G'$, $T$).
6. Add $v_1$. // Theorem 5.

}

## 3.8 An example

To illustrate the algorithm, we give in Figures 3.16 and 3.17 an example of a graph $G$ and an equifacial (i.e., all its interior faces have the same area) orthogonal drawing produced by this algorithm. All figures are to scale. Figure 3.16 shows intermediate steps of the algorithm, where a vertical and a horizontal path is drawn, respectively. Figure 3.17 shows the final drawing of $G$.

## 3.9 Rational coordinates

This section shows that the coordinates of the vertices and edges produced by this algorithm are rational.

**Lemma 12** *Let $T$ be a $T$-staircase with rational coordinates for all its vertices and edges, and $G$ a dart-shaped graph with its outerface vertices pinned on $T$. $G$ can be drawn orthogonally inside $T$ and respecting given rational interior face areas, such that the coordinates of all vertices and bends in the drawing are rational.*

**Proof:** By Lemma 11, $G$ can be drawn recursively inside $T$. In each recursive call, one of two splitting operations is made:

**Algorithm 2** RecursivelyDraw$(G, T)$.

RecursivelyDraw ($G$, $T$) {
// G has dart-shaped orientation with corners $c_l$, $c_r$, $c_t$.
// G is pinned correctly to the T-staircase $T$. $T$ has area $A(G)$.
If the bottom path of $G$ has at least 2 edges {
Let $u$ be a vertex $\neq$ $c_l$, $c_r$ on the bottom path.
Let $P_v$ be the directed (vertical) path from $u$, ending at vertex $v$ that is the first vertex on the outerface.
Let $G_l$ and $G_r$ be dart-shaped graphs into which $P_v$ divides $G$ (Lemma 5.)
Let $A(G_l)$ and $A(G_r)$ the area of $G_l$ and $G_r$, respectively.
If $v = v_i$ has not been pinned to $T$ yet {
// v will be pinned on the top side.
Let $l$ be a vertical line, from point $p_u$ to point $p_v$ that divides $T$ into two T-staircases $T_l$ and $T_r$ of area $A(G_l)$ and $A(G_r)$, respectively (Lemma 7.)
Pin $u$ to $p_u$, $v$ to $p_v$ and all vertices $v_j$ on $P_v$, to $T$ on line $l$ (Lemma 10.)
} else {
// $v_i$ has been pinned to the stairs, say at point $p_{v_i}$.
Let $l$ be an orthogonal path starting at $p_u$, with one bend $b$ and ending at $p_{v_i}$ that divides $T$ into two T-staircases $T_l$ and $T_r$ of area $A(G_l)$ and $A(G_r)$, respectively (Lemma 8.)
Pin $u$ to $p_u$.
If there exists a vertex on $P_v$ with an edge to the right {
Let $w$ be the closest to $v$. Pin $w$ to the bend of $l$ and all vertices $v_j$ on $P_v$ between $u$ and $w$ to the vertical segment of $l$ (Lemma 10.)
}
}
RecursivelyDraw($G_l$, $T_l$) (Lemma 11.)
RecursivelyDraw($G_r$, $T_r$) (Lemma 11.)
}
Else If $G$ has only one interior face return.
Else {
Let $G_b$ be the interior face of $G$ adjacent to $c_l$ and $c_r$.
Let the vertical paths of $G_b$ be from $c_l$ to $c_l'$ and from $c_r$ to $c_r'$.
Let $P_h$ be the horizontal path of $G_b$ from $c_l'$ to $c_r'$.
Let $G_t$ be the dart-shaped graph with base $P_h$ that contains $c_t$ (Lemma 6.)
Let $p_l$ and $p_r$ be the points where $c_l'$ and $c_r'$ are pinned. Pin them if they have not been pinned yet (Lemma 10.)
Let $l$ be an orthogonal path starting at $p_l$ and ending at $p_r$ that divides $T$ into a shape $B$ of area $A(G_b)$ and a T-staircase $T'$ of area $A(G_t)$ (Lemma 9.)
RecursivelyDraw($G_t$, $T'$) (Lemma 11.)
}
}

