國立臺灣大學電機資訊學院電機工程學系

碩士論文

Department of Electrical Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

圖的接觸表示法

Contact Representations of Graphs

張以潤

Yi-Jun Chang

指導教授：顏嗣鈞博士

Advisor: Hsu-Chun Yen, Ph.D.

中華民國 104 年 6 月

June, 2015

致謝

此篇論文記述了我大學四年級以來至今一部分的研究成果，其中部分內容是基於發表於 GD'13、COCOA'14、及 WADS'15 的三篇會議論文。本論文之所以能夠完成，首要感謝顏嗣鈞老師的指導。最初顏老師讓我看的兩篇關於 rectilinear duals 的文章開啟了我對這個議題的興趣，也進而造就了這一系列的研究成果。在這過程中，除了研究上的討論與指導，顏老師也給我研究助理津貼及參加會議的旅費，並讓我很自由地抉擇研究的方向。

我亦得感謝作為口試委員的許聞廉老師、呂學一老師、于天立老師和陳和麟老師。老師們在口試過程中提供了許多建議與指導，讓本篇論文更加完備。此外，我要特別感謝許聞廉老師以及張震華老師，他們充實的圖論課程讓我培養出扎實的基礎能力。如果沒有這些課程的練習，我的研究可能不會那麼順利。我也要特別感謝于天立老師，因為是于老師的計算機概論課程啟發了我對於計算機科學（特別是演算法方面）的興趣。

學術研究領域以外，我想感謝我在其他圈子的朋友及同好，你們的存在讓我在做研究之餘也能過著充實且有趣的生活。我也必須感謝我的家人多年來對我的關愛、支持、自由與包容。

最後，感謝正在閱讀這篇論文的你。希望此篇論文能傳達出我對於研究的熱誠以及這領域的有趣之處。祝你閱讀愉快！

中文摘要

圖 (graph) 是個常見的數學模型，能描述各種複雜的科學和工程問題。圖形繪製 (Graph drawing) 是將一個圖表示在二維或三維空間的過程，讓圖的結構以易於理解的方式呈現出來。由於這議題的重要性，圖形繪製已成為計算機科學中一個快速成長的新興領域。

在眾多圖形繪製的問題中，圖的接觸表示法 (contact representations of graphs) 在近年來受到的關注不斷提升。給定一個圖，它的接觸表示法即是將圖中的各個節點對應到二維或三維空間的幾何物件，其中任兩個節點在圖上相鄰若且唯若他們對應的物件互相接觸。

在許多接觸表示法中，rectilinear dual 是一個經典的繪製風格，能應用於大型積體電路的布圖規劃。在該繪製風格中，圖中的各個節點用正交多邊形來表示，多邊形的相鄰對應於節點在圖中的相鄰關係，並且所有的多邊形合起來恰好成為一個長方形。

在本論文的前半，我們探討如何在多邊形形狀被限制的情形下設計 rectilinear dual。因凸形 (convex shapes) 傾向比非凸形來得好看且單純，我們提出並探討一個新的圖形繪製風格 - orthogonally convex drawing，研究如何設計 rectilinear dual 的演算法使得其中的幾何物件皆為正交凸多邊形 (orthogonally convex polygons)。另外，為了瞭解各形狀在繪製 rectilinear dual 中的功用與限制，我們探討可用形狀被限制下的 rectilinear dual。我們算出 ⊤-free rectilinear dual 的多邊形複雜度，驗證了 ⊤ 形是最有用的八邊形之直覺。

在本論文的後半，我們探討如何將 rectilinear dual 延伸及推廣到二維正交以外的情境上。為了容納凸多邊形 (convex polygons)，我們提出並探討一個新的圖形繪製風格 - convex polygonal dual。針對這項繪製風格，我們提出一些新的技術及 fixed-parameter tractability 的結果。為了容納正交多面體 (orthogonal polyhedra)，我們提出並探討一個新的圖形繪製風格 - 3D floorplan。我們證明了所有 chordal graph 都能畫成 3D-floorplan。我們的畫法不但只需用到兩層，也能夠實現任意對其中多面體指定的體積。

總體來說，在圖的接觸表示法之框架下，本論文提供了許多新的技術及觀點。希望這篇論文指引出的研究方向能讓我們對這個充滿挑戰性的領域有更全面的認識。

關鍵詞: 圖形繪製, 接觸表示法, 平面圖, 示意地圖, 平面規劃

Abstract

Graph represents one of the most popular abstract models in describing complex science and engineering problems. Graph drawing refers to the process of displaying an abstract graph in 2D or 3D, allowing the structure as well as the meaning of the graph to be understood better and easier. As a consequence, the design of graph drawing algorithms has become an emerging and fast growing research area in computer science.

Among problems of interest in the graph drawing community, the topic of contact representations of graphs has received increasing attention over the years. Given a graph, a contact representation of the graph is to map each vertex of the graph to a geometric object in 2D or 3D so that two vertices are adjacent iff their corresponding objects "touch".

A *rectilinear dual*, a classic drawing style which has found applications in VLSI floor-planning, requires that each vertex be drawn as a rectilinear polygon, adjacency in a graph correspond to side-contact in the drawing, and all rectilinear polygons together form a partition of a rectangle.

In the first half of the thesis, we investigate a variety of shape constraints in rectilinear duals. As convex objects tend to be visually more pleasing, the drawing style *orthogonally convex drawing* is proposed and investigated. In addition, we study rectilinear duals using a restricted set of shapes, in order to understand the power and the limitation of different shapes in rectilinear duals. We determine the optimal polygonal complexity of $\top$-free rectilinear dual, justifying the intuition that $\top$-shape is the most useful 8-sided polygon.

In the second half of the thesis, we study possible extensions and generalizations of rectilinear duals beyond the 2D rectilinear setting. To accommodate convex polygons, the drawing style *convex polygonal dual* is proposed and investigated. We demonstrate several new techniques and fixed-parameter tractability results to deal with this drawing style. We also propose and investigate a new drawing style called *3D floorplan*, using rectilinear polyhedra as building blocks. We show that every chordal graph admits a 3D-floorplan which uses only two layers and is also capable of realizing any volume-assignment to its constituent polyhedra.

In summary, the thesis provides a variety of new techniques and new perspectives within the framework of contact representations of graphs. We hope that this study could lead to a better understanding of contact graph representations - an exciting and challenging topic in graph drawing.

Keywords: graph drawing, contact representation, planar graph, cartogram, floorplan

Contents

List of Figures

Chapter 1

Introduction

A *graph* is a mathematical structure which contains a collection of nodes and their pairwise relations. It not only has been a prime topic of study in discrete mathematics for years, but is also a natural model capturing lots of concepts and structures in computer science and electrical engineering. For instance, computer networks, social networks, circuits, and transportation routes can be modeled as graphs.

One very crucial aspect in the study of graphs is their drawings. From a practical point of view, we are frequently asked to find the best drawing of some graph occurring in real-world applications. In the floor-planning phase of the VLSI design, it is critical to find a drawing of the underlying circuit that uses a small chip area while meeting several constraints [45]; for a city having a complicated metro system, it is important to have a nicely drawn metro map that can be read and understood easily [39].

From a theoretical point of view, the investigation of various geometric representations of graphs has led to profound impacts and consequences in graph theory and algorithms. On the one hand, geometry representations have inspired the introduction of some graph classes, like planar graphs and chordal graphs, among others. As they facilitate the combination of graph theoretical and discrete geometric techniques, studying various concepts related to these graph classes has led to the birth of several successful branches in graph theory, like geometric graph theory and graph minor theory. On the other hand, the study of possible geometric representations of a graph class can deepen our understanding of the structures and properties of the graph class.

Basically, drawing a graph consists of three components: an input graph, a drawing style, and a quality measure. The goal of graph drawing is to search for an optimal drawing of the input graph meeting the required drawing style, where the optimality is with respect to the designated quality measure. In many occasions, we require the input graph to be planar, as crossings are not allowed in many drawing styles.

In this thesis, we focus on one particular class of drawing styles called *contact representations*, where all vertices are represented by interior-disjoint geometric objects such that the adjacency relations correspond to contacts between objects. The study of contact representations can be traced back to the following well-known circle packing theorem [29]:

Every planar graph can be drawn as touching circles in the plane.

Following the above celebrated result, a variety of contact representations have been proposed and investigated over the years. Among them, the drawing style *rectilinear dual* has received quite a lot of research efforts in the past two decades:

A rectilinear dual is a contact representation of a graph in which each vertex corresponds to a rectilinear polygon, adjacency in the graph corresponds to side-contact in the drawing, and all rectilinear polygons together form a partition of a rectangle. The definition of rectilinear duals is largely motivated by the floor-planning in VLSI design. As a result, it has not only attracted researchers in the graph drawing community [3,4] but has also received extensive investigation in the VLSI design community [45].

Two important quality measures for designing rectilinear duals are the following:

- *Polygonal complexity*: the polygonal complexity of a rectilinear dual is defined as the maximum number of sides of polygons involved in the rectilinear dual.
- *Area-universality*: a rectilinear dual of a graph G is called *area-universal* iff it can realize any area-assignment $f : V(G) \to \mathbb{R}_{>0}$ in the sense that the polygon corresponding to vertex v has area $f(v), v \in V(G)$.

The study of area-universal drawings is motivated by the need to visualize weighted graphs. For example, a cartogram where the area of each country is scaled to its population size is used to visualize the population of all countries in the world. Area-universality allows the visualization of any possible weight functions, and hence the comparison of different weight functions (such as the change in population over a time period) can be visualized clearly.

A rectilinear dual can be somewhat perceived as the "dual" version of an *orthogonal drawing*, which is a planar drawing such that each edge is composed of a sequence of horizontal and vertical line segments with no crossings. The interplay between rectilinear duals and orthogonal drawings turns out to be rather useful in developing various drawing techniques, as our subsequent discussion reveals.

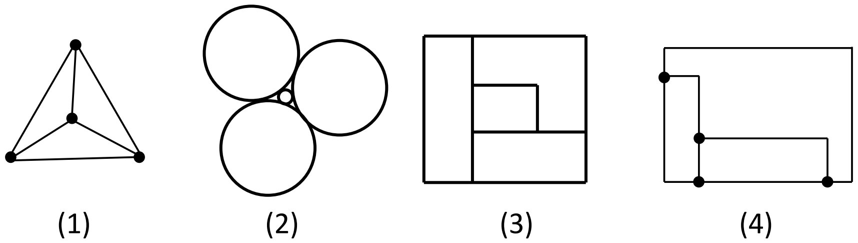


Figure 1.1: Examples of drawing styles

Consider a planar graph G (drawn as a straight line drawing) in Fig. 1.1(1). Fig. 1.1(2) is a contact representation of G using touching circles. A rectilinear dual of G with its polygonal complexity being 6 is displayed in Fig. 1.1(3). Fig. 1.1(4) shows an orthogonal drawing of G. The duality between Fig. 1.1(3) and Fig. 1.1(4) is easy to see by relating the four rectilinear regions of Fig. 1.1(3) to the four nodes in Fig. 1.1(4).

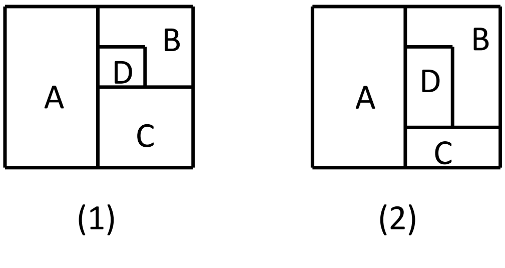


Figure 1.2: Realization of different area-assignments.

The rectilinear dual depicted in Fig. 1.1(3) is actually area-universal. See Fig. 1.2 for its realizations of two different area-assignments: $(A \leftarrow 8, B \leftarrow 3, C \leftarrow 4, D \leftarrow 1)$ and $(A \leftarrow 8, B \leftarrow 4, C \leftarrow 2, D \leftarrow 2)$.

Following a series of recent results, an algorithm for constructing area-universal rectilinear duals with polygonal complexity of 8 for maximal plane graphs was proposed in [4]. This result is tight in the sense that there exists a maximal plane graph that has polygonal complexity of at least 8 in any of its rectilinear duals. Even with such matching lower and upper bounds in polygonal complexity, many interesting issues regarding rectilinear duals of plane graphs remain unanswered or unexplored. Specifically, in the thesis, we initiate the following two lines of research directions:

1. **Imposing geometric constraints on polygons in rectilinear duals.**

   Most of the previous research on rectilinear duals have been focusing on the number of sides of individual rectilinear polygons. Few results are available for tackling geometrical constraints such as orthogonal convexity or shape constraints. To move a step further along this line of research, the following problem is investigated in this thesis:

   Question 1.1. *Given a set of available shapes and an input graph, is it possible to design an efficient algorithm to find a rectilinear dual using only polygons of the available shapes, if it exists?*

2. **Extending rectilinear duals to more general settings.**

   In real life, it is not uncommon to encounter objects displayed as polygons which are not necessarily rectilinear. However, in contrast to the well-studied rectilinear duals, only a scarcity of results and methods are available for tackling cases for polygons that are not necessarily rectilinear. Also, motivated by the emergence of three-dimensional integrated circuits (3D ICs), the study of floor-planning in 3D should be of great interest. In light of the above, we are also concerned with the following problem in the thesis:

   Question 1.2. *Is it possible to design efficient algorithms to construct good floor-plans using convex polygons or 3D rectilinear polyhedra, where the quality measures are area-universality and polygonal complexity?*

Organization of the thesis:

Chapter 2 includes basic definitions, notations and facts that will be used throughout this thesis. We also briefly survey some existing results related to our work.

Chapters 3 and 4 are devoted to the first research direction (Question 1.1).

In Chapter 3, we focus on orthogonal convexity in rectilinear duals. A clean condition for the existence of a rectilinear dual using orthogonally convex polygons subject to a given orthogonally convex boundary constraint is presented. Our new finding relies on the establishment of a close connection between the "dual" setting (i.e.,, the rectilinear dual) and its "primal" version (i.e., the orthogonal drawing) in the study of orthogonal convexity. We propose the drawing style *orthogonally convex drawing* which serves as the orthogonal analogue of the *convex drawing*. To our best knowledge, our effort here is the first time that orthogonal convexity is studied in rectilinear duals and orthogonal drawings. Part of this chapter has appeared in [9].

Motivated by an observation that most algorithms yielding rectilinear duals of low polygonal complexity require the use of ⊤-shape polygons or their extensions, our aim in Chapter 4 is to justify the intuition that ⊤-shape polygons are more powerful than other 8-sided ones. To this end, it is proven that the required polygonal complexity for maximal plane graphs increases from 8 to 12 if ⊤-shape polygons and their extensions are not allowed. We then continue this line of research by studying other constrained rectilinear duals. Part of this chapter has appeared in [10].

Chapters 5, 6, and 7 are devoted to the second research direction (Question 1.2).

In Chapter 5, a new drawing style called the *convex polygonal dual*, serving as the convex polygonal analogue of the rectilinear dual, is proposed. We give a finite combinatorial characterization for plane graphs admitting such drawings. Our characterization not only leads to some fixed-parameter tractability results, but it can also be applied to giving quick alternate proofs for existing results and establishing relationship between rectilinear duals and convex polygonal duals. Part of this chapter has appeared in [11].

In Chapter 6, we give a detailed study of convex polygonal duals for biconnected outer plane graphs. Our study yields a clean condition for the existence of a drawing for a given polygonal complexity. A simple procedure is also given for constructing an area-universal drawing of low polygonal complexity.

In Chapter 7, rectilinear duals are generalized to 3D by representing each vertex of a graph as an orthogonal polyhedron. This study opens the door for non-planar graphs to be accommodated in a floorplan design. We prove that all chordal graphs admit such 3D drawings. This result parallels the well-known fact that all maximal plane graphs admit rectilinear duals, as chordal graphs and maximal plane graphs are regarded as the natural candidates of "triangulated graphs" in the general and the planar settings, respectively.

Finally, Chapter 8 summarizes the results reported in this thesis. Some open problems are posted in this chapter as well.

Chapter 2

Preliminaries

The goal of this chapter is to introduce some basic notations and preliminary results required for the subsequent discussion. We do not intend to give a comprehensive tutorial for each topic we discuss, as there already exist quite a few nicely written literatures devoted to these topics.

For a more comprehensive introduction to graph drawing, the reader is recommended to have a look at the book [34] and the two PhD theses [1, 43]. They not only provide a nice introduction to the field, but they also contain materials intimately related to the content of the thesis. Also, the reader is referred to [44] to learn more about graph theory.

2.1 Graph Theoretic Preliminaries

Given a graph $G = (V, E)$, we write $\Delta(G)$ to denote the maximum degree of G. We write $V(G)$ and $E(G)$ to denote the set of vertices and the set of edges of G, respectively. Graph G is called a d-graph if $\Delta(G) \leq d$.

Note that the definition of the notion G^* changes in different chapters of the thesis, and we usually do not follow the custom to use G^* to denote the dual graph of G.

A graph is *simple* if it contains no self-loop and no multi-edges. A *multi-graph* is a graph where self-loops are disallowed while multi-edges are allowed. If not otherwise stated, all graphs in the thesis are assumed to be simple.

A graph is *k-connected* if it contains at least $k + 1$ vertices, and if removal of any $k - 1$ vertices does not render the graph disconnected. "*biconnected*" and "*triconnected*" are synonyms for "2-connected" and "3-connected", respectively. For $k = 1$, we can simply call the graph *connected*. If not otherwise stated, all graphs in the thesis are assumed to be connected.

A graph is *planar* if it can be drawn on a plane without any edge crossing. A *plane graph* is a planar graph with a fixed combinatorial embedding and a designated outer face F_O. For any vertex and edge, we call it *boundary* if it is located in F_O. Otherwise, it is *non-boundary*.

An *outerplanar graph* is a planar graph with a planar embedding in which all vertices belong to the outer face. An *outerplane graph* is an outerplanar graph in such an embedding.

A graph H is a *subgraph* of G if $V(H) \subseteq V(G), E(H) \subseteq E(G)$. We also write $H \subseteq G$ to denote the subgraph relation.

Given a graph G, a *path* P is a subgraph such that $V(P) = \{v_1, v_2 \ldots, v_k\}$ and $E(P) = \{\{v_i, v_{i+1}\} | 1 \leq i \leq k-1\}$, for some $k > 1$. A *cycle* C is a subgraph such that $V(C) = \{v_1, v_2, \ldots, v_k\}$ and $E(C) = \{\{v_i, v_{i+1}\} | 1 \leq i \leq k-1\} \cup \{\{v_1, v_k\}\}$, for some $k > 1$. For convenience, a path (or cycle) is also written as $(v_1, v_2, \ldots, v_k)$. Unless otherwise stated, repeated vertices are not allowed in paths and cycles in the sense that $v_i \neq v_j$ if $i \neq j$.

If we write $H = G \setminus S$ (or equivalently, $H = G - S$), where S can be a subgraph of G, a subset of $V(G)$, or a subset of $E(G)$, then H is the subgraph of G defined by the following procedure: (1) $V(H) \leftarrow V(G)-$ the vertices in S; (2) $E(H) \leftarrow E(G)-$ the edges in S; (3) removing the isolated vertices (those incident to no edge in $E(H)$) in H.

A cycle is *Hamiltonian* if it contains all vertices in the underlying graph. A graph is *Hamiltonian* if it has a Hamiltonian cycle.

A drawing of a planar graph divides the plane into a set of connected regions, called *faces*. A *contour* of a face F is the cycle formed by vertices and edges along the boundary of F. Sometimes we slightly abuse the terminology to write F to denote its contour. Such a cycle is also called a *facial cycle*. The contour of the outer face F_O is also denoted as C_O.

Given a plane graph G, the *inner* (also known as *internal* or *interior*) region of a cycle C is the region enclosed by C (containing the vertices and the edges located interior of the cycle C), and the *outer* (also known as *external* or *exterior*) region of C is the region outside of C (containing the vertices and the edges located exterior of the cycle C). The inner and outer regions of C are written as $in(C)$ and $out(C)$, respectively. The edges and vertices located along C (i.e. $E(C), V(C)$) are neither in the inner region nor in the outer region of C. We use $G(C)$ to denote the subgraph of G that contains exactly C and vertices and edges residing in its inner region.

2.2 Tree-width and Chordal Graphs

A *tree* is a connected graph without any cycle. It is a basic fact that a tree T has exactly $|V(T)| - 1$ edges (assuming it is simple). However, other than the trees, there are quite a few graph classes (e.g. outerplanar graphs) also having some sort of "tree structure", and the notion *tree-width* formalizes this concept.

A *tree decomposition* of a graph G is a tree T such that the following properties are satisfied:

1. $V(T) = \{X_1, \ldots, X_{|V(T)|}\}$. Each X_i represents a subset of $V(G)$.
2. For each edge $e = \{u, v\} \in E(G)$, there is an X_i such that $u, v \in X_i$.
3. For each vertex $v \in V(G)$, there is an X_i such that $v \in X_i$.
4. If $v \in X_i \cap X_j$, for all X_k in the unique path linking X_i, X_j in T, we have $v \in X_k$.

We denote a vertex in $V(T)$ as a *bag* instead of a vertex to avoid confusion. The width of T is defined to be $\max\{|X_i| - 1 | X_i \in V(T)\}$. The tree-width is then defined as below:

Definition 2.1. *A graph G is said to have tree-width k iff a minimum width tree decomposition of G has width k.*

As an example, each outerplanar graph has tree-width at most two.

A lot of difficult (NP-hard) problems have polynomial time solutions on trees, as it is usually easier to design a dynamic programming algorithm on a tree structure than on a general graph. Intuitively, graphs of bounded tree-width should share this kind of "algorithmic advantage". The famous algorithmic meta-theorem "Courcelle's Theorem" formalizes this intuitive idea:

Theorem 2.1 ([13,14]). *Any graph property expressible in MSO$_2$ is linear time solvable for graphs of bounded tree-width*

Monadic second-order logic, a fragment of second-order logic, allows only quantification over unary relations (i.e., sets). The monadic second-order logic on graph MSO$_2$ includes the following ingredients:

- Variables: vertices, edges, set of vertices, and set of edges.
- Relations: $\in$, $=$, edge-vertex incidence (INC), and adjacency (ADJ).
- Connectives: $\vee, \wedge, \neg, \rightarrow$.
- Quantifiers: $\forall, \exists$ that can be applied to all kinds of variables.

Courcelle's Theorem plays an important role in Chapter 5.

For more about monadic second-order logic on graph structures, the reader is referred to [14, 16].

Besides the above algorithmic application, there are still other topics in graph theory intimately relate to tree-width. A graph is *chordal* if each of its cycle C of length more than 3 has a *chord*, which is an edge $e = \{u, v\} \notin E(C)$ such that $u, v \in V(C)$. As a result, each induced cycle in a chordal graph is necessarily a triangle.

Interestingly, chordal graphs are exactly the ones who admit a tree-decomposition where each bag is a *clique*, which is a subgraph H such that $\forall u, v \in V(H), \{u, v\} \in E(H)$. Such a tree-decomposition is also called a *clique tree*. This concept is crucial in Chapter 7.

2.3 Rectilinear Polygons

A polygon is *rectilinear* if all its edges are parallel to x-axis or y-axis of the Cartesian coordinate system. A rectilinear polygon is also known as an *orthogonal polygon*. In this section we present some notations for rectilinear polygons, which are mostly applied in Chapter 3, 4 of the thesis.

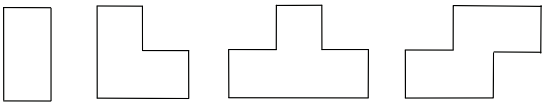


Figure 2.1: I-shape, L-shape, ⊤-shape and Z-shape polygons

Two rectilinear polygons are *combinatorially equivalent* iff they admit the same circular order of angles. When there is no need to differentiate polygons that are combinatorially equivalent to each other, it is without loss of information to use circular order of angles to represent a rectilinear polygon. For example, rectangle (or called I-shape), L-shape, ⊤-shape, W-shape, ⊔-shape, Z-shape can be represented by (V, V, V, V), (V, V, V, C, V, V), (V, V, C, V, V, C, V, V), (V, V, V, C, V, C, V, V), (V, V, V, C, C, V, V, V), (V, V, V, C, V, V, V, C), respectively, where the letters V and C represent convex and concave corners, respectively. See Fig. 2.1 for some illustrations. Given a sequence P of Cs and Vs, we let $\sharp_C(P)$ and $\sharp_V(P)$ denote the numbers of concave and convex corners, respectively.

Here we define a partial order "$\preceq$" on rectilinear polygons as follows:

Definition 2.2. *Let P and Q be two rectilinear polygons. $P \preceq Q$ iff Q can be obtained by iteratively inserting (C, V) or (V, C) into P.*

The drawing style *rectilinear dual* is defined as follows:

Definition 2.3. *A rectilinear dual is a contact representation of a graph meeting the below conditions:*

1. *Each vertex corresponds to a rectilinear polygon.*

2. *Adjacency in the graph corresponds to side-contact in the drawing.*

3. *All rectilinear polygons together form a partition of a rectangle.*

Let R be a rectilinear dual, we call it *Q-free* iff for each polygon of shape P used in R, we have $Q \not\preceq P$. We remark that the partial order "$\preceq$" actually reflects the intuitive idea of degeneracy in the way that $P \preceq Q$ indicates that Q can degenerate to P. Therefore, the notion of "Q-freeness" captures the idea of "Q is not a degenerated form of any rectilinear polygon in the drawing".

A class of rectilinear polygons called *staircase* is defined as follows:

Definition 2.4. *Let P be a rectilinear polygon. P is monotone staircase iff $P = (a, S_1, b, S_2)$ clockwise, where $a = b = V$ and both S_1 and S_2 consist of Cs and Vs appearing alternatively and both start and end with V, where the two points a, b are exactly at the most south-western (i.e., lower left-hand) and the most north-eastern (i.e., upper right-hand) corners.*

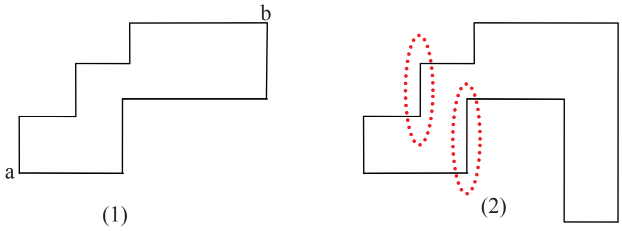


Figure 2.2: Examples of rectilinear polygons.

In other words, a monotone staircase polygon is a polygon formed by two monotonically rising staircases, each of which is a sequence of horizontal and vertical line segments from the bottom-left corner to the top-right corner of the polygon. A *staircase polygon* is a rectilinear polygon resulting from rotating a monotone staircase polygon 90°, 180°, or 270°.

The following facts are easy to observe, and may be explicitly or implicitly applied in the discussion throughout the thesis.

Fact 2.1. $\sharp_V(P) - \sharp_C(P) = 4$ *in any rectilinear polygon P.*

Fact 2.2. *A rectilinear polygon is orthogonally convex iff it does not contain consecutive concave corners.*

Fact 2.3. *A rectilinear polygon P is staircase iff $\top \not\preceq P$ and P is orthogonally convex.*

Fact 2.4. *A rectilinear polygon P satisfies $\top \preceq P$ iff $P = (S_1, a, S_2, b, S_3, S_4)$ such that $a = b = V$ and $\sharp_C(S_2) = \sharp_V(S_2)$, $\sharp_C(S_1) - \sharp_V(S_1) = \sharp_C(S_3) - \sharp_V(S_3) = 1$.*

Fig. 2.2(1) is an example of a staircase polygon. The polygon in Fig. 2.2(2) can degenerate to $\top$-shape by removing the two pairs of corners (circled in the picture); the reader can verify Fact 2.4 by considering its representation $(C, V, V, V, C, C, V, V, V, V, C, V) = (S_1 = (C), a = V, S_2 = (), b = V, S_3 = (V, C, C), S_4 = (V, V, V, V, C, V))$.

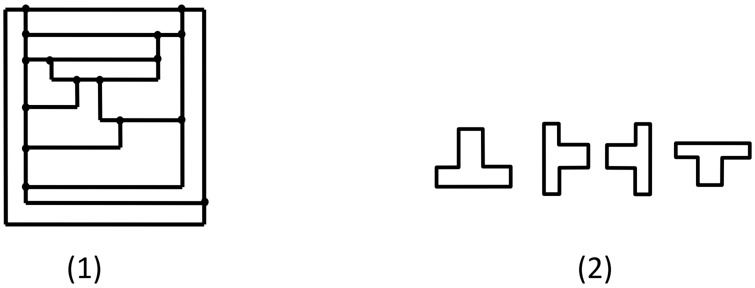


Figure 2.3: A non-rotated $\top$-shape rectilinear dual.

Sometimes there is a need to differentiate combinatorially equivalent polygons that have different orientations. For example, Fig. 2.3(2) shows the four possible orientations of the $\top$-shape. We use the term "*non-rotated*" to describe a situation that exactly one orientation is allowed to appear in a rectilinear dual.

Fig. 2.3 (1) is a rectilinear dual using only non-rotated $\top$-shape polygons, since all the polygons are combinatorially equivalent to (and possibly a degenerate of) exactly one of the four possible $\top$-shape polygons depicted Fig. 2.3(2).

A rectilinear dual using only monotone staircase polygons can be regarded as a non-rotated staircase rectilinear dual.

2.4 Separation Trees

In this section we describe an important technique to construct rectilinear duals.

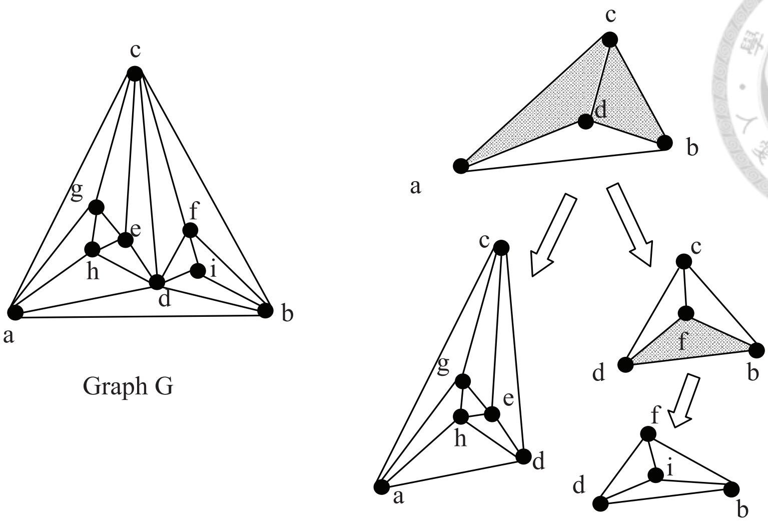


Figure 2.4: A separation tree.

Let $\triangle$ be a triangle (a cycle of length 3 in a graph). We call $\triangle$ a *separating triangle* (also known as a complex triangle) iff $G(\triangle) \neq \triangle$ in any planar embedding of G. $G_{\triangle}$ is defined to be the induced subgraph of the set of vertices $\{v \in V(G(\triangle))|$ for any triangle $\triangle' \neq \triangle$ in $G(\triangle)$, v does not reside in the interior region of $\triangle'\}$. For graph G depicted in Fig. 2.4, $G_{\{a,b,c\}}$ is the subgraph induced by $\{a, b, c, d\}$. The *separation tree* of a maximal plane graph G is defined to be the unique rooted tree whose vertices are separating triangles and the C_O of G, with $\triangle$ being a descendant of $\triangle'$ iff $\triangle$ is contained in $G(\triangle')$. See Fig. 2.4. The reader is referred to [43] for a more detailed introduction to separation trees.

The *contraction* of a triangle $\triangle$ is an operation that replaces $G(\triangle)$ with $\triangle$; the *un-contraction* of a (previously contracted) triangle $\triangle$ is an operation that replaces $\triangle$ with $G_{\triangle}$. The descendants of $\triangle$ remain contracted when we un-contract $\triangle$.

A *rectangular dual* is a rectilinear dual in which each polygon corresponding to a vertex is a rectangle. The following theorem gives a clean characterization of graphs admitting such a drawing [30]:

Theorem 2.2 ([30], and see also [19]). *An internally triangulated plane graph G admits a rectangular dual iff we can augment G with four vertices $\{N, E, S, W\}$ satisfying:*

1. *The new outer face is the quadrangle $\{N, E, S, W\}$;*

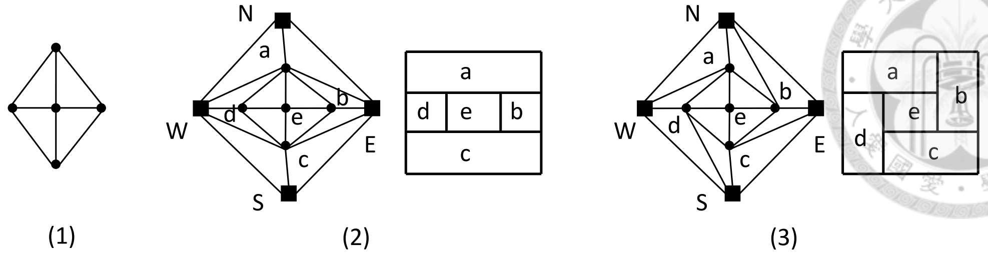


Figure 2.5: Rectangular duals.

2. *The resulting graph is internally triangulated and contains no separating triangle.*

Fig. 2.5(2,3) show two rectangular duals for the graph in Fig. 2.5(1) corresponding to the augmentation of $\{N, E, S, W\}$ depicted.

