Rectangular Layouts and Contact Graphs

Adam L. Buchsbaum[*]   Emden R. Gansner[*]   Cecilia M. Procopiuc[*]

Suresh Venkatasubramanian[*]

May 6, 2005

Abstract

Contact graphs of isothetic rectangles unify many concepts from applications including VLSI and architectural design, computational geometry, and GIS. Minimizing the area of their corresponding *rectangular layouts* is a key problem. We study the area-optimization problem and show that it is NP-hard to find a minimum-area rectangular layout of a given contact graph. We present $O(n)$-time algorithms that construct $O(n^2)$-area rectangular layouts for general contact graphs and $O(n \log n)$-area rectangular layouts for trees. (For trees, this is an $O(\log n)$-approximation algorithm.) We also present an infinite family of graphs (rsp., trees) that require $\Omega(n^2)$ (rsp., $\Omega(n \log n)$) area.

We derive these results by presenting a new characterization of graphs that admit rectangular layouts using the related concept of *rectangular duals*. A corollary to our results relates the class of graphs that admit rectangular layouts to *rectangle of influence drawings*.

1 Introduction

Given a set of objects in some space, the associated *contact graph* contains a vertex for each object and an edge implied by each pair of objects whose boundaries touch. While contact graphs have been extensively studied for objects such as curves, line segments, and even strings (surveyed by Hliněný and Kratochvíl [15]), as has the more general class of *intersection graphs* (surveyed by Brandstädt, Le, and Spinrad [8] and McKee and McMorris [25]), disks appear to be the only **closed** shape so considered in the literature, the genesis of which is Koebe's Theorem [18], which states that any planar graph (and only a planar graph) can be expressed as the contact graph of disks in the plane.[1] In this paper, we consider contact graphs of isothetic rectangles.

Contact graphs of rectangles find critical applications in areas including VLSI design [21, 38–40, 48], architectural design [36], and, in other formulations, computational geometry [9, 28] and geographic information systems [11]. Previous works considered concepts similar to contact graphs but used a variety of notions like rectangular duality [14, 19, 20, 22] and concepts from graph drawing such as orthogonal, rectilinear, visibility, and proximity layouts [3, 4, 32] to achieve their results. Our work is the first to deal directly with contact graphs of rectangles, yielding a simpler foundation for study.

We call a collection of rectangles that realizes a given contact graph G a *rectangular layout*: a set of disjoint, isothetic rectangles corresponding to the vertices of G, such that two rectangles are adjacent if and only if the implied edge exists in G. The associated optimization problem is to find a rectangular

---

[1] This result was lost and recently rediscovered independently by Andreev and Thurston; Sachs [33] provides a history.

layout minimizing some criterion such as area, width, or height. This problem is inherently intriguing, with many enticing subproblems and variations. It arises in practice in the design of an interface to a relational database system used in AT&T to allow customers to model and administer the equipment and accounts in their telecommunications networks. This system allows users to specify their own schemas using entity-relationship models [30]. The system then presents the database entities as rectangular buttons. Clicking on a button provides information related to the corresponding entity type: e.g., detailed descriptions of the entity attributes or information about specific records. Experience indicates the benefit of juxtaposing buttons that correspond to related entities. Viewing the database schema as the obvious graph, with entities as vertices and relations as edges, leads to rectangular layouts. Solving the related optimization problems would automate this part of tailoring the interface to the schema, of significant benefit as otherwise each customer's interface must be built manually.

We solve several important problems concerning rectangular layouts. We give complete existential and constructive characterizations of graphs that admit rectangular layouts, i.e., graphs that are contact graphs of isothetic rectangles. (Any such graph must be planar, but, unlike with disks, not all planar graphs have rectangular layouts.) We show a suite of results concerning the hardness of finding optimal layouts; design algorithms to construct layouts on graphs, and, with better area bounds, trees; and present some worst-case area lower bounds. We detail these results at the end of this section.

1.1   Relationship to Prior Work

Rectangular layouts are perhaps most closely related to *rectangular duals*, which are like layouts except that a rectangular dual must form a dissection of its enclosing rectangle; i.e., it allows no gaps between rectangles. Rectangular duals have a rich history, including much work on characterizing graphs that admit rectangular duals [14, 19, 20, 22], transforming those that do not by adding new vertices [1, 21], and constructing rectangular duals in linear time [5, 14, 17]. The proscription of gaps, however, severely limits the class of graphs that admit rectangular duals; for example, paths are the only trees that have such duals. In general, any (necessarily planar) graph admitting a rectangular dual must be internally triangulated, but no such restriction applies to layouts. This simple distinction yields many advantages to layouts over duals. The cleaner, less specified definition of layouts characterizes a class of graphs that is both more general (including all trees, for example) and also much simpler to formalize. When we discuss area, we will also show that while different variations of rectangular layouts have different degrees of area monotonicity under graph augmentation, rectangular duals do not enjoy any such monotonicity: a small graph might require a significantly larger dual than a larger graph. Additionally, there are graphs that admit asymptotically smaller rectangular layouts than duals.

Another closely related area concerns VLSI *floorplanning*, in which an initial configuration of rectangles (usually a dissection) is given, the goal being to rearrange and resize the rectangles to minimize the area while preserving some properties of the original layout. While there is an extensive literature on floorplanning, there is much divergence within it as to what criteria must be preserved during minimization. These typically include minimum size constraints on the rectangles plus one of various notions of adjacency equivalence. Stockmeyer [37] presents one of the cleanest definitions of equivalence, preserving the notion of relative placement of rectangles (whether one appears left of, right of, above, or below another); other criteria include the preservation of relative area and aspect ratios [46] and that of relative lengths of abutment [40]. These works are further classified by *sliceability*: a floorplan is *sliceable* if it can be recursively deconstructed by vertical and horizontal lines extending fully across the bounding box. Minimizing the area of non-sliceable floorplans has been shown to be NP-hard under various constraints [23, 29, 37], whereas area minimization of sliceable floorplans is tractable [35, 37, 47]. Not all floorplans can be realized by slice-

able equivalents [38, 39], so work exists on isolating and generating sliceable floorplans where possible [48] as well as minimizing the area of non-sliceable floorplans by various heuristics [29, 40, 46].

Two major facets distinguish floorplanning from area-minimization of rectangular layouts. First, floorplanning seeks to minimize the area of an arrangement of rectangles given a priori by some external process (possibly human design); rectangular layouts themselves are determined by corresponding contact graphs. Second, the notion of equivalence among floorplans differs from context to context, whereas contact graph adjacencies strictly identify the equivalence of rectangular layouts. Thus, there can exist different rectangular layouts of the same graph that do not represent equivalent floorplans, even by Stockmeyer's definition; and conversely there can exist equivalent floorplans that are layouts for non-isomorphic graphs. Still, much work in floorplanning uses concepts from rectangular duals, so work on rectangular layouts can also contribute to this area.

Finally, also related is the idea of *proximity drawings* [4, 16], in which a set of objects corresponds to the vertices of a graph, with edges connecting vertices of correspondingly close objects for some definition of proximity. A particularly relevant special case is that of *rectangle of influence drawings* [24], which are (not necessarily planar) straight-line embeddings of graphs such that the isothetic rectangles induced by pairs of vertices contain no other vertices if and only if the corresponding edges exist. Using results of Biedl, Bretscher, and Meijer [6], we show that graphs that admit rectangular layouts are precisely those that admit a weaker variation of planar rectangle of influence drawings, in which induced rectangles may be empty even if the corresponding edges are missing from the graph; i.e., that contact graphs of rectangles also express this variation of rectangle of influence drawings.

1.2   Our Results

We begin in Section **??** by formally defining rectangular layouts and existentially characterizing the class of graphs that admit them. We also define a relaxed *weak rectangular layout*, in which rectangles can be adjacent even if the implied edge is absent from the graph. While weak and (the default) *strong* rectangular layouts might have different areas, we show they have the same expressive power: that they are admitted by the same class of graphs.

In Section **??**, we constructively prove that a graph admits a rectangular layout and thus is a contact graph of rectangles if and only if it has some planar embedding with no filled three-cycle; i.e., if and only if every three-cycle defines an internal face. This is identical to the characterization of planar, weak, closed rectangle of influence drawings [6].

We discuss area issues in Section 6. We show that constructing an optimal rectangular layout is NP-hard. We also define the notion of area monotonicity under vertex and edge augmentation. (Informally, a graph rendering concept is *vertex* (or *edge*) *monotone* if deleting a vertex (or edge) cannot increase the required area.) We show that weak layouts are vertex and edge monotone, strong layouts are vertex but not edge monotone, and rectangular duals are neither. Finally, we show a worst-case linear gap, which is the maximum possible, between strong layouts and duals: there is an infinite family of n-vertex graphs with $O(n)$-area strong layouts but $\Omega(n^2)$-area optimal duals.

In Section 4, we give $O(n)$-time algorithms that construct $O(n^2)$-area weak and strong layouts for any n-vertex contact graph of rectangles. We also show an infinite class of graphs that require this much area. In Section 5, we give a simpler $O(n)$-time algorithm that constructs $O(n \log n)$-area weak and strong layouts for n-vertex trees; show a matching worst-case lower bound; and show how to construct $O(n)$-area layouts for complete trees.

We conclude in Section 7 with an extensive set of open problems and directions for future work.

3

2   Definitions

Throughout, we assume without loss of generality that all graphs are connected and have at least 4 vertices. For a given graph $G = (V, E)$, define $n = |V|$ and $m = |E|$.

A graph G is *k-connected* if the removal of any set of $k - 1$ vertices leaves the remainder of G connected. A *separating triangle* of G is a 3-vertex cycle whose removal disconnects the remainder of G. For an arbitrary but fixed embedding of a planar graph, a *filled triangle* is defined to be a length-3 cycle with at least one vertex inside the induced region. Note that an embedding of a graph might have a filled triangle but no separating triangle (e.g., any embedding of K_4) and vice-versa. A *cubic graph* is a regular degree-3 graph. A *planar triangulation* is a planar graph in which all faces are bounded by 3-vertex cycles. Whitney's 2-Isomorphism Theorem [45] in fact implies that a planar triangulation has a unique embedding up to stereographic projection, which preserves the facial structure, so it is well-defined to talk about a planar graph G itself being a triangulation (or *triangulated*) as opposed to some specific embedding of G being triangulated. Note also that a planar triangulation G has a planar dual, denoted G^*, which is unique up to isomorphism. Clearly, any 4-connected graph has no separating triangle; for planar triangulations, the converse holds as well.

