Aspect Ratio Universal Rectangular Layouts*

Stefan Felsner[1], Andrew Nathenson[2], and Csaba D. Tóth[3]

[1] Institut für Mathematik, Technische Universität Berlin, Berlin, Germany
felsner@math.tu-berlin.de
[2] University of California, San Diego, CA, USA
anathenson@ucsd.edu
[3] California State University Northridge, Los Angeles, CA, USA
csaba.toth@csun.edu

Abstract. A *generic rectangular layout* (for short, *layout*) is a subdivision of an axis-aligned rectangle into axis-aligned rectangles, no four of which have a point in common. Such layouts are used in data visualization and in cartography. The contacts between the rectangles represent semantic or geographic relations. A layout is *aspect ratio universal* if it can realize any assignment of aspect ratios to rectangles. We give two different characterizations for aspect ratio universal layouts, one in terms of geometry and one in terms combinatorics: A layout is aspect ratio universal if and only if it is one-sided and sliceable; and if and only if its extended dual graph admits a unique transversal structure. Furthermore, we describe a quadratic-time algorithm that decides whether a given graph G is the dual graph of an aspect ratio universal layout, and finds such a layout if one exists.

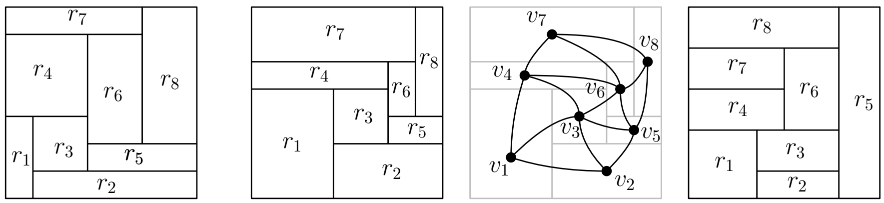
Keywords: rectangular layouts · contact graphs · universality.

1 Introduction

A *rectangular layout* (a.k.a. *mosaic floorplan* or *rectangulation*) is a subdivision of an axis-aligned rectangle into axis-aligned rectangle faces, it is *generic* if no four faces have a point in common. All layouts in this paper are generic unless stated otherwise. In the *dual graph* $G(\mathcal{L})$ of a layout $\mathcal{L}$, the nodes correspond to rectangular faces, and an edge corresponds to a pair of rectangles whose common boundary contains a line segment [5,24,25]. Two layouts are *strongly equivalent* (for short, *equivalent*) if they have isomorphic dual graphs, and the corresponding line segments between rectangles have the same orientation (horizontal or vertical); see Fig. 1 for examples. Rectangular layouts have been studied for more than 40 years, originally motivated by VLSI design [18,20,30] and cartography [23], and more recently by data visualization [15].

An (abstract) graph is called a *proper graph* if it is the dual of a rectangular layout. Every proper graph is a near-triangulation (a plane graph where every bounded face is a triangle, but the outer face need not be a triangle). But not every near-triangulation is a proper graph [24,25]. Ungar [29] gave a combinatorial

* Research on this paper was partially supported by the NSF award DMS-1800734.



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

Fig. 1: (a-b) Two equivalent layouts. (c) Dual graph. (d) Another layout with the same dual graph. The layout in (d) is sliceable, none of them is one-sided.

characterization of proper graphs (see also [14,27]); and they can be recognized in linear time [10,19,21,22].

In data visualization and cartography [15,23], the rectangles correspond to entities (e.g., countries or geographic regions); adjacency between rectangles represents semantic or geographic relations, and the "shape" of a rectangle represent data associated with the entity. It is often desirable to use equivalent layouts to realize different statistics associated with the same entities. Eppstein et al. [5] considered *area universal* layouts $\mathcal{L}$, where any area assignment to the rectangles can be realized by a layout equivalent to $\mathcal{L}$. They showed that a layout is area universal if and only if it is one-sided (defined below). However, no polynomial-time algorithm is known for testing whether a given graph G is the dual of some area-universal layout.

In some applications, the aspect ratios (rather than the areas) of the rectangles are specified. For example, in word clouds adapted to multiple languages, the aspect ratio of (the bounding box of) each word depends on the particular language. The *aspect ratio* of an axis-aligned rectangle r is $\text{height}(r)/\text{width}(r)$. A rectangular layout $\mathcal{L}$ is *aspect ratio universal* if any assignment of aspect ratios to the rectangles can be realized by a layout equivalent to $\mathcal{L}$.

Our results. We characterize aspect ratio universal layouts.