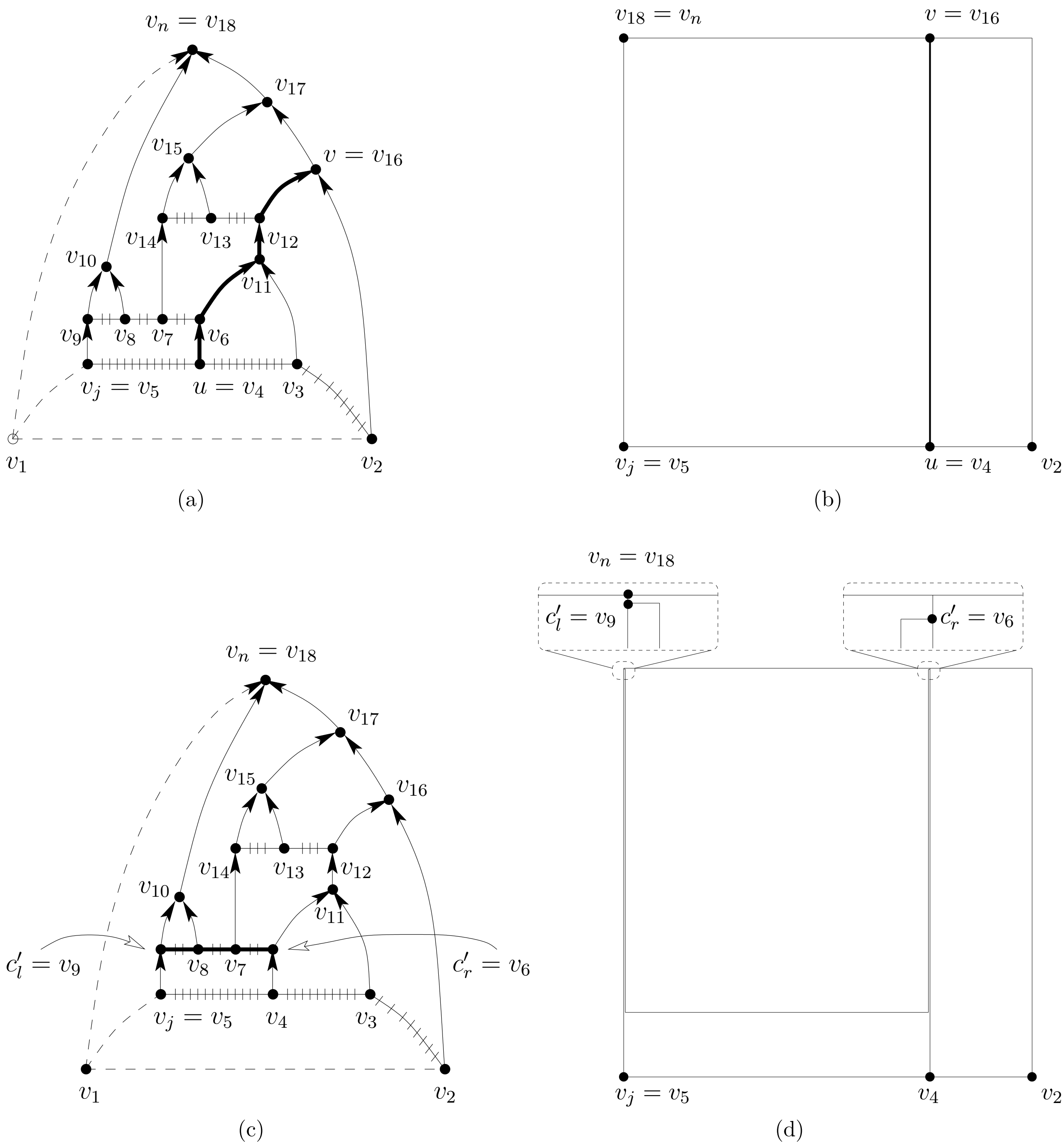



Figure 3.16: (a) $G'$ split by a vertical path. Hashed lines represent horizontal edges, and dashed lines represent edges that were deleted by the algorithm. (b) Vertical path drawn inside $T$. (c) $G'$ split by a horizontal path. (d) Horizontal path drawn inside $T_l$.