Lemma 2.1 *A planar triangulation G is 4-connected if and only if G has no separating triangles.*

Koźmiński and Kinnen [19, Lem. 1] state Lemma 2.1 in terms of a fixed embedding, but the above remarks obviate the issue of fixed embeddings for planar triangulations.

A graph G is *cyclically k-edge connected* if the removal of any $k - 1$ edges either leaves G connected or else produces at least one connected component that contains no cycle, i.e., does not break G into multiple connected components all of which contain cycles. Cyclic 4-edge connectivity is a dual concept to 4-connectivity for planar triangulations. In particular, we will use the following result, which appears to be well known; we prove it in Appendix A for completeness.

Lemma 2.2 *A graph G is a 4-connected planar triangulation if and only if G^* is planar, cubic, and cyclically 4-edge connected.*

Define a *rectangular dissection* to be a partition of a rectangle into smaller rectangles, no four of which meet in any point. A *rectangular representation* of a graph G is a rectangular dissection R such that the vertices of G map 1-to-1 to the intersections of line segments of R minus the four external corners of R; i.e., R is a straight-line, isothetic drawing of G except for the four edges that form right angles at the external corners of R. A *rectangular dual* of a graph G is a rectangular dissection, if one exists, whose geometric dual minus the exterior vertex and incident edges is G.

Definition 2.3 (Rectangular Layout) *A (strong) rectangular layout of a graph $G = (V, E)$ is a set R of isothetic rectangles whose interiors are pairwise disjoint, with an isomorphism $\mathcal{R} : V \to R$ such that for any two vertices $u, v \in V$, the boundaries of $\mathcal{R}(u)$ and $\mathcal{R}(v)$ overlap non-trivially if and only if $\{u, v\} \in E$.*

See Fig. 1. The requirement that **non-trivial** boundary overlaps define adjacencies (that rectangles meeting at corners are not considered adjacent) is significant: allowing corner-touching to imply adjacency changes the class of graphs expressed; cf., Section 7. This therefore obviates the specification that the rectangles be isothetic: any collection of rectangles inducing a **connected** contact graph must be isothetic. In contrast, there is no a priori proscription on trivial corner touching in layouts themselves. We shall show, however, that we can exclude such trivial corner touching without loss of generality.
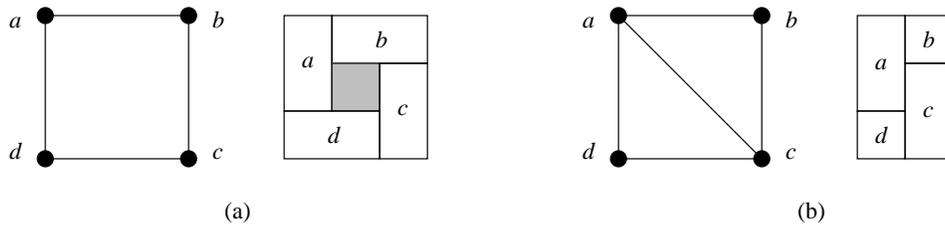
4

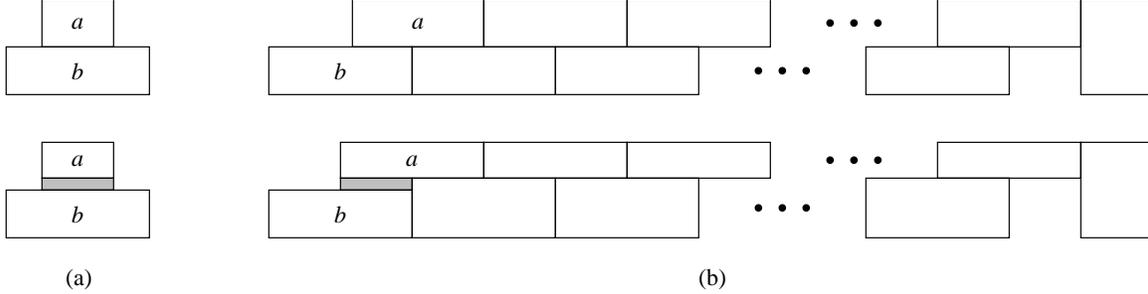Figure 1: Two graphs and associated rectangular layouts.   Shaded regions depict gaps.

Figure 2: Weak layouts (top) and corresponding strong layouts (bottom) after fixing the $\{a, b\}$ violation. (a) The violating boundary of a is nested in that of b. (b) No nesting occurs; shear the projection of the violating boundary until hitting a blocking rectangle. In both cases, there can be other rectangles not shown, but if the new separation ϵ is less than the smallest prior boundary overlap, no other adjacencies are broken.

We also define a relaxed variation: A *weak rectangular layout* of G is a set R of isothetic rectangles whose interiors are pairwise disjoint, with an isomorphism $\mathcal{R} : V \to R$ such that for any edge $\{u, v\} \in E$, the boundaries of $\mathcal{R}(u)$ and $\mathcal{R}(v)$ overlap non-trivially. In a weak layout, two rectangle boundaries may overlap even if the corresponding vertices are not adjacent. For example, in Fig. 1, layout (b) is a weak layout for both graphs (a) and (b) but a strong layout only for graph (b).

Consider a layout drawn on the integral grid. The *area* of the layout is that of the smallest enclosing isothetic bounding box. While corresponding weak and strong layouts may have different areas, from a feasibility standpoint the distinction does not matter, as shown by the following lemma. (Clearly any strong layout is also a weak layout.)

Lemma 2.4 *If G has a weak rectangular layout R, then G has some strong rectangular layout L.*

Proof. Consider any two rectangles a and b in R whose boundaries overlap but such that $\{a, b\} \notin E$. If the overlapping boundary of one (say a) is completely nested in that of the other (b), we can separate the two rectangles by moving the touching boundary of a away from b by some small amount, ϵ, that does not affect any other rectangle adjacency. In particular, let ϵ' be the size of the smallest boundary overlap in R; then let $\epsilon = \epsilon'/2$. (On the integral grid, first scale all coordinates by a factor of 2, and then $\epsilon = 1$.) See Figure 2(a).

Otherwise, neither violating boundary is nested in the other. Assume a is to the right and top of b; other cases are symmetric. (See Figure 2(b).) Consider the projection of the violating boundary rightwards until it hits some other rectangle. Adjust the boundary of all rectangles except b incident to this line segment upwards by ϵ, which fixes the $\{a, b\}$ violation but does not affect any other rectangle adjacency.

Repeating this process to fix each violation produces L. $\qquad\qquad\square$

In the sequel, we will use the term *layout* to refer to strong layouts. Furthermore, the assumption that no two rectangles meet trivially at a corner is without loss of generality. Say rectangles a and b so meet. We can perturb the boundaries by some small amount to make the boundary overlap non-trivial. The layout becomes weak if it was not already. Lemma 2.4 shows that it can be made strong with only non-trivial boundary overlaps.

3 Characterizing Rectangular Layouts

3.1 Background

Thomassen characterizes graphs that admit rectangular layouts (which he calls *strict rectangle graphs*) as precisely the class of proper subgraphs of 4-connected planar triangulations [43, Thm. 2.1]. Together with earlier work [42] his work yields a polynomial-time algorithm for testing a graph G to see if it admits a rectangular layout and, if so, constructing such a layout. He does not analyze the algorithm precisely for running time, however, nor does he bound the layout area at all, two criteria that concern us.

Thomassen's work rests critically on earlier work by Ungar [44]. Ungar defines a *saturated plane map* to be a finite set of non-overlapping regions that partitions the plane and satisfies the following conditions.

1. Precisely one region is infinite.

2. At most three regions meet in any point.

3. Every region is simply connected.

4. The union of any two adjacent regions is simply connected.

5. The intersection of any two regions is either a simple arc or is empty.

An *n-ring* is a set of n regions such that their union is multiply connected.[2] Ungar [44, Thm. A] shows that a saturated plane map that contains no 3-ring is isomorphic to a rectangular dissection. Ungar's Theorem B [44] further implies that for any rectangular dissection R and any two adjacent rectangles a and b in R, there exists a rectangular dissection R' isomorphic to R such that a' (the region in R' corresponding to a in R) is the infinite region and b' (the rectangle in R' corresponding to b in R) has three or four whole sides *fully exposed*: i.e., not overlapping any rectangle other than a'.

We prove the following "folklore" lemma in Appendix A.

Lemma 3.1 *A graph G is planar, cubic, and cyclically 4-edge connected if and only if G is a saturated plane map with no 3-ring.*

Using Lemma 3.1, Thomassen [43] rephrases Ungar's Theorem A as follows.

Lemma 3.2 *Any cubic, cyclically 4-edge connected planar graph has a rectangular representation.*

---

[2]*Connectivity* in this definition and conditions 3–5 above is in the topological sense.

3.2 New Characterization

Koźmiński and Kinnen [19] define a *4-triangulation* to be a 4-connected, planar triangulated graph with at least 6 vertices, at least one of which has degree 4, and they prove [19, Thm. 1] that a cube with one face a rectangular dissection is dual to a planar graph G if and only if G is a 4-triangulation. They also prove [19, Thm. 2] that a planar graph G with all faces triangular except the outside has a rectangular dual if and only if G can be obtained from some 4-triangulation H by the deletion of some degree-4 vertex and all its neighbors. We use the following somewhat weaker result to design an algorithm for constructing rectangular layouts.

Theorem 3.3 *If a planar graph G can be derived from some 4-connected planar triangulation H by the removal of some vertex and its incident edges, then G has a rectangular dual.*

Proof. Let H be any 4-connected planar triangulation and v any vertex of H. By Lemma 2.2, H^* is planar, cubic, and cyclically 4-edge connected, and so by Lemma 3.1 H^* yields a saturated plane map M with no 3-ring. Ungar's Theorem B [44] implies that an isomorphic map M' exists with the external face corresponding to v. M' is thus a rectangular dual for the graph derived from H by removing v and its incident edges. $\square$

The benefit of Theorem 3.3 is that it gives a sufficient condition for rectangular duality in terms of 4-connected triangulations rather than Koźmiński and Kinnen's [19] 4-triangulations. This yields (in Section 4) a simple augmentation procedure for constructing rectangular layouts based on the following alternative characterization of graphs admitting rectangular layouts.