Theorem 1. *For a rectangular layout $\mathcal{L}$, the following are equivalent:*

(i) $\mathcal{L}$ is aspect ratio universal;
(ii) $\mathcal{L}$ is one-sided and sliceable;
(iii) the extended dual of $\mathcal{L}$, $G^*(\mathcal{L})$, admits a unique transversal structure.

The terms in (ii) and (iii) above are defined in Sec. 1.1; and we prove Theorem 1 in Sec. 2. It is not difficult to show that one-sided sliceable layouts are aspect ratio universal; and admit a unique transversal structure. Proving the converses, however, is more involved.

In some applications, the rectangular layout is not specified, and we are only given the dual graph of a layout (i.e., a proper graph). This raises the following problem: Given a proper graph G with n vertices, find an aspect ratio universal layout $\mathcal{L}$ such that $G = G(\mathcal{L})$ or report that none exists. Using structural properties of one-sided sliceable layouts that we develop here, we present an $O(n^2)$-time algorithm for this problem (Theorem 4 in Section 3).

Thomassen [27] gave a linear-time algorithm to recognize proper graphs if the nodes corresponding to corner rectangles are specified, using combinatorial characterizations of layouts [29]. Kant and He [11,13] described a linear-time algorithm to test whether a given graph G^* is the extended dual of a layout, using transversal structures. Later, Rahman et al. [10,19,21,22] showed that proper graphs can be recognized in linear time (without specifying the corners). However, a proper graph may have exponentially many nonequivalent realizations, and prior algorithms cannot report whether one-sided sliceable layout exits. Currently, no polynomial-time algorithm is known for recognizing the duals of one-sided layouts; the problem is fixed-parameter tractable for a parameter related to the number of vertex-disjoint separating 4-cycles [5].

1.1 Background and Terminology

A *rectangular layout* (for short, *layout*) is a rectilinear graph in which each face is a rectangle, the outer face is also a rectangle, and the vertex degree is at most 3. A *sublayout* of a layout $\mathcal{L}$ is a subgraph of $\mathcal{L}$ which is a layout. A layout is *irreducible* if it does not contain any nontrivial sublayout. A *rectangular arrangement* is a 2-connected subgraph of a layout in which bounded faces are rectangles (the outer face need not be a rectangle).

One-sided layouts. A *segment* of a layout $\mathcal{L}$ is a path of collinear inner edges of $\mathcal{L}$. A segment of $\mathcal{L}$ that is not contained in any other segment is maximal. In a *one-sided* layout, every maximal line segment s must be a side of at least one rectangle R; in particular, any other segment orthogonal to s with an endpoint in the interior of s lies in a halfplane bounded by s, and points away from R.

Sliceable layouts. A maximal line segment subdividing a rectangle or a rectangular union of rectangular faces is called a *slice*. A *sliceable layout* (a.k.a. *slicing floorplan* or *guillotine rectangulation*) is one that can be obtained through recursive subdivision with vertical or horizontal lines; see Fig 1(d). The recursive subdivision can be represented by a *binary space partition tree* (*BSP-tree*), which is a binary tree where each vertex is associated with either a rectangle with a slice, or just a rectangle if it is a leaf [3]. For a nonleaf vertex, the two subrectangles on each side of the slice are associated with the two children. The number of (equivalence classes of) sliceable layouts with n rectangles is known to be the nth Schröder number [31]. One-sided sliceable layouts are in bijection with certain pattern-avoiding permutations, closed formulas for their number has been given by Asinowski and Mansour [2]; see also [17] and OEIS A078482 in the on-line encyclopedia of integer sequences (https://oeis.org/) for further references.

A *windmill* in a layout is a set of four pairwise non-crossing maximal line segments, called *arms*, which contain the sides of a central rectangle, and each arm has an endpoint on the interior of another (e.g., the maximal segments around r_3 or r_6 in Fig. 1 (a-b)). We orient each arm from the central rectangle to the other endpoint. A windmill is either *clockwise* or *counterclockwise*. It is well known that a layout is sliceable iff it does not contain a windmill [1].

Transversal structure. The dual graph $G(\mathcal{L})$ of a layout $\mathcal{L}$ encodes adjacency between faces, but does not specify the relative positions between faces (above-below or left-right). The transversal structure (a.k.a. regular edge-labelling) were introduced by He [11,13] for the efficient recognition of proper graphs, and later used extensively for counting and enumerating (equivalence classes of) layouts [9]. The *extended dual graph* $G^*(\mathcal{L})$ is the contact graph of the rectangular faces *and* the four edges of the bounding box of $\mathcal{L}$; it is a triangulation in an outer 4-cycle without separating triangles; see Fig. 2.