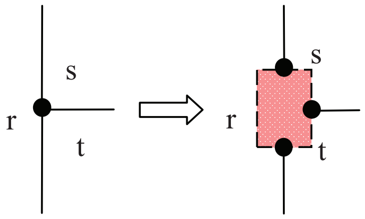


Figure 2.6: Illustration of inserting sub-drawing.

The tight connection between separating triangles and rectangular duals makes separation trees particularly useful in constructing rectilinear duals. A general framework for building rectilinear duals based on separation trees is sketched as follows:

1. Let $\triangle_1, \triangle_2, \ldots, \triangle_k$ be a level-order traversal of the separation tree. We let $G' = \triangle_1$ (C_O, the outer triangle).

2. Construct a rectangular dual of G' as the initial drawing.

3. For $i = 1$ to k, we un-contract $\triangle_i$, and plug-in the rectangular dual of $G_{\triangle_i} \setminus \triangle_i$ to the current drawing.

See Fig. 2.6 for a conceptual illustration of inserting rectangular dual of $G_{\triangle_i} \setminus \triangle_i$ when $\triangle_i = \{s, t, r\}$ is un-contracted. Note that such an insertion inevitably add some concave corners to nearby polygons, as a result, if a graph contains a separating triangle, its polygonal complexity must be at least 6 in any of its rectilinear dual. The reader is referred to [43,45] for a more comprehensive treatment to the approach. See Fig. 2.7 for a full example of a top-down construction of a rectilinear dual based on this approach.

The above framework is adapted in Chapter 4. The "primal version" of the above method is used in Chapter 3.

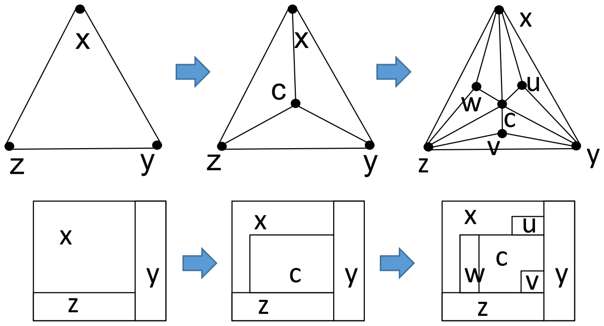


Figure 2.7: Construction of a rectilinear dual.

2.5 Sliceability and Area-universality

In this section we present two important aspects of rectangular duals: sliceability and area-universality. Recall that a rectangular dual is a rectilinear dual where all its polygons are rectangles. These concepts can be generalized or extended to other settings, and this plays an important role in our later discussion.

A rectangular dual is said to be *sliceable* if it can be obtained by recursively cutting a rectangle into two parts by a horizontal or a vertical line. Sliceable rectangular duals enjoy certain nice properties, facilitating global routing by taking advantage of the hierarchical structure of partitioning by the cut lines, for instance. Also see [26] for an application of sliceable rectangular duals in graph drawing.

See Fig. 2.5(2) for an example of a sliceable rectangular dual; and see Fig. 2.5(3) for an example of a non-sliceable rectangular dual.

As a generalization of sliceability in floor-planning, *monotone staircase cuts* have been proposed (see, e.g., [25, 32]), which are able to yield a richer set of floorplan structures while retaining certain attractive properties enjoyed by sliceable floorplans. In fact, floorplans using monotone staircase polygons are exactly those that can be obtained using monotone staircase cuts. This kind of floorplan is investigated in Chapter 4. See Fig. 2.8(2,3) for an example.

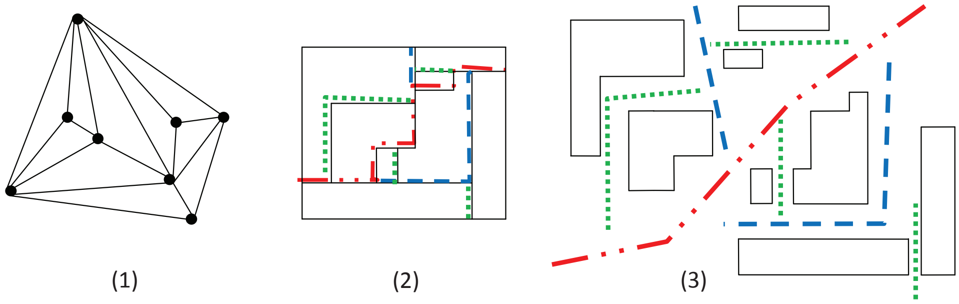


Figure 2.8: A rectilinear dual constructed by monotone staircase cuts.

A rectangular dual (or a rectilinear dual) is *area-universal* if it can realize any area-assignment $f : V(G) \to \mathbb{R}_{>0}$ in the sense that each polygon P corresponding to vertex v has area $f(v)$. One seminal result regarding area-universal rectilinear duals is the following:

Theorem 2.3 ([4]). *For any maximal plane graph, there is an area-universal rectilinear dual of polygonal complexity 8 using only non-rotated* $\top$*-shape polygons.*

This result is tight in the sense that there exists a maximal plane graph where all its rectilinear duals have polygonal complexity of at least 8. However, given a rectilinear dual, currently there is no easy way to decide whether it is area-universal or not. Moreover, even if the rectilinear dual is known to be area-universal, there is no known combinatorial algorithm to realize a given area-assignment.

There are a few positive results for area-universal rectangular duals in literatures. For example, area-universality can be characterized by one-sidedness [19]:

Theorem 2.4 ([19]). *A rectangular dual is area-universal iff it is one-sided.*

A rectangular dual is *one-sided* iff for each straight line in the drawing, one side of it borders exactly one polygonal region. The two rectangular duals in Fig. 2.5 are both one-sided (and hence area-universal).

More interestingly, the aforementioned concepts seem to be able to extend to other settings beyond rectangular duals and rectilinear duals.

Definition 2.5 ([21,28])**.** *A proper touching triangle representation is a contact representation of a graph meeting the below conditions:*

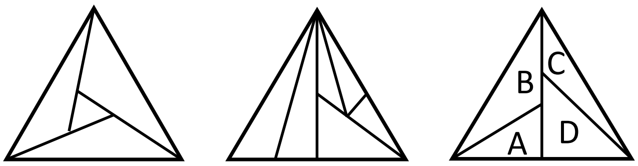
1. *Each vertex corresponds to a triangular region.*
2. *Adjacency in the graph corresponds to side-contact in the drawing.*
3. *All triangular regions together form a partition of a triangle.*

Let $\triangle = \{a, b, c\}$ be a triangle. We define the following two operations which subdivide $\triangle$:

1. Adding a new point d inside of $\triangle$, followed by adding three straight lines linking d to a, b, c.
2. Adding a new point d dividing the line $\overline{bc}$, followed by adding a straight line linking a to d.

We call a proper touching triangle representation *sliceable* iff it can be constructed by applying the above 2 operations to its constituent triangles recursively. A proper touching triangle representation is *one-sided* iff for each straight line in the drawing, one side of it borders exactly one polygonal region. Following basic geometry, the following lemma is easy to observe:

Lemma 2.1. *Every one-sided and sliceable proper touching triangle representation is area-universal. Moreover, if the coordinates of the 3 boundary vertices are fixed, the drawing realizing any given area-assignment is unique.*



(1) (2) (3)

Figure 2.9: Sliceability and one-sidedness in triangle contact representations.

See Fig. 2.9 for illustrations for the above concepts. Fig. 2.9(1) is one-sided but not sliceable; Fig. 2.9(2) is one-sided and sliceable; and Fig. 2.9(3) is not one-sided but sliceable. Note that Fig. 2.9(3) is clearly not area-universal since it cannot realize the area-assignment: $f(A) = f(C) = 0.4, f(B) = f(D) = 0.1$.

See [20] for an interesting result on area-universal drawing in a non-rectilinear setting. They prove the area-universality of their contact representation by refining the drawing to triangles, which are easier to deal with.

Lemma 2.1 is applied in Chapter 6. Sliceability and area-universality are highly relevant to Chapter 4.

2.6 Other Topics

In this section we give a very short introduction to some topics that are important but omitted in the previous sections. For each topic, several good references are provided for interested readers to learn more about them.

Matchings and Flows. In many situations, a graph drawing problem can be reduced to a flow or a matching problem. In these cases, the essential information required in constructing the desired drawing, like the number of bends in each edge and the degree of the angle for each vertex in a face, can be encoded using a matching or a flow. See Section 6.2, 8.2 of [34] for the classic applications of this technique to orthogonal drawings and rectangular drawings, which are the "primal version" of rectilinear duals and rectangular duals, respectively. See [12, 33, 40, 41, 45] for more. This technique is applied in Chapter 3.

Schnyder Labelings. Undoubtedly, Schnyder Labeling [38] is one of the most successful techniques in graph drawing, and it is especially useful in dealing with contact representations. Basically, a Schnyder Labeling is a labeling of corners of a maximal plane graph to $\{1, 2, 3\}$ meeting some conditions. This technique was originally used to construct a straight line drawing in a small grid (see [38] and Chapter 4 of [34]). Since then, it has found applications in varieties of contact representations (see [3, 4, 31] for instances).

Triangular Drawings and Contact Representations. In contrast to the well-studied rectangular duals and rectilinear duals, only a scarcity of results were available in non-rectilinear settings, as it is mathematically easier to handle rectilinear things. Most of the studies in non-rectilinear setting are centered on triangles. The investigation of proper touching triangle representations has been reported in two recent articles [21,28]. Touching triangle representations without boundary constraints have been studied in [22]. In the primal setting, *straight line triangle representation* was proposed and studied in [1, 2]. See Fig 2.10 for a showcase of some triangle contact representations: (1) the 5-cycle, (2) a point-side triangle contact representation, (3) a touching triangle representation without any boundary constraint, (4) a proper touching triangle representation, and (5) a touching triangle representation with a convex polygon boundary.

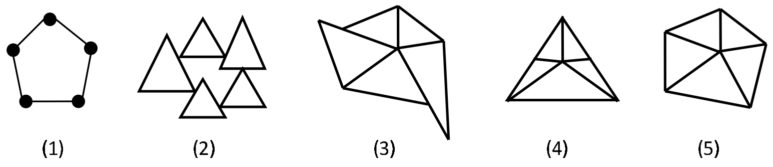


Figure 2.10: The 5-cycle graph and its triangle contact representations.

Chapter 3

Orthogonally Convex Drawings

Both *straight line drawing* and *orthogonal drawing* are very well-studied graph drawing styles (see Fig. 1.1); and convexity is a very important aspect in drawing graphs [34]. However, in contrast to the well-studied *convex drawing*, which is a straight line drawing plus a requirement that each face is drawn as a convex polygon, we know very little about convexity in orthogonal drawing.

Of course, one may simply regard the so-called *rectangular drawing* (an orthogonal drawing where all faces are drawn as rectangles) as the "convex drawing" for orthogonal drawing, since the rectangles are exactly the convex orthogonal polygons. However, this drawing style seems to be too limited. As the following theorem shows, not really many graphs enjoy such a drawing:

Theorem 3.1 ([42]). *Given a plane graph G with four designated vertices on $C_O(G)$, it admits a rectangular drawing with the four designated vertices being the four corner of the outer rectangle iff:*

1. *every 2-legged cycle contains at least two designated vertices, and*

2. *every 3-legged cycle contains at least one designated vertex.*

We note that 3-legged cycles are the "primal" counterpart of the separating triangles, which we present in Section 2.4.

In a plane graph G, an edge $e = \{u, v\} \notin E(G(C))$ is called a *leg* of C if at least one of the two vertices u and v belongs to C. The vertices in $V(C)$ that are incident to some leg of C are called the *legged-vertices* of C. C is *k-legged* if C has exactly k legged-vertices. In a biconnected plane 3-graph, each legged-vertex of C is incident to exactly one leg of C.

As a result, we suggest that *orthogonal convexity* is a more suitable candidate for studying convexity in orthogonal drawings than the traditional convexity.

In view of the above, in this chapter, we study orthogonal convexity in both orthogonal drawings and rectilinear duals.

To be more specific, our contributions are:

1. The drawing style *orthogonally convex drawing* is proposed. A necessary and sufficient condition, along with a linear time testing algorithm, is presented for biconnected plane 3-graphs to admit a no-bend orthogonally convex drawing.

2. We then present an alternative characterization for no-bend orthogonally convex drawings of biconnected plane 3-graphs. It allows us to prove the following results:

   - For any triconnected plane 3-graph, the minimum number of bends remains the same regardless of whether the drawing is orthogonally convex or simply orthogonal.

   - For any subdivision of a triconnected plane 3-graph, its orthogonally convex drawing requires at most one more bend than its orthogonal counterpart.

3. Also based on the above alternative characterization, a flow network algorithm, running in $O(n^{1.5} \log^3 n)$ time, is devised for the bend-minimization problem for biconnected plane 3-graphs.

4. Lastly, we apply our analysis of orthogonally convex drawings to characterizing internally triangulated graphs that admit the so-called Q-floorplans, which are rectilinear duals using only orthogonally convex polygons such that the outer boundary is an orthogonally convex polygon combinatorially equivalent to a given orthogonally convex polygon Q.

3.1 Related Works

The *bend-minimization* problem, a classical optimization problem in orthogonal drawings, is to minimize the total number of *bends* in the drawing. The problem is NP-complete in the most general setting, i.e., for planar graphs of maximum degree 4 [23].

Several subclasses of graphs are known to have polynomial time algorithm to find a bend-minimized orthogonal drawing, including planar graphs of maximum degree 3, series-parallel graphs, and graphs with fixed embeddings [6, 41].

Several attempts have been made to extend the model of orthogonal drawings to better comply with various requirements in practical applications. For example, to improve the readability and aesthetic feel, a new model called the *slanted orthogonal drawing* was introduced in [7]. In this model, a 90° bend is replaced by two 135° bends to smoothen the edges. To allow graphs of degree more than 4 to be drawn, the so-called *quasi-orthogonal drawing* model was invented in [27].

The current approaches for computing orthogonal drawings can be roughly divided into two categories, one uses flow or matching to model the problem (e.g., [6, 12, 41]), while the other tackles the problem in a more graph-theoretic way by taking advantage of structural properties of graphs (e.g., [35--37]). The former usually solves a more general problem, but often requires higher time complexity. On the contrary, algorithms in the latter focus on specific kinds of graphs, resulting in linear time complexity in many cases.

As we shall see in our subsequent discussion, the technique used in this chapter involves a mixture of the above two types of strategies.

For other perspectives of orthogonal drawings, the reader is referred to [18] for a survey chapter.

3.2 Terminologies

For any path, cycle, and face, we call it *boundary* iff it shares some edges with C_O.

A *contour path* P of a cycle C is a subpath of C such that P includes exactly two legged-vertices x and y of C, and x and y are the two endpoints of P. Therefore, each k-legged cycle has exactly k contour paths. If a contour path intersects (i.e., shares some edges with) the outer cycle, we call it *boundary contour path*. In fact, each boundary contour path is a subpath of C_O. Each contour path P of C is incident to exactly one face, denoted as $F_{C,P}$, in the outer region of C.

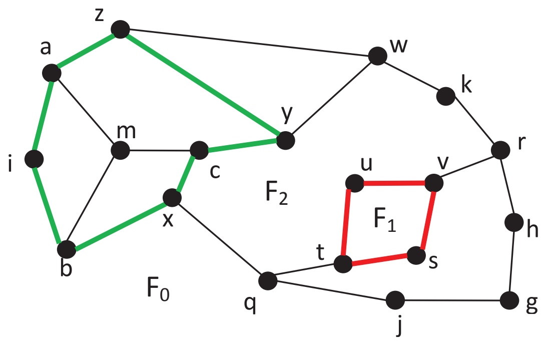


Figure 3.1: Illustration of some terms about cycles and paths.

See Fig. 3.1 for an example. Consider two cycles $C_1 = (s, t, u, v)$ and $C_2 = (x, b, i, a, z, y, c)$ (both drawn in bold line). C_1 is a non-boundary 2-legged cycle, of which two legged-vertices are t and v, and two legs are (t, q) and (v, r). C_1 is also a facial cycle, which is the contour of F_1. C_2 is a boundary 3-legged cycle, of which three legged-vertices are x, y, and z. $P_1 = (t, u, v)$ is a contour path of C_1. $P_2 = (z, a, i, b, x)$ is the boundary contour path of C_2. We have $F_{C_1,P_1} = F_2$ and $F_{C_2,P_2} = F_0$.

Let $D(G)$ be an orthogonal drawing of the plane graph G with outer cycle C_O. Given a cycle C, we use $D(C)$ (or equivalently $D(F)$ if C is the contour of a face F) to denote the drawing of C in $D(G)$.

The orthogonally convex drawing is defined as follows:

Definition 3.1. *$D(G)$ is an* orthogonally convex drawing *of G if $D(F)$ is an* orthogonally convex polygon *for each face F other than the outer one.*

In an orthogonal drawing $D(G)$, $ang_C(v)$ denotes the interior angle of v in polygon $D(C)$. We call v a *convex corner*, *non-corner*, and *concave corner* of C if $ang_C(v)$ is 90°, 180°, and 270°, respectively. A *corner* in the drawing $D(G)$ is either a bend on some edge, or a vertex v of G such that $ang_C(v) \neq 180°$ for some C. If v is a non-corner of C, v is on a side of the polygon $D(C)$.

Given a face F surrounded by its contour cycle C in a drawing D, we use $next_F(v)$ to denote the first corner after v in the counter-clockwise orientation of C; similarly, $prev_F(v)$ is defined to be the first corner after v in the clockwise orientation.

Fact 3.1. *Let v be a non-corner vertex in the common boundary of two faces F and H in an orthogonal drawing. If $next_F(v)$ is concave, $prev_H(v)$ must be convex.*

From Section 3.3 to Section 3.6, graphs under the name G are assumed to be biconnected, $\triangle(G) \leq 3$, and may have multi-edges.

3.3 Review of the Results of Rahman and Nishizeki

Among existing results concerning orthogonal drawings, Rahman et al. [36] gave a necessary and sufficient condition for a biconnected plane 3-graph to admit a no-bend orthogonal drawing, and they devised an algorithm to test the condition, and subsequently constructed such a drawing if one exists.

Theorem 3.2 ([36]). *A biconnected plane 3-graph G has a no-bend orthogonal drawing iff G satisfies the following three conditions:*

(1) There are four or more 2-vertices (i.e., vertices of degree 2) of G on $C_O(G)$.

(2) Every 2-legged cycle contains at least two 2-vertices.

(3) Every 3-legged cycle contains at least one 2-vertex.

Theorem 3.2 obviously holds even when G has multi-edges, as such graphs do not have no-bend orthogonal drawings.

The drawing algorithm in [36] performs the following steps recursively: (1) reducing the original graph G into a structurally simpler graph G^* by collapsing the so-called "bad cycles", (2) drawing G^* in a rectangular fashion, and (3) plugging in the orthogonal drawings of those bad cycles to the rectangular drawing of G^* to yield a no-bend orthogonal drawing of G.

The reader can imagine it as a "primal version" implementation of the general framework for building rectilinear duals based on separation-trees described in Section 2.4.

Algorithm([36]). No-bend-Orthogonal-Draw(G)

1. Determine four 2-vertices on $C_O(G)$ as designated corners.
2. Find the maximal bad cycles C_1, C_2, ..., C_k in G.
3. For each i, $1 \leq i \leq k$, contract cycle C_i to a single vertex v_i.
4. Let G^* be the resulting graph.
5. Find a rectangular drawing $D(G^*)$ by **Rectangular-Draw** such that the four designated corners are the corners of the bounding rectangle.
6. For each i, $1 \leq i \leq k$, extend each v_i in $D(G^*)$ to an appropriate rectangular region, and then patch $D(G(C_i))$ (using the outcome of calling **No-bend-Orthogonal-Draw**$(G(C_i))$) to $D(G^*)$ by identifying the four designated corners of $G(C_i)$ to the corners of the rectangle region.
7. Return the resulting drawing as $D(G)$.

Algorithm **Rectangular-Draw** computes the rectangular drawing of an input graph meeting the conditions in Theorem 3.1.

A key in Algorithm **No-bend-Orthogonal-Draw** above is the identification of a certain type of cycles called *bad cycles*. *Bad cycles* are cycles that are 2-legged or 3-legged if the four designated corner vertices in C_O are considered as leg-vertices. Intuitively, bad cycles are cycles that violate the conditions under which a graph admits a rectangular drawing. For instance, consider the graph in Fig. 3.1. If $\{h, i, j, k\}$ are the 4 designated vertices, then (w, z, a, i, b, x, c, y) (a 3-legged cycle as i is considered a legged-vertex) is a bad cycle, whereas (r, v, u, t, q, j, g, h) (a 4-legged cycle including legged-vertices h and j) is not a bad cycle. *Maximal bad cycles* are bad cycles that are not contained in $G(C)$ for any another bad cycle C. Note that whenever **Rectangular-Draw** is called for G^* in procedure **No-bend-Orthogonal-Draw**, G^* (with each of the maximal bad cycles contracted to a single vertex) always meets the condition for the existence of a rectangular drawing.

The reader is referred to [36] for more.

3.4 No-bend Orthogonally Convex Drawings

Our goal in this section is to give a necessary and sufficient condition for biconnected plane 3-graphs to have no-bend orthogonally convex drawings, in a way similar to Theorem 3.2.

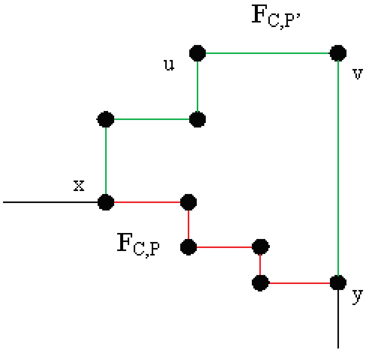


Figure 3.2: Illustration of the proof of Lemma 3.1.

Lemma 3.1. *Consider a no-bend orthogonally convex drawing $D(G)$ of a plane graph G. For every 2-legged cycle C with legged-vertices x and y and a contour path P of C, the number of convex corners of $D(C)$ in $V(P) \setminus \{x, y\}$ must be at least 1 more than that of concave corners, if either*

(1) C is a boundary cycle and P is its boundary contour path, or

(2) C is non-boundary and P is any of its contour paths.

Proof. The result is proven by contradiction. Suppose there exists a contour path P of a cycle C falsifying the lemma. Let P' be the other contour path of C. Clearly P' is a non-boundary contour path, and hence $F_{C,P'}$ is an inner face. Since the number of convex corners in $V(P) \setminus \{x, y\}$ of $D(C)$ is no more than that of the concave corners, to have the total number of convex corners in C 4 more than that of the concave corners (in view of Fact 2.1), the number of convex corners in $V(P') \setminus \{x, y\}$ of $D(C)$ must be at least 2 more than that of concave corners.

In other words, the number of concave corners in $V(P') \setminus \{x, y\}$ of $D(F_{C,P'})$ is at least 2 more than that of convex corners. Therefore, there exist consecutive concave corners in the contour of $F_{C,P'}$, so $D(F_{C,P'})$ is not orthogonally convex (Due to Fact 2.2), contradicting the assumption that $D(G)$ is orthogonally convex.

See Fig. 3.2 for an illustration. As the number of convex corners in $V(P) \setminus \{x, y\}$ is no more than that of the concave corners, there are two consecutive concave corners in $F_{C,P'}$, which are u and v in the figure. □

We are now in a position to present the main result of the section.

Theorem 3.3. *A biconnected plane 3-graph G admits a no-bend orthogonally convex drawing iff the three conditions (1), (2) and (3) in Theorem 3.2 and the following two additional conditions hold:*

(4) every non-boundary 2-legged cycle contains at least one 2-vertex on each of its contour paths, and

(5) every boundary 2-legged cycle contains at least one 2-vertex on its boundary contour path.

Proof. (⇒) Since a convex corner in a contour path not being an endpoint must be a 2-vertex, these two conditions are met according to Lemma 3.1.

(⇐) The sufficiency of the two conditions in the statement of the theorem follows from a modification to the no-bend orthogonal drawing algorithm by Rahman et al.

[36]. The construction of $D(G)$ in the algorithm **No-bend-Orthogonal-Draw** can be seen as a series of operations on the current drawing (initially a rectangular drawing), each of which extending a vertex to a rectangular region, followed by filling in a rectangular drawing.

Now we assume the input plane graph meets the conditions given in our theorem.

Suppose we reach a point at which an orthogonally convex drawing having a maximal bad cycle C_i contracted to a single vertex v_i is given. In the current drawing, no face contains consecutive concave corners. We show how to expand v_i into a rectangular region to accommodate the drawing associated with C_i and its interior.

The procedure refines the strategy developed in [36] to guarantee orthogonal convexity. As in [36], expansion of a vertex v_i can be classified into three cases depicted in Fig. 3.3, 3.4, 3.5.

Case 1: C_i is a 3-legged cycle, and v_i is a 3-vertex in G^*. Let x, y and z be its 3 leg-vertices with x, y, z and a being the 4 designated corners of $G(C_i)$. Depending on the location of a, we have 3 sub-cases, and sub-case 3 allows two alternatives.

In sub-cases 1 and 2, the expansion is straightforward and is the same as the one used in [36] since these sub-cases do not destroy orthogonal convexity of any faces. In sub-case 3, however, care must be taken in order to retain orthogonal convexity. Two cases arise depending on the convexity/concavity of the preceding and subsequent neighbors of v_i. Note that in subcase 3, for face F_1 the corner associated with a is concave.

If $next_{F_1}(v_i)$ is concave, we choose the alternative 3.2. According to Fact 2.2, whenever $next_{F_1}(v)$ is concave, $prev_{F_1}(v)$ must be convex, and hence choosing alternative 3.2 does not generate consecutive concave corners in F_1. Similarly, if $prev_{F_1}(v_i)$ is concave, we can choose the alternative 3.1. In the case both $next_{F_1}(v_i)$ and $prev_{F_1}(v_i)$ are convex, both two alternatives maintain the orthogonal convexity of F_1.

Case 2: C_i is a 2-legged cycle, and v_i is a non-corner 2-vertex in G^*. Let x and y be its 2 leg-vertices with x, y, a and b being the 4 designated corners of $G(C_i)$. Depending on the locations of a and b, we have 2 sub-cases, and sub-case 2 allows two alternatives.

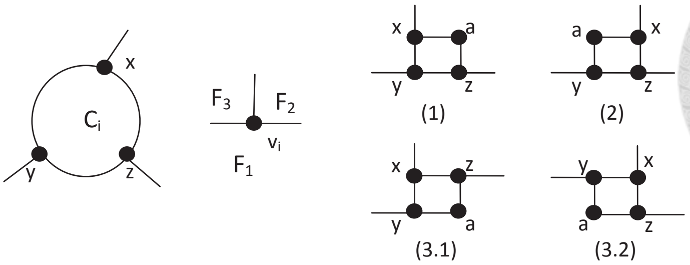


Figure 3.3: Case 1: The drawing w.r.t. a 3-legged cycle C_i.

The expansion of sub-case 1 is a bit tricky. Note that if F_2 is an inner face, orthogonal convexity of F_2 will no longer hold after the expansion. Fortunately we show that we can always choose a and b such that whenever sub-case 1 occurs, F_2 must be the outer face. If C_i is non-boundary, there must be one 2-vertex on each contour path, hence we can choose a and b such that they are on different contour paths, so that sub-case 1 will not occur. If C_i is a boundary cycle, we must have at least one 2-vertex on its boundary contour path, so we can choose a and b such that one of them is in the boundary contour path, and hence we can assure that when sub-case 1 occurs, a and b are both on the boundary contour path of C_i.

Suppose that the sub-case 2 occurs. As in the subcase 3 of Case 1, if $next_{F_1}(v_i)$ is concave, we choose the alternative 2.2, if $next_{F_2}(v_i)$ is concave, we choose the alternative 2.1, otherwise, we can choose either of them. It is easy to deduct from Fact 3.1 that our choices do not generate consecutive concave corners in both F_1 and F_2.

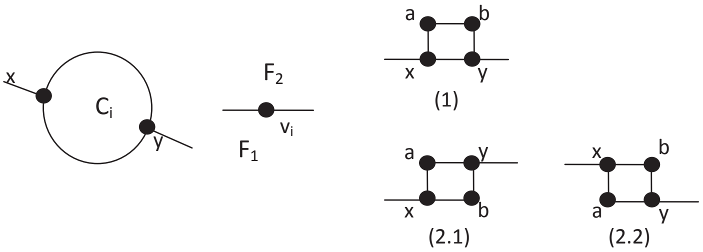


Figure 3.4: Case 2: The drawing w.r.t. a 2-legged cycle C_i with v_i a non-corner 2-vertex.

Case 3: C_i is a 2-legged cycle, and v_i is a corner 2-vertex in G^*. Let x and z be its 2 leg-vertices, y being a corner vertex of the current drawing, and x, y, z and a being the 4 designated corners of $G(C_i)$. Depending on the location of a, we have 3 sub-cases.

All these sub-cases do not destroy orthogonal convexity of any inner faces, since we can apply the same trick for sub-case 1 of case 2 to the sub-cases 2, 3 here, so that F_2 is always the outer one when sub-cases 2, 3 occur.

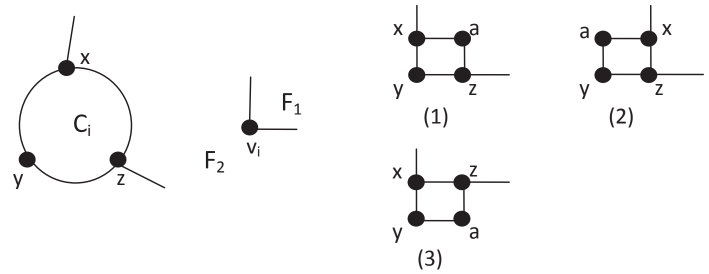


Figure 3.5: Case 3: The drawing w.r.t. a 2-legged cycle C_i with v_i a corner 2-vertex.

In view of the above, by an induction on the number of operations of expanding a vertex to a rectangular region, we obtain a constructive proof for the sufficiency of Theorem 3.3. Hence we conclude the proof. □

Based on a linear time implementation of Algorithm **No-bend-Orthogonal-Draw(G)** described in [36], we have the following result.

Theorem 3.4. *Given a biconnected plane 3-graph G, there is a linear time algorithm to construct a no-bend orthogonally convex drawing $D(G)$ if G admits one.*

3.5 An Alternative Condition

Now we turn our attention to the bend minimization problem w.r.t orthogonally convex drawings of plane graphs. Intuitively, minimizing the number of bends in orthogonal (convex) drawing can be equated with subdividing a minimum number of edges so as to yield a graph having a no-bend orthogonal (convex) drawing.

It is known that network flows, a popular technique for analyzing graph-related problems, are very useful in designing algorithms to minimize bends in orthogonal drawing as reported in, e.g., [18]. It is therefore a natural attempt to see whether the additional orthogonal convexity requirement described in the previous section (in particular, conditions (4) and (5) in Theorem 3.3) can be incorporated into flow networks.

As one can see from Theorem 3.3, contour paths along (boundary or non-boundary) 2-legged cycles play a vital role as far as a graph having an orthogonally convex drawing is concerned. Due to possible overlaps of 2-legged cycles in a plane graph, the corresponding contour paths may intersect each other messily, resulting in difficulties if one attempts capture the amount of convex/concave corners along contour paths by a flow network formulation.

To ease the above problem, we identify a subset of representative paths which are on the one hand, "simple enough" to enable a min-cost flow formulation for the bend-minimization problem, and on the other hand, "sufficient enough" to characterize the presence of no-bend orthogonally convex drawings. The simplicity stems from the following observation: for any paths P and P', if $P \subset P'$, then having a 2-vertex in P guarantees the presence of a 2-vertex in P'.

We are now in a position to identify two types of cycles, namely, *proper* and *improper* cycles, which are later used to characterize the representative paths mentioned above.

A *contraction* of a 2-vertex v with adjacent vertices x and y is to remove v and its incident edges $\{v, x\}$ and $\{v, y\}$ and then add a new edge $\{x, y\}$. Let G^c denote the graph resulting from contracting every 2-vertex of G. Since we require G to have maximum degree 3, G^c must be 3-regular.

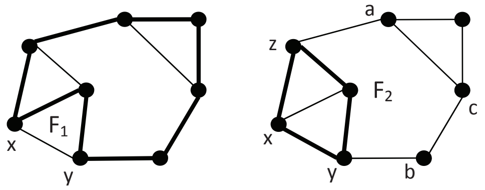


Figure 3.6: Proper and improper 2-legged cycles.

A 2-legged cycle of G is called *improper* if its two legs correspond to the same edge in G^c. A 2-legged cycle is called *proper* if it is not improper. See Fig. 3.6 for illustrations. The left one shows an improper 2-legged cycle (drawn as a bold line) with leg-vertices x, y. The right one shows a proper 2-legged cycle (drawn as a bold line) with leg-vertices y, z.

Due to 3-regularness of G^c and the fact that the two legs of an improper cycle C are the same edge e in G^c, there remains nothing outside $G(C)$ except the leg e. Therefore, an improper cycle must be a boundary cycle, or conversely, all non-boundary 2-legged cycles are proper. It is easy to see that every inner face that intersects C_O in exactly one path yields an improper 2-legged cycle naturally. It is because the intersecting path, which connects the two leg-vertices of a 2-legged cycle, must be contracted into a single edge in G^c. See face F_1 in the left figure of Fig. 3.6. We state these simple but useful observations as a fact.

Fact 3.2. *Let C be a 2-legged cycle of G with two leg-vertices x and y, the following statements are equivalent:*

(1) C is improper.

(2) $E(G) \setminus E(G(C))$, form exactly one path (a subpath of C_O linking x and y).

(3) The two legs correspond to the same edge in G^c.

(4) C is a boundary 2-legged cycle, and boundary of $F_{C,P}$ intersects C_O of G in exactly 1 path, where P is the non-boundary contour path of C.