Theorem 3.4 *A planar graph G is a proper subgraph of a 4-connected planar triangulation if and only if G has an embedding with no filled triangles.*

Proof. Let H be a 4-connected planar triangulation. Let H' be the result of removing any one edge from H. By Lemma 2.1, H, and therefore H', has no separating triangle. Any embedding E' of H' has a non-triangular face; by stereographic projection, we can assume that it is the external face. Because any filled triangle is either a separating triangle or the external face, it follows that E' has no filled triangles, and therefore any proper subgraph of H' also has an embedding with no filled triangles.

Now let G be a planar graph, and let E be some embedding of G with no filled triangles. Assume without loss of generality that G has at least one non-triangular face. Otherwise, G itself is a triangulation, and the assumption that E has no filled triangles implies that G is simply a 3-vertex cycle. We will show how to form by vertex augmentation a proper supergraph G' of G such that G' is a planar triangulation with no separating triangles and hence by Lemma 2.1 is 4-connected.

First, we may assume that G is biconnected. Otherwise, we adapt a procedure attributed to Read [31]. Consider any articulation vertex v, and let u and w be consecutive neighbors of v in separate biconnected components. Add new vertex z and edges $\{z, u\}$ and $\{z, w\}$. Iterating for every articulation point biconnects G without adding separating triangles. Any face in the updated embedding E is then bounded by a simple cycle, and the following procedure is well defined.

Consider any non-triangular facial cycle F in E. Define a *chord* of F to be a non-facial edge connecting two vertices of F. Consider any chord $\{x, y\}$ of F, and let u and v be the neighbors of x on F. There can be no edge $\{u, v\}$ in G, for such an edge would violate planarity. Therefore embedding a new vertex $\nu(x)$ inside F and adding edges $\{\nu(x), u\}$, $\{\nu(x), x\}$, and $\{\nu(x), v\}$ cannot create a separating triangle. Let F' be the new facial cycle defined by replacing the path (u, x, v) in F by $(u, \nu(x), v)$, and iterate until F' has no incident chords. Then, adding a final new vertex $\nu(F)$ with edges to each vertex on F'

completes the triangulation of the original face F without creating separating triangles or modifying other faces. Iterating for all non-triangular facial cycles completes the process, yielding a planar triangulation G' with no additional separating triangles.

Therefore, any separating triangle T in G' must have originally existed in G. Because E had no filled triangles, T must have been embedded as a (triangular) face of E. T remains a face in G', however, and because G' is a triangulation, the removal of any face cannot disconnect G', thereby contradicting the existence of T. □

Corollary 3.5 *A graph G has a rectangular layout if and only if there exists some embedding of G with no filled triangles.*

Proof. This follows from Thomassen's characterizations of graphs that admit rectangular layouts [43, Thm. 2.1] and Theorem 3.4. □

3.3 Rectangle of Influence Drawings

Finally, we link rectangular layouts to another graph visualization technique: rectangle of influence drawings. A *(strong) closed rectangle of influence drawing* of a graph G is a straight-line embedding of G such that the isothetic rectangular region, including the border, induced by any two vertices u and v contains no other vertices if and only if $\{u, v\}$ is an edge in G. A *weak closed rectangle of influence drawing* relaxes the condition so that the isothetic rectangular region, including the border, induced by any two vertices u and v contains no other vertices if $\{u, v\}$ is an edge in G. A *(strong or weak) open rectangle of influence drawing* is one in which all the isothetic rectangular interiors obey the respective emptyness constraints. These drawings are also *planar* if no two edges cross.

Theorem 3.6 *A graph G is a contact graph of rectangles and thus admits a rectangular layout if and only if G has a planar, weak, closed rectangle of influence drawing.*

Proof. This follows from Corollary 3.5 and Theorem 2 of Biedl, Bretscher, and Meijer [6]. □

Rectangular layouts express the same class of graphs under either weak or strong adjacency constraints, but the same is not true of rectangle of influence drawings. For example, a star on three (rsp., five) leaves has a planar, weak, open (rsp., closed) rectangle of influence drawing but no strong, open (rsp., closed) rectangle of influence drawing. Liotta et al. [24] characterize graphs with strong rectangle of influence drawings. This settles an open problem raised by Biedl, Bretscher, and Meijer [6].

4 Layouts for General Graphs

Theorems 3.3 and 3.4 suggest an algorithm for constructing a rectangular layout for an arbitrary input graph G.

1. Construct an embedding E of G with no filled triangles. If no such embedding exists, then G admits no rectangular layout.

2. Vertex-augment E to create a proper supergraph G' of G such that G' is a 4-connected triangulation.

3. Construct a rectangular dual R of $G'' = G' - \{v\}$, where v is any vertex added during the augmentation process in step 2.

4. Replace each rectangle r in R that corresponds to a vertex added during step 2 by a gap. The result is a rectangular layout for G.

Theorem 4.1 *An $O(n^2)$-area rectangular layout can be built in $O(n)$ time for any contact graph G of rectangles. If G is not a contact graph of rectangles, this can be discovered in $O(n)$ time.*

Proof. Biedl, Kant, and Kaufmann [7, Thm. 5.5] show how to construct an embedding E of G with no filled triangles if one exists, or detect if no such embedding exists, both in $O(n)$ time.

The proof of Theorem 3.4 outlines a procedure to effect step 2. Finding articulation points can be done in $O(n)$ time by depth-first search [2]. Representing E by a standard doubly connected edge list [26] then allows all operations to be implemented in $O(n)$ time overall. In particular, after augmenting to assure biconnectivity, iterating over the faces of E takes $O(n)$ time plus the time to process each face. Iterating over the vertices of all the faces takes $O(n)$ time plus the time to process each vertex. Processing each vertex x on each face F involves checking each incident edge e to see if e is a chord of F; each such test takes $O(1)$ time, and each edge in G is checked twice, once for each endpoint, for a total of $O(n)$ time. If e is a chord, augmenting F to replace x with $\nu(x)$ also takes $O(1)$ time. Adding vertex $\nu(F)$ takes time linear in the number of vertices on F; over all faces this is $O(n)$ time. In all, step 2 can be done in $O(n)$ time, yielding graph $G'' \supseteq G$ with $O(n)$ vertices.

Theorem 3.3 asserts that G'' has a rectangular dual. He [14] shows how to construct an $O(n^2)$-area rectangular dual of G'' in $O(n)$ time.[3] During the construction, we simply indicate that any rectangle corresponding to a vertex in $G'' - G$ should instead be rendered as a gap. Since each edge in $G'' - G$ is incident to at least one vertex in $G'' - G$, the result is a rectangular layout for G. □

4.1 General Lower Bound

To show a matching worst-cast lower bound, first define an *n-rung ladder* to be a graph on at least $n + 2$ vertices—L, R, and x_i for $1 \leq i \leq n$—with edges $\{L, x_i\}$ and $\{x_i, R\}$ for $1 \leq i \leq n$ and paths (possibly including additional vertices) connecting x_i to x_{i+1} for $1 \leq i < n$. We call the x_i's the *rungs* and L and R the *struts* of the ladder.

In a rectangular layout, call some rectangle r *above* some rectangle s if the lowest extent of r is no lower than the highest extent of s. Symmetrically define *below*, *right of*, and *left of*. Call a set of rectangles *vertically* (rsp., *horizontally*) *stacked* if their above (rsp., left of) relationships form a total order. A set of rectangles is *vertically* (rsp., *horizontally*) *aligned* if they have pairwise identical projections onto the x-axis (rsp., y-axis). We use the *length* of a rectangle to mean the maximum of its width and height.

Lemma 4.2 *Assume $n \geq 3$. Any rectangular layout for an n-rung ladder has width (or, rsp., height) at least n; rectangles x_i for $1 \leq i \leq n$ all horizontally (or, rsp., all vertically) stacked; and rectangles x_i for $1 < i < n$ all horizontally (or, rsp., all vertically) aligned between L and R.*

Proof. By inspection. See Figure 3. □

Define an (i, j)-*ladder* to be a graph on $i + j + 2$ vertices: an i-rung (*external*) ladder defined by some vertices L, R, and x_k for $1 \leq k \leq i$, united with a j-rung (*internal*) ladder defined by $x_{\lfloor i/2 \rfloor}$, $x_{\lfloor i/2 \rfloor+1}$, and y_k for $1 \leq k \leq j$. See Figure 4.

Theorem 4.3 *For $n \geq 4$, any layout for an (n, n)-ladder has area $\Omega(n^2)$.*

---

[3]He does not explicitly state the area of the dual resulting from his construction, but the bound is easily derived.
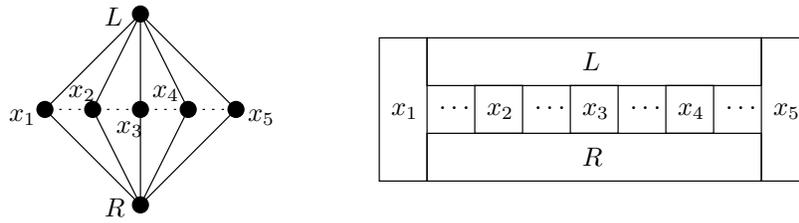
Figure 3: A 5-rung ladder (dotted lines indicate paths) and a possible layout. In any layout, all the x_i must be stacked, and the rectangles x_2, x_3, and x_4 must be aligned as shown between L and R. The only other variations also align rectangle(s) x_1 and/or x_5, but the width remains at least 5.
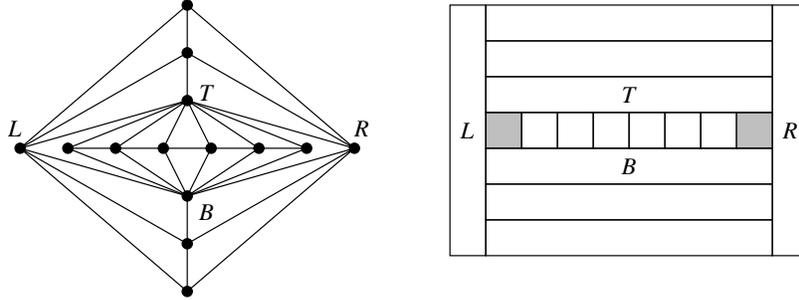

Figure 4: A $(6, 6)$-ladder and a possible layout.