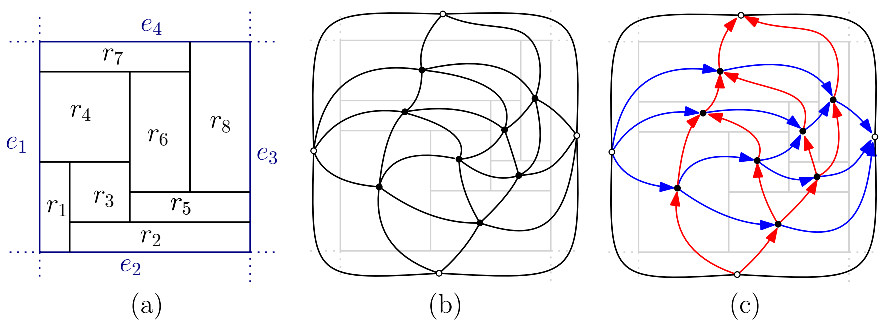


Fig. 2: (a) A layout $\mathcal{L}$ bounded by $e_1, \ldots, e_4$. (b) Extended dual graph $G^*(\mathcal{L})$ with an outer 4-cycle $(e_1, \ldots, e_4)$. (c) Transversal structure.

A layout $\mathcal{L}$ is encoded by a *transversal structure* that comprises $G^*(\mathcal{L})$ and an orientation and bicoloring of the inner edges of $G^*(\mathcal{L})$, where blue (resp., red) edges correspond to above-below (resp., left-to-right) relation between two objects in contact. An (abstract) *transversal structure* is defined as a graph G^*, which is a 4-connected triangulation of an outer 4-cycle (W, N, E, S), together with a bicoloring and orientation of the inner edges of G^* such that the all inner edges incident to W, N, E, and S, respectively, are outgoing blue, outgoing red, incoming blue, and incoming red; and at each inner vertex the counterclockwise rotation of incident edges consists of four nonempty blocks of outgoing blue, outgoing red, incoming blue, and incoming red edges.

Flips and Alternating 4-Cycles. It is known that transversal structures are in bijection with the equivalence classes of layouts [7,9,13]. Furthermore, a sequence of flip operations can transform any transversal structure with n inner vertices into any other [6,9]. Each *flip* considers an *alternating 4-cycle* C, which comprises red and blue edges alternating, and changes the color of every edge in the interior of C; see Fig. 3. If, in particular, there is no vertex in the interior of C, then the flip changes the color of the inner diagonal of C. Furthermore, every flip operation yields a valid transversal structure on $G^*(\mathcal{L})$, hence a new layout $\mathcal{L}'$ that is not equivalent to $\mathcal{L}$. We can now establish a relation between geometric and combinatorial properties.

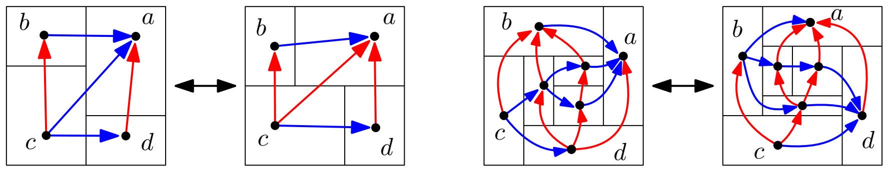


Fig. 3: A flip of an empty (left) and a nonempty (right) alternating cycle.

Lemma 1. *A layout $\mathcal{L}$ is one-sided and sliceable if and only if $G^*(\mathcal{L})$ admits a unique transversal structure.*

Proof. Assume that $\mathcal{L}$ is a layout where $G^*(\mathcal{L})$ admits two or more transversal structures. Consider a transversal structure of $G^*(\mathcal{L})$. Since any two transversal structures are connected by a sequence of flips, there exists an alternating 4-cycle. Any alternating 4-cycle with no interior vertex corresponds to a segment in $\mathcal{L}$ that is two-sided. Any alternating 4-cycle with interior vertices corresponds to a windmill in $\mathcal{L}$. Consequently, $\mathcal{L}$ is not one-sided or not sliceable.