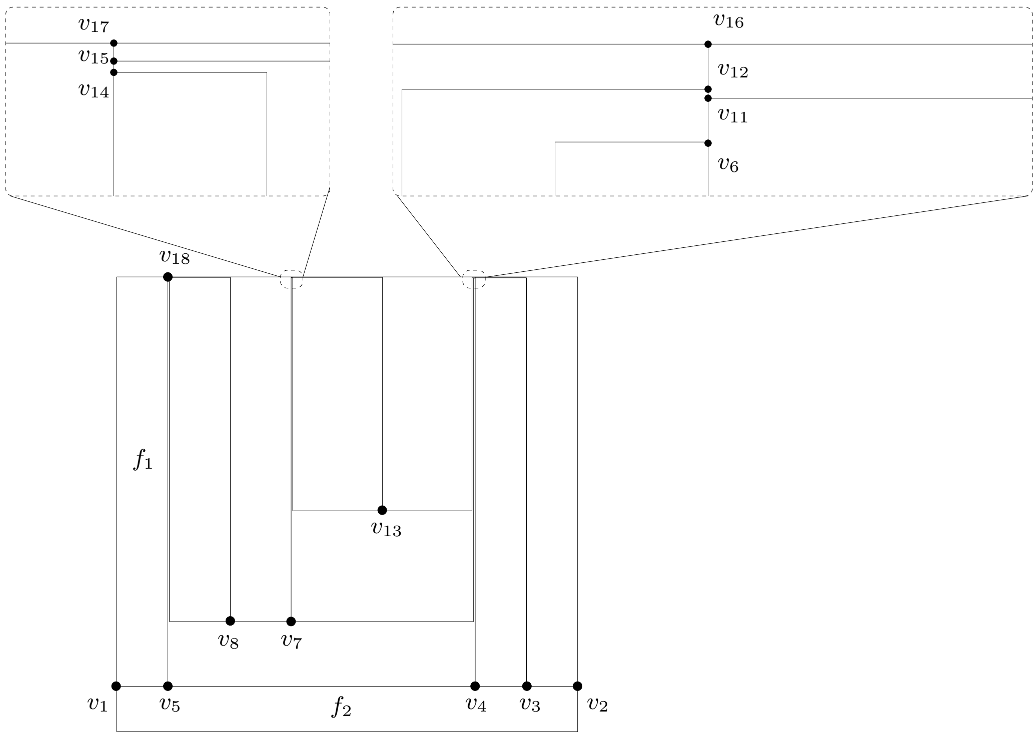



Figure 3.17: Equifacial orthogonal drawing of $G$.

(a) If there is at least one vertex $u$ on $P_b$ between $c_l$ and $c_r$, there are three cases, depending on where $v$ was assigned:

i. $v$ was assigned to a point on the allowed segment on the top side of $T$ (Algorithm 2, lines 6 and 7.)

In this case, it follows immediately that the $y$-coordinate of both $u$ and $v$ is rational. To see that it is also true for their $x$-coordinate $X$, divide the T-staircase to the left of $l$, $T_l$, into rectangles, by drawing a vertical segment from each bend on the stairs to the top side. See also Figure 3.18a. Let $R$ be the rectangle adjacent to $l$. Since the coordinates of all the bends in $T$ are rational, all rectangles different from $R$ have rational area. The area of $R$ is $A(T_l)$ minus the area of the other rectangles, hence rational and then the height and width of $R$ are also rational. If $x_0$ is the $x$-coordinate of the left most point of the base of $T$, $X$ equals the width of $R$ plus $x_0$. Thus, $X$ is rational.

ii. $v$ was assigned to a point on a vertical segment or a non-convex corner of the left stairs (Algorithm 2, lines 9 – 14.) Let $(x_v, y_v)$ be the coordinates of $v$, and $(x_b, y_v)$ the coordinates of the bend of $l$. If $v$ was pinned already, it has rational coordinates. If it was not, the position to pin it is selected according to Lemma 10. In this case, since $x_v$ is the same as the $x$-coordinate of a bend of $T$, it is rational. Also, we can pick $y_v$, as long as $v$ is placed correctly, as explained in Lemma 10. Thus, we can pick $y_v$ to be rational. To see that $x_b$ is also rational, proceed in the same fashion as in case i. Divide $T_l$ into rectangles, this time with a vertical segment from each of the bends below $v$ on the left stairs, to the horizontal segment of $l$. Each of those rectangles has rational area, and therefore $x_b$ is also rational. See also Figure 3.18b.

iii. $v$ was assigned to a point on a vertical segment of the right stairs. This case is symmetric to ii.