Proof. Lemma 4.2 applied to the external ladder shows that the height (or, rsp., width) of any layout is at least n and that the rectangles $x_{\lfloor n/2 \rfloor}$ and $x_{\lfloor n/2 \rfloor+1}$ are vertically (or, rsp., horizontally) aligned between L and R. Say the height is n; the other case is symmetric. Then Lemma 4.2 shows that the width induced by the internal ladder is at least n. The theorem follows.   □

5   Layouts for Trees

We present an algorithm that constructs $O(n \log n)$-area rectangular layouts for trees. We then show a matching worst-case lower bound. A trivial lower bound is $\max \left\{ n, \sum_{v \in V} \left\lceil \frac{deg(v)}{4} \right\rceil, \sum_{v \in V} \left\lfloor \frac{deg(v)-2}{2} \right\rfloor \right\}$, where $deg(v)$ is the degree of vertex v. There do exist trees with better layouts, however, and we constructively show an infinite class of trees that have $O(n)$-area layouts. As with general graphs, this leaves open the problem of devising better approximation algorithms for trees.

5.1   General Algorithm

Given an undirected tree, $T = (V, E)$, assume T is rooted at some vertex r; if not, pick an arbitrary root. A simple Algorithm A lays out T as follows. For all $v \in V$, let $desc(v)$ be the number of descendents of v. Each vertex v is represented as a rectangle of height 1 and width $desc(v)$. For any v, the rectangles for its children are placed under the rectangle for v. The rectangle for the root appears on top. For simplicity, we allow corners to meet trivially; with our construction, we can eliminate this problem at a constant-factor area penalty. Similarly, we ignore the issue of strong versus weak layouts.

Lemma 5.1 *Algorithm A produces a layout of T of area $n \cdot depth(T)$.*

Proof. The assertion about area is straightforward. Correctness follows from the fact that each rectangle is wide enough to touch all the rectangles for its vertex's children. □

Consider a partition of T into a collection of vertex-disjoint paths. For a path $P = (u_0, \ldots, u_k)$, in top-down order, define $t(P) = u_0$, i.e., the vertex in P of maximum height. Define $desc(P) = desc(t(P))$. Define $s(u_k) = desc(u_k)$, and for $0 \leq i < k$, define $s(u_i) = desc(u_i) - desc(u_{i+1})$. Note that $\sum_{x \in P} s(x) = desc(P)$. Algorithm A generalizes into Algorithm B as follows.

Each path $P = (u_0, \ldots, u_k)$ is represented as a rectangle of height 1 and width $desc(P)$, which is partitioned into rectangles of width $s(u_0), \ldots, s(u_k)$, each representing the corresponding $u_i \in P$. The rectangle for path P with $t(P) = r$ is placed on top. Under the rectangle for each $u_i \in P$ (for each P) are placed the rectangles for each path P' such that $t(P')$ is a child of u_i.

Lemma 5.2 *Algorithm B produces a layout of T.*

Proof. Consider any vertex x. We need only account for the adjacency $(x, p(x))$ (assuming $x \neq r$), because any child u of x is accounted for by the adjacency $(u, p(u))$.

Denote by $B(x)$ the rectangle representing x. If $x \neq r$, then $B(x)$ overlaps $B(p(x))$, for either x and $p(x)$ are in a common path, in which case their rectangles share a vertical side, or else $B(x)$ is layed out underneath $B(p(x))$. The construction assures that $B(p(x))$ is wide enough in this latter case. □

Consider the *compressed tree* $C(T)$, formed by compressing each path P into a super-vertex, with edges to each super-vertex P' such that $t(P')$ is a child of some $x \in P$.

Lemma 5.3 *Algorithm B produces a layout of T of area $n \cdot depth(C(T))$.*

Proof. Algorithm B produces a one-unit high collection of rectangles for each distinct depth in $C(T)$. Each such collection is of width no more than n (the total number of descendents of the paths at that depth). □

We use the heavy-path partition of T, as defined by Harel and Tarjan [13] and later used by Gabow [10].[4] Call tree edge $(v, p(v))$ *light* if $2 \cdot desc(v) \leq desc(p(v))$, and *heavy* otherwise. Since a heavy edge must carry more than half the descendents of a vertex, each vertex can have at most one heavy edge to a child, and therefore deletion of the light edges produces a collection of vertex-disjoint *heavy paths*. (A vertex with no incident heavy edges becomes a singleton, called a *trivial heavy path*.)

Theorem 5.4 *Algorithm B applied to the heavy-path partition of T produces a layout of area $O(n \log n)$ in $O(n)$ time.*

Proof. The compressed tree, $C(T)$, is constructed by contracting each heavy path in T into a single super-vertex. Each tree edge in $C(T)$ corresponds to a light edge of T. Since there are $O(\log n)$ light edges on the path from any vertex to the root of T, $C(T)$ has depth $O(\log n)$. The area bound follows from Lemmas 5.2 and 5.3. $C(T)$ can be built in $O(n)$ time after a depth-first search; the rest of the algorithm performs $O(1)$ work per vertex. □

---

[4]Tarjan [41] originally introduced heavy-path partitions, but defined in different terms; Schieber and Vishkin [34] later used yet another variant.
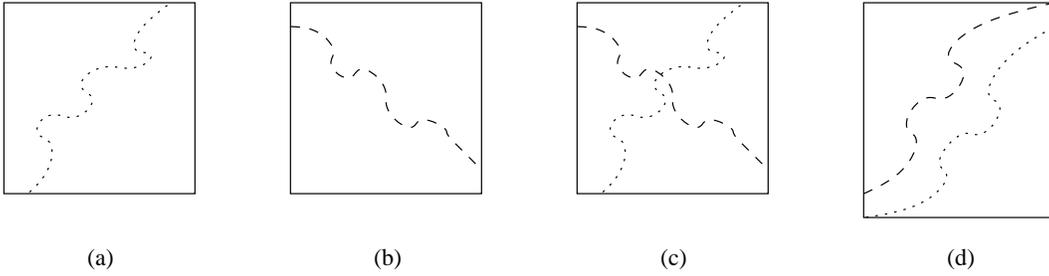
Figure 5: Bounding boxes and extremal paths for various layouts. (a) Layout L_1 for G_1; (b) Layout L_2 for G_2; each of size d-by-d. (c) Impossible d-by-d layout for G containing G_1 and G_2 as subgraphs; any corresponding extremal paths would have to cross. (d) d-by-$(d+1)$ layout for for G. The extremal path for L_1 is dotted, and that for L_2 is dashed. By extending G in one dimension, the extremal paths need not cross.

5.2 General Lower Bound

We show there exists an infinite family of trees that require $\Omega(n \log n)$ area for any layout. First we show that any layout of a binary tree has $\Omega(\log n)$ length in each dimension.

We define the notion of paths in layouts. A *path* in layout L is a sequence $(r_1, \ldots, r_\ell)$ of rectangles in L such that for each $1 \le i < \ell$, the boundaries of r_i and r_{i+1} overlap. A path in a strong layout thus corresponds to a path in the underlying graph. A *vertical extremal path* of L is a path that touches both the top and bottom of L's bounding box B; similarly, define a *horizontal extremal path* to touch the left and right sides of B. An extremal path that touches opposite corners of B is both vertical and horizontal. By definition, every layout has at least one vertical and at least one horizontal extremal path, possibly identical.

Consider graph G with some layout L and some subgraph G'. L contains a sub-layout L' for G'. Any extremal path in L' induces a path in L. Two sub-layouts are *disjoint* if their induced subgraphs are disjoint. Extremal paths for disjoint sub-layouts may not cross in L, for this would imply two non-disjoint rectangles, a fact codified as follows.

Fact 5.5 *Consider graph G, some layout L of G, and any two disjoint, connected subgraphs G_1 and G_2 of G. Extremal paths for the corresponding sub-layouts L_1 and L_2 may not cross in L.*

Lemma 5.6 *Let G_1 and G_2 have minimal area layouts with length at least d in each dimension. Let G contain both G_1 and G_2 as subgraphs. Then any layout L of G has length at least $d+1$ in at least one dimension.*

Proof. Since L contains sub-layouts L_1 and L_2 for G_1 and G_2, rsp., by assumption L has length at least d in both dimensions. If L is d-by-d, then any horizontal extremal path for L_1 must cross any vertical extremal path for L_2, contradicting Fact 5.5. Thus, L must have width or height at least $d+1$. (See Figure 5.) □

Lemma 5.7 *Let L_1 and L_2 be layouts for graphs G_1 and G_2, each of minimum dimension d. Let G be formed by adding a new vertex r, adjacent to one vertex in each of G_1 and G_2. Then in any layout L of G with minimum dimension d, r cannot be incident to a length-d side of the bounding box of L. Furthermore there exist extremal paths in the length-d dimension to either side of r.*

Proof. (Refer to Figure 6(a).) Assume to the contrary that L has width d and r is adjacent to the bottom (sym., top) of L; the argument for height d is symmetric. Let r be adjacent to a rectangle ℓ of L_1 (sym.,
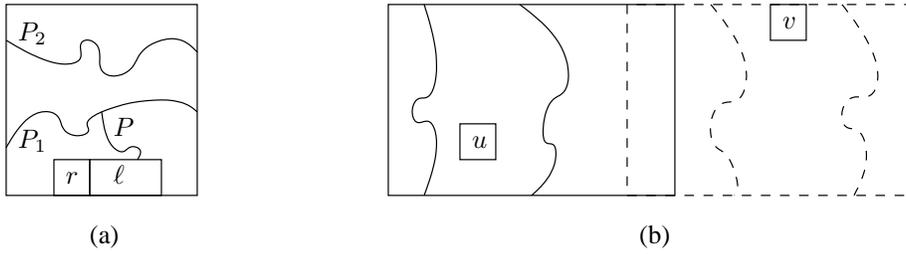
Figure 6: (a) A bounding box of width d containing a layout L that includes two sub-layouts L_1 and L_2, each of minimum dimension d. If r is adjacent to the bottom of L, let ℓ be a rectangle in L_1 adjacent to r. There is some path P in L_1 connecting ℓ to a horizontal extremal path P_1 of L_1 and L. No horizontal extremal path of L_2 can intersect the closed curves formed by P, P_1, and the bounding box of L. Thus P_2 is above P_1, but then r cannot be connected to P_2 without crossing P_1. (b) Bounding boxes for layouts of two T_{i-1}'s, rooted at u (solid) and v (dashed). If both layouts are of height $\lfloor \frac{i-2}{2} \rfloor$, then extremal paths (dashed for the T_{i-1} rooted at v) separate u from v, and r cannot be placed within this height.