Consider Fig. 3.6 for examples of proper and improper cycles. F_1 and F_2 are two boundary faces corresponding to the $F_{C,P}$ of the 2-legged cycles C drawn as bold lines and their non-boundary contour paths P in the left and right figures, respectively. The contour of F_1 intersects C_O in exactly one path (x, y), whereas the contour of F_2 intersects C_O in two paths (z, a) and (y, b, c). Note that both left and right illustrations show the whole graph instead of a subgraph.

Recall from Theorem 3.3 that contour paths of boundary or non-boundary 2-legged cycles are keys to orthogonal convexity in no-bend orthogonal drawings. In what follows, we replace Conditions (4) and (5) of Theorem 3.3 with new conditions on two sets of paths, namely, *critical paths* and paths in S_G, which are contour paths of proper cycles and improper cycles, respectively. As we shall see later, these two sets of representative paths enjoy the following nice properties, which further facilitate a flow network formulation for bend minimization:

- paths in S_G are mutually edge-disjoint,
- critical paths are mutually edge-disjoint, and
- any path in S_G is either contained in a critical path or contained in no critical path.

We are now in a position to give the definitions of critical paths and S_G.

Definition 3.2. *A path P of G is called* critical *if there is a proper 2-legged cycle C such that:*

(1) P is a contour path of C,

(2) if C is a boundary 2-legged cycle, then P is the boundary contour path of C, and

(3) P does not edge-intersect any proper 2-legged cycle C' that is contained in $G(C)$.

To proceed further, we require the following two lemmas.

Lemma 3.2. *For any biconnected plane 3-graph G, the critical paths of G are edge-disjoint.*

Proof. Assume that there exist two different critical paths P and P' edge-intersecting each other. Let C and C' be two proper 2-legged cycles such that P and P' are contour paths of C and C', respectively. As a boundary contour path does not edge-intersect any non-boundary contour path, we divide the situation into the following two cases.

Case 1: P and P' are both non-boundary contour paths. In this case, C and C' are non-boundary 2-legged cycles with $E(G(C)) \cap E(G(C')) \neq \emptyset$.

In what follows, we show that either $G(C) \subseteq G(C')$ or $G(C') \subseteq G(C)$ is true, and hence the condition (3) in Definition 3.2 is violated.

Suppose that $G(C) \nsubseteq G(C')$ and $G(C') \nsubseteq G(C)$. Then, obviously, the two legs e_1, e_2 of C' satisfies $e_1, e_2 \in E(G(C))$:

- If there is no leg of C' contained in $E(G(C))$, then $G(C)$ is disconnected, as there is no edge in $E(G(C))$ connecting $V(G(C)) \setminus V(G(C'))$ and $V(G(C)) \cap V(G(C'))$.

- If there is only one leg e_1 of C' with $e_1 \in E(G(C))$, then $G(C)$ is not biconnected, as e_1 is the only edge in $E(G(C))$ connecting $V(G(C)) \setminus V(G(C'))$ and $V(G(C)) \cap V(G(C'))$.

Similarly, the two legs of C belongs to $E(G(C'))$. This implies that there is no path linking vertices in $V(G)\setminus(V(G(C))\cup V(G(C')))$ and vertices in $V(G(C))\cup V(G(C'))$, and hence G is not connected, a contradiction.

Case 2: P and P' are both boundary contour paths. In this case, C and C' are boundary 2-legged cycles with $E(G(C)) \cap E(G(C')) \neq \emptyset$.

First of all, in view of our argument for Case 1, "$G(C) \not\subseteq G(C')$ and $G(C') \not\subseteq G(C)$" requires "$V(G(C)) \cup V(G(C')) = V(G)$".

Suppose that that there is an edge $e \notin E(G(C)) \cup E(G(C'))$, then it must be a leg of both C and C'. This is impossible as the two legs of C belongs to $E(G(C'))$. Therefore, we can infer that $E(G) \setminus E(G(C)) \subseteq E(G(C'))$.

Since C is proper, $E(G)\setminus E(G(C))$ must not be a single path. Therefore, there exists a proper boundary 2-legged cycle C'' such that $E(G(C'')) \subseteq E(G)\setminus E(G(C)) \subseteq E(G(C'))$. This contradicts the assumption that P' is a critical path. □

Lemma 3.3. *Let P be a path satisfying (1) and (2) in Definition 3.2. If P is not critical, there must be a critical path P' such that $P' \subset P$.*

Proof. Assume that there exist some non-critical paths that contradict the statement of the lemma. We choose the path P to be the shortest among them. Let C be the proper 2-legged cycle having P as its contour path. Since P is not critical, there must be a proper 2-legged cycle C' that is contained in $G(C)$, and P edge-intersects with C'. P must edge-intersect a contour path P' of C'. It is easy to see that P' satisfies (1) and (2) of Definition 3.2, and $P' \subset P$.

From our assumption, P' is not critical. But according our choice of P, there must be a critical path P'' such that $P''\subset P'\subset P$, which is a contradiction to the choice of P. □

Given a path P with endpoints x and y, we write $P_{(x\frown y)}$ to denote the "open" version of P, i.e., excluding x and y. That is, $P_{(x\frown y)}$ consists of $V(P)\setminus\{x,y\}$ and $E(P)$.

We now define S_G, a set of paths associated with improper 2-legged cycles in graph G, as follows:

$$S_G = \{\ C_O \backslash P_{(x\frown y)} \mid P \text{ is a boundary contour path of } C, \text{ where } C \text{ is an improper 2-legged cycle with two legged-vertices } x \text{ and } y \text{ in } G\ \}.$$

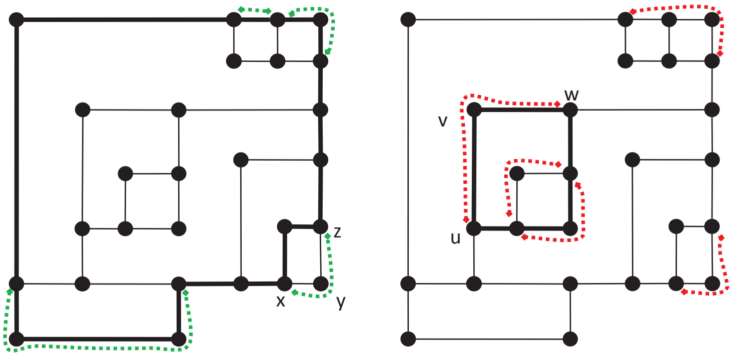


Figure 3.7: Critical paths and S_G in a plane graph.

We note that the paths in (2) of Fact 3.2 are exactly paths in S_G. Therefore, intermediate vertices in paths of S_G are 2-vertices, and paths in S_G must be a subpath of C_O, and hence $P\in S_G$ iff P is a boundary contour path of a facial cycle having only one boundary contour path. The following fact summarizes the above observations.

Fact 3.3. *Let P be a path of G with two end-vertices x and y, the following statements are equivalent:*

(1) P is in S_G.

(2) P is the boundary contour path of a facial cycle C that intersects C_O of G in exactly one path.

(3) P is $C_O \backslash P'_{(x \frown y)}$, for some P' which is the boundary contour path of some improper 2-legged cycle with two legged-vertices x and y.

To have a better grasp of critical paths and S_G, consider Fig. 3.7 in which a no-bend orthogonally convex drawing of a plane graph G is shown. In the left figure, the four dotted paths are those in S_G, which are edge-disjoint. Let C be the 2-legged cycle drawn in bold, and P be its boundary contour path. We have $C_O \backslash P_{(x \frown z)} = (x, y, z)$. In the right figure, the five dotted paths are critical paths, which are edge-disjoint. Let C be the 2-legged cycle drawn in bold. We have (1) the path (u, v, w) is one of its contour paths, (2) C is a non-boundary 2-legged cycle, and (3) P does not edge-intersect any proper 2-legged cycle other than C that is contained in $G(C)$. A path in S_G is either contained in exactly one critical path or intersects with no critical path.

The following theorem, which is the main result in this section, enables us to characterize no-bend orthogonally convex drawings in terms of critical paths and S_G.

Theorem 3.5. *Suppose a biconnected plane 3-graph G has a no-bend orthogonal drawing. G has a no-bend orthogonally convex drawing iff the following conditions are satisfied:*

(1) Every critical path of G contains at least one 2-vertex.

(2) For each $P \in S_G$, $V(C_O) \backslash V(P)$ contains at least one 2-vertex.

Proof. It suffices to prove that the conditions stated here imply the two additional conditions stated in Theorem 3.3 since the other direction is straightforward. Condition (2) simply means that every improper 2-legged cycle contains at least one 2-vertex on its boundary contour path. According to Lemma 3.3, for every non-critical path P that is a contour path of a proper 2-legged cycle, there must be a critical path P' that is a subpath of P. Therefore, if condition (1) is satisfied, every path that is a contour path of a non-boundary 2-legged cycle, or a boundary contour path of a proper boundary 2-legged cycle, contains at least one 2-vertex. □

We note that both S_G and the set of all critical paths can be found easily in linear time. Since they are edge-disjoint sets of paths, a contour edge-traversal for each face suffices to list all of them.

In the following we demonstrate several interesting results which can be established using Theorem 3.5.

Theorem 3.6. *Given a triconnected plane 3-graph G, let $SD(G)$ be the set of graphs obtained by subdividing edges of G such that either (1) there are at most 3 2-vertices in C_O or (2) there is no facial cycle intersecting C_O that contains all 2-vertices in C_O. For each $G' \in SD(G)$, suppose b_1 (resp., b_2) is the minimum number of bends needed to construct orthogonal drawing (resp., orthogonally convex drawing) of G', then $b_1 = b_2$.*

Proof. For each $G' \in SD(G)$, we first construct an orthogonal drawing $D(G')$ using b_1 bends. Let G_1 be the graph resulting from making each bend in $D(G')$ a new 2-vertex. As a result, G_1 has a no-bend orthogonal drawing, and $|V(G_1)| = |V(G')| + b_1$.

We claim that G_1 contains no critical paths. Suppose not, then it must have a proper 2-legged cycle C. According to the (3) of Fact 3.2, the two legs of C correspond to different edges in ${G_1}^c$. Therefore, removing these two edges suffices to disconnect ${G_1}^c$, implying that ${G_1}^c$ is not 3-edge-connected. As edge-connectivity $\geq$ vertex-connectivity, ${G_1}^c$ is not triconnected, contradicting that G_1 is a subdivision of triconnected plane 3-graph.

As a result, according to Theorem 3.5, G_1 has a no-bend orthogonally convex drawing iff $V(C_O(G_1)) \backslash V(P)$ contains at least one 2-vertex for every $P \in S_{G_1}$. As a 2-vertex belongs to at most one path in S_{G_1}, the above condition essentially requires that there is no path in S_{G_1} containing all 2-vertices in $C_O(G_1)$. If there is indeed no such path, the theorem follows. Hence, in the next, we assume that there is such a path P.

In the following we construct a graph G_2 which is also a subdivision of G' with $V(G_2) = V(G') + b_1$. As G_1 has a no-bend orthogonal drawing, there are at least 4 2-vertices in $C_O(G_1)$ (and hence in P). Apparently there is a facial cycle (which contains P as a subpath) intersecting $C_O(G')$ contains all 2-vertices in $C_O(G')$, so there are at most 3 2-vertices in $C_O(G')$ due to the statement of this theorem. Therefore, there is a 2-vertex v in P which is introduced by a subdivision (i.e., $v \in V(G_1)$ but $v \notin V(G')$). We let G_2 be the resulting graph of contracting v and subdividing an edge e in $E(C_O(G_1)) \setminus E(P)$.

It is easy to see that G_2 satisfies Theorem 3.5 and hence admits a no-bend orthogonally convex drawing $D'(G_2)$. It is immediately that D' is also an orthogonally convex drawing of G' using b_1 bends. As $b_2 \geq b_1$, we have $b_1 = b_2$.

□

The following two results immediately follow from Theorem 3.6.

Corollary 3.1. *For any triconnected plane 3-graph G, the minimum number of bends needed to construct an orthogonally convex drawing is the same as that of an orthogonal drawing.*

Corollary 3.2. *For any subdivision of a triconnected plane 3-graph G, the minimum number of bends needed to construct an orthogonally convex drawing is at most one more than that of an orthogonal drawing.*

See Fig. 3.8 for an example of a subdivision of a triconnected plane 3-graph whose bend-minimized orthogonally convex drawing has exactly one more bend than its bend-minimized orthogonal drawing.

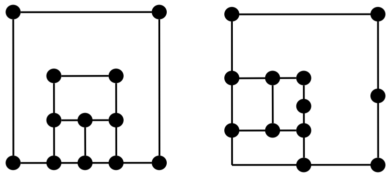


Figure 3.8: An example for Corollary 3.2.

In contrast of the above results, there are infinite number of biconnected plane 3-graphs whose bend-minimized orthogonally convex drawings have $V(G)/2 - O(1)$ more bends than their bend-minimized orthogonal drawings. See Fig. 3.9 for a series of plane graphs achieving this difference in their bend-minimized orthogonal drawings and bend-minimized orthogonally convex drawings.

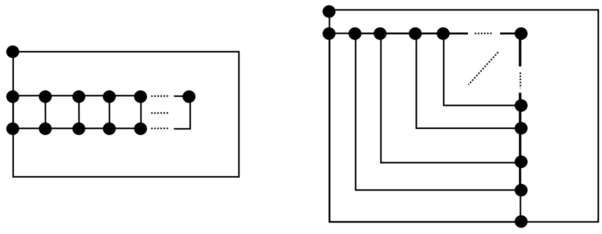


Figure 3.9: Bend-minimized orthogonal drawings and bend-minimized orthogonally convex drawings.

3.6 Flow Formulation for Bend-minimization

In this section, we tailor the planar min-cost flow formulation originally designed for orthogonal drawing [41] to coping with orthogonal convexity. Our strategy is considerably different to most of the previous approaches using flow networks to design graph drawing algorithms. Instead of finding a desired drawing directly by applying min-cost flow, our flow network solves the problem indirectly in the sense that it adds a minimum number of new 2-vertices to the input graph to satisfy Theorem 3.5. A bend-minimized drawing can then be constructed using Theorem 3.4.

To make our subsequent discussion clear, we use *arc* and *node* instead of edge and vertex, respectively, in describing a flow network. A *min-cost flow network* is a directed multi-graph $N = (W, A)$ associated with four functions: *lower bounds* $\lambda : A \rightarrow \mathbb{Z}_{\geq 0}$, *capacities* $\mu : A \rightarrow \mathbb{Z}_{\geq 0} \cup \{\infty\}$, *costs* $c : A \rightarrow \mathbb{Z}_{\geq 0}$, *demands* $b : W \rightarrow \mathbb{Z}$. A map $f : A \rightarrow \mathbb{Z}_{\geq 0}$ is a *flow* if the following constraints are met:

$$\forall v \in W,\ b(v) + \sum_{(u,v)\in A} f(u,v) - \sum_{(v,u)\in A} f(v,u) = 0, \quad \forall a \in A,\ \lambda(a) \leq f(a) \leq \mu(a)$$

The cost of a flow f is $c(f) = \sum_{a\in A} f(a) \times c(a)$. We first describe the flow network $N_G = (W_G, A_G)$ associated with a biconnected plane 3-graph G in which each flow in N_G corresponds to an orthogonal drawing of G:

- $W_G = W_V \cup W_F$, where W_V and W_F are the vertex set and face set (including the outer face) of G, respectively, Furthermore, $\forall u_v \in W_V$, $b(u_v) = 2$ if $deg_G(v) = 3$; $b(u_v) = 0$ if $deg_G(v) = 2$. $\forall u_F \in W_F$, $b(u_F) = -4$ if F is an inner face; $b(u_F) = 4$ if F is the outer face.

- $A_G = A_V \cup A_F$, where
  - $A_V = \{(u_v, u_F), (u_F, u_v) | deg(v) = 2\} \cup \{(u_v, u_F) | deg(v) = 3\}$, where $v \in V(G), F \in \text{face}(G), v$ incident to F. $\forall a \in A_V$, $\lambda(a) = 0$, $\mu(a) = 1$, and $c(a) = 0$.
  - $A_F = \{(u_F, u_H) | F, H \in \text{face}(G),$ and F adjacent to $H\}$ is a multi-set of arcs between faces, and the number of (u_F, u_H) in A_F equals the number of shared edges in contours of F and H. We use $(u_F, u_H)_e$ to indicate the specific arc that corresponds to the shared edge e. $\forall a \in A_F$, $\lambda(a) = 0$, $\mu(a) = \infty$, and $c(a) = 1$.

Although our definition of N_G is slightly different from the original one given in [41], the validity of N_G is apparent as the following explains. Every flow f in N_G corresponds to an orthogonal drawing $D(G)$, and vice versa, such that

- $f(u_v, u_F) - f(u_F, u_v) = -1, 0, 1$ means v is a concave corner, non-corner, convex corner in $D(F)$, respectively,
- $f(u_F, u_H)_e$ is the number of bends on e that are concave corners in $D(F)$ and convex corners in $D(H)$, and
- the total number of bends in $D(G)$ equals $c(f)$.

The reader is referred to Fig. 3.10 for an illustration of flow network N_G: The up-left picture is the graph G, the right one is the flow network, and the down-left one is a bend-minimized orthogonal drawing. Every bi-directed arc represents two arcs with opposite direction. The flow f representing the drawing is defined as follows: $f(F_0, F_1) = 2, f(F_0, F_2) = f(F_0, F_3) = f(x, F_2) = f(x, F_3) = f(y, F_1) = f(y, F_3) = f(z, F_1) =$

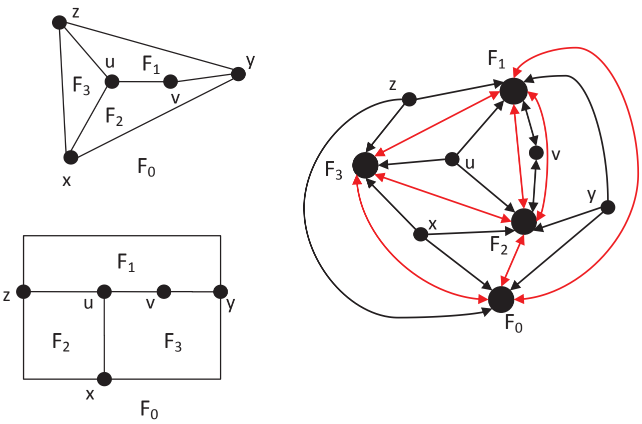


Figure 3.10: Illustration of the flow network N_G.

$f(z, F_2) = f(u, F_2) = f(u, F_3) = 1$, and $f(a) = 0$ for the remaining arcs a. The cost of f is 4.

The following fact is an immediate consequence of the above discussion. It establishes a correspondence from "the difference between the numbers of convex corners and concave corners in a portion of a contour of a face" to "the amount of flow passing through it".

Fact 3.4. *Let S_1 (resp., S_2) be any subset of edges (resp., vertices) along the contour of a face F. For any $e \in S_1$, we write F_e to denote the face incident to e other than F. For a flow f in N_G and its corresponding orthogonal drawing D, we must have $\sum_{e \in S_1}[f(u_{F_e}, u_F)_e - f(u_F, u_{F_e})_e] + \sum_{v \in S_2}[f(u_v, u_F) - f(u_F, u_v)]$ equaling the difference between the numbers of convex corners and concave corners in the portion $S_1 \cup S_2$ of $D(F)$.*

We have the following lemma, which reduces the existence of a no-bend orthogonally convex drawing to the existence of a no-bend orthogonal drawing plus some constraints about numbers of convex corners and concave corners in some paths. This lemma together with the above fact lay down the foundation for us to find bend-minimized orthogonally convex drawings based on a modification to the above flow network N_G.

Lemma 3.4. *A biconnected plane 3-graph G admits a no-bend orthogonally convex* *drawing iff there is a no-bend orthogonal drawing (not necessarily orthogonally convex) such that*

(1) for every critical path P along a contour path of 2-legged cycle C, $\#_{cc}(P_{(x\frown y)})$ $> \#_{cv}(P_{(x\frown y)})$ in $F_{C,P}$, and

(2) for every P in S_G, $\#_{cc}(P_{(x\frown y)}) \leq 3 + \#_{cv}(P_{(x\frown y)})$ in the outer face,

where P has endpoints x and y, and $\#_{cv}(\cdot)$ and $\#_{cc}(\cdot)$ represent the numbers of convex and concave corners, respectively.

Proof. ($\Leftarrow$) It is easy to see that the two conditions imply the two conditions in Theorem 3.5 (In the first condition, concave corners of $P_{(x\frown y)}$ in $F_{C,P}$ must be 2-vertices; and the second condition implies that there must be a concave corner in $C_O \setminus P_{(x\frown y)}$, which must be a 2-vertex, too); hence, we conclude the "if" part of the lemma.

($\Rightarrow$) Now, suppose G admits a no-bend orthogonally convex drawing. According to Lemma 3.1, the first condition is always satisfied for every no-bend orthogonally convex drawing. Therefore, it suffices to show that there exists a no-bend orthogonally convex drawing $D(G)$ such that the second condition is satisfied. We show that if there is a path P contradicting the second condition, we can modify the drawing in a way that orthogonal convexity of each face is preserved. Let F be the face incident to P other than the outer face, and C be its contour.

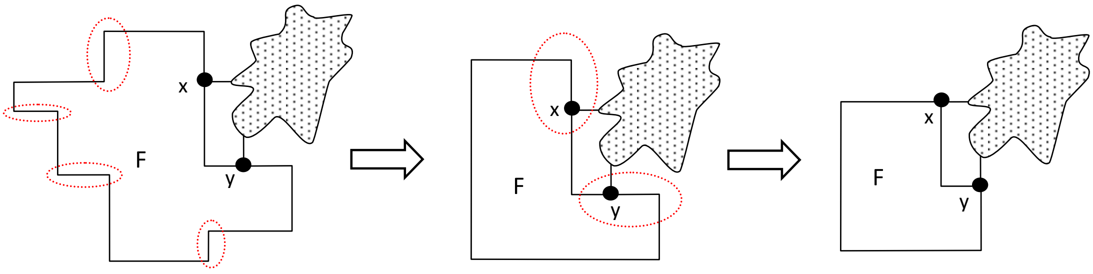


Figure 3.11: Illustration of the proof of Lemma 3.4.

Suppose the second condition of the lemma does not hold. Then, the number $k = \#_{cv}(P_{(x\frown y)}) = \#_{cc}(P_{(x\frown y)})$ in F is more than 3. Note that a concave (resp., convex) corner of $P_{(x\frown y)}$ in the outer face must be convex (resp., concave) in F.

It is clear that k cannot be more than 5; otherwise there must be consecutive concave corners in $C \setminus P$ in F. If there is any concave corner v of $P_{(x \frown y)}$ in F, let w be its nearest corner vertex in P, which must be convex in F. We can make them both non-corner in F without introducing any consecutive concave corners. Therefore, we can assume there are exactly 4 or 5 convex corners of $P_{(x \frown y)}$ in F. If the number is 4, one of x and y must be non-corner in F, and if it is 5, then both x and y are non-corner in F; otherwise there must be consecutive concave corners in $C \setminus P$ in F. Note that x and y cannot be concave in F since they are of degree 3. Suppose x (or y) is non-corner in F, and let z be any convex corner of $P_{(x \frown y)}$ in F. We can make x (or y) convex in F and z non-corner in F without introducing any consecutive concave corners. Hence we can reduce the number k to be at most 3, which concludes the proof. (We actually prove a stronger result than the statement of lemma in the "only if" part, since the drawing constructed is orthogonally convex.) See Fig. 3.11 for a graphical illustration of removing additional convex corners. □

In what follows, we show how to construct a flow network N'_G from N_G in such a way that a flow of N'_G corresponds to an orthogonal drawing meeting the conditions stated in Lemma 3.4.

We use Fig. 3.12 as a graphical illustration of the procedure. Note that Fig. 3.12(1) shows a portion the a graph G with F_0 the outer face, $P_1 = (x, y, z)$ and $P_2 = (z, w)$ the two paths in S_G, and $P_3 = (x, y, z, w)$ a critical path; Fig. 3.12(2) shows its corresponding portion in N_G.

Initially we set $N'_G = N_G$.

- $\forall P \in S_G$ with endpoints x, y, let the outer face be F', and let $S_{P,F'}$ denotes the set of faces bordering F' along some edges in the path P.
  - add a new node u_P to $W(N'_G)$, and two arcs $(u_{F'}, u_P)$, $(u_P, u_{F'})$ to $A(N'_G)$.
  - set $b(u_P) = 0$, $\lambda(u_{F'}, u_P) = \lambda(u_P, u_{F'}) = 0$, $\mu(u_{F'}, u_P) = 3$, $\mu(u_P, u_{F'}) = \infty$, and $c(u_{F'}, u_P) = c(u_P, u_{F'}) = 0$.

  - redirect all the arcs in the current $A(N'_G)$ of the following forms: $(u_{F'}, u_v)$, $(u_v, u_{F'})$, $(u_{F'}, u_F)_e$, $(u_F, u_{F'})_e$ for all $v \in V(P) \setminus \{x, y\}$, $F \in S_{P,F'}$, $e \in E(P)$ by replacing $u_{F'}$ with u_P.

  See Fig. 3.12(3) for the modification to N_G for the two paths P_1 and P_2 in S_G, in which the newly added arcs are drawn as dotted lines. In view of Fact 3.4, such a modification makes the orthogonal drawing corresponding to any flow f of N'_G in compliance with the condition 2 of Lemma 3.4 (view it as a no-bend drawing by treating all bends as 2-vertices).

- $\forall$ critical path P with endpoints x, y, C the 2-legged cycle for which P is its contour path, and S the set of faces in $G(C)$ that border P,

  - add a new node u_P to $W(N'_G)$, and a new arc $(u_{F_{C,P}}, u_P)$ to $A(N'_G)$.
  - set $b(u_P) = 0$, $\lambda(u_{F_{C,P}}, u_P) = 1$, $\mu(u_{F_{C,P}}, u_P) = \infty$, and $c(u_{F_{C,P}}, u_P) = 0$.
  - redirect all the arcs in the current $A(N'_G)$ of the following forms: $(u_{F_{C,P}}, u_{P'})$, $(u_{P'}, u_{F_{C,P}})$, $(u_{F_{C,P}}, u_v)$, $(u_v, u_{F_{C,P}})$, $(u_{F_{C,P}}, u_F)_e$, $(u_F, u_{F_{C,P}})_e$ for all $P' \in S_G$ such that $P' \subseteq P$, $v \in V(P) \setminus \{x, y\}$, $F \in S$, $e \in E(P)$ by replacing $u_{F_{C,P}}$ with u_P.

  See Fig. 3.12(4) for the modification to N_G for the critical path P_3, in which the newly added arc is drawn as a dashed line. Similarly, due to Fact 3.4, condition 1 of Lemma 3.4 holds for any orthogonal drawing corresponding to a flow f in the modified network (view it as a no-bend drawing by treating all bends as 2-vertices).

Since critical paths are mutually edge-disjoint according to Lemma 3.2, and since every path in S_G is either a subpath of a critical path or intersects no critical path, the construction process is valid and can be done in linear time, the planarity of N'_G is preserved, and the number of newly added arcs and nodes is linear in $|V(G)|(= n)$. Note that the maximum possible value of the minimum cost is also $O(n)$. Therefore, an

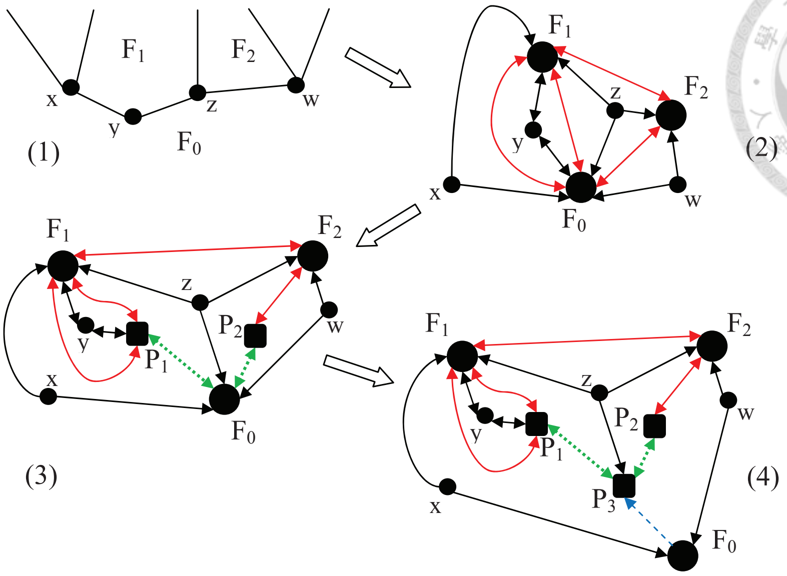


Figure 3.12: Illustration of the construction of N'_G.

optimal solution of N'_G can be solved using the $O(n^{1.5}\log^3 n)$ time algorithm for planar min-cost flow with cost $O(n)$ described in [12].

Note that since we cannot make the flow uncapacitated, a more efficient algorithm of $O(n^{1.5})$ complexity also described in [12] cannot be adapted.

We are in a position to prove the main theorem in this section.

Theorem 3.7. *For any biconnected plane 3-graph G, a bend-minimized orthogonally convex drawing of G can be constructed in $O(n^{1.5}\log^3 n)$ time.*

Proof. Consider an orthogonal drawing D of G corresponding to a min-cost flow of N'_G. According to our construction of N'_G, D satisfies Lemma 3.4 (view it as a no-bend drawing by treating all bends as 2-vertices). Such a drawing is the one using a minimum number of bends, say s, among all drawings of G satisfying Lemma 3.4. Let G_1 be the graph resulting from making all bends in the drawing D of G as 2-vertices. According to Lemma 3.4, G_1 has a no-bend orthogonally convex drawing D_1 which can be constructed in linear time according to Theorem 3.4. It is clear that D_1 is an orthogonally convex drawing of G using s bends.

We claim that D_1 is a bend-optimal orthogonally convex drawing of G. Suppose that there exists an orthogonally convex drawing D_2 of G using t bends such that $t < s$.

Let G_2 be the graph resulting from making all bends in the drawing D_2 of G as 2-vertices. Then, G_2 has a no-bend orthogonal drawing D_3 meeting the two conditions in Lemma 3.4. Therefore, D_3 corresponds to a feasible flow of N'_G whose cost value is t. This contradicts the fact that the orthogonal drawing D corresponds to an optimal solution of N'_G as $t < s$. Therefore, D_1 is indeed bend-optimal.

The time complexity of the above procedure involves (1) construction of N'_G, (2) calculation of min-cost flow of N'_G, and (3) construction of D_1 based on Theorem 3.4. Both (1) and (3) take linear time. Task (2) is the bottleneck which requires $O(n^{1.5} \log^3 n)$ time. Hence the theorem is concluded. □

3.7 Orthogonal Convexity in Rectilinear Duals

In this section, we study orthogonal convexity in rectilinear duals.

The graph class under investigation in this section is the class of simple, connected, and internally triangulated plane graphs. A plane graph is *internally triangulated* if all the inner faces are triangles.

Definition 3.3. *Let Q be an orthogonal polygon, we write Q-*floorplan *to denote a rectilinear dual whose outer boundary is combinatorially equivalent to Q (the rectangular boundary constraint in Definition 2.3 is relaxed). A Q-floorplan is orthogonally convex if its polygons are orthogonally convex.*

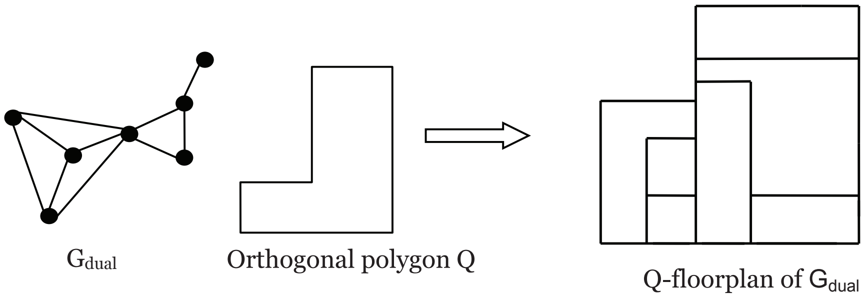


Figure 3.13: An example of a Q-floorplan.

In this section, graphs under the name G_{dual} are assumed to be simple, connected, internally triangulated plane graphs. Our goal is to give a necessary and sufficient condition to test whether a given graph G_{dual} has an orthogonally convex Q-floorplan w.r.t. a given orthogonally convex polygon Q. See Fig. 3.13 for an example. As we shall see later, it is often the case that problems about rectilinear duals can be re-stated as problems about orthogonal drawings. As in the last section, Theorem 3.5 and the construction process of no-bend orthogonally convex drawings described in Section 3.4 play important roles in proving the desired result.

We note that the *weak dual* of a plane graph G is the subgraph of the dual graph that excludes the vertex v in the dual graph that corresponds to the outer face $F_O(G)$ and all the edges in the dual graph incident to v.

Lemma 3.5. *For any simple, connected, internally triangulated plane graph G_{dual}, there is a unique biconnected 3-regular plane multi-graph G_{primal} such that G_{dual} is the weak dual of G_{primal}, and the following properties hold:*

(1) G_{primal} does not have any non-boundary 2-legged cycle, and

(2) internal faces (which are orthogonal polygons) of an orthogonal drawing of G_{primal} form a rectilinear dual of G_{dual}.

Proof. We use the following procedure to construct G_{primal} from a given G_{dual}, and then show that G_{primal} indeed satisfies the conditions stated in the lemma.