Conversely, if $\mathcal{L}$ is not one-sided (resp., sliceable), then the transversal structure of $G^*(\mathcal{L})$ contains an alternating 4-cycle with no interior vertex (resp., with interior vertices). Consequently, we can perform a flip operation, and obtain another transversal structure for $G^*(\mathcal{L})$. □

2 Aspect Ratio Universality

An *aspect ratio assignment* to a layout $\mathcal{L}$ is a function that maps a positive real to each rectangle in $\mathcal{L}$. An aspect ratio assignment to $\mathcal{L}$ is *realizable* if there exists an equivalent layout $\mathcal{L}'$ with the required aspect ratios (a *realization*). A layout is *aspect ratio universal* if every aspect ratio assignment is realizable. In this section, we characterize aspect ratio universal layouts (Theorem 2). We start with an easy observation about sliceable layouts.

Lemma 2. *Let $\mathcal{L}$ be a sliceable layout. If an aspect ratio assignment for $\mathcal{L}$ is realizable, then there is a unique realization up to scaling and translation. Furthermore, for every $\alpha > 0$ there exists a realizable aspect ratio assignment for which the bounding box of the realization has aspect ratio α.*

Proof. To prove the first claim, we proceed by induction on k, the height of the BSP-tree representing $\mathcal{L}$. **Basis step:** A layout of height 0 comprises a single rectangle, which is uniquely determined by its aspect ratio up to scaling and translation. **Induction step:** Assume, for induction, that every sublayout at height k in the tree admits a unique realization in which all rectangles at the leaves of the BSP-tree have the required aspect ratios. A rectangle r at height $k+1$ of the BSP-tree is composed of two rectangles at height k, say r_1 and r_2, that share an edge. Given a realization of r_1, there is a unique scaling and translation that attaches r_2 to r_1, and identifies their matching edges. Consequently, the sublayout at height $k+1$ has a unique realization up to scaling and translation.

The second claim follows trivially: Start with a bounding box of aspect ratio α, subdivide it recursively into a layout equivalent to $\mathcal{L}$, and define an aspect ratio assignment using the aspect ratios of the resulting leaf rectangles. □

Corollary 1. *If $\mathcal{L}$ is one-sided and sliceable, then it is aspect ratio universal.*

Proof. Let α be an aspect ratio assignment to a one-sided sliceable layout $\mathcal{L}$. A (unique) realization $\mathcal{L}'$ can be constructed by the induction in the proof of Lemma 2. Indeed, in the basis step, a layout with a single rectangle is aspect ratio universal. In the induction step, when a rectangle is composed of two rectangles $r = r_1 \cup r_2$ separated by a line segment ℓ, then r_1 or r_2 is a leaf of the BSP-tree, since $\mathcal{L}$ is one-sided. Consequently, the pairs of adjacent rectangles on opposite sides of ℓ do not depend on the aspect ratio assignment, and the resulting layout $\mathcal{L}'$ is equivalent to $\mathcal{L}$. □

2.1 Sliceable and One-Sided Layouts

Next we show that any sliceable layout that is aspect ratio universal must be one-sided. We present two simple layouts that are not aspect ratio universal, and then show that all other layouts that are not one-sided or not sliceable can be reduced to these prototypes.

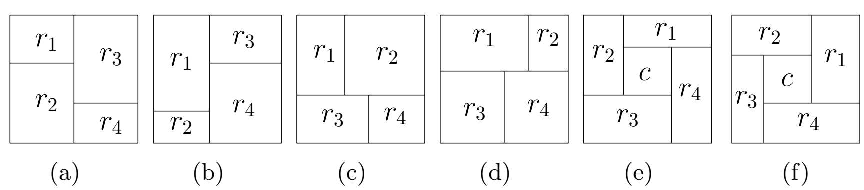


Fig. 4: Prototype layouts that are not aspect ratio universal: (a)–(d) brick layouts are sliceable but not one-sided; (e)–(f) windmills are one-sided but nonsliceable.

Lemma 3. *The layouts in Figure 4 are not aspect ratio universal.*

Proof. Suppose w.l.o.g. that a brick layout $\mathcal{L}_0$ in Fig. 4a is aspect ratio universal. Then there exists an equivalent layout $\mathcal{L}$ for the aspect ratio assignment $\alpha(r_2) = \alpha(r_3) = 1$ and $\alpha(r_1) = \alpha(r_4) = 2$. Since $\mathrm{width}(r_1) = \mathrm{width}(r_2)$ and $\alpha(r_1) = 2\alpha(r_2)$, then $\mathrm{height}(r_1) = 2\,\mathrm{height}(r_2)$, and the left horizontal slice is below the median of $r_1 \cup r_2$. Similarly, width(r_3) = width(r_4) and $\alpha(r_4) = 2\alpha(r_2)$ imply that the right horizontal slice is above the median of $r_3 \cup r_4$. Consequently, r_1 and r_4 are in contact, and $\mathcal{L}$ is not equivalent to $\mathcal{L}_0$, which is a contradiction.