L_2). Then there must be a path P from ℓ that intersects a horizontal extremal path P_1 of both L_1 and L. The union of P and one side of this extremal path forms a closed curve with the bounding box that contains r.

Now consider a horizontal extremal path P_2 of L_2, which is also a horizontal extremal path of L. There must be a path P' in L_2 connecting P_2 to r. P_2 cannot intersect the closed curve defined above without creating non-disjoint rectangles; thus P_2 is above P_1. But then by the Jordan curve theorem [27, Section 8-13], P' itself must cross the curve to reach P_2, which again would create non-disjoint rectangles.

If r is above P_2, a similar contradiction holds. Thus, r must be between P_1 and P_2. $\qquad \square$

Define T_i to be a complete binary tree on 2^i vertices.

Lemma 5.8 *Any layout for T_i has length at least $\lfloor i/2 \rfloor$ in each dimension.*

Proof. The theorem is true for $i = 0$ and $i = 1$. Assuming it is true up to $i - 1$, we prove it by induction for $i \geq 2$. Denote by r the root of T_i and by u and v the roots of the T_{i-1}'s rooted at the children of r. By induction, each T_{i-2} rooted at children of u and v has minimum dimension at least $\lfloor \frac{i-2}{2} \rfloor$. By Lemma 5.6 therefore, the layouts for the T_{i-1} subtrees rooted at u and v each have at least one dimension of length $\lfloor \frac{i-2}{2} \rfloor + 1 = \lfloor i/2 \rfloor$. If either sub-layout has both dimensions this large, we are done. If one sub-layout has width $\lfloor i/2 \rfloor$ and the other height $\lfloor i/2 \rfloor$, we are similarly done.

Therefore, assume the sub-layouts for both T_{i-1}'s have height $\lfloor \frac{i-2}{2} \rfloor$. (Symmetrically argue if both widths are this small.) Also assume that in the layout for T_i, r is not placed on top of the two sub-layouts, or the height grows by the required one unit. By Lemma 5.7, neither u nor v may be adjacent to the left or right side of their layouts, and furthermore, there are vertical extremal paths due to their own children that separate u from v. But then by Jordan curve arguments, there cannot be paths connecting both u and v to r without violating rectangle disjointedness. (See Figure 6(b).) $\qquad \square$

Assume $n = 2^k$ for some integer $k \geq 1$. Let S_n be a tree formed by linking the root of T_k to the root of a star with n leaves.

Theorem 5.9 *Any layout L_n for S_n has $\Omega(n \log n)$ area.*

Proof. L_n includes sub-layouts for T_k and the star on n leaves. Lemma 5.8 implies that the length of each dimension of L_n is $\Omega(\log n)$. The only layout for a star with n leaves is a rectangle for the root with
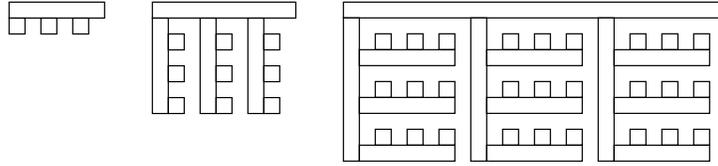
13

Figure 7: Layouts L_3^3 (left), L_9^3 (middle), and L_{27}^3 (right).

n rectangles around its perimeter. Thus, at least one dimension of the star layout is of length $\Omega(n)$. The theorem follows. □

5.3 Linear Area for Complete Trees

Let T_n^k be a complete tree of arity k on n leaves and L_n^k a corresponding strong layout. L_1^k is the unit square. For higher n, construct L_n^k as follows. Recursively construct the k $L_{n/k}^k$ sub-layouts, assuming the root of each one has unit length in one dimension. Attach the roots of the sub-layouts by their unit-length sides to a root rectangle of height 1 and appropriate width, leaving one unit of horizontal space to the right of each sub-layout. See Figure 7.

Theorem 5.10 L_n^k is of area $O(n)$.

Proof. For clarity, we drop the superscripts. Denote by h_n the height and w_n the width of L_n. Then $h_1 = w_1 = 1$; $h_k = 2$, $w_k = 2k$; and for higher n, we have the following *type-1 recurrence*:

$$
\begin{aligned}
h_n &= 1 + w_{n/k}; \\
w_n &= kh_{n/k} + k.
\end{aligned}
$$

Simplifying:

$$
\begin{aligned}
h_n &= kh_{n/k^2} + k + 1; \\
w_n &= kw_{n/k^2} + 2k.
\end{aligned}
$$

Solving these recurrences for $n = k^c$ yields

$$
h_n = \Theta\left(k^{\lfloor c/2 \rfloor}\right); w_n = \Theta\left(k^{\lceil c/2 \rceil}\right).
$$

Therefore, the area, which is $h_n w_n$, is $\Theta(n)$. □

An alternative layout yields a family of layouts of linear area, but with elastic widths and heights. Rather than attaching the roots of the sub-layouts to the root of the layout by their unit-length sides, consider attaching them by their other sides. This establishes a recurrence of the form

$$
\begin{aligned}
h_n &= 1 + h_{n/k}; \\
w_n &= kw_{n/k} + k;
\end{aligned}
$$

which yields the solution $h_n = \Theta(\log_k n)$, $w_n = \Theta(n)$. We call this a *type-2 recurrence*.

Combining the above two recurrences, we can prove the following.

Theorem 5.11 *For any constant $\alpha \in [\frac{1}{2}, 1)$, a layout for T_n^k exists with width $\Theta(n^\alpha)$ and height $\Theta(n^{1-\alpha})$.*

14

Proof. If we apply a type-1 recurrence $2m$ times, we obtain the recurrence

$$
\begin{aligned}
h_n &= (1+k)(1+k+\cdots+k^{m-1}) + k^m h_{n/k^{2m}}; \\
w_n &= 2k(1+k+\cdots+k^{m-1}) + k^m w_{n/k^{2m}};
\end{aligned}
$$

which simplified yields

$$
\begin{aligned}
h_n &= \Theta(k^m) + k^m h_{n/k^{2m}}; \\
w_n &= \Theta(k^m) + k^m w_{n/k^{2m}}.
\end{aligned}
$$

Similarly, a type-2 recurrence applied 2ℓ times yields

$$
\begin{aligned}
h_n &= 2\ell + h_{n/k^{2\ell}}; \\
w_n &= k^{2\ell} + k^{2\ell} w_{n/k^{2\ell}}.
\end{aligned}
$$

Consider a layout in which we apply a type-1 construction $2m$ times and then a type-2 construction 2ℓ times and then repeat. The recurrence governing this construction is given by a combination of the above two recurrences:

$$
\begin{aligned}
h_n &= \Theta(k^m) + k^m(2\ell + h_{n/k^{(2\ell+2m)}}); \\
w_n &= \Theta(k^m) + k^m(k^{2\ell} + k^{2\ell} w_{n/k^{(2\ell+2m)}});
\end{aligned}
$$

which simplifies to

$$
\begin{aligned}
h_n &= \Theta(k^m \ell) + k^m h_{n/k^{(2\ell+2m)}}; \\
w_n &= \Theta(k^{m+2\ell}) + k^{m+2\ell} w_{n/k^{(2\ell+2m)}}.
\end{aligned}
$$

Solving these recurrences and setting $\alpha = 1 - m/(2m+2\ell)$, we get $h_n = \Theta(n^{1-\alpha}(1+\ell))$ and $w_n = \Theta(n^\alpha)$. For any $1/2 \le \alpha < 1$ we can find constants ℓ and m that satisfy this equation. $\square$

6 Area Optimality

6.1 NP-Hardness of Generating Optimal Layouts

Recall the problem of *numerical matching with target sums* [12]. Given are disjoint sets X and Y, each of m elements, a size $s(a) \in \mathbb{Z}^+$ for each $a \in X \cup Y$, and a target vector $B = (B_1, \ldots, B_m)$ with each $B_i \in \mathbb{Z}^+$. The problem is to determine if $X \cup Y$ can be partitioned into m disjoint sets $A_1, \ldots, A_m$, each containing exactly one element from each of X and Y, such that for $1 \le i \le m$, $\sum_{a \in A_i} s(a) = B_i$. The problem is strongly NP-hard in general. Consider some instance $\mathcal{I}$ of numerical matching with target sums. Assume without loss of generality that

$$
\sum_{x \in X} s(x) + \sum_{y \in Y} s(y) = \sum_{i=1}^m B_i, \tag{1}
$$

or else $\mathcal{I}$ has no solution. We will construct a graph $G(\mathcal{I})$ that has an optimal layout of certain dimensions if and only if $\mathcal{I}$ has a solution.
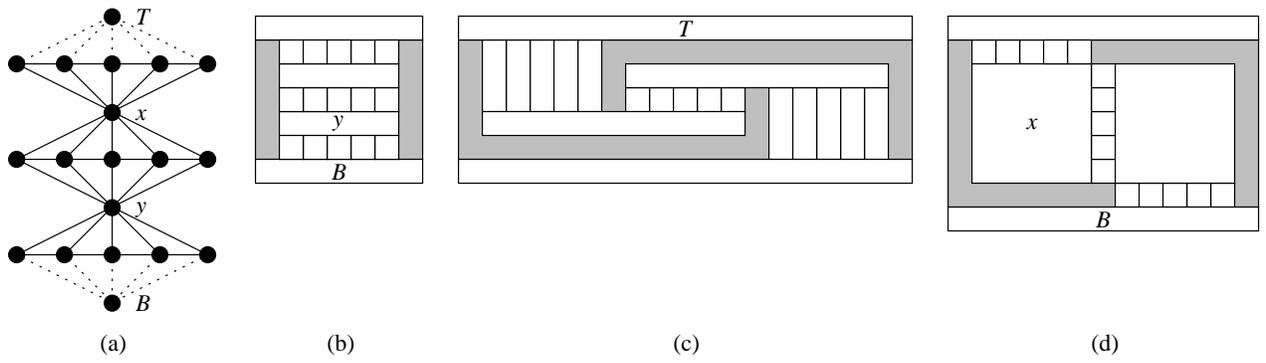
15

Figure 8: (a) A 5-accordion enclosed by T and B. The accordion edges are solid; the enclosing edges are dotted. (b)–(d) Layouts with (b) y below x; (c) x below y; and (d) neither x nor y below the other.