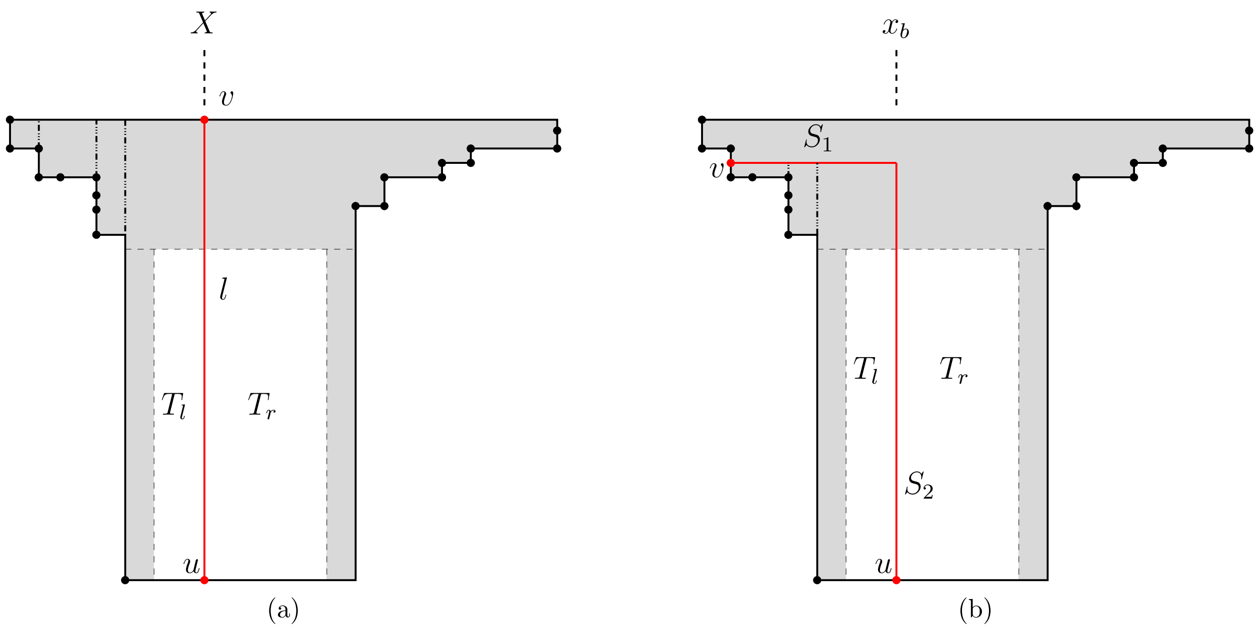



Figure 3.18: All vertices and bends have rational coordinates when splitting $G$ with a vertical path: (a) $X$ is rational when $v$ is assigned to the top side of $T$. (b) $x_b$ is rational when $v$ is assigned to the left stairs of $T$.

(b) If there are no vertices on $P_b$ between $c_l$ and $c_r$ (Algorithm 2, lines 20 – 28), there are four cases, depending on where $c_l'$ and $c_r'$ have been assigned. Vertex $c_l'$ can be either on the interior of a vertical segment of the left stairs, on a reflex corner, or on the allowed segment, and $c_r'$ can be either on the interior of a vertical segment of the right stairs, on a reflex corner, or on the allowed segment. We will only argue that the coordinates of $c_l'$ and $c_r'$ and all bends of $l$ are rational for the case when $c_l'$ is on the interior of a vertical segment of the left stairs and $c_r'$ is on the allowed segment; the other cases are very similar.

Let $(x_l', y_l')$ and $(x_r', y_r')$ be the coordinates of $c_l'$ and $c_r'$, respectively. Let $(x_1, y_1)$, $(x_2, y_2)$ and $(x_3, y_3)$ be the coordinates of the three bends of $l$. See also Figure 3.19. The following coordinates are rational: $x_l'$, because it is the $x$-coordinate of a bend of $T$; $y_l'$, because we can pick it to be rational; $x_1$, because it is the $x$-coordinate of the rightmost side of the left $\varepsilon$-region; $y_1$ is the same as $y_l'$; $x_2$ is the same as $x_1$; $x_3$ is the $x$-coordinate of the leftmost side of the right $\varepsilon$-region; $x'_r$ is the same as $x_3$; and $y'_r$, because it is the $y$-coordinate of the top side.

Therefore, it is left to prove that $y_2$ and $y_3$ (which are the same) are rational. This argument is very similar to the ones in case (a). Divide $B$ into three pieces by extending $S_2$ and $S_4$ to the base of $T$. Then $B$ is divided into rectangles that all have rational areas since their corners have rational coordinates; with the exception of rectangle $B'$ below $S_3$. But since $B$ has rational rational area, $B'$ must have rational area, and so $S_3$ has a rational $y$-coordinate. □

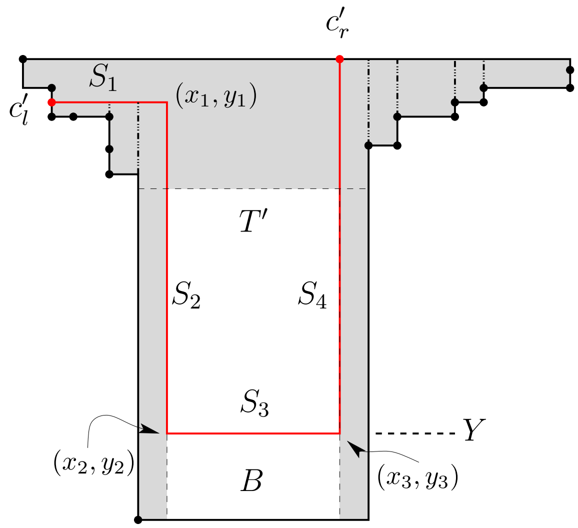



Figure 3.19: All vertices and bends have rational coordinates when splitting $G$ with a horizontal path.

A natural question is whether the size of the denominator can be bounded, similarly as we did for planar 3-trees in Section 2.3. We tried to prove such a bound (guessing it to be $n!$), but failed. The main difficulty lied on finding the correct form of $X$ and $Y$, so that a proof by induction could go through. We leave this for future research.

## 3.10 Time complexity

In this section, we prove that our algorithm creates orthogonal drawings for 3-connected planar graphs with maximum degree 3 that respect given interior face areas in $O(n \log n)$ time.