Input: G_{dual} - a simple, connected and internally triangulated plane graph

Output: G_{primal}

- Suppose $C_O = (v_1, v_2, \ldots, v_s)$ is the outer cycle of G_{dual}, which may have repeated vertices.

- Add a new vertex t in the outer face of G_{dual}, and then triangulate the outer face by adding edge $\{v_i, t\}$ for $1 \leq i \leq s$ to construct a triangulated plane multi-graph G'

- Take the dual of G' to yield G_{primal}.

See Fig. 3.14 for examples of G_{dual}, G' and G_{primal}.

It is clear that G_{primal} is biconnected (otherwise G_{dual} is not connected) and 3-regular (otherwise G_{dual} is not internally triangulated).

Suppose G_{primal} has a non-boundary 2-legged cycle C. Let F_1 and F_2 be its two neighboring faces in the outer region of C. Let v_1 and v_2 be two vertices in G_{dual} that correspond to F_1 and F_2, respectively. Then there must be multi-edges linking v_1 and v_2 in G_{dual} since the two legs of C both border F_1 and F_2, which contradicts the fact that G_{dual} is a simple graph. Hence G_{primal} does not have any non-boundary 2-legged cycle. The fact that internal faces of an orthogonal drawing of G_{primal} form a rectilinear dual of G_{dual} directly follows from the weak duality between G_{primal} and G_{dual}. □

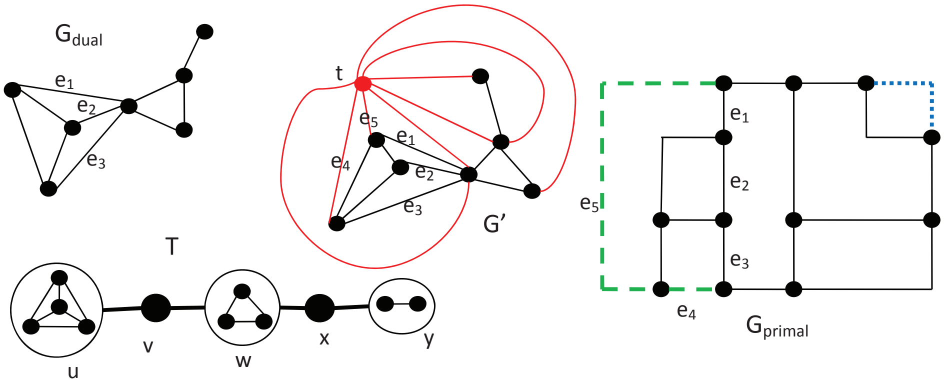


Figure 3.14: The construction of G_{primal} and the block-cutvertex tree of G_{dual}.

Recall that our goal is to characterize graphs G_{dual} that admit orthogonally convex Q-floorplans given an orthogonally convex polygon Q, and subsequently realize such floorplans. We use $\mathrm{numSide}(P)$ to denote the number of sides of the polygon P with non-corner vertices neglected. As we shall see later, the number of the boundary critical paths of G_{primal} is the key behind realizability of a Q-floorplan.

Lemma 3.6. *Let G be a biconnected plane 3-graph (may have multi-edges) with k boundary critical paths, and Q be an orthogonally convex polygon. We have*

(1) $\min\{\mathrm{numSide}(D(C_O)) \mid D$ *is an orthogonally convex drawing of* $G\} = \max\{4, 2k - 4\}$, *and*

(2) if $\mathrm{numSide}(Q) \geq \max\{4, 2k-4\}$*, there is an orthogonally convex drawing* $D(G)$ *such that* $D(C_O)$ *is combinatorially equivalent to* Q*.*

Proof. First, we show that for any orthogonally convex drawing $D(G)$, $\mathrm{numSide}(D(C_O)) \geq \max\{4, 2k-4\}$. According to Lemma 3.1, for any boundary critical path P of two ends x and y, the number of convex corners located in the $P_{(x\frown y)}$ of $D(C_O)$ is at least one more than that of concave corners. Since all critical paths are edge-disjoint by Lemma 3.2, the total number of convex corners in $D(C_O)$ must be at least k. Therefore, the total number of corners of $D(C_O)$ is at least $k + (k-4) = 2k-4$ (since number of convex corners must be four more than that of concave corners in an orthogonal polygon), and so is the number of sides. Since each orthogonal polygon must contains at least 4 sides, we conclude that $\mathrm{numSide}(D(C_O)) \geq \max\{4, 2k-4\}$.

Second, we show that for any orthogonally convex polygon Q of $\mathrm{numSide}(Q) \geq \max\{4, 2k-4\}$, we can construct an orthogonally convex drawing $D(G)$ such that $D(C_O)$ is combinatorially equivalent to Q. Let the circular order (in counter-clockwise orientation) of corners of Q be $(v_0, v_1, \ldots, v_{s-1})$. Since the number of convex corners is exactly four more than the number of concave corners, there exist four indices $0 \leq i_0 < i_1 < i_2 < i_3 \leq s-1$ such that v_{i_t}, $v_{(i_t-1) \bmod s}$ are convex corners, and $s_t = (v_{i_t}, \ldots, v_{i_{(t+1) \bmod 4}-1})$ is a sequence of alternation of convex and concave corners, for $0 \leq t \leq 3$.

Now we are in a position to start the construction, which is based mainly on the algorithm described in Section 3.4. Since the number of bends inside Q is irrelevant, we can add to G a sufficient large amount of 2-vertices by subdividing edges not in C_O. Every 2-vertex in C_O is removed by contraction, and add to each boundary critical path P a 2-vertex by subdividing an edge in P.

If $k < 4$, $4-k$ 2-vertices are added to C_O at arbitrary positions, as long as we do not put all the k 2-vertices in $P_{(x\frown y)}$ for any P in S_G whose two ends are x and y. This is easy since all the paths in S_G are edge-disjoint.

The next step is to verify that the current graph admits a no-bend orthogonally convex drawing by examining the conditions in Theorems 3.2 and 3.5. Conditions in

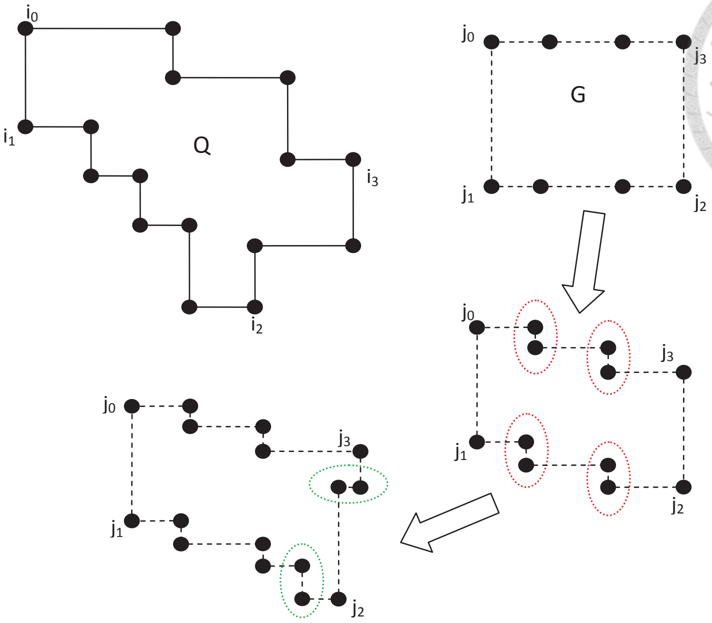


Figure 3.15: Illustration of the proof of Lemma 3.6.

Theorem 3.2 are automatically satisfied since an arbitrary amount of 2-vertices for edges not in C_O can be added. Condition 1 in Theorem 3.5 is also met. Now assume there exists a P in S_G contradicting condition 2 in Theorem 3.5, that is, $C_O \setminus P$ contains no 2-vertices. Since all the 2-vertices are located in P, every boundary critical path must edge-intersect with P, and hence contain P as a subpath. Due to the fact that critical paths are mutually edge-disjoint, G contains at most 1 boundary critical path. According to our strategy of adding additional 2-vertices when $k < 4$, P cannot contain all the 2-vertices in C_O, which is a contradiction to the assumption that $C_O \setminus P$ contains no 2-vertices. Hence condition 2 in Theorem 3.5 must be met, too.

Next, we construct a desired drawing based on the drawing algorithm described in the proof of Theorem 3.3.

Similarly, we let $(u_0, u_1, \ldots, u_{r-1})$ be the circular order(in counter-clockwise orientation) of 2-vertices in C_O. Four indices $0 \leq j_0 < j_1 < j_2 < j_3 \leq r-1$ are chosen such that $(j_{(t+1) \mod 4} - 1 - j_t) \mod r$ is less than or equal to the number of concave corners in s_t, for $0 \leq t \leq 3$. Let d_t denote the difference of the number of concave corners in s_t and $(j_{(t+1) \mod 4} - 1 - j_t) \mod r$.

Consider the three cases and their sub-cases in the construction detailed in the proof of Theorem 3.3. Let $u_{j_0}, u_{j_1}, u_{j_2}, u_{j_3}$ be the four designated vertices of G. Since we have a sufficient amount of 2-vertices for every non-boundary path, when case 1 occurs for the outer face being F_1, we can always forbid sub-case 3, and similarly sub-case 3 of case 3 and sub-case 1 of case 2 can be excluded.

Whenever case 2 occurs for the outer face being F_1, we always choose sub-case 2.2 (it is easy to see that sub-case 2.2 is always available if we never choose sub-case 2.1 whenever F_1 is the outer face). If the algorithm is executed in this way, it is easy to see that when the algorithm terminates, all the 2-vertices in C_O become convex corners.

Due to the preference of sub-case 2.2 and the orthogonally convexity of $D(C_O)$, the portion of $D(C_O)$ between u_{j_t} and $u_{j_{(t+1) \bmod 4}}$ contains exactly $(j_{(t+1) \bmod 4} - 1 - j_t) \bmod r$ pairs of convex corners and concave corners, for $0 \leq t \leq 3$. Therefore, our result follows after adding d_t pairs of convex and concave corners to the portion of $D(C_O)$ between u_{j_t} and $u_{j_{(t+1) \bmod 4}}$, for each $0 \leq t \leq 3$.

See Fig. 3.15 for a graphical illustration for the proof. For simplicity, only the boundary contour of G is drawn, omitting everything other than the r 2-vertices described in the proof. The corners circled by the dotted ellipses are produced by the extension of sub-case 2.2 of case 2; the corners circled by the dash-dotted ellipses are added in the last, after the execution of the no-bend orthogonally convex drawing algorithm. □

The concept of critical paths turns out to be pretty clean in the dual setting. We use T_G to denote the block-cutvertex tree of G. As we shall see later in Lemma 3.7, leaves in $T_{G_{dual}}$ can be put into one-to-one correspondence with critical paths in G_{primal}.

Let $\{v, u\}$ be an edge in $E(T_{G_{dual}})$ such that v is a cut-vertex. Now u must be a block. Let $V_{v,u}$ be the vertex set of the component in $G_{dual} \setminus \{v\}$ that contains some vertices in block u, and $F_{v,u}$ denote the corresponding face set in G_{primal}. Since G_{dual} is internally triangulated, the edges in $E(G_{dual})$ that link v to vertices in $V_{v,u}$ must be located consecutively in the circular list of edges incident to v that describes the combinatorial embedding of G_{dual}. We denote such an edge set as $E_{v,u}$. According to the definition of duality of plane graphs and the algorithm for constructing G_{primal} from G_{dual}, these edges form a path in G_{primal}. We write $C_{v,u}$ to denote the cycle that is the boundary of the union of faces in $F_{v,u}$. For instance, in Fig. 3.14 the set $E_{v,u}$ is $\{e_1, e_2, e_3\}$, which forms the non-boundary contour path with respect to $C_{v,u}$=$(e_1, e_2, e_3, e_4, e_5)$.

Lemma 3.7. *$\{$Boundary contour path of $C_{v,u} \mid u$ is a leaf of $T_{G_{dual}}, \{v, u\} \in E(T_{G_{dual}})\}$ is the set of boundary critical paths in G_{primal}.*

Proof. We first prove that a 2-legged cycle is proper if and only if it is $C_{v,u}$ for some edge $\{v, u\}$ in $E(T_{G_{dual}})$ such that v is a cut-vertex.

Following the respective definitions, it is easy to verify that $P_{v,u}$ is the intersection of $C_{v,u}$ and the contour of F_v (i.e. the face in G_{primal} corresponding to the vertex v in G_{dual}), and F_v is the only inner face that borders $C_{v,u}$ in its outer region. Therefore, $C_{v,u}$ is a boundary 2-legged cycle, with $P_{v,u}$ being its non-boundary contour path, and $F_{C_{v,u},P_{v,u}}$ being F_v. Since $G_{dual} \setminus \{v\}$ has more than 1 component, there must be some face not in $F_{v,u}$ that also borders F_v, and hence the contour of F_v intersects C_O of G_{primal} in more than one path. Therefore, $C_{v,u}$ is proper.

Conversely, let C be a proper 2-legged cycle in G_{primal}, and P being its non-boundary contour path. Due to the properness of C, the boundary of $F_{C,P}$ intersects C_O of G_{primal} in more than one path. Let v be the vertex in G_{dual} to which $F_{C,P}$ corresponds, and U be the subset of $V(G_{dual})$ that corresponds to the faces in $G(C)$. It is easy to see that v is a cut-vertex, and U is a component in $G_{dual} \setminus \{v\}$, and hence $C = C_{v,u}$ for some block u (the block neighboring v that belongs to the component).

We define a partial order $\prec$ on proper 2-legged cycles in G_{primal} such that $C_1 \prec C_2$ if and only if $G(C_1) \subseteq G(C_2)$. From the definition of critical paths and the fact that G_{primal} does not contain any non-boundary 2-legged cycle, a path P is a critical path if and only if it is a boundary contour path of a proper 2-legged cycle C such that C is minimal with respect to $\prec$. To conclude the proof, it suffices to show that $\{C_{v,u} \mid u$ is a leaf of $T_{G_{dual}}, \{v, u\} \in E(T_{G_{dual}})\}$ contains the minimal elements.

According to basic properties of block-cutvertex trees, let $\{v_1, u_1\}$, $\{v_2, u_2\}$ be two edges in $E(T_{G_{dual}})$ such that v_1 and v_2 are cut-vertices, is easy to see that $F_{v_1,u_1} \subset F_{v_2,u_2}$ if and only if there is a path $(v_1, u_1, \ldots, v_2, u_2)$ in $T_{G_{dual}}$. Since $G(C_{v,u})$ is exactly the union of faces in $F_{v,u}$, we conclude that $\{C_{v,u} \mid u \text{ is a leaf of } T_{G_{dual}}, \{v, u\} \in E(T_{G_{dual}})\}$ contains the minimal elements. □

In Fig. 3.14, the boundary contour paths of $C_{v,u}$ and $C_{x,y}$ are the paths drawn in dashed and dotted lines, respectively. These two paths are the boundary critical paths of G_{primal}. Following Lemmas 3.5, 3.6, 3.7 and Theorem 3.4, we have

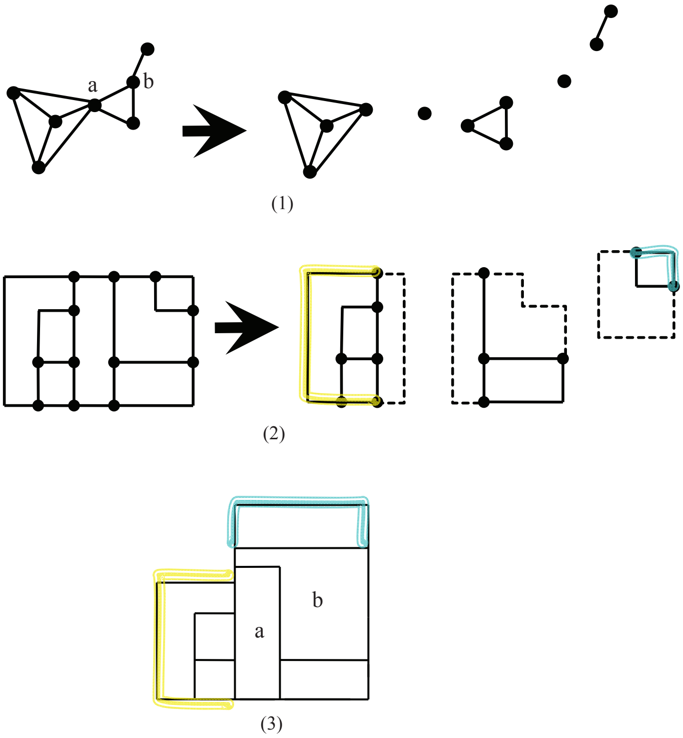


Figure 3.16: Key concepts in Q-floorplanning.

Theorem 3.8. *For any internally triangulated graph G_{dual} and orthogonally convex polygon Q, let k be the number of leaves in the block-cutvertex tree of G_{dual}. G_{dual} admits an orthogonally convex Q-floorplan iff* $\mathrm{numSide}(Q) \geq \max\{4, 2k - 4\}$. *The floorplan can be constructed in linear time.*

Fig. 3.16 summarizes the key concepts presented above. Fig. 3.16(1) is a decomposition of G_{dual} based on the block-cutvertex tree. Fig. 3.16(2) shows the respective decomposition in G_{promal}, in which the two boundary critical paths corresponding to the two cut vertices are clearly marked. The final Q-floorplan is displayed in Fig. 3.16(3), where Q is an L-shape polygon.

Chapter 4

Rectilinear Duals without T-shape

Most algorithms in recent years leading to low polygonal complexity rectilinear duals require the use of ⊤-shape polygons or their extensions [3, 4, 8].

In particular, the following theorem was proven:

Theorem 4.1 ([4]). *Every maximal plane graph admits an area-universal rectilinear dual using only non-rotated ⊤-shape polygons.*

See Fig. 2.3 for an example of such rectilinear dual. Please refer to Section 2.3 for the definition of non-rotated ⊤-shape polygon.

Theorem 4.1 is tight in the sense that there exists a maximal plane graph such that all its rectilinear duals have polygonal complexity at least 8, and we know that ⊤-shape has 8 sides.

This gives rise to an intriguing question whether other 8-sided polygons than ⊤-shape (such as Z-shape) and their degenerated cases are sufficient in constructing rectilinear duals of any maximal plane graphs. In simple words, is ⊤-shape really the most powerful 8-sided polygon?

As it turns out, in this chapter, we are able to answer the question by showing that the polygonal complexity of ⊤-free rectilinear dual of maximal plane graph is 12 by proving the following:

1. There exists a maximal plane graph such that all its ⊤-free rectilinear duals have polygonal complexity at least 12.

2. Every maximal plane graph admits a rectilinear duals using only monotone staircase polygons of at most 12 sides.

As a result, indeed $\top$-shape is the most powerful 8-sided polygons. Without its presence, the required polygonal complexity increases by $12 - 8 = 4$.

Again see Section 2.3 for the definition of $\top$-free rectilinear duals and monotone staircase polygons. For the sake of convenience, we restate the definition of Q-free rectilinear duals here:

Definition 4.1. *Let R be a rectilinear dual, we call it Q-*free *iff for each polygon of shape P used in R, we have $Q \not\preceq P$.*

Recall that Theorem 2.4 gives a very clean characterization for area-universal rectangular duals. In contrast, we still know very little for the following problems:

1. Characterize the rectilinear duals that are area-universal.
2. Characterize the graphs that admit an area-universal rectilinear dual.

In particular, we still do not know whether every maximal plane graph admit a $\top$-free rectilinear dual. In other words, is $\top$-shape really essential in constructing an area-universal rectilinear dual for maximal plane graph?

In an attempt to (partially) solve the above problems, in this chapter we also prove the following results:

1. There exists a maximal plane graph that does not admit any monotone staircase area-universal rectilinear dual.
2. For Hamiltonian maximal plane graphs, we can easily construct an rectilinear dual using only non-rotated Z-shape polygons.

4.1 Related Works

Yeap and Sarrafzadeh [45] showed that every maximal plane graph admits a rectilinear dual using polygons of at most eight sides, which matches the lower bound. Liao *et al.* [31] later improved the above result by showing that it suffices to use only I-shape, L-shape, and ⊤-shape polygons, whereas in [45], Z-shape polygons are also required. See Fig. 2.1 for these four types of polygons.

In fact, I-shape and L-shape polygons are degenerated cases of ⊤-shape polygons, as an I (resp., L) can be obtained from a ⊤ by chopping off two ends (resp., one end) of the horizontal segment of the ⊤.

In [19], rectangular duals under some constraints about relative positions between objects were studied. In a broader sense, this work fits into the line of research on constrained rectilinear duals, where the usable shapes or their relations are constrained. Our characterization for orthogonally convex rectilinear duals in Section 3.7 is also related to the line of research in this chapter.

4.2 Lower Bound of Polygonal Complexity

In this section, we prove that ⊤-free rectilinear duals of maximal plane graphs have polygonal complexity of at least 12, which is higher than the 8 in the general case when ⊤-shape polygons are allowed.

We define the plane graph H_0 in Fig. 4.1, which is a key structure behind the higher polygonal complexity of ⊤-free rectilinear duals. The following lemma indicates that a presence of the structure H_0 in a graph inevitably requires two concave corners in any of its ⊤-free rectilinear duals.

Lemma 4.1. *Let H be a subgraph of a maximal plane graph G such that H is isomorphic to H_0 (with x, y and z being the three vertices on the outer cycle) and $G(H) = H$. For any ⊤-free rectilinear dual of G, there must be at least two concave corners in polygons associated with x, y, and z which are located along the border between the region $\{x, y, z\}$ and the region $\{u, v, w, c\}$.*

Proof. Clearly there must be at least one concave corner. If there is only one such concave corner, without loss of generality, we let x be the one containing the concave corner. Now, the boundary of the region of $\{u, v, w, c\}$ must be a rectangle, as illustrated in the upper drawings of Fig. 4.1. Since the polygon c in Fig. 4.1 touches x and since x

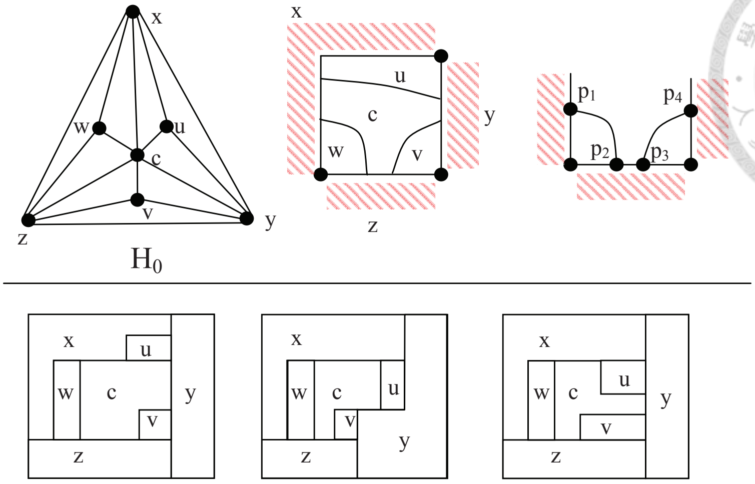


Figure 4.1: Definition of H_0 and illustration of the proof of Lemma 4.1.

borders the rectangular region $\{u, v, w, c\}$ along its west and north sides, by symmetry, we can assume that c touches the west boundary without loss of generality. We identify four points p_1, p_2, p_3 and p_4 on the boundary of c as illustrated in Fig. 4.1. By setting $a = p_2, b = p_3$, S_1 = the sequence of points between p_1 and p_2, $S_2 = ()$, and S_3 = the sequence of points between p_3 and p_4, it is easy to see that if all the polygons are drawn rectilinearly, such assignments must satisfy the statement of Fact 2.4, and hence we must have $\top \preceq c$, which is a contradiction. □

With the help of the above lemma, we are in a position to prove the following main theorem of this section by a simple counting method.

Theorem 4.2. *There exists a maximal plane graph G such that every* $\top$*-free rectilinear dual of G must have polygonal complexity of at least 12.*

Proof. Let $G_0 = (V, E)$ be an n-node maximal plane graph with $n > 10$. We replace each inner face of G_0 with a copy of H_0 by adding new vertices and edges. Let the resulting graph be G_1, and we let R be any $\top$-free rectilinear dual of G_1. According to Lemma 4.1, since the number of inner faces in G_0 is $2n - 5$, the number of concave corners in polygons associated with vertices of V must be at least $2 \times (2n - 5) = 4n - 10 > 3n = 3|V|$. Therefore, there must be a polygon in R containing at least 4 concave corners. By Fact 2.1, such a polygon has at least $4 + (4 + 4) = 12$ corners. □

In the lower part of Fig. 4.1, we give three examples of rectilinear duals of H_0. The left two drawings are both $\top$-free and contain two concave corners in polygons associated with x, y, and z; the rightmost one contains only one concave corner but the polygon associated with c is a $\top$-shape. These two $\top$-free drawings serve as prototypical concepts of the algorithm presented in the next section.

4.3 Construction of 12-sided $\top$-free Rectilinear Duals

In this section, we present an algorithm to construct 12-sided $\top$-free rectilinear duals for maximal plane graphs. Our construction uses only monotone staircases, which are orthogonally convex ones that cannot degenerate to $\top$-shape (Fact 2.3).

Our algorithm is an inductive approach based on the separation-trees described in Section 2.4. All we need to do is to devise a method inserting the rectangular dual of $G_\triangle \setminus \triangle$ to the current rectilinear dual during the course of the construction meeting the following:

1. Every polygon preserves the shape of a monotone staircase;

2. The total number of concave corners on the boundary of each polygon is at most 4.

4.3.1 Un-contracting a Separating Triangle

When we un-contract a triangle $\triangle = \{x, y, z\}$, a rectangular space is allocated to accommodate a rectangular dual of $G_\triangle \setminus \triangle$, which in turn imposes (at least) a concave corner to one of $\{x, y, z\}$. Without loss of generality, we assume that such a concave corner is associated with the polygon x.

As observed in Section 4.2, one concave corner in $\{x, y, z\}$ may not be enough in some cases. In order to enforce the staircase constraint, we further annotate one of its four sides as *"allowed to add a concave corner"*, which is indicated by an arrow in our illustrations. See Fig. 4.2.

Since polygon x is a monotone staircase, x borders either the entire west and north boundary or the entire east and south boundary of the rectangular space. Therefore, there are eight cases in total since the arrow can point to any one of the four sides of the rectangular boundary. It is sufficient to consider the following two cases (see the left illustration of Fig. 4.2):

1. Polygon x borders the west and the north sides, and the arrow points to the north side;

2. Polygon x borders the west and the north sides, and the arrow points to the east side.

The remaining cases are symmetric to one of the above (by flipping the entire drawing around the north-west to south-east line or the north-east to south-west line).

We also fix polygon y to be the one that borders the east side of the rectangular space.

A key in our un-contracting process is to identify three special vertices associated with each separating triangle. Consider Fig. 4.2. Let u, v, and w be the three vertices in $G_\triangle \setminus \{x, y, z\}$ such that u, v, and w are adjacent to $\{x, y\}$, $\{y, z\}$, $\{x, z\}$, respectively. It is easy to see that u, v, and w are uniquely determined; otherwise, there must be a separating triangle in $G_\triangle$, which contradicts its definition.

Unless $|V(G_\triangle)| = 4$ (in this case, $u = v = w$), u, v and w must be different from each other (otherwise, a separating triangle can be found in $G_\triangle$).

When the rectangular dual of $G_\triangle \setminus \{x, y, z\}$ is constructed to fill the rectangular space, we further assume the rectangular dual to have polygon w adjacent to the entire west side. Such a drawing must exist since there is no separating triangle inside the quadrangle $\{x, y, z, w\}$ (See Fig. 4.2).

If we consider the children of the node associated with $\triangle = \{x, y, z\}$ in the separation tree, there are two types of separating triangles:

1. Separating triangles that are either $\{x, y, u\}$, $\{y, z, v\}$, or $\{x, z, w\}$. (Note that some of these three triangles may not be separating triangles.)

2. Separating triangles in the subgraph surrounded by vertices $\{x, u, y, v, z, w\}$. See Fig. 4.2.

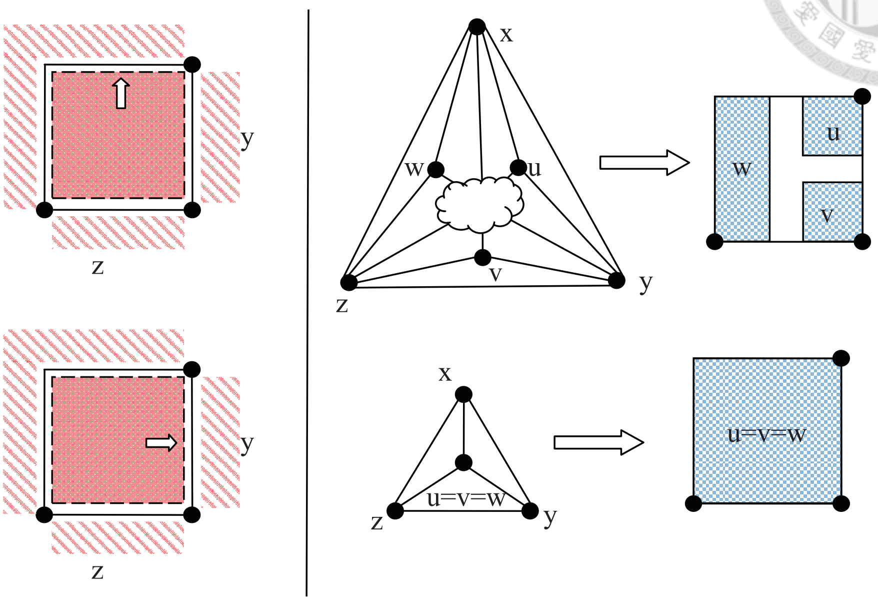


Figure 4.2: Location of u, v, w in the rectangular space for separating triangle $\{x, y, z\}$.

Now, we are in the position to present how we make a rectangular space for each separating triangle.

Type 1 separating triangles. We first consider the Type 1 separating triangles, i.e., the separating triangles that are either $\{x, y, u\}$, $\{y, z, v\}$, or $\{x, z, w\}$.

Depending on whether the arrow points to the north side or the east side of the rectangular space for $G_{\{x,y,z\}}$, the solutions are depicted in the upper and lower parts of Fig. 4.3, respectively.

The rectangular regions surrounded by dashed boundaries in the rightmost figures of Fig. 4.3 are the rectangular spaces for $\{x, y, u\}$, $\{y, z, v\}$, and $\{x, z, w\}$ when we un-contract them in later iterations, with some of which possibly be void if they are not separating triangles.

Special attention should be given to the directions of the arrows in those regions.

The checkered regions represent those allocated for the special vertices u, v, and w.

The white spaces in Fig. 4.3 are parts of rectangular duals of the subgraph surrounded by vertices $\{x, u, y, v, z, w\}$, and the white dots indicate points at which there may be separating triangles when Type 2 separating triangles are included.

Type 2 separating triangles. Recall from Fig. 2.6 that when a separating triangle is un-contracted, a rectangular region is inserted at the juncture of the three polygons associated with the three vertices of the separating triangle. Depending on the orientation of the three polygons, there are four cases as illustrated in the upper part of Fig. 4.4.

Attention should be given to the arrows in those regions, which indicate the sides where additional concave corners are possible during the course of future un-contraction.

Although Fig. 4.4 shows the general rule for allocating spaces to accommodate Type 2 separating triangles, there is an exception. Consider the white point incident to both u and x in the upper illustration of Fig. 4.3 (see also the lower illustration of Fig. 4.4). One can see from Fig. 4.3 that the polygon u stretches upwards in order to make room for the separating triangle $\{x, u, y\}$ (i.e., the top-most red region in the upper illustration of Fig. 4.3). As a result, we apply rule (3) in Fig. 4.4, as opposed to rule (4), as illustrated in the lower illustration of Fig. 4.4. Such special care prevents the creation of an additional (undesired) concave corner to $\{x, y, z\}$. A similar situation occurs at the juncture between u and y in the lower illustration of Fig. 4.3.

Bounding Polygonal Complexity. It is clear from the above that all operations preserve monotone staircase shape. What remains to do is to count the number of concave corners in each polygon s.

(Case: $s \notin \{u, v, w\}$) In Fig. 4.4, when we make a rectangular space at point p, if p is a non-corner of polygon s, no concave corner is imposed on s. Therefore, for any polygon s not belonging to $\{u, v, w\}$, the number of concave corners imposed on s is at most four since a rectangle has four corners.

(Case: $s = u = v = w$) If $s = u = v = w$, it is easy to see that we also impose at most four concave corners on s. In Fig. 4.3, we make one concave corner and three arrows (which may potentially become concave corners) to s.

(Case: $s \in \{u, v, w\}$, u, v, w are distinct vertices) For this case, the results are summarized in the following, which can be easily observed in Fig. 4.3:

u: 0 concave corner, 1 arrow, and 3 white dots; the total amount is 4.

v: 1 concave corner, 1 arrow, and 3 white dots; the total amount is 5.

w: 0 concave corner, 1 arrow, and 2 white dots; the total amount is 3.

So far, our algorithm can compute a monotone staircase rectilinear dual that uses polygons of at most 14 (=2 × 5 + 4) sides, as the number of sides = 2 × (the number of concave corners) + 4. To lower the polygonal complexity from 14 to 12, our approach is to transfer one concave corner from v to w. Our solution is presented in the following subsection.