Define an *n-accordion* to be a graph on $3n+2$ vertices: three disjoint, simple paths of length n each; two additional vertices x and y; an edge between x and each vertex on the first and second paths; and an edge between y and each vertex on the second and third paths. An *enclosed accordion* is an accordion augmented (*enclosed*) by two additional vertices T and B: T adjacent to each vertex on the first path, and B adjacent to each vertex on the third path. See Figure 8(a).

Lemma 6.1 *Assume $n \geq 3$. In any layout of an enclosed n-accordion such that T appears above (sym., below) B and no rectangle corresponding to an accordion vertex is left or right of T or B, the accordion rectangles form a bounding box of height at least 5 and width at least n; furthermore, to achieve height 5 and width n simultaneously, the accordion rectangles must form a dissection. Symmetrically, in any layout of an enclosed n-accordion such that T appears left (sym., right) of B and no rectangle corresponding to an accordion vertex is above or below T or B, the accordion rectangles form a bounding box of height at least n and width at least 5; furthermore, to achieve height n and width 5 simultaneously, the accordion rectangles must form a dissection.*

Proof. We prove the case in which T is above B; the other cases are symmetric. Refer to Figure 8. The width lower bound follows, because all vertices on the first path must be adjacent to T's bottom boundary. By assumption, in any layout x must be below T and y above B. Lemma 4.2 implies that the width or height of the ladder between x and y must be at least n. If the height is at least n (as in Figure 8(d)), the height lower bound follows from the mutual non-adjacency of x, y, T, and B and the lower bound on n. Otherwise, either x must be above y (as in Figure 8(b)) or y above x (as in Figure 8(c)); again the height lower bound follows. To achieve height 5 and width n simultaneously requires the accordion rectangles to be configured as in Figure 8(b), which is a dissection. □

Let $x_1, \ldots, x_m$ (rsp., $y_1, \ldots, y_m$) denote the sizes of the elements of X (rsp., Y) in $\mathcal{I}$; define $B^* = \max\{B_i : 1 \leq i \leq m\}$; and define $\Delta_i = 2B^* - B_i \geq 1$ for $1 \leq i \leq m$. Graph $G(\mathcal{I})$ is formed from the following components.

- vertices X, Y, t, b, g^*, and g_i for $0 \leq i \leq m$;

- an x_i-rung ladder R_i and a y_i-rung ladder S_i for $1 \leq i \leq m$;

- a Δ_i-accordion, denoted $\mathcal{A}_i$, for $1 \leq i \leq m$;

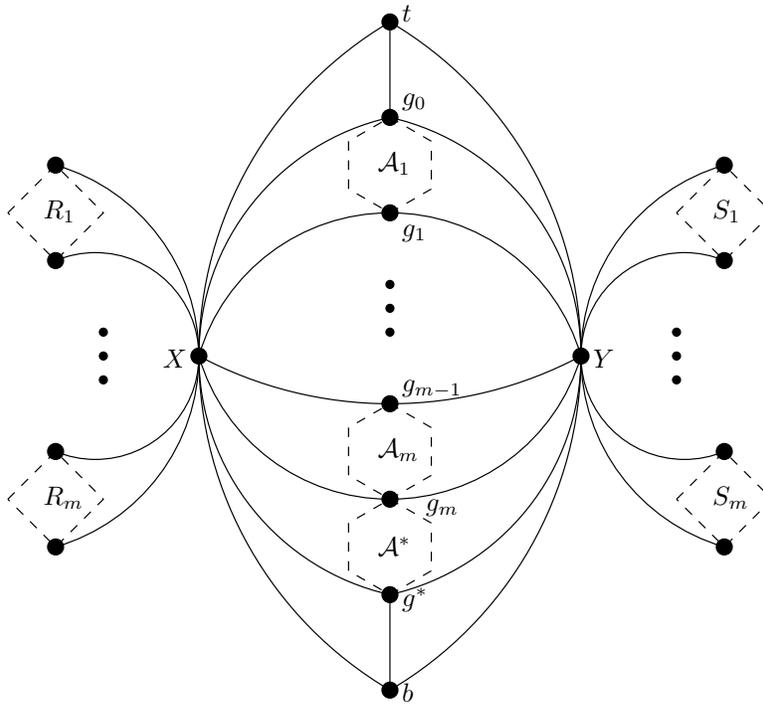- a $(2B^* + 2)$-accordion, denoted $\mathcal{A}^*$.

16

Figure 9: The graph $G(\mathcal{I})$ for some instance $\mathcal{I}$ of numerical matching with target sums. The accordions are denoted by dashed hexagons and the ladders by dashed diamonds.

The components are arranged as follows. (See Figure 9.)

- X and Y are each adjacent to t, b, g^* and g_i for $0 \leq i \leq m$;

- t is adjacent to g_0 and b to g^*;

- g_{i-1} and g_i enclose $\mathcal{A}_i$ for $1 \leq i \leq m$;

- g_m and g^* enclose $\mathcal{A}^*$;

- X is adjacent to the struts of R_i for $1 \leq i \leq m$;

- Y is adjacent to the struts of S_i for $1 \leq i \leq m$.

Lemma 6.2 *Let $\mathcal{I}$ be an instance of numerical matching with target sums. $\mathcal{I}$ has a solution if and only if $G(\mathcal{I})$ has a rectangular layout of width no more than $2B^* + 6$ and height no more than $6m + 9$ (or vice versa).*

Proof. The subgraph of $G(\mathcal{I})$ induced by X, Y, t, b, g^*, the g_i's, and paths through the accordions connecting t, g_0, etc. through b, create a global $(m + 4)$-rung ladder. Scaling if necessary, we can assume all $\Delta_i \geq 3$. Lemmas 4.2 and 6.1 together imply that any layout for $G(\mathcal{I})$ respects the following (up to width/height symmetry). (Refer to Figure 10.)
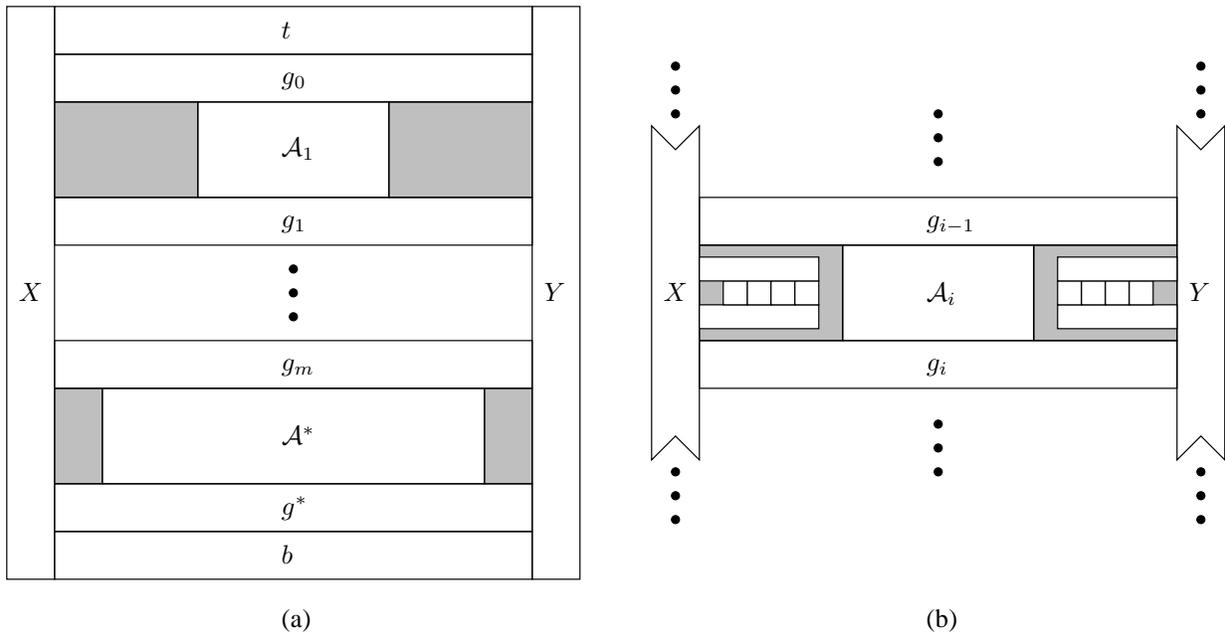
Figure 10: (a) An incompressible sub-layout for $G(\mathcal{I})$. (b) Filling the gaps in layer i by some R_i and S_i.

1. The overall height is at least $6m + 9$, because the $m + 1$ enclosed accordions, each accordion itself of height 5, are all aligned, and the $m + 4$ rungs of the global ladder are stacked.

2. The overall width is at least $2B^* + 6$: $2B^* + 2$ for $\mathcal{A}^*$, 2 to separate it from X and Y, and 1 each for X and Y.

(1) is true regardless of whether rectangle t (rsp., b) is above (rsp., below) X and Y or in between. That is, the subgraph induces an *incompressible* sub-layout: these components must have the indicated heights and/or widths in any layout.

Refer to the sub-layout including g_{i-1}, $\mathcal{A}_i$, and g_i as *layer* i. Accordion $\mathcal{A}_i$ creates gap(s) of total width $w(i) = 2B^* + 4 - \Delta_i$ between X and Y at layer i. The idea is to fill each such gap with a sub-layout for some ladder adjacent to X or Y. Each such ladder sub-layout must consist of two struts, attached to X (or Y), with the rungs between them (except for possibly one of them) but separated from X and Y. See Figure 10(b). Ladder R_i has x_i rungs, and its sub-layout must therefore have length at least $x_i + 1$ (x_i for rungs plus one unit to separate the rungs from X). Similarly, ladder S_i's sub-layout must have length at least $y_i + 1$.