For the analysis, assume that it is possible to access the number in the canonical order, the coordinates and the neighbours of a vertex in constant time. This is true by using hash

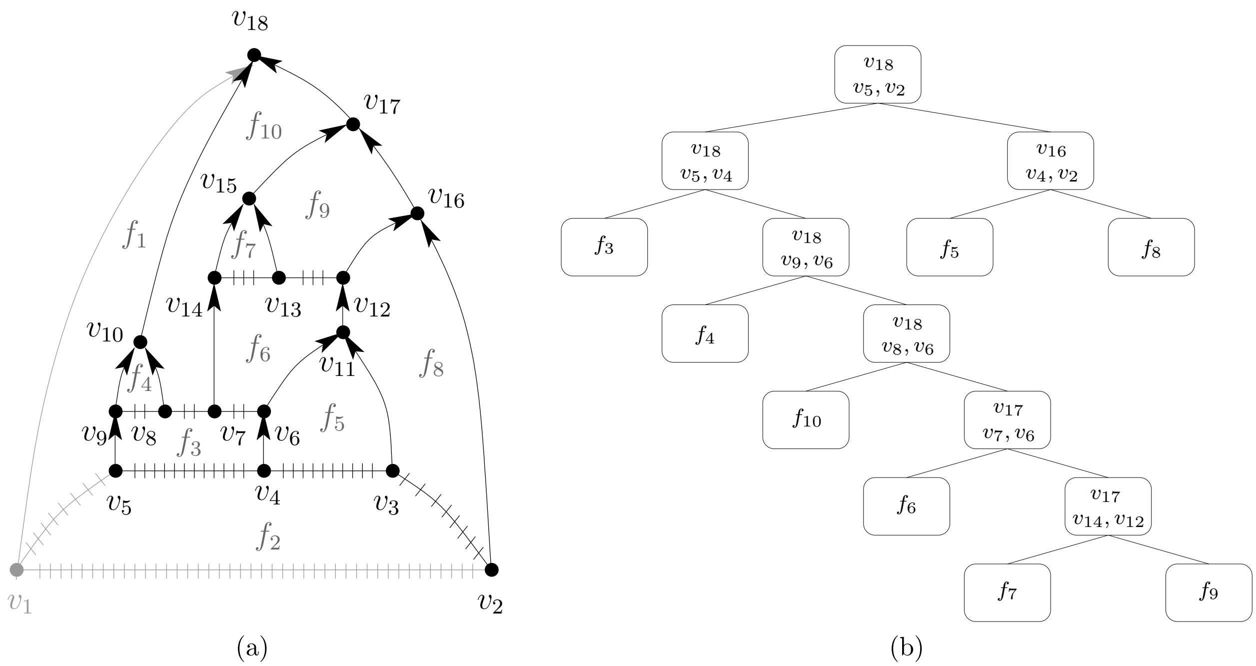



Figure 3.20: (a) Dart-shaped graph $G$. (b) Splitting tree of $G$.

tables for the canonical order and the coordinates and an adjacency list for the neighbours (because $G$ is 3-regular.) The adjacency list keeps the neighbours of a vertex in clockwise order around it. We will also maintain an adjacency list for the dual graph $G^*$ of $G$ with pointers from the edges in $G$ to their dual edges in $G^*$. Vertices in $G^*$ have weights which are the area of their corresponding face in $G$.

Since $G$ is planar, both the number of edges and the number of faces is $O(n)$.

In RecursivelyDraw$(G, T)$, all steps can clearly be done in $O(n)$ time. In fact, almost all of them take amortized time $O(1)$, which would lead to $O(n)$ time complexity: Operations (3), (22) and (23) take time proportional to the number of edges added; since each edge is used only once or twice, this leads to $O(n)$ in total. Operations (8), (13) and (25) take time proportional to the number of vertices pinned; since each vertex is pinned only once, this leads to $O(n)$ in total. Operations in line (26) take time proportional to the number of vertices removed from the T-staircase since vertices used to calculate the coordinates of $p_{c'_l}$ and $p_{c'_r}$ belong to a single face; this leads to $O(n)$ in total as well.

The area necessary for operation (5) can be precomputed by creating a *splitting tree* similar to the one used in Chapter 2, Section 2.3. In this case, the splitting tree is a binary tree of the graphs into which $G$ is divided during the algorithm. The tree can be created in a top-down fashion by using the left, right and top corners of the dart-shape, $c_l$, $c_r$ and $c_t$ respectively. Figure 3.20 shows an example of a dart-shaped graph $G$ and its splitting tree. Then, traverse the tree bottom-up to obtain the faces that belong to each subgraph of $G$, and therefore its area.

Precomputing the tree takes $O(n)$ time. RecursivelyDraw will do one top-down scan to obtain the area values for all recursive calls, thus it will take $O(n)$ time in total.