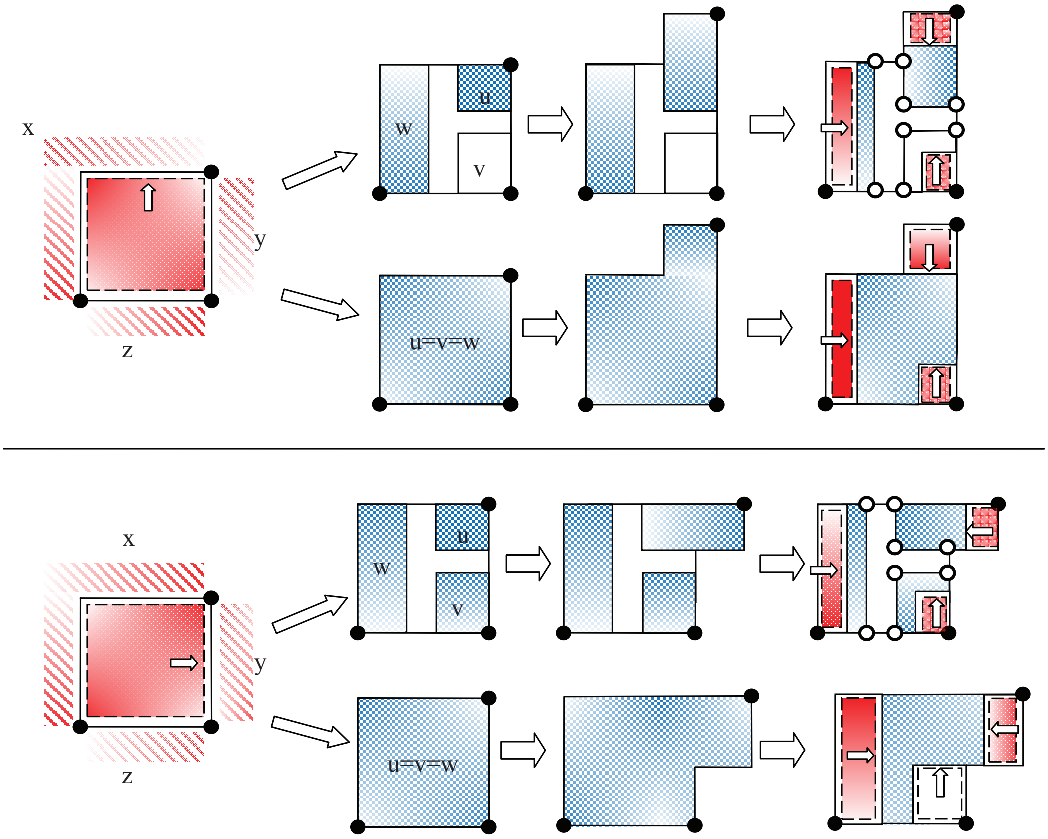


Figure 4.3: Illustration of un-contracting Type 1 triangles.

4.3.2 Transferring Concave Corners

Let $S = V(G_\triangle) \setminus \{x, y, z\}$, and given a rectangular dual R_0 of $G_\triangle \setminus \{x, y, z\}$ meeting the conditions described in Section 4.3.1:

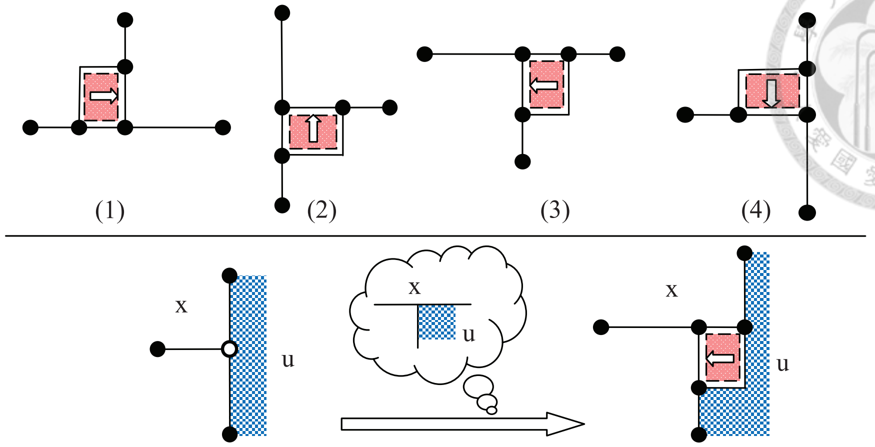


Figure 4.4: Illustration of un-contracting Type 2 triangles.

1. The west and north boundaries of R_0 are adjacent to x, the east and south boundaries of R_0 are adjacent to y and z, respectively.

2. w (the unique vertex adjacent to x and z) borders the entire west boundary of R_0.

We define a relation "$\leftarrow$" on S:

Definition 4.2. *Given S and R_0, "$\leftarrow$" is a relation on S such that: $s \leftarrow s'$ iff (1) the west side of s is more west than (i.e., on the left-hand side of) the west side of s', and (2) there is a point p in R_0 such that p is a 180° corner in s and a 90° corner in s'.*

Regarding the separating triangle $\{x, y, z\}$ discussed in Section 4.3, the following lemmas are easy to observe. Lemma 4.2 directly follows from the fact that w is the unique vertex adjacent to x and z.

Lemma 4.2. *w is the only vertex that touches both the north boundary and the south boundary of R_0.*

Lemma 4.3. *There exists a path $v = s_1, s_2, \ldots, s_k = w$ in S such that $s_{i+1} \leftarrow s_i$ for $1 \leq i \leq k-1$.*

Proof. If such a path does not exist, there must exist a vertex $t \neq w$ such that $s \not\leftarrow t$ for all $s \in S$. It means that the north-west corner of t touches the north boundary of

R_0 and the south-west corner of t touches the south boundary of R_0, which contradicts Lemma 4.2. □

Let $s_1, s_2, \ldots, s_k$ be the path that satisfies Lemma 4.3. Our concave corner transfer algorithm works as follows: For $i = 1$ to k, if there is a separating triangle $\triangle' = \{s_i, s_{i+1}, t\}$ for some $t \in S$, we re-build the rectangular space for $\triangle'$ as depicted in Fig. 4.5, which is capable of "shifting" 1 concave corner (or arrow) from s_i to s_{i+1}.

The procedure terminates if there is no such triangle; in this case, the number of concave corners in s_i must be smaller than four before the execution of this algorithm. Therefore, all polygons must have at most four concave corners after the concave corner transfer algorithm ends.

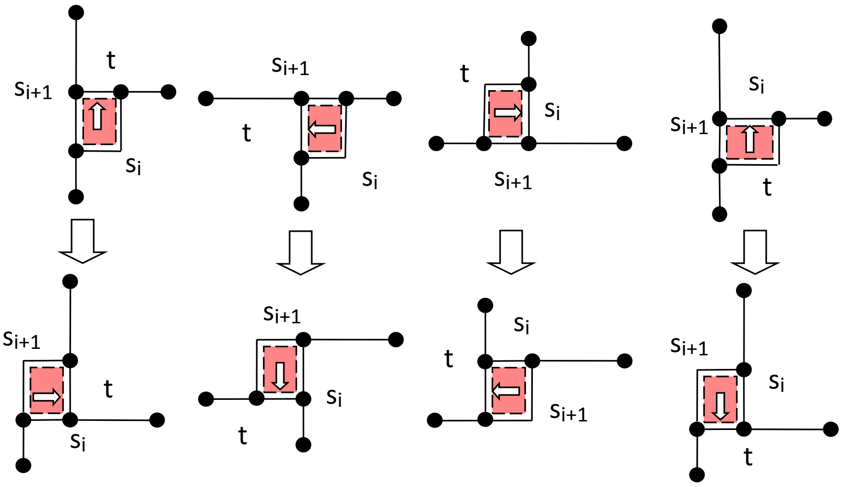


Figure 4.5: Illustration of transferring concave corners.

It is easy to see that the algorithm presented in this section for constructing monotone staircase rectilinear duals can be implemented in linear time. As a result, we conclude the following main theorem of the chapter:

Theorem 4.3. *⊤-free rectilinear duals for maximal plane graphs have polygonal complexity of at most 12. Moreover, there is a linear time algorithm that constructs a monotone staircase rectilinear dual for any maximal plane graph.*

4.4 Area-universal Drawings

In view of our earlier discussion, it is natural to investigate how or whether area-universal rectilinear duals can be constructed in the absence of $\top$-shape polygons.

In this section we show that restricting usable shapes to monotone staircases is insufficient to construct area-universal rectilinear duals for maximal plane graphs in general.

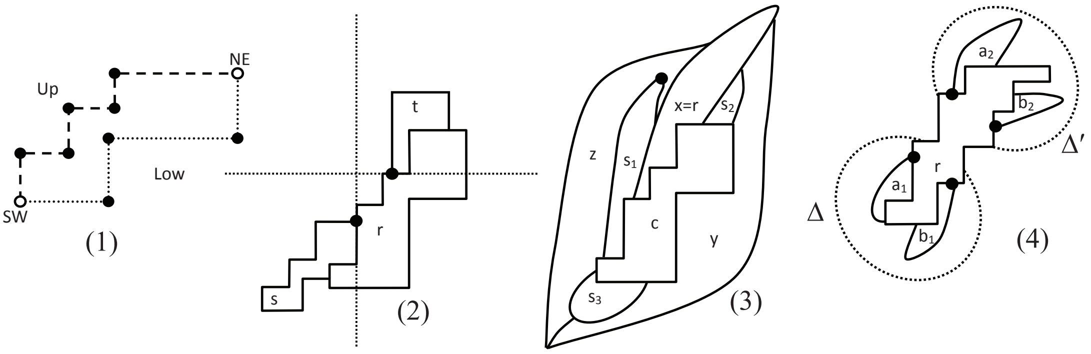


Figure 4.6: Illustration of concepts in Section 4.4.

To proceed further, we require some definitions. We denote the most south-western point and the most north-eastern point of a monotone staircase as SW and NE, respectively. Separated by these 2 points, the boundary of the polygon is divided into the upper part and lower part naturally, which we denote as Up and Low, respectively. See Fig. 4.6(1) for an illustration. We define relations $\xrightarrow[X,Y]{}$ for $X \in \{SW, NE\}$, $Y \in \{Up, Low\}$, and give a condition for a rectilinear duals to be not area-universal.

Definition 4.3. *For $X \in \{SW, NE\}$, $Y \in \{Up, Low\}$, and any two polygons s, t of monotone staircase shape, $s \xrightarrow[X,Y]{} t$ iff $X(s)$ is located in $Y(t)$, where $X(s)$ and $Y(t)$ denote the X point of polygon s and the Y boundary of polygon t, respectively.*

Lemma 4.4. *For any monotone staircase rectilinear dual R, if there exist three polygons r, s, t and $Y \in \{Up, Low\}$ such that $s \xrightarrow[NE,Y]{} r$, $t \xrightarrow[SW,Y]{} r$, and $NE(s)$ is more south-west than $SW(t)$, then R is not area-universal.*

Proof. For any drawing of R, consider a Cartesian system formed by setting x-axis to be the east-west line passing through $NE(s)$ and setting y-axis to be the north-south line passing through $SW(t)$, then s, t must be confined in quadrant I, III, respectively (see Fig. 4.6(2) for an illustration). It is easy to see that any area-assignment where both the area of s and the area of t surpass 25% of the area of the rectilinear dual is not realizable. □

Similar to what we have done in Section 4.2, let G be a maximal plane graph and H be a sub-graph of G such that H is isomorphic to H_0 and $G(H) = H$. We consider a monotone staircase rectilinear dual R of G. It is easy to observe that the border between 2 monotone staircase polygons cannot intersect with both Up and Low of one of them (otherwise, the other cannot be monotone staircase). Therefore, by the pigeonhole principle, in H we have that two of $\{u, v, w\}$ border c in one of $\{Up(s), Low(s)\}$. We denote these two vertices as s_1, s_2, and the border between s_1, c is more south-west than that of s_2, c. Let $r \in \{x, y, z\}$ be the unique vertex adjacent to both s_1 and s_2, the next lemma reveals a relationship between s_1, s_2, c, and r.

Lemma 4.5. *If we require R to be area-universal, exactly one of the following must be satisfied for H:*

1. *$SW(r)$ is located in $Up(c)$, $s_1 \xrightarrow[NE,Up]{} r$, and $s_2 \xrightarrow[NE,Low]{} r$.*
2. *$NE(r)$ is located in $Up(c)$, $s_1 \xrightarrow[SW,Low]{} r$, and $s_2 \xrightarrow[SW,Up]{} r$.*
3. *$SW(r)$ is located in $Low(c)$, $s_1 \xrightarrow[NE,Low]{} r$, and $s_2 \xrightarrow[NE,Up]{} r$.*
4. *$NE(r)$ is located in $Low(c)$, $s_1 \xrightarrow[SW,Up]{} r$, and $s_2 \xrightarrow[SW,Low]{} r$.*

Proof. If both $SW(r)$ and $NE(r)$ are not located in one of $Up(c)$ and $Low(c)$, it is easy to see that setting $(s, t, r) = (s_1, s_2, c)$ satisfies the condition in Lemma 4.4, and hence R is not area-universal. Therefore, the first part (which is concerned with the location of $SW(r)$ or $NE(r)$) of one of the four statements is satisfied. It remains to prove that for each statement, the first part implies the remaining parts; By symmetry, all four statements are inherently the same, therefore it suffices to consider statement (1) only, that is, showing that $SW(r)$ is located in $Up(c)$ implies $s_1 \xrightarrow[NE,Up]{} r$ and $s_2 \xrightarrow[NE,Low]{} r$.

Let s_3 be the only vertex in $\{u, v, w\} \setminus \{s_1, s_2\}$, and we, without loss of generality, let $x = r$. Since $SW(r)$ is located in $Up(c)$, x cannot touch the most south-western point of $c \cup s_3$, and we denote this point as p. It forces both y and z touch p since if it is touched by only one of $\{y, z\}$, that one cannot remain monotone staircase. It is not hard to see in Fig. 4.6(3) that if $s_1 \xrightarrow[NE,Up]{} r$ is not satisfied, then $NE(s_1)$ is a locally most north-eastern point in $Low(z)$, which makes z not monotone staircase. Similar contradiction can be made for y when $s_2 \xrightarrow[NE,Low]{} r$ is not satisfied. Hence the lemma follows. □

Theorem 4.4. *There exists a maximal plane graph G such that every monotone staircase rectilinear dual of G is not area-universal.*

Proof. The idea hehind this proof is similar to that of Theorem 4.2. Let G_0, G_1, S, and n be the same as what they are in the proof of Theorem 4.2. Let R be a monotone staircase rectilinear dual of G_1. According to Lemma 4.5, for each separating triangle $\triangle$ in G_1 such that $G(\triangle)$ is isomorphic to H_0, we define a function f that chooses a vertex $f(\triangle) = r \in \triangle$ such that there are two vertices $a, b \in V(G(\triangle)) \setminus \triangle$ satisfying either $a \xrightarrow[NE,Up]{} r$ and $b \xrightarrow[NE,Low]{} r$ or $a \xrightarrow[SW,Up]{} r$ and $b \xrightarrow[SW,Low]{} r$.

Since the number of $\triangle$ in G_1 such that $G(\triangle)$ is isomorphic to H_0 is $2n - 5 > n$, we can find two such triangles $\triangle$, and $\triangle'$ such that $f(\triangle) = f(\triangle')$. As illustrated in Fig. 4.6(4), the condition stated in Lemma 4.4 must be satisfied, and hence, the drawing of G_1 must not be area-universal. □

4.5 More about Staircase Polygons

Following the impossibility result in the previous section, we study monotone staircase polygons in a more restricted setting. It is easy to observe that Theorem 4.4 still holds even restricting to plane 3-trees since replacing a triangle with H_0 preserves the property of being a plane 3-tree. Beside plane 3-trees, Hamiltonian maximal plane graphs, which subsume maximal outer plane graphs and 4-connected plane graphs, are another important sub-class of maximal plane graphs.

Contrasting the above results, the following theorem can be shown easily by a slight modification to the $\sqcup$-shape cartogram drawing algorithm described in [4] (by reversing the construction order of the right part, we can get a non-rotated Z-shape drawing).

Theorem 4.5. *All Hamiltonian maximal plane graphs admit 8-sided area-universal monotone staircase rectilinear duals.*

Combining the result in [4], we conclude that all Hamiltonian maximal plane graphs admit rectilinear duals meeting the following requirements:

- The drawing consists of non-rotated $\top$-shape (non-rotated $\sqcup$-shape, or non-rotated Z-shape) only.
- The drawing is area-universal.

The above gives rise to the problem of whether every Hamiltonian maximal plane graph admits a non-rotated W-shape rectilinear dual (as W-shape is the only 8-sided rectilinear polygon other than $\top$-, $\sqcup$-, Z- shapes). See Fig. 4.7(1) for a picture of a W-shape. We denote a monotone staircase whose Low contains exactly one convex corner and no concave corner as a *monotone strict staircase*. It is easy to see that a non-rotated W-shape is exactly the 8-sided monotone strict staircase. We prove the following result:

Theorem 4.6. *For any k, there is a Hamiltonian maximal plane graph that does not admit any k-sided monotone strict staircase rectilinear dual.*

Proof. Let G be a Hamiltonian maximal plane graph, and let $\{x, y, z\}$ be its outer cycle. Given any monotone strict staircase rectilinear dual R for G, without loss of generality, we let the polygon associated with z be the one touching the south-east corner of the rectangular boundary of R. The region surrounded by polygons x, y, z must be an upside-down monotone strict staircase, as depicted in Fig. 4.7(2).

Let $\triangle = \{a, b, z\}$ be any separating triangle such that $\{a, b\} \cap \{x, y\} = \emptyset$. Since polygons of a, b are both monotone strict staircases, there must be at least one concave corner of polygon z residing in the border between polygon z and the region interior

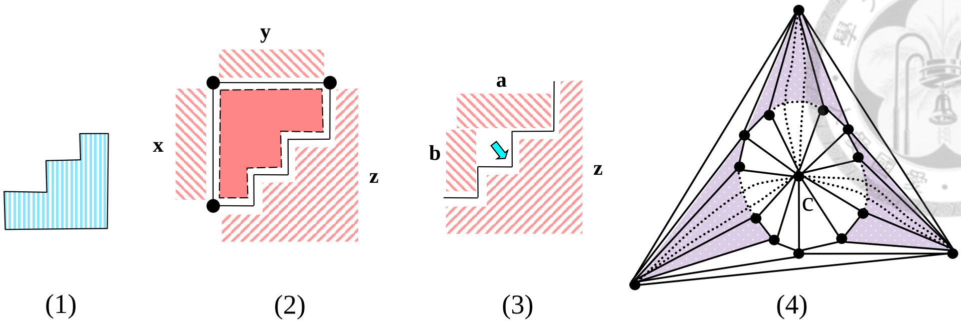


Figure 4.7: Illustration of the proof of Theorem 4.6.

of $\triangle$ (otherwise, the region interior of $\triangle$ cannot be enclosed by polygons $\{a, b, z\}$). See Fig. 4.7(3). As a result, a necessary condition for R to be a k-sided monotone strict staircase rectilinear dual is that the number of interior-disjoint separating triangles $\triangle = \{a, b, z\}$ such that $\{a, b\} \cap \{x, y\} = \emptyset$ is at most $(k - 4)/2$ (recall that the number of convex corners of a rectilinear polygon is four plus the number of concave corners, and that the number of sides equal the number of corners). Next, we show that for any number t, there is always a Hamiltonian maximal plane graph G in which each boundary vertex is contained in at least t such triangles. Fig. 4.7(4) showcases such an example. In this scheme, we can have as many interior-disjoint triangles containing a boundary vertex as we want by adding more dotted-lines linking a boundary vertex to the center vertex c. Also, the graph remains to be Hamiltonian if we add a new vertex v inside each triangle $\triangle$ in the shaded region and add an edge between each vertex in $\triangle$ and v (making $\triangle$ a separating triangle). As a result, we conclude the proof. □

However, if unbounded polygonal complexity is allowed (i.e., $k \to \infty$), we have the following theorem:

Theorem 4.7. *Every maximal plane graph admits a monotone strict staircase rectilinear dual, having possibly unbounded polygonal complexity.*

Proof. For any maximal plane graph G, we construct a desired drawing using the framework of building rectilinear duals based on separation-trees described in Section 2.4. To meet the requirement of using only monotone strict staircases, it suffices to make sure that during the construction, each addition of a concave corner to a polygon s preserves the property that s is a monotone strict staircase.

We only add concave corners when a separating triangle $\triangle_i$ is un-contracted, since we need to make a rectangular space to plug-in the rectangular drawing of $G_{\triangle_i} \setminus \triangle_i$ to the current drawing (see Fig. 2.6 for an conceptual illustration). It is easy to see that if we make rectangular space according to the rule described in Fig. 4.8, the property that all polygons are monotone strict staircase can be preserved. □

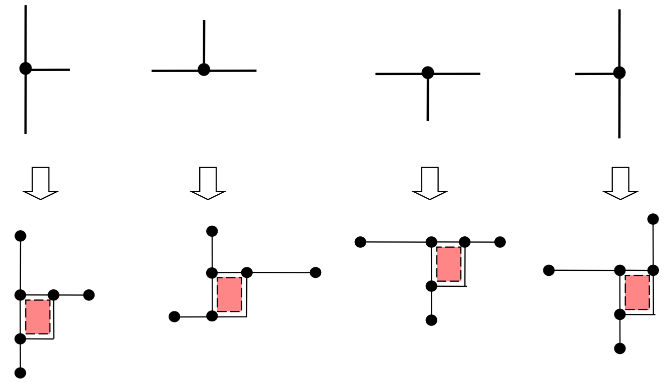


Figure 4.8: Illustration of the proof of Theorem 4.7.

We remark that for the special case of $k = 6$, a 6-sided monotone strict staircase is exactly non-rotated L-shape, and the maximal plane graphs admitting rectilinear duals using such shape have been characterized based on Schnyder labeling (Theorem 2.1.7 of [43]).

Chapter 5

Convex Polygonal Duals

Recall that in Chapter 1 we mentioned that there is a lack of results and tools to deal with contact representations of geometric objects that are not necessarily rectilinear.

The goal of the chapter is to extend the study of rectilinear dual to convex polygonal objects. In particular, new tools for contact representations using convex polygons are presented, and some links between the rectilinear setting and the convex polygonal setting are established.

Our contributions include the followings:

- A very general drawing style called *convex polygonal dual*, which subsumes well-studied drawing styles like proper touching triangle representation and rectangular dual, is proposed.
- We characterize graphs admitting *straight-line convex t-gon representations* and *straight-line t-gon representations*, which can be regarded as a primal version of convex polygonal duals. This greatly extends the main result of [2].
- Based on the above result, a characterization for a plane graph to admit a t-sided convex polygonal dual is presented.
- Using Courcelle's theorem, we derive some useful fixed-parameter tractability results for convex polygonal duals.
- To further demonstrate the usefulness of our approach, we give quick alternate proofs for the following existing results based on our techniques:
  - Each maximal plane graph admits a 6-sided convex polygonal dual [17].

  - Each triconnected cubic plane graph admits a proper touching triangle representation [28].

5.1 Related Works

There are a few works on contact representations for triangles. Please refer to Section 2.6. Also see the PhD thesis of Aerts [1].

For convex polygons beyond triangles, [17] showed an algorithm to a construct contact representations for any plane graphs using convex polygons of at most 6 sides. They also proved in [17] that the polygonal complexity of 6 meets the lower bound by constructing a series of planar graphs that cannot be represented by polygons of at most 5 sides. Much different to our setting in the thesis, they allow the presence of holes in their drawings.

It has been proved that all Hamiltonian maximal plane graphs admit contact representations using convex polygons of at most 5 sides (Corollary 2.2.5 of [43]). As the drawings they constructed are based on a modification to L-shape rectilinear duals, there is no hole in their drawings.

To our best knowledge, [20] is the only work on area-universal drawing regarding geometric objects that are not necessarily rectilinear.

From a primal point of view, the drawing style *convex drawing* is already well-studied. See Chapter 5 of [34].

5.2 Terminologies

Some definitions presented below can be seen as extensions or generalizations of the similar ones in [2].

Definition 5.1. *Given a biconnected plane graph $G = (V, E)$ such that all degree 2 vertices are in $V(F_O(G))$, a t-*flat angle assignment *(t-*FAA*, for short) is a mapping from a subset of $V \setminus \{v | v \in V(F_O(G))\}$ to inner faces of G such that:*

1. *Each vertex is assigned at most once;*

2. *Each inner face F is assigned at least $|V(F)| - t$ times;*

3. *for each mapping associating a vertex v to a face F, we have $v \in V(F)$.*

Intuitively speaking, the idea behind assigning v to a face F in a t-FAA is to capture the presence of a 180^o angle surrounding v in the face F in a drawing. Condition (2) is to ensure that each inner face is drawn as a convex polygon which has at most t convex corners.

Definition 5.2. *A* straight line t-gon representation *(t-SLR, for short) is a planar drawing such that:*

1. *each inner face is a polygon of at most t sides, and*

2. *the outer face is a convex polygon.*

A straight line convex t-gon representation *(t-convex-SLR, for short) is a t-SLR with an additional constraint that each inner face is convex.*

FAAs are also closely related to the so-called *contact systems of pseudo-segments* [15], each of which is a set of non-crossing Jordan arcs where any two of them intersect in at most one point, and each intersecting point is internal to at most one arc. A contact system is *stretchable* if there exists a homeomorphism transforming the contact system into a drawing where each arc is a straight line. Stretchable contact systems of pseudo-segments were characterized in [15] based on the notion of *extremal points*.

Definition 5.3. *A point p is an extremal point of a contact system S of pseudo-segments if the following three conditions are satisfied:*

1. *p is an endpoint of a pseudo-segment in S.*

2. *p is not interior to any pseudo-segment in S.*

3. *p is incident to the unbounded region of S.*

Theorem 5.1 ([15]). *A contact system S of pseudo-segments is stretchable iff each of its subsystems (i.e., subsets of pseudo-segments) S' of cardinality greater than 1 has at least 3 extremal points.*

It is not difficult to see that a t-FAA of a plane graph naturally defines a contact system of pseudo-segments in which each pseudo-segment is associated with a path $(v_1, \ldots, v_k)$ meeting the below conditions:

1. $k \geq 2$.
2. $\forall 1 < j < k, v_j$ is assigned to a face containing the two edges $\{v_{j-1}, v_j\}, \{v_j, v_{j+1}\}$.
3. v_1 (v_k) is either unassigned or assigned to a face not containing $\{v_1, v_2\}$ ($\{v_{k-1}, v_k\}$), respectively.

Such a pseudo-segment is said to be *induced* by an edge $\{v_j, v_{j+1}\}$, for any $1 \leq j \leq k-1$. Note that an edge induces exactly one pseudo-segment.

For ease of explanation, we write S_C to denote the set of pseudo-segments induced by the edges in the cycle C w.r.t. a given t-FAA.

It is clear that a graph admits a t-FAA corresponding to a stretchable contact system of pseudo-segments iff it admits a t-convex-SLR. With respect to a t-FAA, we call a corner of an inner face a *combinatorial convex corner* if it is not assigned to the face.

For a more detailed exposition, the reader is referred to [2].

5.3 Characterizing t-sided Convex Polygonal Duals

A *t-sided convex polygonal dual* is a contact representation of a plane graph defined as follows:

Definition 5.4. *A t-sided convex polygonal dual is a contact representation of a plane graph meeting the below conditions:*

1. *Each vertex corresponds to a convex polygon of at most t sides.*

2. *Adjacency in the graph corresponds to side-contact in the drawing.*

3. *All convex polygons together form a partition of a convex polygon.*

The goal in this section is to give a combinatorial characterization for plane graphs admitting such drawings.

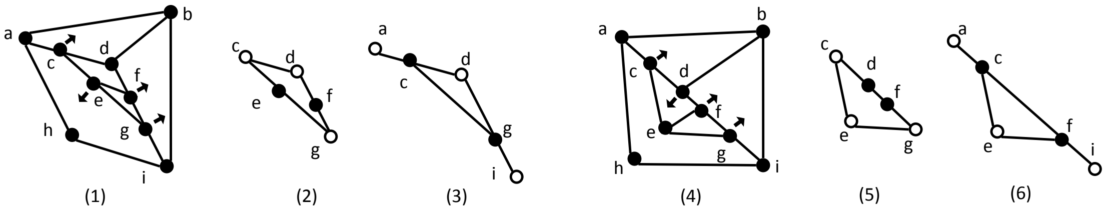


Figure 5.1: Illustration of concepts introduced in Section 5.3.

In what follows we first derive a characterization for a graph admitting a t-convex-SLR based on the notion of t-FAAs.

Definition 5.5. *Let C be a cycle in a biconnected plane graph G whose degree 2 vertices are all in $V(F_O)$, and let v be a vertex in C. Given a t-FAA, we call v* free *in C if one of the following conditions is satisfied:*

1 *v is unassigned, or*

2 *v is assigned to a face F in $out(C)$, and F is not the only face to which v is incident in $out(C)$.*

Moreover, v is strongly-free *if Condition 1 above is replaced by*

1' *v is unassigned, and v is either in the outer face or incident to more than one face in $out(C)$*

Intuitively speaking, a free vertex (strongly-free vertex) of a cycle C indicates a corner (convex corner) in $in(C)$. Fig. 5.1(2) is a cycle C in Fig. 5.1(1), which is drawn in 5-convex-SLR. The vertices c, d, and g are strongly-free vertices of the cycle C. Fig. 5.1(3) shows the set of pseudo-segments S_C for the cycle C. The vertices a, d, and i are the extremal points in S_C. Fig. 5.1(4) is a 6-SLR. As we shall prove in the following theorem, the FAA described in Fig. 5.1(4) cannot be a convex-SLR since the cycle (c, d, f, g, e) only has 2 strongly-free vertices c and g. Note that the vertex e is free but not strongly-free. In any drawing realizing that FAA, e must be a concave corner in the face interior to the cycle (a, c, e, g, i, h).

The following key theorem, one of the main results of this chapter, characterizes graphs admitting t-SLR and t-convex-SLR in terms of FAAs.

Theorem 5.2. *Let G be a biconnected plane graph whose degree 2 vertices are all in $V(F_O)$. G admits a t-convex-SLR (resp., t-SLR) iff there exists a t-FAA such that each cycle has at least 3 strongly-free (resp., free) vertices.*

Proof. (Idea) As the detailed proof is quite lengthy, we only describe the intuitive idea here. The full proof is presented in the next section. From our previous discussion, it is clear that a t-FAA of a plane graph naturally induces a contact system of pseudo-segments. For deciding whether the contact system is stretchable (implying that the plane graph admits a t-SLR), a direct application of Theorem 5.1 requires checking *all* sub-systems of pseudo-segments for the availability of 3 extremal points. The current theorem shows a simpler characterization, i.e., examining only subsets of pseudo-segments of the form S_C for some cycle C is sufficient. Furthermore, we are able to relate the availability of 3 extremal points of pseudo-segments of S_C to the presence of at least 3 free vertices along cycle C. See Fig. 5.1(2, 3, 5, 6) for instance.

If each face is further required to be a convex polygon, we need to prevent a vertex from causing a face to be a concave polygon, like the vertex e in the cycle depicted in Fig. 5.1(5). It turns out that adding the constraint forcing each free vertex to be incident to more than one face in $out(C)$ (see Condition (1') in Definition 5.5) leads to a necessary and sufficient characterization. □

It is easy to extend Theorem 5.2 to all biconnected plane graphs by modifying the definition of FAAs to handle degree 2 inner vertices. However, as the situation would not be encountered throughout the chapter, we omit it in order to reduce complication.

To give a characterization of convex polygonal duals, in what follows we establish a link between t-sided convex polygonal duals and its primal counterpart, t-convex-SLRs.

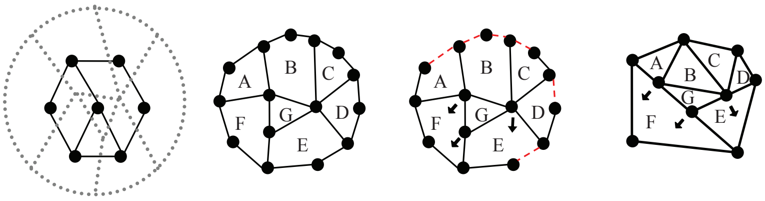
Given a plane graph G, one may hope to find some sort of a "dual" graph G^* such that any t-convex-SLR of G^* is also a t-sided convex polygonal dual of G. Unfortunately, this kind of a reduction strategy turns out to be more complex than it appears on the surface, as the polygon associated with a vertex $v \in F_O(G)$ may touch the boundary of the t-sided convex polygonal dual of G on 0, 1, ..., $t-1$ sides (see Fig. 5.2(4)). As an attempt to resolve such a difficulty, we define the G^* associated with a graph G as follows:

Definition 5.6. *Given a plane graph G and an integer t, the graph G^* is defined to be the result of the following construction steps:*

1. *Add a new vertex s in the unbounded face of G, and add an edge between s and each vertex in the outer face.*
2. *Take the dual, and the new outer face is designated to the one corresponding to s.*
3. *Subdivide each edge in the outer face to transform it to a path of $t-1$ edges.*

The following result is then straightforward.

Theorem 5.3. *A plane graph G admits a t-sided convex polygonal dual iff there is a graph G' that results from contracting some edges along the outer face of G^*, admitting a t-convex-SLR.*



(1) (2) (3) (4)

Figure 5.2: Concepts in Section 5.3.

Theorems 5.2 and 5.3 relate the problem of finding a convex polygonal dual to finding a set of boundary edges to be contracted and a corner labeling satisfying some constraints. In comparison with previous techniques designed for contact graph representations, the greatest advantage of Theorem 5.3 is that it turns a geometry problem to a purely graph-theoretic one. This, in conjunction with Theorem 5.2, allows us to get rid of any tedious and laborious geometric construction process when designing algorithms for contact graph representations.