Assume $\mathcal{I}$ has a solution. Re-index the elements of X and Y so that $x_i + y_i = B_i$ for $1 \le i \le m$. Then ladders R_i and S_i fit in the layer-i gaps without increasing the width of each layer between X and Y from the minimum $2B^* + 4$. To see this, note that the width required for the sub-layouts at layer i is $x_i + y_i + \Delta_i + 4 = 2B^* + 4$: $x_i + 1$ for R_i, $y_i + 1$ for S_i, Δ_i for $\mathcal{A}_i$, and 2 units to separate $\mathcal{A}_i$ from R_i and S_i. Also, each ladder sub-layout fits within the height-5 lower bound of layer i. Therefore, the width remains $2B^* + 6$, the height remains $6m + 9$, and so the desired layout exists.

Assume $G(\mathcal{I})$ has a layout of the hypothesized dimensions. The incompressibility argument above implies that each ladder must fit into a gap in one of the layers. No two ladders may be placed in the same gap, for the height of the layer would have to grow, violating the height assumption. Re-index the elements of X and Y so that for $1 \le i \le m$, x_i and y_i are the elements whose ladders, R_i and S_i, fit into layer i.

18

The gap-height constraint implies that the bounding boxes for R_i and S_i are neither above or below each other. Their widths are thus bounded by the same horizontal level of rectangles in $\mathcal{A}_i$. Therefore, we can assume without loss of generality that $\mathcal{A}_i$ is layed out as a dissection and thus by Lemma 6.1 in width Δ_i. Because R_i and S_i fit into the layer-i gaps, it follows that $x_i + y_i + 4 \leq w(i) = 2B^* + 4 - \Delta_i$ and hence $x_i + y_i \leq B_i$. As this is true for all layers, Equation (1) implies that $x_i + y_i = B_i$ for all i, which gives a solution to $\mathcal{I}$. $\qquad\square$

Theorem 6.3 *Given a graph G and values $W, H, A \in \mathbb{Z}^+$, determining if G has a (strong or weak) rectangular layout of (1) width no more than W and height no more than H or (2) area no more than A is NP-complete.*

Proof. Theorem 4.1 implies that the problem is in NP. Lemma 6.2 provides a P-time reduction showing NP-completeness, because the number of vertices and edges in $G(\mathcal{I})$ is poly(m) and numerical matching with target sums is strongly NP-complete: for problem (1), set $W = 2B^* + 6$ and $H = 6m + 9$, and for problem (2), set $A = (6m + 9)(2B^* + 6)$. A similar reduction using height-3 accordions can be used for weak layouts. $\qquad\square$

Corollary 6.4 *Given a graph G and value $L \in \mathbb{Z}^+$, it is NP-hard to determine the minimum width (rsp., height) layout of G such that the height (rsp., width) does not exceed L.*

6.2 Area Monotonicity

For any graph $G = (V, E)$, $V' \subset V$, and $E' \subset E$, define $G_{V'}$ to be the induced subgraph on $V \setminus V'$ and $G_{E'}$ that on $E \setminus E'$ (removing isolated vertices). By Corollary 3.5, we know that both $G_{V'}$ and $G_{E'}$ have layouts if G has a layout. The same does not necessarily hold for rectangular duals, however. In general, given a rendering strategy we say that a vertex or edge subset is *rendering preserving* if the corresponding subgraph defined above admits such a rendering.

Consider monotonicity of areas under augmentation. For a given rendering strategy such that $A^*(G)$ is the area of an optimal rendering of G (assuming G admits a rendering), we say the rendering is *vertex* (rsp., *edge*) *monotone* if for any graph G and any rendering-preserving vertex subset V' (rsp., edge subset E') it is true that $A^*(G_{V'}) \leq A^*(G)$ (rsp., $A^*(G_{E'}) \leq A^*(G)$).

Theorem 6.5

1. *Weak layouts are vertex and edge monotone.*

2. *Strong layouts are vertex monotone but not edge monotone.*

3. *Rectangular duals are neither vertex nor edge monotone.*

Proof. (1) A weak layout for G is also a weak layout for any subgraph of G.

(2) Given a strong layout L of G, removing the rectangle corresponding to v yields a strong layout of $G_{\{v\}}$ of no greater area; hence, strong layouts are vertex monotone. Figure 1 disproves edge monotonicity, however: by inspection, any strong layout for the graph in Figure 1(a) must have area at least 9, whereas the edge-augmented graph in Figure 1(b) has an area-6 strong layout.

(3) For any $n \in \mathbb{Z}^+$, Figure 11 shows a graph G on $2n + 7$ vertices and a rectangular dual of area $6(n + 2)$. In any rectangular dual of $G_{\{p\}}$, however, depicted in Figure 12(a), rectangles a, b, c, and d must be the corners of the dissection, for each has degree only 2. Thus, the height and width must each be $\Omega(n)$, as exemplified in Figure 12(b). The area is thus $\Omega(n^2)$, which disproves vertex monotonicity and also edge monotonicity, because the latter implies the former. $\qquad\square$
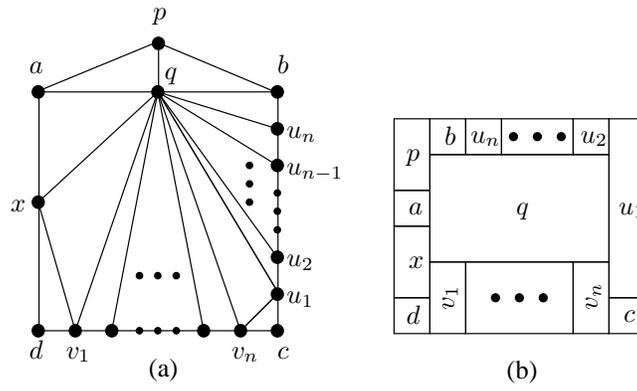
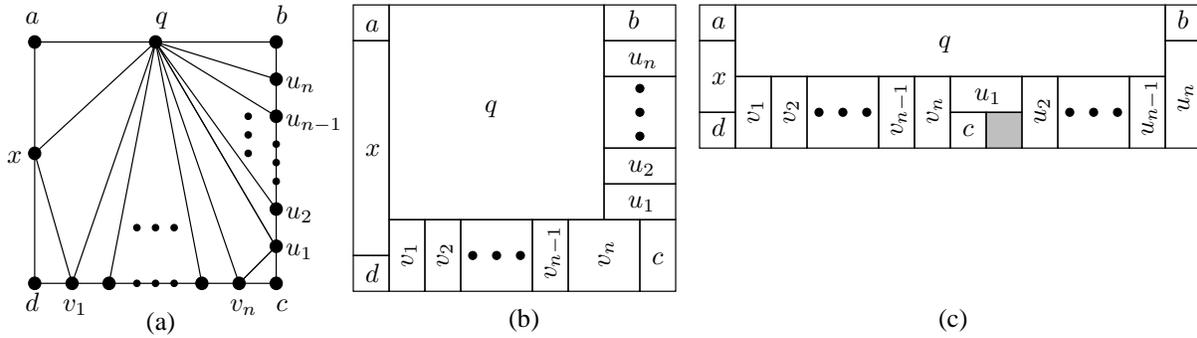Figure 11: (a) A graph on $2n + 7$ vertices. (b) A rectangular dual of area $6(n + 2)$.

Figure 12: (a) The graph on $2n+6$ vertices derived by deleting vertex p from Figure 11(a). (b) A rectangular dual of area $(n + 3)(n + 2)$. (c) A strong rectangular layout of area $4(n + 2)$.

6.3 Gaps between Layouts and Duals

Consider the potential gap between minimum-area layouts and rectangular duals of the same graph. Define $A_s^*(G)$ (rsp., $A_D^*(G)$) to be the area of an optimal strong rectangular layout (rsp., rectangular dual) of G.

Theorem 6.6 *There exists an infinite family $\mathcal{G}$ of graphs such that for any $G \in \mathcal{G}$, $A_s^*(G) = O(n)$ but $A_D^*(G) = \Omega(n^2)$.*

Proof. Define $\mathcal{G}$ to contain the graph shown in Figure 12(a) for each $n \in \mathbb{Z}^+$. Consider some $n \in \mathbb{Z}^+$ and the corresponding $G \in \mathcal{G}$. As argued in the proof of Theorem 6.5, any rectangular dual of G has area $\Omega(n^2)$, while Figure 12(c) depicts a strong rectangular layout of height 4 and width $n + 2$. $\square$

7   Conclusion

We have presented a new characterization of contact graphs of isothetic rectangles in terms of those planar graphs that can be embedded with no filled three-cycles. We have shown the general area and constrained-width (and -height) optimization problems for rectangular layouts to be NP-hard and provided $O(n)$-time

algorithms to construct $O(n^2)$-area rectangular layouts for graphs and $O(n \log n)$-area rectangular layouts for trees.

Many open problems remain. What is the hardness of approximating the minimum-area rectangular layout? Are there better approximation algorithms than the ones we presented here ($O(n)$-approximation for graphs; $O(\log n)$ for trees)? Is approximating the minimum dimension (width or height) easier than approximating the area? This problem is motivated by applications on fixed-width, scrollable displays. Also, does the NP-hardness result extend to rectangular duals? Since graphs that admit such duals are internally triangulated (triangulated except for the outer face), the freedom to place components to satisfy partition-type reductions does not seem to exist.

Can our techniques be applied to study contact graphs on other closed shapes: for example, squares, arbitrary regular polygons, arbitrary convex polygons, and higher-dimensional shapes? Also, allowing corner touching to imply adjacency changes the class of graphs described by layouts. For example, K_4 can be expressed by four rectangles meeting at a corner, and K_6, which is not planar, can be expressed by triangles. For $k \geq 4$, however, no layout on k-gons can express a non-planar graph. Allowing corner touching also opens the question of allowing non-isothetic rectangles, which can represent embeddings with filled triangles. Finally, is there a class of polygonal shapes other than disks whose contact graphs are the planar graphs?