Therefore, the only operations that are left are lines (7) and (10), which involve obtaining the coordinates of some of the vertices or bends of the orthogonal line that splits $T$ into two T-staircases. This operation depends on vertices pinned before, and may take $\Omega(n)$ time if the T-staircase has many corners.

We tried, but did not succeed, in devising a data structure that can reduce the running time to $O(1)$ amortized time, and leave this as an open problem. However, we can obtain $O(\log n)$ amortized time as follows: Each time we split a T-staircase into two with a vertical path, we can calculate the $x$-coordinate of $p_u$ in two ways. One option is to calculate the area $A(T_l)$ as a function of the $x$-coordinate of $p_u$ as explained in Section 3.9, and obtain the correct $p_u.x$ from this. The second option is to calculate the area $A(T_r)$ to the right of $l$ as a function of the $x$-coordinate of $p_u$, which can be done similarly by splitting $T_r$ into rectangles. Calculating $A(T_l)$ and $A(T_r)$ takes time proportional to the number of corners in the stairs of $T_l$ and $T_r$, respectively. Since we know how many faces are on each side of $l$ (we can store this with the splitting tree), we use the side with smaller number of faces to calculate $p_u.x$. Then, if a corner is used to calculate $p_u.x$, the number of interior faces of the T-staircase it belongs to will be reduced by a factor of 2 or more, and therefore it can be used at most $O(\log n)$ times. Since there are $O(n)$ corners, the total time is $O(n \log n)$.

In Draw($G$), all operations are clearly $O(1)$, except for obtaining the canonical order, which is is $O(n)$ by Kant [25], and RecursivelyDraw($G, T$), which takes $O(n \log n)$ time as discussed before.

Therefore, the total running time for our algorithm is $O(n \log n)$.

# Chapter 4

# Conclusions and open problems

In this thesis, we studied drawings of planar graphs with prescribed face areas. Our results can be divided into two main parts:

1. Straight-line drawings of planar graphs with prescribed face areas.

   We proved that all planar partial 3-trees can be drawn respecting given face areas. If the assigned areas are rational, our algorithm leads to rational coordinates. We gave bounds on the size of the grid of such drawings. However, these bounds are huge, which leaves the question of whether tighter bounds can be found.

   There are many paths that can be taken from here.

   One is to relax the straight-line constraint, by allowing a very small number of bends (e.g., one bend per edge.) One could start by studying planar 4-trees. Due to Ringel [34], we know that planar 4-trees cannot always be drawn so that given areas for the faces are respected. Thus, it would be interesting to study if allowing one bend per edge would make such drawings possible for all graphs, or at least all partial 4-trees.

   Another possibility would be to relax the area constraint, i.e., allow face $f$ to have area $A(f) \pm \varepsilon$ for small enough $\varepsilon$. If this is done, rational coordinates for cubic graphs would follow immediately, since vertices with non-rational coordinates would be moved to the closest rational position. Then, it would be interesting to ask what bounds can be imposed on the area error. And what can be proved for other graph classes?

   Also, recall Conjecture 1: Does every planar graph with maximum degree 3 have a straight-line drawing that respects given rational face areas and where coordinates of the vertices are rational? This question remains open.

   Finally, one can study the complexity of testing whether a planar graph has a straight-line drawing with prescribed face areas.

2. Orthogonal cartograms.

We showed an algorithm to create orthogonal drawings of 3-connected planar graphs with maximum degree 3 and prescribed face areas. This algorithm works by splitting the graph recursively into dart-shaped graphs and drawing the pieces into a T-staircase. We hope that this way of decomposing a graph can be used independently for other graph drawing applications. We studied various aspects of the drawings resulting from our algorithm. The maximum number of bends per edges produced is 4, and the maximum number of bends per face is 8. Since these numbers represent a nice improvement from previous results, they make one wonder what are the lower bounds for them. Is there a triconnected cubic graph that requires 8 bends in a face to respect given face areas? Sometimes there is a variation on the number of total bends obtained if the paths used to split the graph are selected in different order. Can we find the optimal order to split a graph to minimize the number of bends? Also, in our algorithm all vertices and bends have rational coordinates, but we did not give bounds on the size of the grid. Thus, a natural question would be, what bounds can be found for it? Alternatively, one can ask what bounds can be imposed on the minimum feature size and minimum edge length? Also, can the running time for the algorithm be brought down to $O(n)$?

Our algorithm works for graphs with maximum degree 3. The degree is crucial when a dart-shaped graph is divided by a horizontal path. Thus, one could try to modify the algorithm to allow vertices with degree 4. It is known [28] that orthogonal drawings respecting given face areas are possible, but the number of bends is not constant, it depends on the number of vertices of degree 4. Can we do it with $O(1)$ bends per face always? The next step beyond that would be to study graphs with higher degree, where vertices use rectangles, instead of points.