By offering the possibility of contracting boundary edges of G^*, polygons associated with vertices in $F_O(G)$ can touch the boundary of the convex polygonal dual of G on 0, 1, …, $t-1$ sides.

Fig. 5.2 illustrates the concepts presented in the section: (1) a graph G, (2) its associated G^* (for $t = 3$), (3) applying edge contraction to the dashed edges along the outer face of G^*, (4) a 3-convex-SLR of G^* which is also a 3-sided convex polygonal dual of G.

In Fig. 5.2(3) and Fig. 5.2(4), flat angle assignments are annotated by arrows. In Fig. 5.2(4), faces B, C and F touch the boundary on 0, 1, or 2 sides, respectively. Note that an edge contraction has the same effect of a corner assignment in $F_O(G^*)$. Therefore, we can assume that no assignment occurs in $F_O(G^*)$ since edge contraction already handles it.

(Remark) Theorem 5.2 is of independent interest as it improves the main result in [2] (i.e., Theorem 2.10 in [2]) in the following way: (i) We check only simple cycles instead of all outline cycles; (ii) the result holds for all t-FAAs instead of 3-FAAs only; and (iii) we are able to deal with both polygons and convex polygons.

5.4 Proof of Theorem 5.2

In this section we give the full proof of Theorem 5.2. This section can be skipped without loss of continuity.

It is evident from the definition of t-FAA that, if the drawing can be straighten to satisfy the assignment, each inner face is assured to have at most t sides. Therefore, to prove Theorem 5.2, the only two things we need to deal with are (i) stretchability constraint and (ii) convexity constraint.

(Terminologies) We extend the graph-theoretic definitions of cycle, internal region, and outer region to pseudo-segments. Given a plane graph G and an FAA with Σ_G the contact system of pseudo-segments corresponding to the FAA, a subset S' ($\subseteq \Sigma_G$) of pseudo-segments is a *cycle* if there exists a cycle C in G which induces S' (i.e., $S_C = S'$). Given a subset $S \subseteq \Sigma_G$, the *internal region* of S, denoted by $in(S)$, is the set of faces in G surrounded by S. The *outer region* of S is the set of faces in G excluding the ones in $in(S)$.

We deal with (i) stretchability constraint first.

Lemma 5.1. *Let G be a biconnected plane graph whose degree 2 vertices are all in $V(F_O)$. It admits a t-SLR iff there exists a t-FAA such that each cycle has at least 3 free vertices.*

Proof. Let Σ_G denotes the contact system of pseudo-segments corresponding to the given FAA. The statement of this lemma is equivalent to "Σ_G is stretchable iff each cycle in G has at least 3 free vertices".

We first prove the following claim:

Claim: Let S be a set of connected pseudo-segments and $s \notin S$ be a segment such that $in(S) = in(S \cup \{s\})$. Then $S \cup \{s\}$ has at least as many extremal points as S does.

Proof of Claim. Based on the location of s with respect to S, there are two cases:

1. s is located in $in(S)$. Certainly the claim holds as the extremal points of S and $S \cup \{s\}$ are identical. See Fig. 5.3(2) for an example. A new segment $\{g, h\}$ is

added to the set $S = \{\{a,b\},\{c,d\},\{e,f\}\}$ depicted in Fig. 5.3(1). After the inclusion, the extremal points remain to be $\{a,c,e\}$.

2. s is located in $out(S)$. Since the inclusion of s does not enlarge $in(S)$, s touches S in at most 1 point; otherwise, a new cycle enlarging the internal region is formed. Therefore, the inclusion of s can only make at most 1 extremal point in S non-extremal, and it happens only when an extremal point in S touches the interior portion of s. However, since at least 1 of the 2 endpoints of s does not touch S, a new extremal point must be generated after the inclusion of s. See Fig. 5.3(3) for an example. The addition of the new segment $s = \{i,j\}$ changes the set of extremal points from $\{a,c,e\}$ to $\{c,e,i,j\}$.

As we have checked all the cases, the claim holds. See also Fig. 5.3(4) for a demonstration of a situation that the inclusion of s lowers the number of extremal points but increases the internal region. □

We write $m(S)$ to denote the number of maximal edge-connected set of faces in $in(S)$, and we let $B_1, B_2, \ldots, B_m$ be these sets of faces. We write C_i to denote the cycle enclosing B_i. In view of the above claim, the following statement is true:

To search for a subset of pseudo-segments of cardinality greater than 1 that has at most 2 extremal points, it suffices to check only candidate set X (set of set of pseudo segments) such that, for each $S \in X$:

1. Any proper subset of S encloses a smaller internal region than $in(S)$;

2. S is connected (since the number of extremal points of S is the sum of that of its connected components).

Note that the first condition implies $S = \bigcup_{i\in\{1,\ldots,m(S)\}} S_{C_i}$.

Our next task is to narrow down the candidate set to cycles only (i.e. the ones satisfying $m(S) = 1$ in the above candidate sets). For any $S \in X$ such that $m(S) > 1$, we show that it is always possible to find another set of pseudo segments $\tilde{S} \in X$ with

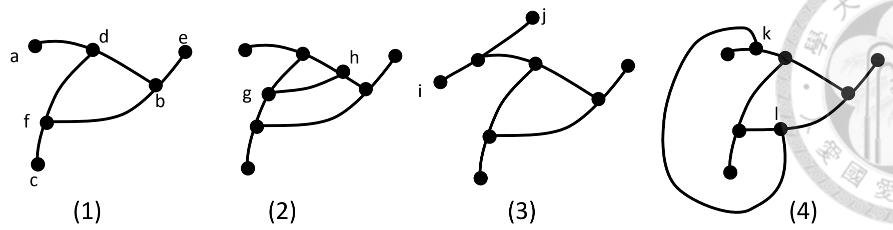


Figure 5.3: Illustration of inclusion of a new segment into a set of pseudo segments.

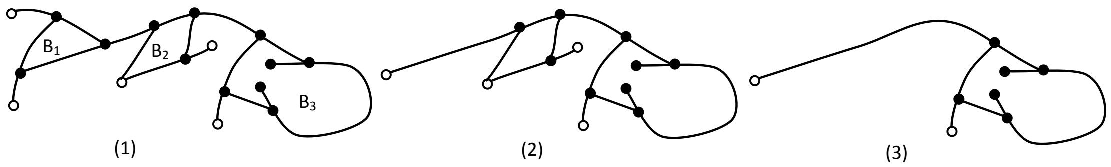


Figure 5.4: Illustration of finding $\tilde{S}$.

smaller number m such that the number of extremal points in $\tilde{S} \leq \max(2,$ the number of extremal points in $S)$.

Suppose that $S \in X$, and $m(S) > 1$. We choose an integer k such that the set $S' = \bigcup_{i \in \{1,\ldots,m(S)\}, i \neq k} S_{C_i}$ remains a connected set of pseudo segments. Since each B_i is chosen to be a maximal edge-connected set of faces, $S' \cap S_{C_k}$ contains exactly 1 segment. Furthermore, if we let $S' \cap S_{C_k} = \{s'\}$, $S_{C_k} - \{s'\}$ does not touch s' in both of its two endpoints. We divide the situation into two cases:

1. There is an extremal point of S_{C_k} located in $S_{C_k} - \{s'\}$. Then, S' contains no more extremal points than S does;

2. There is no extremal point of S_{C_k} located in $S_{C_k} - \{s'\}$. Then S_{C_k} has at most 2 extremal points.

As a result, choosing $\tilde{S}$ to be one of S', S_{C_k} that has smaller number of extremal points always works. The two cases are depicted in Fig. 5.4, where white dots indicate extremal points. Fig. 5.4(1) shows $S = \bigcup_{i \in \{1,\ldots,3\}} S_{C_i}$. Fig. 5.4(2) depicts the situation that $S' = S_{C_2} \cup S_{C_3}$ is chosen to be $\tilde{S}$ when $k = 1$; Fig. 5.4(3) shows the case of $k = 3$, in which $\tilde{S}$ is set to be S_{C_3}.

As we have narrowed down the candidate set X to cycles only, we now know that only cycles matter in deciding stretchability of Σ_G. To complete the proof, It suffices to prove the following two statements:

1. Given a cycle C in G, the number of the extremal points in S_C is smaller than or equal to the number of free vertices in C.

2. Given a subset of pseudo segments S that is a cycle, we can find a cycle C in G such that $S = S_C$, and that the number of the extremal points in S equals the number of free vertices in C.

First of all, we identify three types of points p in S_C:

1. p is located in C, and p is an extremal point of S. Then, it must be also a free vertex which is either unassigned or assigned to a face not containing any edge in C.

2. p is located in C, and p is an endpoint of a segment $s' \in S$ and is interior to another segment $s \in S$. Then, it must be also a free vertex which is assigned to a face containing an edge in C.

3. p is not located in C, and p is an extremal point of S.

Take Fig. 5.5 as an example. Let $C = (b, c, e, g, h)$, then $S_C = \{\{a, c\}$, $\{b, h\}$, $\{a, g\}$, $\{d, e\}$, $\{g, f\}\}$. The points a, b, c, d, e, f, g, h are of types $3, 2, 2, 3, 2, 3, 1, 2$, respectively. Note that a free vertex is either of type 1 or of type 2; an extremal point is either of type 1 or of type 3.

We define a mapping from free vertices of C to extremal points in S as follows:

1. For a type 1 free vertex v, we map it to itself.

2. For a type 2 free vertex v, we map it to the type 3 endpoint u of the segment s such that v is an interior point of s and that the path $(u, \ldots, v)$ along s does not share any edge with the cycle C.

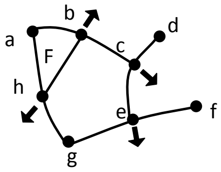


Figure 5.5: Illustration of relating extremal points to free vertices.

Also consider the above example, the free vertices b, c, e, g, h are mapped to endpoints a, d, f, g, a, respectively.

Proof of Statement 1. It suffices to show that the above mapping is onto. Suppose that an endpoint v is not being mapped to. It must be of type 3. Let it be an endpoint of a segment s. We walk along the segment s from v until we reach C. We must stop at an junction point v' which is interior to s and is an endpoint of another segment s'. It is immediate that v' should have been mapped to v. □

Proof of Statement 2. We choose C to be the cycle that encloses $in(S)$ (i.e. $in(S) = in(C)$), It suffices to show that the above mapping is an 1-1 correspondence between extremal points in S and free vertices in C. As we already argued that the mapping is onto, we only need to show that it is 1-1.

The only situation that violates the 1-1 condition is that a type 3 point p is being mapped by several type 2 points through different segments. This is not possible according to our choice of C since two of these segments and a portion of C must form a region belonging to $in(S) \setminus in(C)$. Consider again the same example, type 3 extremal point a is mapped to two type 2 free vertices b, h through segments $\{a, c\}, \{a, g\}$, and there is a face F surrounded by these two segments and a portion of the cycle belonging to $in(S) \setminus in(C)$. □

Hence the lemma is concluded. □

Our next task is to deal with (ii) convexity constraint. The idea of our strategy is to construct a modified t-FAA $\hat{X}$ on a modified graph $\hat{G}$ such that each vertex $v \in V(G)$ is automatically forced to be a convex corner in any face F it incident to.

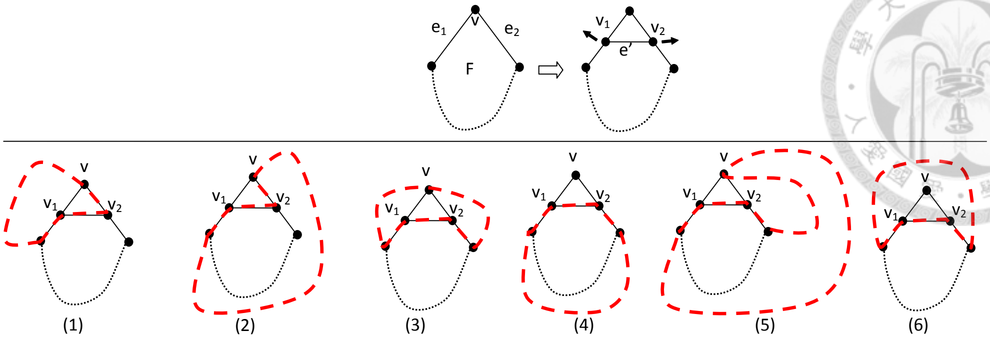


Figure 5.6: Illustration of proof of Lemma 5.2.

Given a biconnected plane graph G whose degree 2 vertices are all in $V(F_O)$, and given its t-FAA X, we define $\hat{G}$ and $\hat{X}$ as follows. For each vertex $v \in V(G)$ of degree > 2, for each inner face F incident to v such that v is not assigned to F, let e_1, e_2 be the two edges in $E(F)$ that incident to v, we subdivide these two edges. Let the two new vertices introduced by subdividing e_1, e_2 be v_1, v_2, respectively. We add a new edge $e' = \{v_1, v_2\}$, which divides F into two regions. Lastly, we force v_1, v_2 to be assigned to the face that is not formerly a region of F. See the upper part of Fig. 5.6 for an illustration.

It can be easily seen that X can be straighten in the way that each face in G is a convex polygon iff $\hat{X}$ is stretchable. Therefore, to complete the proof of Theorem 5.2, it suffices to prove the following lemma:

Lemma 5.2. *Let G be a biconnected plane graph whose degree 2 vertices are all in $V(F_O)$, and let X be a t-FAA of G. $\hat{X}$ is stretchable iff each cycle in G has at least 3 strongly-free vertices with respect to X.*

Proof. ($\Rightarrow$) Regarding necessity, it suffices to prove that, for each cycle C in G, we can find a cycle C' in $\hat{G}$ such that the number of free vertices in C' equals the number of strongly-free vertices in C. This statement implies that, when $\hat{X}$ is stretchable, all cycles in G has at least 3 strongly-free vertices with respect to X.

The procedure of finding C' is described as follows. Let v be an unassigned vertex in $V(C)$ incident to exactly one face F in $out(C)$. In other word, v is a free but not strongly-free vertex in C. Let e', e_1, e_2, v_1, v_2 be the edges and vertices described in the construction of $\hat{G}$ (for our chosen v, F). We first let $C' = C$. Then, for each such v, we replace (v_1, v, v_2) with $e = (v_1, v_2)$ in C'. It is clear from the definition of assignment $\hat{X}$ that both v_1, v_2 are not free in C'. As a result, a vertex in C' is free iff it is strongly-free in C.

($\Leftarrow$) For sufficiency, we prove that, when $\hat{X}$ is not stretchable, we can always find a cycle C' in G that has at most 2 strongly-free vertices with respect to X. We first select C to be a cycle in $\hat{G}$ of at most 2 free vertices. Among those cycles of at most 2 free vertices, we choose C to be the one having the least number of newly added edges (i.e. the e' described in the construction of $\hat{G}$).

If there is no newly added edges in C, simply setting $C' = C$ suffices. In the below, we assume there are some newly added edges in C.

Let e' the a newly added edge in C, and let e_1, e_2, v_1, v_2 be the relevant edges and vertices as defined before. All possible situations are enumerated in the lower part of Fig. 5.6. For each case except the last one, we describe an modification leading to another a cycle $\tilde{C}$ having at most 2 free vertices but not containing e'.

For (1),(2), we replace the portion (v_1, v_2, v) in C with (v_1, v). For (3),(4) we replace the portion (v_1, v_2) in C with (v_1, v, v_2). For (5), we replace the path $(v_1, v_2, \ldots, v)$ with (v_1, v). The change of free vertices is summarized in the below table:

Case	Among $\{v, v_1, v_2\}$, the ones "must" be free in C.	Among $\{v, v_1, v_2\}$, the ones "may" be free in $\tilde{C}$.
1	v_2	v
2	v_1	v
3	None	v
4	v_1, v_2	v
5	v_1, v_2	v

Note that we get two cycles after the modification for case (3). Since v may become a new free vertex in both new cycles, and since the number of free vertices in C is at most 2, the total number of free vertices in these two new cycles is at most 4. As a result, one of them contains at most 2 free vertices. Choosing $\tilde{C}$ to be this one suffices. For cases (1),(2),(4),(5), it is immediate from the table that the resulting cycle $\tilde{C}$ contains at most 2 free vertices.

Therefore, that the existence of configurations of cases (1-5) contradicts our choice of C is concluded. The following modification construct the desired cycle C' from C which is free of any configuration of cases (1-5). For each newly added edges e' in C, its local structure must obey case (6). We modify the cycle by replacing (v_1, v_2) with (v_1, v, v_2). After the modification, both v_1, v_2 remains non-free. However, v may become a new free vertex. As the face (v_1, v, v_2) is the only face v incident to in the outer region of the resulting cycle, v is not qualified to be strongly-free. By doing so for all e' in C, we get a desired cycle C'.

□

5.5 Fixed-parameter Tractability Results

Recall from Theorem 5.3 that a plane graph G admitting a t-sided convex polygonal dual can be characterized by the presence of a t-convex-SLR of G' (a graph resulting from applying some edge contraction in F_O of G^*), and the latter can further be captured by t-FAAs (Theorem 5.2). Like many graph structures expressible in MSO$_2$, it turns out that such a characterization can be formulated in the framework of MSO$_2$ (See Section 2.2). More precisely, we have:

Theorem 5.4. *Given a plane graph G, one can construct a graph $\tilde{G}$ along with a designated set of vertices F_{IN}, a designated vertex F_O, and a formula φ in MSO$_2$ such that G has a t-sided convex polygonal dual iff $(\tilde{G}, F_{IN}, F_O) \models \varphi$.*

Proof. First note that the parameter t in a t-sided convex polygonal of the plane graph G is considered a fixed constant.

The graph $\tilde{G}$ is constructed from G^* using the following procedure: (1) add a new vertex for each face in G^*; (2) for each newly added vertex v and its associated face

F, for all $u \in V(F)$, add an edge $\{u, v\}$. In setting up φ, we allow some designated vertices, edges, set of vertices, and set of edges to be associated with free variables. We use F_{IN} to denote the designated set of vertices in $V(\tilde{G})$ corresponding to inner faces in G^*, and $\mathrm{F_O}$ to denote the designated vertex in $V(\tilde{G})$ corresponding to the outer face of G^*.

We define the formula $\textsc{Corner}(e)$ which is true iff e is an edge incident to a vertex in $V(G^*)$ and a vertex in F_{IN}.

$$\textsc{Corner}(e) \equiv (\exists u, v)[\textsc{inc}(e, u) \wedge \textsc{inc}(e, v) \wedge (u \in \mathrm{F}_{\mathrm{IN}}) \wedge (v \notin \mathrm{F}_{\mathrm{IN}})]$$

We use a subset U of $\{e \in E(\tilde{G}) | \textsc{Corner}(e)\}$ to encode a t-FAA and a subset R of edges in the outer face of G^* to encode edge contraction.

Our goal is to define φ as $(\exists U, R)t\text{-}\textsc{Valid}\text{FAA}(U, R)$, where $t\text{-}\textsc{Valid}\text{FAA}(U, R)$ is true iff U, together with R, represents a t-FAA such that each cycle has at least 3 strongly-free vertices.

$$t\text{-}\textsc{Valid}\text{FAA}(U, R) \equiv t\text{-FAA}(U, R) \wedge (\forall C)\{\textsc{Cycle}(C, R)$$

$$\to \bigcup_{k=0,\ldots,3} [(3-k)\text{-}\textsc{BoundaryCorners}(C, R) \wedge (\exists v_1, \ldots, v_k) \bigwedge_{i=1,\ldots,k} \textsc{sFree}(v_i, C, U)]\}$$

t-FAA(U, R) is used to capture Definition 5.1. $\textsc{Cycle}(C, R)$ is to ensure that C is a cycle after applying edge contraction R. $i\text{-}\textsc{BoundaryCorners}(C, R)$ is true iff the number of vertices in $V(C) \cap V(F_O)$ remains at least i after applying the edge contraction R (these vertices are strongly-free boundary vertices in C). $\textsc{sFree}(v, C, U)$ is true iff v is strongly-free and non-boundary in C under the FAA U. Note that $\bigwedge_{i=1,\ldots,k}$ $\textsc{sFree}$ (v_i, C, U) is vacuously true if $k = 0$.

We leave the detailed description of these formulas to the next section. □

We are in a position to give our main result in this section (See Section 2.2 for the definition of tree-width and the definition of tree decomposition).

Theorem 5.5. *For any t, it can be decided in polynomial time whether a plane graph G admits a t-sided convex polygonal dual if there is a constant k such that:*

1. *tree-width of $G \leq k$, or*

2. *For all $v \in V(G)$ such that deg$(v) > 3$, there is a path linking v to the outer face of length $\leq k$.*

Proof. First of all, we can assume that every pair of vertices in G^* belong to at most two common inner faces, since otherwise those faces cannot be drawn as convex polygons simultaneously, implying that G has no convex polygonal dual. Testing whether every pair of vertices in G^* belong to at most two inner faces can be done easily in polynomial time.

For the first condition, according to Theorem 5.4, it suffices to show that, if the tree-width of G is k, then the tree-width of $\tilde{G}$ is $O(k^2)$. In the construction process of G^* described in Definition 5.6, the first step increases the tree-width by at most 1 since only 1 new vertex is added to the graph. The second step also increases the tree-width by at most 1 according to a well-known fact that the tree-widths of a graph and of its dual differ by at most 1. The third step does not increase the tree-width since subdivision does not increase the tree-width. Therefore, we conclude that the tree-width of G^* is at most $k + 2$.

The construction of $\tilde{G}$ adds a new vertex v per each face F, and then links v to all vertices in $V(F)$. We make the following claim:

Claim: Let T be a tree-decomposition of G^*, and let F be a face in G^*. We can find a subtree T_F in T such that (1) each bag in T_F contains at least two vertices in $V(F)$, and that (2) for each edge $\{v, w\}$ in $E(F)$, there is a bag in T_F containing v and w.

Proof of Claim. Let $v \in V(G^*)$, we denote T_v the subset of bags in $V(T)$ that contains v. It is clear from the definition of tree decomposition that T_v is a subtree of T. We define T' to be the union of all T_v such that $v \in V(F)$. It is clear that T' is also a subtree of T and that T_F, if exists, is a subtree of T'.

Suppose that there is no such T_F, then any subset satisfying (1) and (2) is not a subtree of T'. Hence we are able to find two edges $e_1 = \{u_1, v_1\}, e_2 = \{u_2, v_2\} \in V(F)$ and a bag $X \in V(T')$ such that (i) X contains only one vertex in $V(F)$, and that (ii) $T_{u_1} \cap T_{v_1}$ and $T_{u_2} \cap T_{v_2}$ are subtrees of different components of $T' \backslash \{X\}$.

Without loss of generality, we assume, for $i = 1, 2$, v_i immediately follows u_i in the clockwise circular order of vertices in F. Therefore, there are two disjoint paths $P_1 = (v_1, \ldots, u_2)$ and $P_2 = (v_2, \ldots, u_1)$ in F. Due to (ii), $\bigcup_{v \in P_i} T_v$ contains X for $i = 1, 2$, which contradicts with the definition of X. This completes the proof of our claim. □

We then construct a tree decomposition of $\tilde{G}$ based on any tree decomposition of G^*. We simply add each $v \in V(\tilde{G})$ corresponding to a face F of G^* to all bags in T_F. It is an easy routine to verify that the resulting tree is a valid tree decomposition of $\tilde{G}$.

There are only $O(k^2)$ distinct pairs of vertices in each bag of tree decomposition of G^*. Due to the above claim and the assumption that every pair of vertices in G^* belong to at most two common inner faces, we can increase the size of each bag by $O(k^2)$ during the construction of tree decomposition of $\tilde{G}$. Therefore, we conclude the proof that tree-width of $\tilde{G}$ is bounded for the case that condition 1 is met.

For the second condition, we first note that each inner vertex (i.e. not located in the outer face) in $V(G)$ must have degree at least 3, since otherwise it is clear that the graph has no convex polygonal dual. Also, it is easy to see that if this condition is satisfied, in $\tilde{G}$, the distance between $\mathrm{F_O}$ and any vertex in $\mathrm{F_{IN}}$ of degree at least 4 is $O(k)$.

Let S be a subgraph corresponding to a maximal connected set of triangles in G^*. For any $v \in V(S)$ such that any path linking the outer boundary of S and v has length at least 2, it is easy to see that any cycle in G^* passing v has at least 3 free vertices. It is due to the fact that, no flat angle assignment can be made in a triangle. Let S' be the subgraph induced by those vertices. See Fig. 5.7 for an example. The left illustration is a maximal connected set of triangles S. The white vertices are ones in S'. We contract each connected component of S' into a vertex. See the right illustration of Fig. 5.7 for an example. Since cycles that may have less than 3 strongly-free vertices do not pass

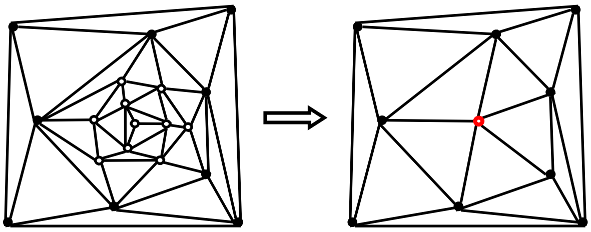


Figure 5.7: Illustration of the proof of Theorem 5.5.

through S', such contraction preserves the property of having t-convex-SLR. Hence we can use the $\tilde{G}$ constructed from the modified G^* instead of the original one. It is obvious from the definition of S and S' that the resulting $\tilde{G}$ constructed from the modified G^* has bounded outer planarity. Therefore, we conclude the proof as bounded outer planarity implies bounded tree-width. □

Theorem 5.5 implies polynomial time algorithms for many important graph classes appearing frequently in the literature [21, 28]. We have:

Corollary 5.1. *Deciding whether a plane graph admits a t-sided convex polygonal dual is solvable in polynomial time for graphs of max degree 3, partial 3-trees, and* k*-outerplane graphs.*

5.6 Exact Definition of the Formula t-ValidFAA

In this section, we give the exact definition for the formula t-ValidFAA. This section can be skipped without loss of continuity.

We adapt the following convention: C is an edge set intended to be a cycle. P is an edge set intended to be a path. u, v are vertices. e is an edge. U, R are reserved for representing FAA and edge contraction, respectively. W is reserved for set of vertices. H, Q are reserved for set of edges.

5.6.1 t-FAA

The two conditions in the definition of t-FAA can be expressed as follows. $\mathrm{C1}(U)$ ensures that the FAA U satisfies the condition "each vertex is assigned at most once". t-C2(U, R) ensures that each inner face F is assigned at least $|V(F)| - t$ times. In other words, t-C2(U, R) iff each inner face has at most t combinatorial convex corners.

$$\mathrm{C1}(U) \equiv (\forall v \in V(\tilde{G}) \setminus (\mathrm{F}_{\text{IN}} \cup \{\mathrm{F_O}\}))(\forall e_1, e_2 \in U)[\neg(\textsc{inc}(e_1, v) \wedge \textsc{inc}(e_2, v))]$$

$$t\text{-C2}(U, R) \equiv (\forall v \in \mathrm{F}_{\text{IN}})$$

$$\bigcup_{0 \leq a,b; a+b \leq t} (a\text{-C2-}\textsc{assignment}(v, U) \wedge b\text{-C2-}\textsc{contraction}(v, R))$$

To define t-C2(U, R), we introduce two formulas a-C2-$\textsc{assignment}(v, U)$ and b-C2 - $\textsc{contraction}(v, R)$ which together divide the task of t-C2(U, R). We assume that v represents an inner face F. We have a-C2-$\textsc{assignment}(v, U)$ iff the number of combinatorial convex corners in F not located in the boundary of drawing is at most a. Similarly, b-C2-$\textsc{contraction}(v, R)$ iff the number of combinatorial convex corners in F located in the boundary of drawing is at most b.

$$n\text{-C2-}\textsc{assignment}(v, U) \equiv (\neg\exists e_1, \ldots, e_{n+1})\{ \bigwedge_{1 \leq i < j \leq n+1} [e_i \neq e_j] \wedge$$

$$\bigwedge_{i=1,\ldots,n+1} [e_i \notin U \wedge \textsc{inc}(e_i, v)]\}$$

$$n\text{-C2-}\textsc{contraction}(v, R) \equiv \neg \bigcup_{k=1,\ldots,n+1} \bigcup_{1 \leq s_1 \leq \ldots \leq s_k, \sum_{i=1}^{k} s_i = n+1} (\exists P_1, \ldots, P_k)$$

$$\textsc{DisjointEdgeSets}(P_1, \ldots, P_k) \wedge$$

$$\bigwedge_{j=1,\ldots,k} (\textsc{Max-BoundaryPath}(P_j, v) \wedge \textsc{AtLeast-}s_j\textsc{-Corners}(P_j, R))$$

The formula Max-BoundaryPath(P, C), where P (C) is a set of edges representing a path (cycle) in $E(G^*)$, expresses that P is a maximal subpath of $C \cap E(F_O(G^*))$. Similarly, if $v \in \mathrm{F_{IN}}$ represents an inner face F in G^*, Max-BoundaryPath(P, v) is the same as Max-BoundaryPath$(P, E(F))$. The formula AtLeast-n-Corners(P, R), where P is a subpath of $E(F_O(G^*))$, expresses that the number of combinatorial convex corners (including the two endpoints) in P is at least n. DisjointEdgeSets$(P_1, \ldots, P_n)$ is true iff $P_1, \ldots, P_n$ are disjoint edge sets.

Satisfying both C1(U) and t-C2(U, R) is still not sufficient to guarantee that the current edge contraction and assignment together form an t-FAA. We still need to check that (1) each face remains a cycle after edge contraction and that (2) the corner assignment only involves non-boundary vertices (i.e. those not in $V(F_O(G^*))$). The above (1) and (2) are captured by the formula Validity(U, R).

Therefore, the following formula determines whether the given U, R satisfy the definition of t-FAA:

$$t\text{-FAA}(U, R) \equiv \textsc{Validity}(U, R) \wedge \text{C1}(U) \wedge t\text{-C2}(U, R)$$

5.6.2 t-ValidFAA

Similarly, we can express the notion of strongly-freeness in an MSO$_2$ sentence sFree (v, C, U) which is true iff v, an inner vertex in G^*, is a strongly-free vertex in the cycle C under the FAA U. To deal with corners in F_O, the formula n-BoundaryCorners(C, R) is true iff, given the edge contraction R, the number of vertices in $V(C) \cap V(F_O)$ remains at least n. To ensure that C is a cycle after applying edge contraction R, we use the formula Cycle(C, R).

Prior to the definition of sFree(v, C, U), we define Inside(u, C) (Outside(u, C)) which is true iff u is a vertex located inside (outside) of the cycle C.

$$\textsc{Inside}(u, C) \equiv (\forall e \in C)\neg\textsc{inc}(e, u) \wedge (\exists W \subseteq V(\tilde{G}))\{$$

$$(\forall v)[(\exists e \in C)(\textsc{inc}(e, v)) \rightarrow (v \notin W)] \wedge (u \notin W)$$

$$\wedge(\textsc{f}_{\textsc{o}} \in W) \wedge (\forall v_1 \in V(\tilde{G}) \setminus W, v_2 \in W)[\textsc{adj}(v_1, v_2) \rightarrow (\exists e \in C)(\textsc{inc}(e, v_1))]\}$$

$$\textsc{Outside}(u, C) \equiv \neg\textsc{Inside}(u, C) \wedge (\forall e \in C)\neg\textsc{inc}(e, u)$$

$$\textsc{sFree}(u, C, U) \equiv [(\exists e \in C)\textsc{inc}(e, u)] \wedge [\neg\textsc{adj}(\textsc{f}_{\textsc{o}}, u)]$$

$$\wedge\{[(\exists e \in U)(\exists v)\textsc{inc}(e, v) \wedge \textsc{inc}(e, u) \wedge \textsc{Outside}(v, C)] \vee [(\neg\exists e \in U)\textsc{inc}(e, u)]\}$$

$$\wedge(\exists v_1, v_2 \in \textsc{f}_{\textsc{in}})[\textsc{Outside}(v_1, C) \wedge \textsc{Outside}(v_2, C) \wedge \textsc{adj}(u, v_1) \wedge \textsc{adj}(u, v_2) \wedge (v_1 \neq v_2)]$$

The sentence of $\textsc{sFree}(u, C, U)$ is the conjunction of four components. The first one says that u is in cycle C. The second one further ensures that u is not in $V(F_O)$. The third one expresses that u is either unassigned or assigned to a face in $out(C)$. The fourth one then requires u to be incident to more than one face in $out(C)$. It is easy to see that this sentence exactly matches the definition of strongly-free vertex in Definition 5.5 for the case of inner vertex.

Finally, according to the Theorem 5.3, we define t-$\textsc{Valid}$FAA as follows.

$$t\text{-}\textsc{Valid}\text{FAA}(U, R) \equiv t\text{-FAA}(U, R) \wedge (\forall C)\{\textsc{Cycle}(C, R)$$

$$\rightarrow \bigcup_{k=0,\ldots,3} [(3-k)\text{-}\textsc{BoundaryCorners}(C, R) \wedge (\exists v_1, \ldots, v_k) \bigwedge_{i=1,\ldots,k} \textsc{sFree}(v_i, C, U)]\}$$

Note that $\bigwedge_{i=1,\ldots,k} \textsc{sFree}(v_i, C, U)$ is vacuously true if $k = 0$.

The missing descriptions of formulas are described in the next.