Acknowledgements

References

[1] A. Accornero, M. Ancona, and S. Varini. All separating triangles in a plane graph can be optimally "broken" in polynomial time. *International Journal of Foundations of Computer Science*, 11(3):405–21, 2000.

[2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, MA, 1974.

[3] G. Di Battista, P. Eades, R. Tamassia, and I. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, Upper Saddle River, NJ, 1999.

[4] G. Di Battista, W. Lenhart, and G. Liotta. Proximity drawability: A survey. In *Proc. Graph Drawing, DIMACS Int'l. Wks. (GD'94)*, volume 894 of *Lecture Notes in Computer Science*, pages 328–39. Springer-Verlag, 1994.

[5] J. Bhasker and S. Sahni. A linear algorithm to find a rectangular dual of a planar triangulated graph. *Algorithmica*, 3:247–78, 1988.

[6] T. Biedl, A. Bretscher, and H. Meijer. Rectangle of influence drawings of graphs without filled 3-cycles. In *Proc. 7th Int'l. Symp. on Graph Drawing '99*, volume 1731 of *Lecture Notes in Computer Science*, pages 359–68. Springer-Verlag, 1999.

[7] T. Biedl, G. Kant, and M. Kaufmann. On triangulating planar graphs under the four-connectivity constraint. *Algorithmica*, 19:427–46, 1997.

[8] A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph Classes: A Survey*. SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, PA, 1999.

[9] M. de Berg, S. Carlsson, and M. H. Overmars. A general approach to dominance in the plane. *Journal of Algorithms*, 13(2):274–96, 1992.

[10] H. N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In *Proc. 1st ACM-SIAM Symp. on Discrete Algorithms*, pages 434–43, 1990.

[11] K. R. Gabriel and R. R. Sokal. A new statistical approach to geographical analysis. *Systematic Zoology*, 18:54–64, 1969.

[12] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.

[13] D. Harel and R. E. Tarjan. Fast algorithms for finding nearest common ancestors. *SIAM Journal on Computing*, 13(2):338–55, 1984.

[14] X. He. On finding the rectangular duals of planar triangular graphs. *SIAM Journal on Computing*, 22(6):1218–26, 1993.

[15] P. Hliněný and J. Kratochvíl. Representing graphs by disks and balls (a survey of recognition-complexity results). *Discrete Mathematics*, 229(1-3):101–24, 2001.

[16] J. W. Jaromczyk and G. T. Toussaint. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE*, 80:1502–17, 1992.

[17] G. Kant and X. He. Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems. *Theoretical Computer Science*, 172:175–93, 1997.

[18] P. Koebe. Kontaktprobleme der konformen Abbildung. *Berichte über die Verhand-lungen der Sächsischen, Akad. Wiss., Math.-Phys. Klass*, 88:141–64, 1936.

[19] K. Koźmiński and W. Kinnen. Rectangular duals of planar graphs. *Networks*, 15:145–57, 1985.

[20] K. Koźmiński and W. Kinnen. Rectangular dualization and rectangular dissections. *IEEE Transactions on Circuits and Systems*, 35(11):1401–16, 1988.

[21] Y.-T. Lai and S. M. Leinwand. Algorithms for floorplan design via rectangular dualization. *IEEE Transactions on Computer-Aided Design*, 7:1278–89, 1988.

[22] Y.-T. Lai and S. M. Leinwand. A theory of rectangular dual graphs. *Algorithmica*, 5:467–83, 1990.

[23] A. S. LaPaugh. *Algorithms for Integrated Circuit Layout: An Analytic Approach*. PhD thesis, M.I.T., 1980.

[24] G. Liotta, A. Lubiw, H. Meijer, and S. H. Whitesides. The rectangle of influence drawability problem. *Computational Geometry: Theory and Applications*, 10:1–22, 1998.

[25] T. A. McKee and F. R. McMorris. *Topics in Intersection Graph Theory*. SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, PA, 1999.

[26] D. E. Muller and F. P. Preparata. Finding the intersection of two convex polyhedra. *Theoretical Computer Science*, 7(2):217–36, 1978.

[27] J. R. Munkres. *Topology: A First Course*. Prentice-Hall of India, New Delhi, 1999.

[28] M. H. Overmars and D. Wood. On rectangular visibility. *Journal of Algorithms*, 9(3):372–90, 1988.

[29] P. Pan, W. Shi, and C. L. Liu. Area minimization for hierarchical floorplans. *Algorithmica*, 15:550–71, 1996.

[30] R. Ramakrishnan and J. Gehrke. *Database Management Systems*. McGraw-Hill, Boston, MA, 2000.

[31] R. C. Read. A new method for drawing a graph given the cyclic order of the edges at each vertex. *Congressus Numerantium*, 56:31–44, 1987.

[32] P. Rosenstiehl and R. E. Tarjan. Rectilinear planar layouts and bipolar orientations of planar graphs. *Discrete Computational Geometry*, 1:343–53, 1986.

[33] H. Sachs. Coin graphs, polyhedra, and conformal mapping. *Discrete Mathematics*, 134:133–8, 1994.

[34] B. Schieber and U. Vishkin. On finding lowest common ancestors: Simplification and parallelization. *SIAM Journal on Computing*, 17(6):1253–62, 1988.

[35] W. Shi. A fast algorithm for area minimization of slicing floorplans. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(12):1525–32, 1996.

[36] P. Steadman. Graph-theoretic representation of architectural arrangement. In L. March, editor, *The Architecture of Form*, pages 94–115. Cambridge University Press, London, New York, Melbourne, 1976.

[37] L. Stockmeyer. Optimal orientations of cells in slicing floorplan designs. *Information and Control*, 57:91–101, 1983.

[38] S. Sur-Kolay and B. B. Bhattacharya. Inherent nonslicability of rectangular duals in VLSI floorplanning. In *Proc. 8th Conf. on Foundations of Software Technology and Theoretical Computer Science*, volume 338 of *Lecture Notes in Computer Science*, pages 88–107. Springer-Verlag, 1988.

[39] S. Sur-Kolay and B. B. Bhattacharya. On the family of inherently nonslicible floorplans in VLSI layout design. In *Proc. IEEE Int'l. Conf. on Circuits and Systems*, volume 5, pages 2850–3, 1991.

[40] K. Tani, S. Tsukiyama, S. Shinoda, and I. Shirakawa. On area-efficient drawings of rectangular duals for VLSI floor-plan. *Mathematical Programming*, 52:29–43, 1991.

[41] R. E. Tarjan. Applications of path compression on balanced trees. *Journal of the ACM*, 26(4):690–715, 1979.

[42] C. Thomassen. Plane representations of graphs. In J. A. Bondy and U. S. R. Murty, editors, *Progress in Graph Theory*, pages 43–69. Academic Press, Canada, 1984.

[43] C. Thomassen. Interval representations of planar graphs. *Journal of Combinatorial Theory (B)*, 40:9–20, 1988.

[44] P. Ungar. On diagrams representing maps. *Journal of the London Mathematical Society*, 28:336–42, 1953.

[45] H. Whitney. 2-isomorphic graphs. *American Journal of Mathematics*, 55:245–54, 1933.

[46] S. Wimer, I. Koren, and I. Cederbaum. Floorplans, planar graphs, and layouts. *IEEE Transactions on Circuits and Systems*, 35(3):267–78, 1988.

[47] G. K. Yeap and M. Sarrafzadeh. A unified approach to floorplan sizing and enumeration. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(12):1858–67, 1993.

[48] G. K. Yeap and M. Sarrafzadeh. Sliceable floorplanning by graph dualization. *SIAM Journal on Discrete Mathematics*, 8(2):258–80, 1995.

A   Proofs of Lemmas

A.1   Proof of Lemma 2.2

Proof. G is a planar triangulation if and only if G^* is planar and cubic. It therefore suffices to prove that (1) if G is a planar triangulation that is not 4-connected, then G^* is not cyclically 4-edge connected, and (2) if G^* is cubic and planar but not cyclically 4-edge connected, then G is not 4-connected.

Assume that G is a planar triangulation that is not 4-connected. By Lemma 2.1 G has some separating triangle (a, b, c). In G^*, therefore, there are edges $\overline{\{a, b\}}$, $\overline{\{b, c\}}$, and $\overline{\{c, a\}}$, each connecting the two faces in G incident upon edges $\{a, b\}$, $\{b, c\}$, and $\{c, a\}$, resp. These three edges separate G^* into two components, G_1^* and G_2^*. That (a, b, c) is a separating triangle and G is triangulated implies that G_1^* and G_2^* contain cycles. Hence G^* is not cyclically 4-edge connected.

Assume that G^* is cubic and planar but not cyclically 4-edge connected. Then there exist three edges e_1, e_2, and e_3 that separate G^* into two or three components, each of which contains a cycle. Assume for now a separation into two components, G_1^* and G_2^*. Without loss of generality, each of e_1, e_2, and e_3 has an endpoint in each component. Therefore there are vertices v_1, v_2, and v_3 in G that correspond to the three faces induced in G^* between pairs of e_1, e_2, and e_3. Furthermore, (v_1, v_2, v_3) forms a separating triangle, because the cyclic nature of G_1^* and G_2^* implies that the corresponding components G_1 and G_2 in G are non-empty. By Lemma 2.1, therefore, G, which is a planar triangulation, is not 4-connected. A similar argument holds when G^* is separated into three components, in which case some pair among e_1, e_2, and e_3 separates G^* into two components. $\square$

A.2   Proof of Lemma 3.1

Proof.   Assume G is a saturated plane map with no 3-ring. By definition, G is cubic and planar, and so G^* is a planar triangulation. Furthermore, because a 3-ring in G induces a separating triangle in G^* and vice-versa, it follows that G^* has no separating triangle. By Lemma 2.1, therefore, G^* is 4-connected, and by Lemma 2.2, G is cubic, cyclically 4-edge connected.

Assume G is planar, cubic, and cyclically 4-edge connected. Then by Lemma 2.2 G^* is a 4-connected planar triangulation. By Lemma 2.1, G^* has no separating triangle, and so G has no 3-ring. That G is a saturated plane map follows the assumption by definition. $\square$