Also, our algorithm works for 3-connected graphs. This condition is necessary to obtain the canonical orientation of a graph. It is possible to extend this result to 2-connected graphs by introducing extra vertices and edges until the graph is 3-connected, and removing them in the final drawing. However, with this approach, the number of bends per face is unbounded. One approach to keep the number of bends small could be to draw 3-connected components separately, maintaining a compatible pinning of the shared vertices, so that the edges connecting different components can be drawn. How can we ensure this compatible pinning?

Our algorithm assures that all faces are $x$-monotone. However, not all faces are $y$-monotone. Is it possible to guarantee both?

Finally, area-respecting drawings could also be studied in the hexagonal grid, where edges can be horizontal, at 60 and at 120 degrees. Whether planar graphs with prescribed face areas can be drawn in a hexagonal grid, what would be the size of such grid, and which bounds could be obtained on the number of bends per edge and per face, all remain as interesting topics for future study.

# Bibliography


[1] Stefan Arnborg and Andrzej Proskurowski. Linear time algorithms for NP-hard problems restricted to partial $k$-trees. *Discrete Applied Mathematics*, 23(1):11–24, 1989. 10

[2] Therese Biedl and Goos Kant. A better heuristic for orthogonal graph drawings. In *Comput. Geom. Theory Appl*, volume 855, pages 24–35. Springer-Verlag, 1998. 2

[3] Therese Biedl and Lesvia Elena Ruiz Velázquez. Drawing planar 3-trees with given face-areas. In *Proc. 17th International Symposium on Graph Drawing (GD'09), Lecture Notes in Computer Science*, volume 5849, pages 316–322. Springer, 2010. 10

[4] Therese C. Biedl. Optimal orthogonal drawings of triconnected plane graphs. In *5th Scandinavian Workshop on Algorithm Theory (SWAT '96), Lecture Notes in Computer Science*, volume 1097, pages 333–344, 1996. 2

[5] Therese C. Biedl. New lower bounds for orthogonal drawings. *J. Graph Algorithms Appl*, 2:2–7, 1998. 2

[6] Hans L. Bodlaender. Dynamic programming on graphs with bounded treewidth. In *Proceedings of the 15th International Colloquium on Automata, Languages and Programming (ICALP '88), Lecture Notes in Computer Science*, volume 317, pages 105–118, London, UK, 1988. Springer-Verlag. 10

[7] Hans L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 226–234, New York, NY, USA, 1993. ACM. 10, 21

[8] H.L. Bodlaender. Planar graphs with bounded treewidth. Technical Report RUU-CS-88-14, Department of Information and Computing Sciences, Utrecht University, 1988. 11, 23

[9] Marek Chrobak and Shin ichi Nakano. Minimum-width grid drawings of plane graphs. In *Graph Drawing (GD '94), Lecture Notes in Computer Science*, volume 894, pages 104–110. Springer, 1994. 27

[10] Marek Chrobak and Goos Kant. Convex grid drawings of 3-connected planar graphs. *Int. Journal of Computational Geometry and Applications*, 7(3):211–223, 1997. 27

[11] Mark de Berg, Elena Mumford, and Bettina Speckmann. On rectilinear duals for vertex-weighted plane graphs. In *Proc. 13th International Symposium on Graph Drawing (GD'05), Lecture Notes in Computer Science*, volume 3843, pages 61–72. Springer, 2006. 5, 6, 26

[12] Mark de Berg, Elena Mumford, and Bettina Speckmann. Optimal BSPs and rectilinear cartograms. In *GIS '06: Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, pages 19–26, New York, NY, USA, 2006. ACM. 5

[13] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10:41–51, 1990. 2

[14] Hubert de Fraysseix, János Pach, and Richard Pollack. Small sets supporting fary embeddings of planar graphs. In *STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 426–433, New York, NY, USA, 1988. ACM.

[15] Giuseppe di Battista, Peter Eades, Roberto Tamassia, and Tollis Ioannis G. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice Hall, 1998. 2

[16] Vida Dujmović, David Eppstein, Matthew Suderman, and David R. Wood. Drawings of planar graphs with few slopes and segments. In *Computational Geometry Theory and Applications*, volume 38, pages 194–212. Springer, 2005. 27

[17] E. S. El-Mallah and C. J. Colbourn. On two dual classes of planar graphs. *Discrete Math.*, 80(1):21–40, 1990. 15

[18] Ehab S. El-Mallah and Charles J. Colbourn. Partitioning the edges of a planar graph into two partial $k$-trees. In *19th Southeastern Conference on Combinatorics, Graph Theory, and Computing*, volume 66, pages 69–80. Congressus Numerantium, 1988. 11

[19] David Eppstein, Elena Mumford, Bettina Speckmann, and Kevin Verbeek. Area-universal rectangular layouts. In *Proc. 25th ACM Symposium on Computational Geometry (SoCG)*, pages 267–276, 2009. 5

[20] I. Fáry. On straight line representation of planar graphs. *Acta. Sci. Math. Szeged*, 11:229–233, 1948. 1

[21] Sander Florisson, Marc van Kreveld, and Bettina Speckmann. Rectangular cartograms: construction & animation. In *Proc. 21st ACM Symposium on Computational Geometry (SoCG)*, pages 372–373, 2005.