5.6.3 Remaining Formulas

We first define some useful formulas. $\textsc{incEdge}(e_1, e_2)$ is true iff the two edges share a vertex. $\textsc{MiddleEdge}(e, H)$ is true iff the two vertices in e are of degree 2 in H. $\textsc{EndEdge}\ (e, H)$ is true iff the two vertices in e are of degree 1,2, respectively, in H.

$$\textsc{incEdge}(e_1, e_2) \equiv (\exists u)[\textsc{inc}(e_1, u) \wedge \textsc{inc}(e_2, u)]$$

$$\textsc{MiddleEdge}(e, H) \equiv (\exists e_1, e_2)(\exists u_1, u_2)(e_1 \in H) \wedge (e_2 \in H) \wedge \textsc{inc}(e_1, u_1)$$

$$\wedge \textsc{inc}(e_2, u_2) \wedge \textsc{inc}(e, u_1) \wedge \textsc{inc}(e, u_2) \wedge (e_1 \neq e) \wedge (e_2 \neq e) \wedge (u_1 \neq u_2)$$

$$\wedge (\forall e_3)\{[(e_3 \in H) \wedge (e_3 \neq e) \wedge (\textsc{incEdge}(e, e_3))] \rightarrow [(e_3 = e_1) \vee (e_3 = e_2)]\}$$

$$\textsc{EndEdge}(e, H) \equiv (\exists e_1)(e_1 \in H) \wedge (e_1 \neq e) \wedge \textsc{incEdge}(e, e_1)$$

$$\wedge (\forall e_2)\{[(e_2 \in H) \wedge (\textsc{incEdge}(e, e_2)) \wedge (e_2 \neq e)] \rightarrow (e_1 = e_2)\}$$

$$\textsc{ConnectedEdges}(H) \equiv (\neg\exists Q)\{(Q \subseteq H) \wedge (H \neq Q)$$

$$\wedge (\forall e_1 \in Q, e_2 \in H \setminus Q)(\neg\exists u)[\textsc{inc}(e_1, u) \wedge \textsc{inc}(e_2, u)]\}$$

$$\textsc{DisjointEdgeSets}(H_1, \ldots, H_k) \equiv (\forall e) \bigwedge_{1 \leq i < j \leq k} \neg[(e \in H_i) \wedge (e \in H_j)]$$

$$\textsc{Cycle}(C) \equiv \textsc{ConnectedEdges}(C) \wedge (\forall e \in C)\textsc{MiddleEdge}(e, C)$$

$$\textsc{Path}(P) \equiv \textsc{ConnectedEdges}(P) \wedge (\exists e_1, e_2)\{\textsc{EndEdge}(e_1, P) \wedge \textsc{EndEdge}(e_2, P)$$

$$\wedge (e_1 \neq e_2) \wedge (\forall e \in P)[((e \neq e_1) \wedge (e \neq e_2)) \rightarrow \textsc{MiddleEdge}(e, P)]\}$$

Next, we deal with boundary paths (i.e. the subpaths of F_O).

$$\textsc{BoundaryVertex}(v) \equiv \textsc{adj}(v, \mathrm{F_O})$$

$$\textsc{BoundaryEdge}(e) \equiv (\exists u, v)[(u \neq v) \wedge \textsc{BoundaryVertex}(u)$$

$$\wedge \textsc{BoundaryVertex}(v) \wedge \textsc{inc}(e, u) \wedge \textsc{inc}(e, v)]$$

$$\textsc{BoundaryPath}(P) \equiv \textsc{Path}(P) \wedge (\forall e \in P)\textsc{BoundaryEdge}(e)$$

$$\textsc{BoundaryPath}(P, C) \equiv \textsc{BoundaryPath}(P) \wedge (P \subseteq C)$$

$$\textsc{Max-BoundaryPath}(P, C)$$

$$\equiv (\forall P_2)[(\textsc{BoundaryPath}(P, C) \wedge \textsc{BoundaryPath}(P_2, C) \wedge (P \subseteq P_2)) \rightarrow (P = P_2)]$$

The following formula $\textsc{InnerFace}(C, v)$ is true iff v represents an inner face ($v \in$ $\mathrm{F_{IN}}$) corresponding to the cycle C.

$$\textsc{InnerFace}(C, v) \equiv \textsc{Cycle}(C) \wedge (\forall e \in C)[(\exists u_1, u_2)\textsc{adj}(u_1, v) \wedge \textsc{inc}(e, u_1)$$

$$\wedge \textsc{adj}(u_2, v) \wedge \textsc{inc}(e, u_2) \wedge (u_1 \neq u_2)]$$

$$\textsc{Max-BoundaryPath}(P, v) \equiv (\exists C)[\textsc{InnerFace}(C, v) \wedge \textsc{Max-BoundaryPath}(P, C)]$$

Note that v in the above formula is intended to be a vertex in $\mathrm{F}_{\textsc{in}}$ (i.e. representing an inner face).

$$\textsc{AtMost-}n\textsc{-Corners}(P, R) \equiv (\neg\exists e_1, \ldots, e_n \in P)[\bigwedge_{i=1,\ldots,n} (e_i \notin R) \wedge \bigwedge_{1\leq i<j\leq n} (e_i \neq e_j)]$$

The number of combinatorial convex corners in the boundary subpath P of any cycle equals $1+$ the number of boundary edges in P not contracted. Therefore, there are at most n such corners iff it is impossible to find n edges on P not contracted.

The next formula n-$\textsc{BoundaryCorners}(C, R)$ is intended to capture the situation that the number of boundary combinatorial convex corners in C after applying edge contraction described by R is at least n. We first define $\textsc{AtLeast-}n\textsc{-Corners}(P, R)$ which is sort of negation of $\textsc{AtMost-}n\textsc{-Corners}(P, R)$. Then, n-$\textsc{BoundaryCorners}(C, R)$ can be defined in a way analogues to the definition of n-C2-$\textsc{contraction}(v, R)$.

$$\textsc{AtLeast-}n\textsc{-Corners}(P, R) \equiv \neg\textsc{AtMost-}(n-1)\textsc{-Corners}(P, R)$$

$$n\textsc{-BoundaryCorners}(C, R) \equiv \bigcup_{k=1,\ldots,n} \bigcup_{1\leq s_1\leq\ldots\leq s_k, \sum_{i=1}^{k} s_i = n} (\exists P_1, \ldots, P_k)$$

$$\textsc{DisjointEdgeSets}(P_1, \ldots, P_k) \wedge$$

$$\bigwedge_{j=1,\ldots,k} (\textsc{Max-BoundaryPath}(P_j, C) \wedge \textsc{AtLeast-}s_j\textsc{-Corners}(P_j, R))$$

The next formula $\textsc{Cycle}(C, R)$ is intended to capture the situation that the cycle C remains to be a cycle (i.e. has no repeated vertices) after the edge contraction R.

Repeated vertices appear only when there is a boundary path $P = (u, \ldots, v)$ such that $u, v \in V(C)$, P and C share no edge, and all edges in P belong to R. Moreover, in order to make sure that C is a cycle in G^*, we require two formulas $\textsc{InG*}(C)$ and $\textsc{Cycle}(C)$.

$$\textsc{EdgeSet-Vertex-contain}(H, v) \equiv (\exists e)[(e \in H) \wedge \textsc{inc}(e, v)]$$

$$\textsc{InG*}(H) \equiv (\forall v)(v \in \mathrm{F}_{\text{IN}} \vee v = \mathrm{F_O}) \rightarrow (\neg\exists e)[e \in H \wedge \textsc{inc}(e, v)]$$

Note that $\textsc{InG*}(H)$ is true iff H does not contain any edge not in $E(G^*)$.

$$\textsc{EndPoint}(v, P)$$

$$\equiv \textsc{EdgeSet-Vertex-contain}(P, v) \wedge (\exists e)(\forall e_1 \in P)(\textsc{inc}(e, v) \leftrightarrow e = e_1)$$

As its name suggests, $\textsc{EndPoint}(v, P)$ is true iff v is one of the two endpoints of the path P.

We are now in a position to define $\textsc{Cycle}(C, R)$.

$$\textsc{Cycle}(C, R) \equiv \textsc{Cycle}(C) \wedge \textsc{InG*}(C)$$

$$\wedge\neg(\exists P)\{\textsc{BoundaryPath}(P) \wedge (P \subseteq R) \wedge \textsc{DisjointEdgeSets}(P, C)$$

$$\wedge(\forall v)[(\textsc{EdgeSet-Vertex-contain}(C, v) \wedge \textsc{EdgeSet-Vertex-contain}(P, v))$$

$$\leftrightarrow \textsc{EndPoint}(v, P)]\}$$

We are in a position to define $\textsc{Validity}(U, R)$. The formula can be divided to two sub-formulas regarding U and R, respectively. For FAA U, we need to make sure that the flat angle assignment only involves non-boundary vertices (i.e. those in $V(G^*) \setminus V(F_O(G^*))$). For edge contraction R, we require that each face (including the outer one) remains a cycle after edge-contraction.

$$\textsc{Validity-assignment}(U) \equiv (\forall e \in U)\textsc{Corner}(e)$$

$$\wedge(\forall v)[\textsc{EdgeSet-Vertex-contain}(U, v) \rightarrow (v \neq \textsc{f}_{\textsc{o}} \wedge \neg\textsc{adj}(\textsc{f}_{\textsc{o}}, v))]$$

$$\textsc{Validity-contraction}(R) \equiv (\forall v \in \mathrm{F}_{\mathrm{IN}})(\exists C)[\textsc{Cycle}(C, R) \wedge \textsc{InnerFace}(C, v)]$$

$$\wedge(\exists e_1, e_2, e_3)[\bigwedge_{i\in\{1,2,3\}} ((e_i \notin R) \wedge \textsc{BoundaryEdge}(e_i)) \wedge \bigwedge_{1\leq i<j\leq 3} (e_i \neq e_j)]$$

$$\textsc{Validity}(U, R) \equiv \textsc{Validity-assignment}(U) \wedge \textsc{Validity-contraction}(R)$$

5.7 Further Applications of Our Technique

In addition to the fixed-parameter tractability results derived in the previous sections, in this section we give alternate proofs for some interesting existing results using the technique we have developed. First, we give a simple proof for a result of [17]:

Theorem 5.6 ([17]). *Each maximal plane graph admits a 6-sided convex polygonal dual.*

Proof. Our alternative proof relies on Theorem 4.1, which guarantees that maximal plane graphs admit rectilinear duals using only upside-down T-shapes and their degenerations. Fig. 5.8(1.1) lists the set of allowed shapes, while shapes listed in Fig. 5.8(1.2) are not allowed.

Given a maximal plane graph G (see Fig. 5.8(2.1)), an FAA of G^* is constructed naturally according to a rectilinear dual as shown in Fig. 5.8(2.2). To be precise, we make an assignment at each 180° corner not in the boundary of the drawing. Note that the concave corners (bends) of any rectilinear polygon are not vertices in G^*. As such an FAA may not lead to a stretchable drawing, we do some adjustments by unassigning some vertices according to the rules specified in Fig. 5.8(3). See Fig. 5.8(2.3) for an example of an FAA after the adjustment.

It is easy to see that the resulting FAA is a 6-FAA. Prior to the adjustment, it is a 6-FAA since only a convex corner of a polygon can be a combinatorial convex corner, and since each polygon has at most 6 convex corners. Though each adjustment increases the number of combinatorial convex corner of a face by one, we can apply it only when we have a nearby convex corner that is not a vertex in G^*. Therefore, the FAA after the adjustment is still a 6-FAA.

What is left to be done is to show that each cycle has at least 3 strongly-free vertices. Let C be any cycle in G^*. Consider the sub-drawing, which is a rectilinear polygon, of C in the rectilinear dual. Let $\overline{ab}$ and $\overline{cd}$ be its highest and lowest horizontal segments, respectively, as shown in Fig. 5.8(4). It is immediate that a and b are strongly-free vertices of C. Suppose that there is no strongly-free vertex on $\overline{cd}$. Then, c and d must be bends in the drawing (i.e. not a vertex in G^*), and no adjustment is applied on $\overline{cd}$. This implies that there is no line segment touching $\overline{cd}$ from $out(C)$, meaning that there is a non-convex polygon F in $out(C)$ incident to $\overline{cd}$, which is a contradiction to the allowed set of shapes (i.e., upside-down T-shapes and their degeneracies). Therefore, we conclude that there is a strongly-free vertex in $\overline{cd}$, and hence C has at least 3 strongly-free vertices. See Fig. 5.8(2.4) for the resulting convex polygonal dual. □

As each Hamiltonian maximal plane graph admits a rectilinear dual using only L-shape and rectangles [43], following a similar approach, our technique can also be utilized to give a simple proof for the following:

Theorem 5.7 ([43]). *Each Hamiltonian maximal plane graph admits a 5-sided convex polygonal dual.*

Next, we showcase a quick proof for the main result of [28]:

Theorem 5.8 ([28]). *Each triconnected cubic plane graph admits a proper touching triangle representation.*

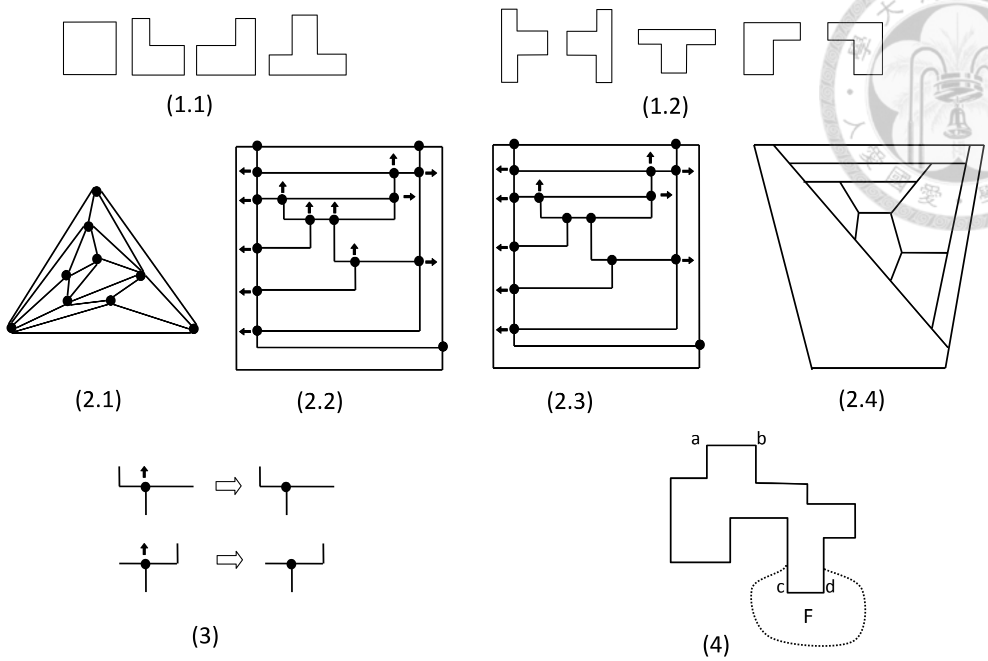


Figure 5.8: Illustration of the proof of Theorem 5.6.

Proof. A proper touching triangle representation is just a 3-sided convex polygonal dual whose boundary is a triangle.

Let G be a triconnected cubic plane graph, and we construct its associated G^* as described in Section 5.3. We let $F_O(G) = (v_1, v_2, \ldots, v_s)$ be the outer face of G. Note that we must have $s \geq 3$ since G is simple. It is easy to see that the subgraph H of G^* induced by the faces corresponding to vertices in $V(G) \setminus V(F_O(G))$ (the shaded area in Fig. 5.9) is biconnected, since otherwise G is not triconnected.

We contract most of the boundary edges, only leaving a boundary edge for each of F_1, F_2, and F_3, where F_i is the face in G^* corresponding to v_i. We let the 3-FAA contain only $u_i \to F_i, i \in \{1, 2, 3\}$, where $u_i \in V(G^*)$ is the shared non-boundary vertex of F_i and F_{i+1}. See Fig. 5.9 for an illustration. We claim that our edge contraction and FAA work.

It is immediate that the assignment is a 3-FAA such that the boundary in the resulting drawing is a triangle whose three corners are c_1, c_2 and c_3 in Fig. 5.9. What remains to be done is to verify that each cycle C has 3 free vertices:

- If C contains none of c_1, c_2 and c_3, it belongs entirely to H (the shaded area).

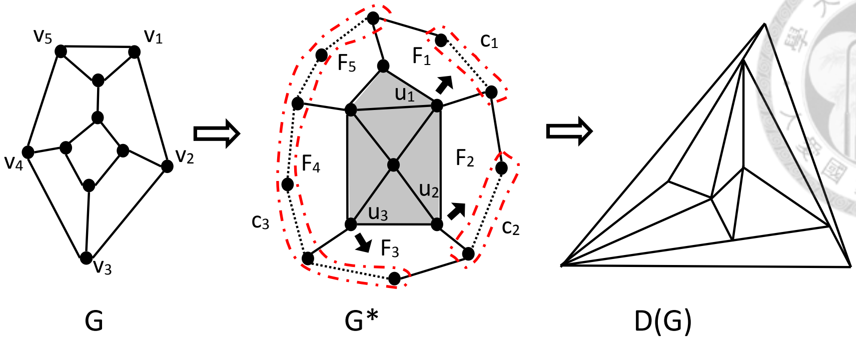


Figure 5.9: Illustration of the proof for Theorem 5.8.

Then, certainly all its vertices are free, as they are not assigned to $in(C)$.

- If C contains exactly one of c_1, c_2 and c_3, the one it contains must be c_3 (since c_1, c_2 have only one adjacent non-boundary vertex). Let x and y be the two neighboring vertices of c_3 in C. It is clear that x, c_3 and y are 3 free vertices in C, Since x and y are either unassigned or assigned to $out(C)$, and since c_3 is unassigned.

- If C contains exactly two of c_1, c_2 and c_3, as these two corners already contribute two free vertices to C, the only situation that makes C to have less than 3 free vertices is that all vertices in $C \setminus \{c_1, c_2, c_3\}$ are assigning to $in(C)$. However, since only u_1, u_2 and u_3 are involved in our FAA (i.e. $V(C) \subseteq \{c_1, c_2, c_3, u_1, u_2, u_3\}$), we can assure that it never happen by examining a small bounded amount of possibilities.

- If C contains all of c_1, c_2 and c_3, these three corners form 3 free vertices of C.

□

Adapting our approach, the laborious process of explicitly assigning positions for all junction points and all segments to construct a drawing, which inevitably appears in many works on contact graph representations in non-rectilinear situation, can be prevented.

Chapter 6

Area-universal Drawings of Biconnected Outerplane Graphs

We continue the study of convex polygonal dual in this chapter.

Polygonal complexity and area-universality are two prime quality measure in contact graph representations. As a complement to the last chapter which was centered only on polygonal complexity, in this chapter we study convex polygonal duals of biconnected outerplane graphs for both polygonal complexity and area-universality.

Our contributions are:

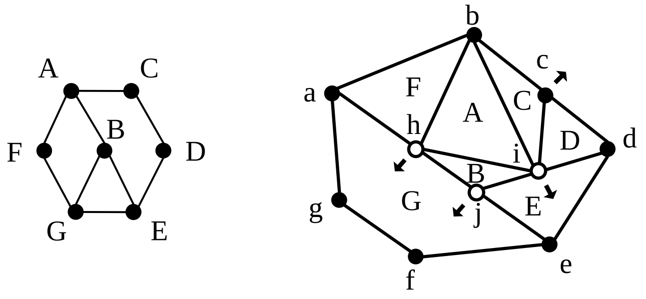
1. A clean necessary and sufficient condition for the existence of a convex polygonal dual of a given polygonal complexity.
2. An simple algorithm for constructing an area-universal convex polygonal dual of low polygonal complexity.

6.1 Terminologies

Our interests in this chapter are biconnected outerplane graphs having k-sided convex polygonal duals with their boundary polygons being t-sided. We abbreviate such a representation as t-TkR, where T stands for touching and R stands for representation. As an example, Fig. 6.1(2) shows a 6-T4R of the plane graph depicted in Fig. 6.1(1).

Note that 3-T3R coincides with the proper touching triangle graph representation investigated in [21, 28].

We write *junction points* to denote the points that are endpoint of some segments in the drawing. In Fig. 6.1(2), there are 10 junction points. Among them, a, b, c, d, e, f, g



(1) (2)

Figure 6.1: A graph G and its convex polygonal dual.

are the boundary ones, and i, j, k are the non-boundary ones. Note that c is interior to the side (b, d) of the boundary polygon. The arrows in the drawing indicate $180°$ angles.

6.2 Drawing Biconnected Outerplane Graphs

We first prove the following lemma which gives an upper bound of the number of sides of the boundary polygon:

Lemma 6.1. *Let G be a biconnected outerplane graph. If G admits a $t-$TkR, then $3 \leq t \leq (k-1)|V(G)| - |E(G)| + 1$. Moreover, the equality $t = (k-1)|V(G)| - |E(G)| + 1$ holds iff in the drawing, (1) each polygon is $k-$sided, and (2) each non-boundary junction point is interior to a side of a polygon.*

Proof. It is clear that $t \geq 3$ since every polygon must have at least 3 sides.

Let N be the total number of corners of polygons representing vertices in $V(G)$, which is at most $k|V(G)|$. Since G is a biconnected outerplane graph, each polygon must intersect the boundary of the drawing in one connected path or a point. Otherwise, the vertex v corresponding to the polygon will be a cut-vertex in G. Since a path of s sides constitutes $s + 1$ corners, when a polygon contains s sides on the boundary of the drawing, it has at most $k - s - 1$ corners located not in the boundary of the drawing.

Let $N = N_O + N_I$, where N_O denotes the total number of corners located in the boundary of the drawing (i.e. at boundary junction points), and where N_I denotes the total number of corners located interior of the drawing (i.e. at non-boundary junction points).

First, we have $N_O \geq |V(G)| + t$: We write N_v to denote the number of sides in the boundary of the drawing intersecting the polygon corresponding to v. In view of above, $N_O = \sum_{v \in V(G)}(N_v + 1) = \sum_{v \in V(G)} N_v + |V(G)| \geq t + |V(G)|$ (a side can intersect more than one polygon).

We write C to denote the total number of non-boundary junction points. It is obvious that C is an upper bound of number of $180°$ angles at non-boundary junction points, since each junction point can have at most one. Let $D =$ summation of deg(p), for all non-boundary junction points p. Then, clearly we have $D - C$ as an lower bound of N_I.

Exact value of C, D can be expressed in terms of $|V(G)|$ and $|E(G)|$. C is the number of inner faces in G. Therefore, according to Euler's formula, $C = |E(G)| - |V(G)| + 1$. D is simply number of boundary edges plus two times of the number of non-boundary edges in G, so we have $D = 2|E(G)| - |E(F_O(G))|$. Since G is an outerplane graph, $|E(F_O(G))| = |V(G)|$. To sum up, $N_I \leq [2|E(G)| - |V(G)|] - [|E(G)| - |V(G)| + 1] = |E(G)| - 2|V(G)| - 1$.

Then, since it is clear that $k|V(G)| \geq N$, we have $k|V(G)| \geq N = N_O + N_I \geq [|V(G)| + t] + [|E(G)| - 2|V(G)| - 1]$. By re-ordering the terms, we get $t \leq (k-1)|V(G)| - |E(G)| + 1$.

The equality $t = (k-1)|V(G)| - |E(G)| + 1$ is reached iff the two equalities $k|V(G)| = N$, $N_I = |E(G)| - 2|V(G)| - 1$ are met. The first one is met iff each polygon is $k-$sided. The second one is met iff each non-boundary junction point has an $180°$ angle (i.e. is interior to a side of a polygon). □

The above lemma can be seen as a necessary condition for a biconnected outerplane graph to have a $t-$TkR. Surprisingly, the simple condition is also sufficient when $k \geq 4$, which we will prove later.

Some definitions are needed before proceeding further. Given a biconnected outer-

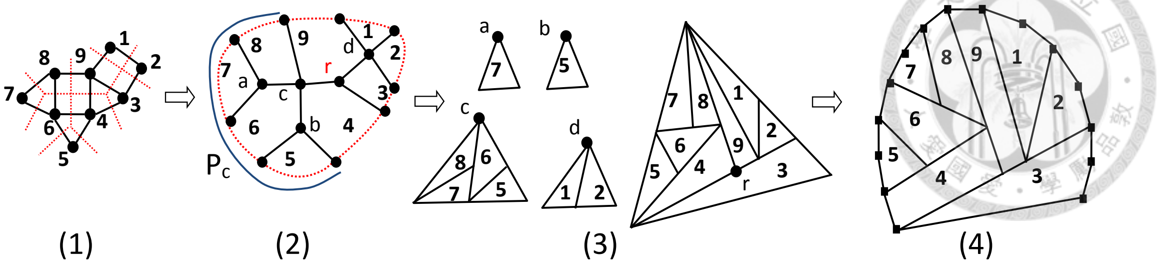


Figure 6.2: The construction of an area-universal t-T4R.

plane graph G, the plane graph G^* (not the dual graph) is defined as the graph resulting from the following operations:

1. Start from the dual graph of G, and let t be the vertex in the dual graph corresponding to $F_O(G)$.

2. Divide t into $|N(t)|$ vertices, each of which is adjacent to a distinct vertex incident in $N(t)$.

3. Link all these $|N(t)|$ vertices together to form a cycle, which is set to the outer cycle of the graph G^*.

As an illustrating example, Fig. 6.2(2) shows the plane graph G^* (the outer cycle is depicted in a dotted-line) associating with the plane graph G depicted in Fig. 6.2(1). The subgraph of G^* that excludes the edges in the outer cycle is called the *skeleton*. For the case G is a biconnected outerplane graph, its skeleton is a tree. It is easy to see that each non-leaf vertex in the skeleton has degree at least 3.

A skeleton can be regarded as a rooted tree by selecting any non-leaf vertex r as its root. For any non-leaf vertex v in the skeleton, we define T_v as the sub-tree rooted at v. We let F_v be the set of faces in G^* such that all their non-boundary edges are contained in $E(T_v)$. For instance, $F_c = \{5, 6, 7, 8\}$ in Fig. 6.2(2). We write P_v to denote the sub-path of $F_O(G^*)$ formed by including all boundary edges contained in some face $F \in F_v$. See Fig. 6.2(2) for P_c.

We are now in a position to prove one of the main results of the chapter:

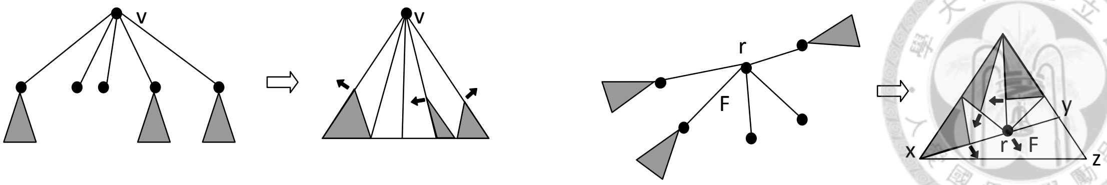


Figure 6.3: Illustration of Procedures 1 and 2.

Theorem 6.1. *Every biconnected outerplane graph admits an area-universal 3-T4R.*

Proof. The basic idea is that G^* can be regarded as a "sketch" of a contact representation of G. All we have to do is to find a drawing of G^* meeting the requirement of the theorem.

The proof is based on a bottom-up approach operating on the skeleton of the input biconnected outerplane graph. When a vertex v in the skeleton is encountered, all vertices in $V(T_v) \setminus \{v\}$ have been processed before. During each iteration, the following invariant is kept:

- For each non-leaf vertex $u \neq r$ that has been processed, the sub-graph G_u of G^* induced by F_u is drawn as an area-universal drawing satisfying:

    1. Each face (in F_u) is either a triangle or a convex quadrangle.
    2. Each non-boundary vertex in $V(T_u) \setminus \{u\}$ is a junction point having an 180° angle in the current sub-drawing.
    3. The outer boundary of the sub-drawing of G_u is a triangle in which u is one of its corner and P_u is one of its sides (hence u is not an 180° corner in any face in F_u).

Let v be the vertex currently being processed. If v is a leaf, we do nothing. If v is non-leaf vertex that is not the root, we do the following:

Procedure 1 (See Fig. 6.3(left))

1. Let $u_1, \ldots, u_s$ be the children of v.

2. For each u_i that is not a leaf, if one of the two faces incident to the edge $\{v, u_i\}$ is not contained in F_v, we make u_i an 180° corner of the face (indicating by an arrow in the illustration). If both faces are contained in F_v, the choice is arbitrary.
3. For each face $F \in F_v \setminus \bigcup_{1 \le i \le s} F_{u_i}$, we contract as many its boundary edges as possible such that F has at least 3 sides in the drawing.
4. Straighten the path P_v.

It is easy to see that the resulting drawing of G_v satisfies the invariant from the illustration (left one of Fig. 6.3). To see that the drawing is area-universal, we first divide each quadrangle into two triangles by adding a straight line linking v to the opposite corner on the boundary of the drawing. Then, if we treat each sub-drawing of G_{u_i} as a single triangle, the drawing of G_v is clearly a one-sided and sliceable 3-T3R (and hence area-universal (Lemma 2.1)).

What remains to be done is the case when the root r is encountered.

Procedure 2 (see Fig. 6.3(right))

1. Let $u_1, \ldots, u_s$ be the children of r.
2. Choose a designated face $F \in (F_r \setminus \bigcup_{1 \le i \le s} F_{u_i})$.
3. Assign r to be an 180° corner of F; remove F from F_r.
4. Do (2),(3) of Procedure 1 (with $v = r$).
5. Subdivide the boundary edge $\{x, y\}$ of F, resulting in two edges $\{x, y\}, \{y, z\}$.
6. Select 3 designated vertices on the boundary cycle such that x, z are selected, y is not selected.

7. Straighten everywhere on the boundary cycle except those 3 selected vertices (making the boundary triangular).

Similar to Procedure 1, it can be easily seen that the resulting drawing after applying Procedure 2 is an area-universal drawing. The outer boundary of the drawing is a triangle. Each inner face is drawn as a triangle or a convex quadrangle. Hence, the theorem holds.

□

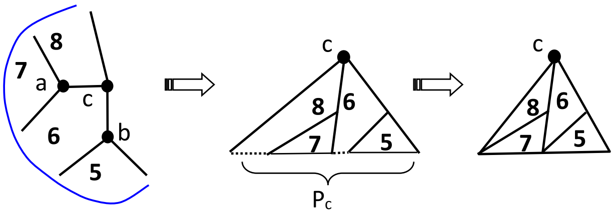


Figure 6.4: Applying Procedure 1 to the subtree rooted at c in Fig.6.2(2).

See Fig. 6.2(1-3) for a full example of the above algorithm, and see Fig. 6.4 for a showcase of applying Procedure 1 for the subtree rooted at c in Fig. 6.2(2).

Note that Theorem 6.1 is tight in the sense that it fails in general when the underlying graph class is changed to either biconnected 2-outerplane graphs or 1-connected outerplane graphs. Also, for biconnected outerplane graphs, in general, 3-sided polygons are not sufficient to construct convex polygonal duals.

Combining the above algorithm and Lemma 6.1, we prove the other main theorem of the chapter which provides a simple necessary and sufficient condition for a biconnected outerplane graph G to admit a $t-$TkR, for $k > 3$:

Theorem 6.2. *For a biconnected outerplane graph G, and for $k > 3$, G admits a $t-TkR$ iff* $3 \leq t \leq (k-1)|V(G)| - |E(G)| + 1$.

Proof. We show only the "if" part as the "only-if" part follows from Lemma 6.1. The case $t = 3$ is a direct result of Theorem 6.1. We observe that in the output drawing of the algorithm in the proof of Theorem 6.1, each non-boundary vertex in G^* is assigned to be a 180° corner for some face. Therefore, for the case of $t > 3$, a desired drawing can be constructed by adding sufficient number of additional corners in the boundary of the output drawing while maintaining $k-$sidedness and convexity for each polygon (in view of Lemma 6.1). It can be achieved by some slight perturbation in the boundary (See Fig. 6.2(4)). Hence the theorem is concluded.

□

(Remark.) We note that we may use FAA and its related terminologies presented in the last chapter to make the proofs in this chapter more formal. However, as we would like the chapter to be more self-contained and the proofs more intuitive, we did not take this approach.

Chapter 7

3D Floorplans

In this chapter, we generalize rectilinear duals to 3D by allowing each object to be an orthogonal polyhedron.

We define the *3D floorplans*, which is the natural generalization of rectilinear duals to 3D, as follows:

Definition 7.1. *A 3D floorplan R of a graph G is a contact representation in which each vertex $v \in V(G)$ corresponds to an axis-aligned genus 0 simple orthogonal polyhedron such that two polyhedra have a surface contact (of non-zero area) iff their corresponding vertices are adjacent in G. Furthermore, polyhedra in R form a partition of a box, and hence the representation has no hole.*

We show that all chordal graphs admit 3D floorplans that use only two layers. Furthermore, such 3D floorplans are volume-universal.

To the best of our knowledge, this is the first attempt to investigate the 3D generalization of rectilinear duals.

7.1 Related Works

For recent related research in the literature, [5] is concerned with proportional contact graph representations using orthogonal polyhedra in low complexity. The main difference between our work and [5] is that in our 3D floorplans, modules associated with vertices of a graph must fill the entire bounding box, whereas in [5], holes and/or empty spaces are allowed as long as the contact of modules respects the adjacency of vertices. From a combinatorial viewpoint, the work of [24] focuses on a bound on the number of a specific type (including general and two-layer mosaic) of 3D floorplans using boxes.

7.2 The Drawing Algorithm

In the two dimensional case, the graph class receiving the most attention in the study of contact graph representations is undoubtedly the class of maximal planar graphs, in which each of its internal faces is a triangle corresponding to a T-junction in the rectilinear dual of the graph. As 3D allows us to capture a richer graph class in terms of contact representations, it is natural to begin with a class of non-planar graphs with some sort of a flavor of a "triangulation". *Chordal graphs* (also named as *triangulated graphs* in some literature) is a good candidate for this purpose. See Section 2.2 for the definition of the chordal graphs.