[22] Michael T. Gastner and M. E. J. Newman. Diffusion-based method for producing density equalizing maps. *Proceedings of the National Academy of Sciences of the United States of America*, 101:7499–7504, 2004. 3

[23] Jim Geelen, Anjie Guo, and David McKinnon. Straight line embeddings of cubic planar graphs with integer edge lengths. *Journal of Graph Theory*, 58(3):270–274, 2008. 2

[24] Vít Jelínek, Eva Jelínková, Jan Kratochvíl, Bernard Lidický, Marek Tesař, and Tomáš Vyskočil. The planar slope number of planar partial 3-trees of bounded degree. In *Proc. 17th International Symposium on Graph Drawing (GD '09), Lecture Notes in Computer Science*, volume 5849, pages 304–315. Springer, 2010. 10

[25] G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16(1):4–32, 1996. 6, 26, 27, 56

[26] Goos Kant. Hexagonal grid drawings. In *Graph-Theoretic Concepts in Computer Science, 18th International Workshop (WG '92), Lecture Notes in Computer Science*, volume 657, pages 263–276. Springer, 1992. 29

[27] Goos Kant and Xin He. Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems. *Theor. Comput. Sci.*, 172(1-2):175–193, 1997. 4

[28] Akifumi Kawaguchi and Hiroshi Nagamochi. Orthogonal drawings for plane graphs with specified face areas. In *Proc. 4th Int. Conf. on Theory and Applications of Models of Computation, Lecture Notes in Computer Science.*, volume 4484, pages 584–594. Springer, 2007. 5, 26, 58

[29] Yanpei Liu, Aurora Morgana, and Bruno Simeone. A linear algorithm for 2-bend embeddings of planar graphs in the two-dimensional grid. *Discrete Applied Mathematics*, 81(1-3):69–91, 1998. 2

[30] Takao Nishizeki and Md. Saidur Rahman. *Planar Graph Drawing (Lecture Notes Series on Computing).* World Scientific Publishing Company, 2004. 26

[31] Md. Saidur Rahman, Kazuyuki Miura, and Takao Nishizeki. Octagonal drawings of plane graphs with prescribed face areas. *Comput. Geom. Theory Appl.*, 42(3):214–230, 2009. 5

[32] Md. Saidur Rahman, Takao Nishizeki, and Mahmuda Naznin. Orthogonal drawings of plane graphs without bends. *Journal of Graph Algorithms and Applications*, 7(4):335–362, 2003. 2

[33] Erwin Raisz. The rectangular statistical cartogram. *Geographical Review*, 24(3):292–296, 1934. 4

[34] Gerhard Ringel. Equiareal graphs. *Contemporary Methods in Graph Theory. BI Wissenschaftsverlag*, pages 503–505, 1990. 2, 10, 24, 57

[35] Daniel P. Sanders. On linear recognition of tree-width at most four. *SIAM J. Discret. Math.*, 9(1):101–117, 1996. 10, 21

[36] Walter Schnyder. Embedding planar graphs on the grid. In *SODA '90: Proceedings of the first Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 138–148, Philadelphia, PA, USA, 1990. Society for Industrial and Applied Mathematics. 2, 14

[37] S. Stein. Convex maps. *American Mathematical Society*, 2:464–466, 1951. 1

[38] Roberto Tamassia, Ioannis G. Tollis, and Jeffrey Scott Vitter. Lower bounds for planar orthogonal drawings of graphs. *Information Processing Letters*, 39:35–40, 1991. 2

[39] Carsten Thomassen. Plane cubic graphs with prescribed face areas. *Combinatorics, Probability & Computing*, 1:371–381, 1992. 2, 10, 22

[40] Peter Ungar. On Diagrams Representing Maps. *Journal of the London Mathematical Society*, s1-28(3):336–342, 1953. 2, 4

[41] Marc van Kreveld and Bettina Speckmann. On rectangular cartograms. In *Proc. 12th European Symposium on Algorithms (ESA), Lecture Notes in Computer Science.*, volume 3221, pages 724–735. Springer, 2004. 5

[42] K. Wagner. Bemerkungen zum Vierfarbenproblem. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 46:26–32, 1936. 1

[43] Hassler Whitney. A set of topological invariants for graphs. *American Journal of Mathematics*, 55(1):231–235, 1933. 7