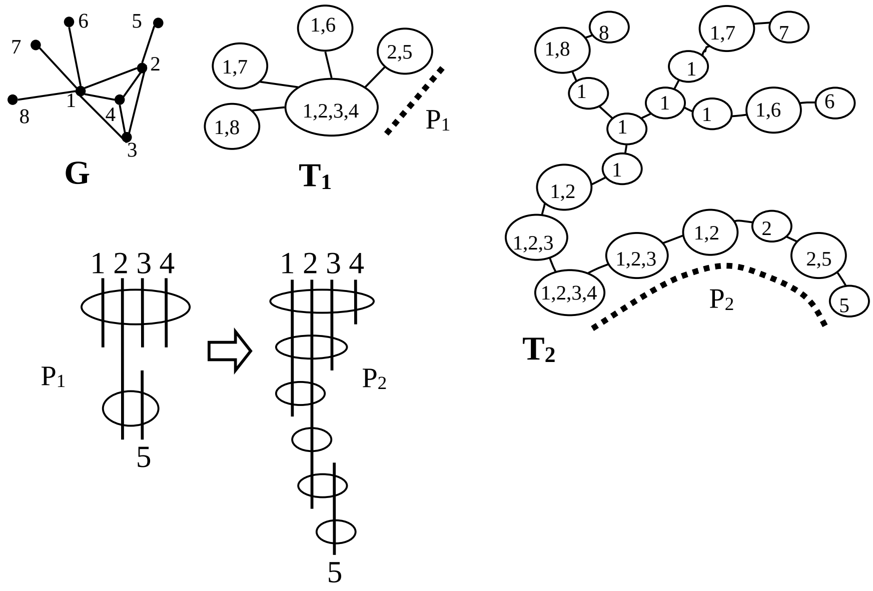


Figure 7.1: A chordal graph G and trees T_1 and T_2.

Chordal graphs can be characterized as intersection graphs of subtrees of a tree. The following result is well-known.

Theorem 7.1. *For a (connected) chordal graph G, we can construct a tree* T_1 *where each* $X \in V(T_1)$ *is a subset of* $V(G)$ *such that the following conditions are met:*

1. *$\forall v \in V(G)$, the set $\{X \in V(T_1) | v \in X\}$ forms a subtree of T_1.*

2. *$\forall v_1, v_2 \in V(G)$, $\{v_1, v_2\} \in E(G)$ iff $\exists X \in V(T_1)$ such that $\{v_1, v_2\} \subseteq X$.*

It is noted that the tree T_1 in the above theorem is also known as a *clique tree* of G, as each node of T_1 corresponds to a maximal clique of G. See graph G and tree T_1 in Fig. 7.1.

To facilitate the construction of a 3D floorplan of a chordal graph, we slightly modify the clique tree in the above theorem to yield a new tree satisfying four additional properties, as shown in the following theorem.

Theorem 7.2. *For a (connected) chordal graph G, we can construct a tree T_2 where each $X \in V(T_2)$ is a subset of $V(G)$ such that the following conditions are met:*

1. *$\forall v \in V(G)$, the set $\{X \in V(T_2) | v \in X\}$ forms a subtree of T_2.*

2. *$\forall v_1, v_2 \in V(G)$, $\{v_1, v_2\} \in E(G)$ iff $\exists X \in V(T_2)$ such that $\{v_1, v_2\} \subseteq X$.*

3. *T_2 is of maximum degree 3.*

4. *If $X \in V(T_2)$ is of degree 3, then $\forall v \in V(G)$, $\forall \{X, X'\} \in E(T_2)$, $v \in X$ iff $v \in X'$.*

5. *If $X \in V(T_2)$ is of degree 1, $|X| = 1$.*

6. *$\forall \{X, X'\} \in E(T_2)$, $|X \setminus X'| \leq 1$.*

To understand the difference between T_1 and T_2, first recall that a chordal graph can be captured as the intersection graph of subtrees of a tree. See Fig. 7.1 for illustrations of T_1 and T_2 satisfying the conditions in Theorem 7.1 and Theorem 7.2, respectively. The vertical line segments in the lower left of Fig. 7.1 shows a portion of the subtrees corresponding to vertices $1, 2, 3, 4, 5$ within the pathes P_1, P_2. For instance, considering the intersection graph of subtrees of T_1, $\{2, 5\}$ is an edge in G since line segments associated with 2 and 5 overlap at some node, as our drawing of P_1 indicates. To meet the four additional conditions in Theorem 7.2, P_2 is obtained from P_1 by stretching some of the line segments of P_1 (while introducing additional nodes in the tree) so that the difference between two adjacent nodes in P_2 is exactly one, and P_2 exhibits exactly the same intersection relation as P_1. Adapting a similar approach for other portion of T_1, a tree (T_2) satisfying Theorem 7.2 can be constructed easily. It is not difficult to see that T_2 in Fig. 7.1 satisfies all the conditions in Theorem 7.2. In view of the above, Theorem 7.2 should be obvious.

In what follows, we show that any chordal graph G admits a 3D floorplan R using only two layers. Intuitively, a 2-layer floorplan is referred to one that can be obtained by gluing two 2D floorplans together. We write $\mathbb{N}, \mathbb{E}, \mathbb{S}, \mathbb{W}, \mathbb{U}$, and $\mathbb{L}$ to denote the northern, eastern, southern, western, upper, and lower faces of the boundary box of R, respectively. Each of them can be seen as a 2D floorplan of some rectilinear polygons corresponding to some vertices in G, where several rectilinear polygons may correspond to the same vertex. We write $R(X), X \in \{\mathbb{N}, \mathbb{E}, \mathbb{S}, \mathbb{W}, \mathbb{U}, \mathbb{L}\}$, to denote such a 2D drawing. We note that a 3D floorplan R using only two layers is completely describable by the two overlapping 2D floorplans $R(\mathbb{U})$ and $R(\mathbb{L})$.

See Fig. 7.2 for an illustration of a 3D floorplan consisting of a box labeled 2 lying on two L-shape polyhedra labeled 1 and 3. $R(X), X \in \{\mathbb{N}, \mathbb{E}, \mathbb{S}, \mathbb{W}, \mathbb{U}, \mathbb{L}\}$, are also depicted in the figure. One can easily observe that the information provided by $R(\mathbb{U})$ and $R(\mathbb{L})$ are sufficient to describe the floorplan completely as it uses only two layers. Hence, to simplify the illustration of our algorithm, instead of giving a 3D drawing of the floorplan, we only draw $R(\mathbb{U})$ and $R(\mathbb{L})$ in our subsequent discussion.

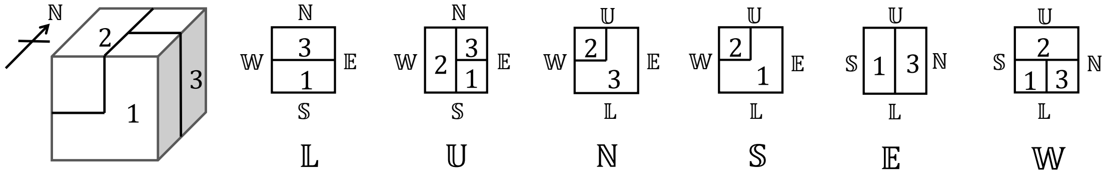


Figure 7.2: Illustration of a 3D floorplan.

Given a chordal graph G, our approach is an inductive construction based on a tree T guaranteed by Theorem 7.2. We make T a rooted tree by designating one of its vertices of degree 1 as the root.

(**Invariant**) For any subtree T' whose root is X, we write G_X to denote the subgraph of G induced by $\{v \in V(G) | \exists X \in V(T'), v \in X\}$. The invariant (i.e., inductive hypothesis) of our construction is described as follows:

Given any designated vertex $v \in X$, there is a 2-layer 3D floorplan $R_{X,v}$ of G_X satisfying the following requirements:

1. Each of $R_{X,v}(\mathbb{E})$ and $R_{X,v}(\mathbb{W})$ contains exactly one rectangle; and the rectangle corresponds to v.

2. If $|X| = 1$, $R_{X,v}(\mathbb{S})$ contains exactly one rectangle; and the rectangle corresponds to v.

3. If $|X| > 1$, $R_{X,v}(\mathbb{S})$ contains exactly $|X| + 1$ rectangles, each of which touches the upper and the lower sides of the outer boundary. The west-most and the east-most rectangles correspond to v. The remaining $|X| - 1$ ones correspond to the $|X| - 1$ vertices in $X \setminus \{v\}$, respectively.

(**Base case**) X is a leaf in T. According to Theorem 7.2, $|X| = 1$, it is immediate that a 3D floorplan containing exactly one box for the unique vertex $v \in X$ works.

(**Induction step**) Suppose X is not a leaf. Since the maximum degree of T is at most 3, and since the root of T is of degree 1, every non-leaf vertex in T contains either one or two children.

We first deal with the situation when X has exactly one child X_1. By induction hypothesis, for any designated vertex $v \in X_1$, there is a drawing $R_{X_1,v}$ of G_{X_1} satisfying the above requirements. If $X = X_1$, taking the drawing $R_{X,v} = R_{X_1,v}$ suffices. Therefore, we assume that $X \neq X'$. In view of Theorem 7.2, we must have either $X_1 = X \cup \{v'\}$ for some $v' \in X_1 \setminus X$ (Case A) or $X = X_1 \cup \{v'\}$ for some $v' \in X \setminus X_1$ (Case B).

Case A: For any $v \in X$, we must have $v \in X_1$. A desired $R_{X,v}$ can be constructed by extending the southern part of $R_{X_1,v}$ with an operation which eliminates v' from

$R_{X,v}(\mathbb{S})$. Adding a bend to a module near the module corresponding to v' suffices to prevent the presence of v' in $R_{X,v}(\mathbb{S})$. See Fig. 7.3 for an illustration of such a removal operation.

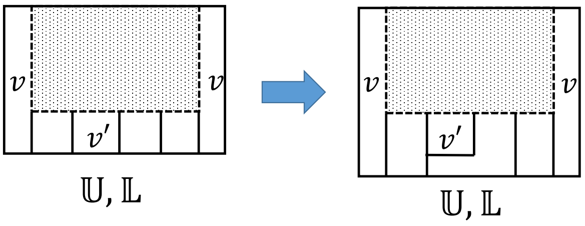


Figure 7.3: Illustration of the removal operation

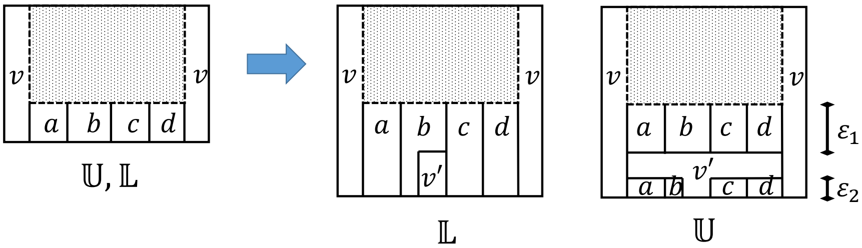


Figure 7.4: Illustration of the insertion operation

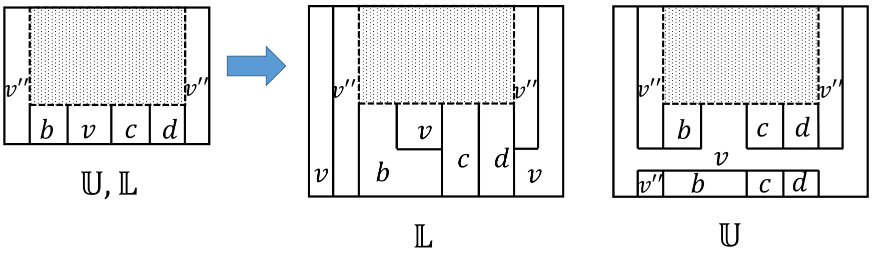


Figure 7.5: Illustration of the operation that changes the outer module

Case B: For any $v \in X$, we must have either $v \in X_1$ or $v = v'$. Similarly, if $v \in X_1$, a desired $R_{X,v}$ can be constructed by extending the southern part of $R_{X_1,v}$ with an operation which introduces the new orthogonal polyhedron corresponding to v' and makes it appears in $R_{X,v}(\mathbb{S})$. Such an insertion operation is illustrated in Fig. 7.4. If $v = v'$, we start with $R_{X_1,v''}$ for any $v'' \in X_1$. After introducing the new orthogonal polyhedron corresponding to v' using the insertion operation in Fig. 7.4, we still need to make the module of $v = v'$ to be the outer one (i.e., occupying the entire $R_{X,v}(\mathbb{E})$ and the entire $R_{X,v}(\mathbb{W})$). The operation to fulfill this task is described in Fig. 7.5. It is easy to verify that the resulting drawing meets all the requirements and that the drawing is a 3D floorplan of G_X (the module corresponding to v' is guaranteed to contact all other modules of vertices in X in the upper layer).

Next, we deal with the remaining situation where X has two children X_1 and X_2 (Case C). In view of Theorem 7.2, we always have $v \in X_1$ and $v \in X_2, \forall v \in X$.

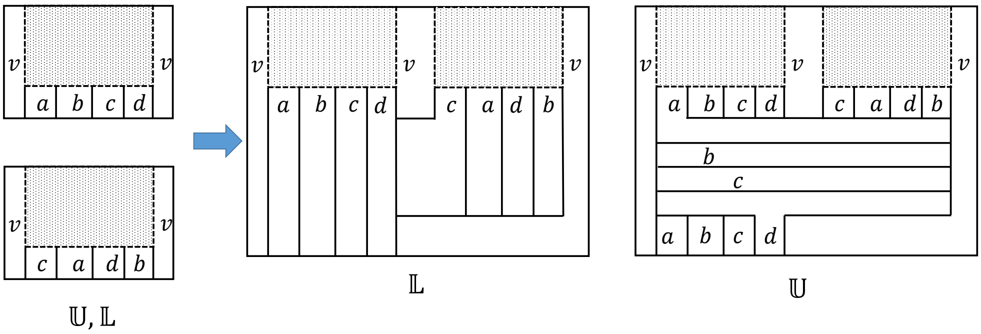


Figure 7.6: Illustration of the merging operation

Case C: For any $v \in X$, a desired $R_{X,v}$ can be constructed by the following procedure. First, we combine $R_{X_1,v}$ and $R_{X_2,v}$ together by gluing $R_{X_1,v}(\mathbb{E})$ with $R_{X_2,v}(\mathbb{W})$. Then, we extend the southern boundary to link the polyhedra in $R_{X_1,v}$ and in $R_{X_2,v}$ corresponding to the same vertex. The detail of the merging operation is described in Fig. 7.6.

By carrying out the inductive construction in a bottom up fashion, the following main theorem of the section is obtained:

Theorem 7.3. *Every (connected) chordal graph G admits a 2-layer 3D floorplan.*

Before ending this section, more is said about our algorithm in the following.

First, one can easily see that the drawing produced by our algorithm is "volume-universal" (i.e., can realize any volume assignment to the objects) by the following reasoning. We make the lower layer to be negligibly thin enough. The added volume introduced by any operations other than the insertion described in Fig. 7.4 is also set to be negligibly small. Whenever a new polyhedron corresponding to a vertex is added to the drawing, the lengths ε_1 and ε_2 (see Fig. 7.4) are made negligibly small. As a result, the insertion of v' is now "equivalent" to adding a box corresponding to v' gluing to the southern face of the current 3D floorplan. These boxes are the only parts of the entire drawing that contributes non-negligible volume. It is easy to see that the volume of such a box can be set arbitrarily during the insertion operation. Hence, we have the following:

Corollary 7.1. *Given a (connected) chordal graph G, there is a 2-layer 3D volume-universal floorplan that respects any weight assignment to vertices in G, in the sense that the volume of an orthogonal polyhedron in the floorplan equals the assigned weight of the associated vertex in G.*

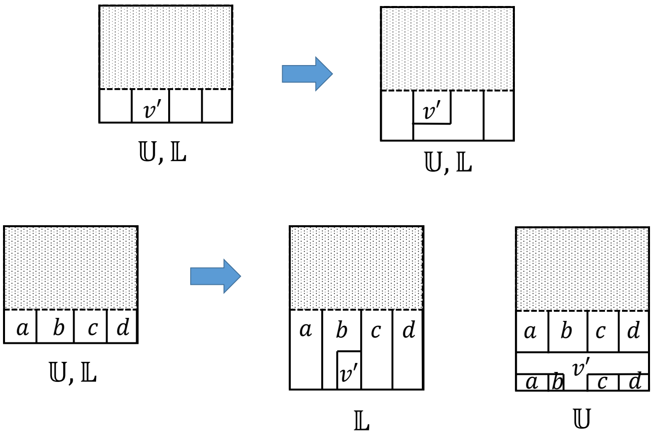


Figure 7.7: Illustration of the simplified operations for interval graphs

Second, if we restrict the graph class to the interval graphs (the intersection graph of subpaths in a path), the drawing resulting from our algorithm has width $O(k)$ (i.e., the number of layers needed in the west-east direction), where k is the size of the maximum clique in G. For an interval graph G, and for its corresponding T is a path, Case C never occurs. As a result, the requirement in the invariant that a designated polyhedron occupying the entire east and the entire west sides is redundant. Therefore, the only operations needed are the insertion (Fig. 7.4) and the removal (Fig. 7.3). In fact, these operations can be simplified when the aforementioned requirement is relaxed. See

Fig. 7.7 for an illustration of the relaxed version of these operations. Since the maximum size of $X \in V(T)$ equals the maximum size of cliques in G, the following result easily follows from our algorithm, with coordinates being properly adjusted and assigned.

Corollary 7.2. *Every interval graph G admits a 2-layer 3D floorplan which fits in a $O(1) \times O(k) \times O(|V(G)|)$ grid, where k is the size of the maximum clique in G.*

Chapter 8

Conclusion and Future Perspectives

At the beginning of the thesis, we studied various shape constraints in rectilinear duals and orthogonal drawings. Next, we moved on to investigate possible extension and generalization of rectilinear duals to other settings. In particular, a few results for convex polygons and for rectilinear polyhedra were presented.

Our study has opened up a few exciting new research directions. In the below we briefly describe some of them, with a few interesting and approachable open problems highlighted.

Orthogonally Convex Drawings. In Chapter 3, our study of orthogonally convex drawing can be seen as the first ever attempt to combine the flow network approach [6, 12,41] with the combinatorial study of orthogonal drawing by Rahman and Nishizeki [35, 36] to yield interesting consequences. We really feel that the result of Rahman and Nishizeki should be useful in improving the current orthogonal drawing algorithms and tackling other variants of orthogonal drawing.

A natural future work direction regarding orthogonally convex drawing is to design polynomial time algorithms for broader graph classes (than biconnected plane 3-graphs) or to prove that it is NP-complete to do so.

Question 8.1. *What is the time complexity of constructing a bend-minimized orthogonally convex drawing for a plane 4-graph?*

Question 8.2. *What is the time complexity of constructing a bend-minimized orthogonally convex drawing for a planar 4-graph?*

For orthogonal drawing, there is already an efficient flow-network based algorithm for plane 4-graphs [12]; however, it is NP-complete to find a bend-minimized orthogonal drawing for planar 4-graphs [23].

Note that the proof of NP-completeness in [23] does not naturally carry over to orthogonally convex drawings, so it would not be surprising if it turns out that bend-minimization of orthogonally convex drawings for planar 4-graphs is polynomial time solvable.

Rectilinear Duals without T-shape. At a high level, the results presented in chapter 4 increase our understanding of the power and limitation of different shapes of polygons in constructing rectilinear duals. Our results are largely related to [4, 43].

A natural direction for future research is to investigate other kinds of previously unstudied restrictions to usable shapes.

We proved that $\top$-free polygons suffice to construct rectilinear duals for maximal plane graphs, and we even showed that the optimal polygonal complexity is 12. However, unlike [4], our result does not carry over to area-universal ones. In fact, the following question is open.

Question 8.3. *Is there a maximal plane graph such that all its area-universal rectilinear duals must contain some $\top$-shape polygon or its extension?*

Convex Polygonal Duals. In chapter 5, we proposed a new approach for tackling a wide range of problems of contact graph representations. In addition to the facilitation of Courcelle's Theorem in the framework of monadic second-order logic to yield some fixed-parameter tractability results, the usefulness of this new technique is further amplified through several short proofs of some interesting existing results. Some intriguing questions and open problems still remain.

In particular, though we presented some fixed-parameter tractability results for convex polygonal duals, the NP-completeness proof for the general problem is still lacking.

Question 8.4. *Given a plane graph G and a number k, is it NP-complete to decide whether G admits a k-sided convex polygonal dual?*

If it turns out that the answer to Question 8.4 is NO, a lot portion of our work will be meaningless. We conjecture that the answer is YES even for some restricted versions.

Other interesting future work directions are listed as follows:

- Is there a general approach to deal with the case when holes are allowed? Also, how about other types of contact styles?
- As the huge constant involved in Courcelle's Theorem makes the FPT algorithm practically unusable, it would be helpful to have a practically usable solution.
- The work of [21] showed that a special subclass of outerplanar graphs enjoys proper touching triangle representations. Is it possible to extend the result to a broader graph class such as the entire class of outerplanar graphs?
- In view of Theorem 5.6, 5.7, it will be interesting to see more results linking rectilinear contact representations to non-rectilinear ones.

Area-universal Drawings of Biconnected Outerplane Graphs. In Chapter 6, we presented some very clean and quick results for convex polygonal duals of outerplane graphs: Theorem 6.1 allows us to construct an area-universal drawing of low complexity, and Theorem 6.2 gives a simple condition for us to examine whether a drawing exists under a certain polygonal complexity requirement.

Though Theorem 6.2 only holds for $k > 3$, the case for $k = 3$ has been solved in [1], where a much different necessary and sufficient condition is given.

Theorem 6.1 implies that all biconnected outerplane graphs admit a 4-sided convex polygonal dual. However, for the case of 3-sided convex polygonal duals, the issue becomes much more complicated.

Question 8.5. *What is the time complexity to decide whether a biconnected outerplane graph admits an area-universal 3-sided convex polygonal dual?*

Though Corollary 5.1 already carry over k-outerplane graphs, it is still of interest to extend Theorem 6.2 to broader graph classes.

Question 8.6. *For an outerplane graph G, and for two positive integers k, t, is it possible to give a simple necessary and sufficient condition to test whether G admits a t−TkR?*

3D Floorplans. In Chapter 7, we showed that all chordal graphs admit a 3D floorplan. However, the construction process seems to require pretty high polygonal complexity. We feel that there is a possibility to design a better algorithm.

Future work along the line of research on 3D floorplans includes minimizing the complexity of the drawing (measured, for instance, in terms of the size of the bounding box or the number of faces/sides/corners in each constituent orthogonal polyhedron). Finding a broader class of graphs admitting 3D floorplans is also of interest.

Note that we can define the polygonal complexity of a 3D floorplan naturally as $\max\{$ the number of edges of $P \mid P$ is a polyhedron corresponding to a vertex in the drawing. $\}$.

Question 8.7. *What is the optimal polygonal complexity for 3D floorplans for chordal graphs?*

Question 8.7 remains interesting even for interval graphs.

In addition to polygonal complexity, the volume of the drawing, which can be measured by the size of the underlying grid or the number of grid planes, is another important quality measure.

In Corollary 7.2, we showed that interval graphs admit a 2-layer 3D floorplan which fits into a $O(1) \times O(k) \times O(|V(G)|)$ grid, where k is the size of the maximum clique in G. In other words, it requires $O(|V(G)|)$ grid planes. We conjecture that it is optimal.

Question 8.8. *Is it possible to construct a 3D floorplan using $o(|V(G)|)$ grid planes for every interval graph G?*

The following research directions are also interesting:

1. improving the constant behind Corollary 7.2; and

2. reducing the number of grid planes in our algorithm for constructing 3D floorplans of chordal graphs.

Bibliography

[1] Nieke Aerts. *Geometric Representations of Planar Graphs*. PhD thesis, Technischen Universität Berlin, 2015.

[2] Nieke Aerts and Stefan Felsner. Straight line triangle representations. In Stephen Wismath and Alexander Wolff, editors, *Graph Drawing*, volume 8242 of *Lecture Notes in Computer Science*, pages 119--130. Springer International Publishing, 2013.

[3] Md. Jawaherul Alam, Therese Biedl, Stefan Felsner, Andreas Gerasch, Michael Kaufmann, and Stephen G. Kobourov. Linear-time algorithms for hole-free rectilinear proportional contact graph representations. *Algorithmica*, 67(1):3--22, 2013.

[4] Md. Jawaherul Alam, Therese Biedl, Stefan Felsner, Michael Kaufmann, Stephen G. Kobourov, and Torsten Ueckerdt. Computing cartograms with optimal complexity. *Discrete & Computational Geometry*, 50(3):784--810, 2013.

[5] Md. Jawaherul Alam, Stephen G. Kobourov, Giuseppe Liotta, Sergey Pupyrev, and Sankar Veeramoni. 3d proportional contact representations of graphs. In *The 5th International Conference on Information, Intelligence, Systems and Applications (IISA 2014)*, pages 27--32, 2014.

[6] Giuseppe Di Battista, Giuseppe Liotta, and Francesco Vargiu. Spirality and optimal orthogonal drawings. *SIAM Journal on Computing*, 27(6):1764--1811, 1998.

[7] Michael A. Bekos, Michael Kaufmann, Robert Krug, Stefan Näher, and Vincenzo Roselli. Slanted orthogonal drawings. In Stephen Wismath and Alexander Wolff, editors, *Graph Drawing*, volume 8242 of *Lecture Notes in Computer Science*, pages 424--435. Springer International Publishing, 2013.

[8] Therese Biedl and Lesvia Elena Ruiz Velázquez. Orthogonal cartograms with few corners per face. In Frank Dehne, John Iacono, and Jörg-Rüdiger Sack, editors, *Algorithms and Data Structures*, volume 6844 of *Lecture Notes in Computer Science*, pages 98--109. Springer Berlin Heidelberg, 2011.

[9] Yi-Jun Chang and Hsu-Chun Yen. On orthogonally convex drawings of plane graphs. In Stephen Wismath and Alexander Wolff, editors, *Graph Drawing*, volume 8242 of *Lecture Notes in Computer Science*, pages 400--411. Springer International Publishing, 2013.

[10] Yi-Jun Chang and Hsu-Chun Yen. Rectilinear duals using monotone staircase polygons. In Zhao Zhang, Lidong Wu, Wen Xu, and Ding-Zhu Du, editors, *Combinatorial Optimization and Applications*, volume 8881 of *Lecture Notes in Computer Science*, pages 86--100. Springer International Publishing, 2014.

[11] Yi-Jun Chang and Hsu-Chun Yen. A new approach for contact graph representations and its applications. To be presented in the 14th Algorithms and Data Structures Symposium (WADS'15), LNCS 9214, 2015.

[12] Sabine Cornelsen and Andreas Karrenbauer. Accelerated bend minimization. In Marc van Kreveld and Bettina Speckmann, editors, *Graph Drawing*, volume 7034 of *Lecture Notes in Computer Science*, pages 111--122. Springer Berlin Heidelberg, 2012.

[13] Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Information and Computation*, 85(1):12--75, 1990.

[14] Bruno Courcelle and Engelfriet Joost. *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*. Cambridge University Press, 2012.

[15] Hubert de Fraysseix and Patrice Ossona de Mendez. Barycentric systems and stretchability. *Discrete Applied Mathematics*, 155(9):1079 -- 1095, 2007.

[16] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Springer-Verlag London, 2013.

[17] Christian A. Duncan, Emden R. Gansner, Yifan Hu, Michael Kaufmann, and Stephen G. Kobourov. Optimal polygonal representation of planar graphs. *Algorithmica*, 63(3):672--691, 2012.

[18] Christian A. Duncan and Michael T. Goodrich. Planar orthogonal and polyline drawing algorithms. In *Handbook of Graph Drawing and Visualization*, chapter 7. CRC Press.

[19] David Eppstein, Elena Mumford, Bettina Speckmann, and Kevin Verbeek. Area-universal and constrained rectangular layouts. *SIAM Journal on Computing*, 41(3): 537--564, 2012.

[20] William Evans, Stefan Felsner, Michael Kaufmann, Stephen G. Kobourov, Debajyoti Mondal, Rahnuma Islam Nishat, and Kevin Verbeek. Table cartograms. In Hans L. Bodlaender and Giuseppe F. Italiano, editors, *Algorithms – ESA 2013*, volume 8125 of *Lecture Notes in Computer Science*, pages 421--432. Springer Berlin Heidelberg, 2013.

[21] J. Joseph Fowler. Strongly-connected outerplanar graphs with proper touching triangle representations. In Stephen Wismath and Alexander Wolff, editors, *Graph Drawing*, volume 8242 of *Lecture Notes in Computer Science*, pages 155--160. Springer International Publishing, 2013.

[22] Emden R. Gansner, Yifan Hu, and Stephen G. Kobourov. On touching triangle graphs. In Ulrik Brandes and Sabine Cornelsen, editors, *Graph Drawing*, volume 6502 of *Lecture Notes in Computer Science*, pages 250--261. Springer Berlin Heidelberg, 2011.

[23] Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM Journal on Computing*, 31(2):601--625, 2001.

[24] Paul Horn and Gabor Lippner. Two layer 3d floor planning. *The Electronic Journal of Combinatorics*, 20(4):P16, 2013.

[25] Bapi Kar, Susmita Sur-Kolay, Sridhar H. Rangarajan, and Chittaranjan R. Mandal. A faster hierarchical balanced bipartitioner for vlsi floorplans using monotone staircase cuts. In Hafizur Rahaman, Sanatan Chattopadhyay, and Santanu Chattopadhyay, editors, *Progress in VLSI Design and Test*, volume 7373 of *Lecture Notes in Computer Science*, pages 327--336. Springer Berlin Heidelberg, 2012.

[26] Akifumi Kawaguchi and Hiroshi Nagamochi. Drawing slicing graphs with face areas. *Theoretical Computer Science*, 410(11):1061--1072, 2009.

[27] Gunnar W. Klau and Petra Mutzel. Quasi–orthogonal drawing of planar graphs. Technical Report MPI-I-98-1-013, Max-Planck-Institut für Informatik, Saarbrücken, 1998.

[28] Stephen G. Kobourov, Debajyoti Mondal, and Rahnuma Islam Nishat. Touching triangle representations for 3-connected planar graphs. In Walter Didimo and Maurizio Patrignani, editors, *Graph Drawing*, volume 7704 of *Lecture Notes in Computer Science*, pages 199--210. Springer Berlin Heidelberg, 2013.

[29] Paul Koebe. Kontaktprobleme der konformen abbildung. *Ber. Verh. Sächs. Akademie der Wissenschaften Leipzig, Math.-Phys.*, Klasse 88:141--164, 1936.

[30] Krzysztof Koźmiński and Edwin Kinnen. Rectangular duals of planar graphs. *Networks*, 15(2):145--157, 1985.

[31] Chien-Chih Liao, Hsueh-I Lu, and Hsu-Chun Yen. Compact floor-planning via orderly spanning trees. *J. Algorithms*, 48(2):441--451, 2003.

[32] Subhashis Majumder, Susmita Sur-Kolay, Bhargab B. Bhattacharya, and Swarup Kumar Das. Hierarchical partitioning of vlsi floorplans by staircases. *ACM Trans. Des. Autom. Electron. Syst.*, 12(1):7:1--7:19, 2007.

[33] Kazuyuki Miura, Hiroki Haga, and Takao Nishizeki. Inner rectangular drawings of plane graphs. *International Journal of Computational Geometry & Applications*, 16(02n03):249--270, 2006.

[34] Takao Nishizeki and Md. Saidur Rahman. *Planar Graph Drawing, Lecture Notes Series on Computing 12*. World Scientific, 2004.

[35] Md. Saidur Rahman, Shinichi Nakano, and Takao Nishizeki. A linear algorithm for bend-optimal orthogonal drawings of triconnected cubic plane graphs. *J. Graph Algorithms Appl*, 3:31--62, 1999.

[36] 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.

[37] Md.Saidur Rahman, Noritsugu Egi, and Takao Nishizeki. No-bend orthogonal drawings of series-parallel graphs. In Patrick Healy and Nikola S. Nikolov, editors, *Graph Drawing*, volume 3843 of *Lecture Notes in Computer Science*, pages 409--420. Springer Berlin Heidelberg, 2006.

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

[39] Jonathan Stott, Peter Rodgers, Juan Carlos Martínez-Ovando, and Stephen G. Walker. Automatic metro map layout using multicriteria optimization. *Visualization and Computer Graphics, IEEE Transactions on*, 17(1):101--114, 2011.

[40] Yachyang Sun and Majid Sarrafzadeh. Floorplanning by graph dualization: L-shaped modules. In *IEEE International Symposium on Circuits and Systems*, volume 4, pages 2845--2848, 1990.

[41] Roberto Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM Journal on Computing*, 16(3):421--444, 1987.

[42] Carsten Thomassen. Plane representations of graphs. In J.A. Bondy and U.S.R. Murty, editors, *Progress in Graph Theory*, pages 43--69. Academic Press, Toronto, 1984.

[43] Torsten Ueckerdt. *Geometric Representations of Graphs with Low Polygonal Complexity*. PhD thesis, Technischen Universität Berlin, 2011.

[44] Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, 2nd edition, 2001.

[45] Kok-Hoo Yeap and Majid Sarrafzadeh. Floor-planning by graph dualization: 2-concave rectilinear modules. *SIAM Journal on Computing*, 22(3):500--526, 1993.