Suppose w.l.o.g. that the layout $\mathcal{L}_1$ in Fig. 4e is aspect ratio universal. Then there exists an equivalent layout $\mathcal{L}$ for the aspect ratio assignment $\alpha(c) = \alpha(r_1) = \alpha(r_2) = \alpha(r_3) = \alpha(r_4) = 1$. In particular, $r_1, \ldots, r_4$ are squares; denote their side lengths by s_i, for $i = 1, \ldots, 4$. Note that one side of r_i strictly contains a side of r_{i-1} for $i = 1, \ldots, 4$ (with arithmetic modulo 4). Consequently, $s_1 < s_2 < s_3 < s_4 < s_1$, which is a contradiction. □

Lemma 4. *If a layout is sliceable but not one-sided, then it is not aspect ratio universal.*

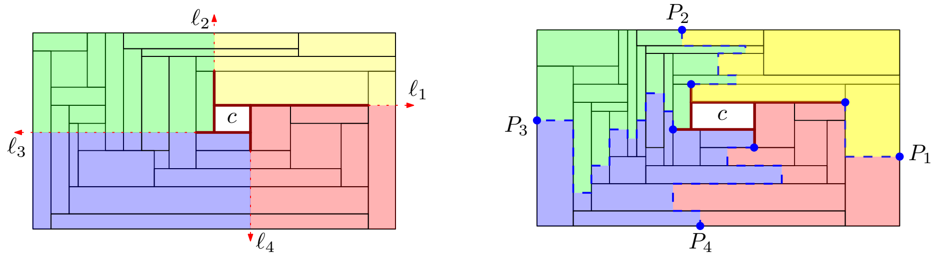
Proof. To show that a layout is not aspect ratio universal, it is sufficient to show that any of its sublayouts are not aspect ratio universal, because any non-realizable aspect ratio assignment for a sublayout can be expanded arbitrarily to an aspect ratio assignment for the entire layout.

Let $\mathcal{L}$ be a sliceable but not one-sided layout. We claim that $\mathcal{L}$ contains a sublayout equivalent to a layout in Figs. 4a–4d. Because $\mathcal{L}$ is not one-sided, it contains a maximal line segment ℓ which is not the side of any rectangle. Because $\mathcal{L}$ is sliceable, every maximal line segment in it subdivides a larger rectangle into two smaller rectangles. We may assume w.l.o.g. that ℓ is vertical. Because ℓ is not the side of any rectangle, the rectangles on the left and right of ℓ must be subdivided horizontally in the recursion. Let ℓ_{left} and ℓ_{right} be the first maximal horizontal line segments on the left and right of ℓ, respectively. Assume that they each subdivide a rectangle adjacent to ℓ into r_1 and r_2 (on the left) and r_3 and r_4 on the right. These rectangles comprise a layout equivalent to the one in Figs. 4a–4d; but they may be further subdivided recursively. By Lemma 2 there exists an aspect ratio assignment to $\mathcal{L}$ not realizable by an equivalent layout. □

In the remainder of this section, we prove that if a layout is not sliceable, then it contains a sublayout similar, in some sense, to a prototype in Figs. 4e–4f. In a nutshell, our proof goes as follows: Consider an arbitrary windmill in a nonslicable layout $\mathcal{L}$. We subdivide the exterior of the windmill into four *quadrants*, by extending the arms of the windmill into rays $\ell_1, \ldots, \ell_4$ to the bounding box; see Fig. 5. Each rectangle of $\mathcal{L}$ lies in a quadrant or in the union of two consecutive quadrants. We assign aspect ratios to the rectangles based on which quadrant(s) it lies in. If these aspect ratios can be realized by a layout $\mathcal{L}'$ equivalent to $\mathcal{L}$, then the rays $\ell_1, \ldots, \ell_4$ will be "deformed" into x- or y-monotone paths that subdivide $\mathcal{L}'$ into the center of the windmill and four arrangements of rectangles, each incident to a unique corner of the bounding box. We assign the aspect ratios for the rectangles in $\mathcal{L}'$ so that these arrangements can play the same role as rectangles $r_1, \ldots, r_4$ in the prototype in Figs. 4e–4f. We continue with the details.

We clarify what we mean by a "deformation" of a (horizontal) ray ℓ.

Lemma 5. *Let a ray ℓ be the extension of a horizontal segment in a layout $\mathcal{L}$ such that ℓ does not contain any other segment and it intersects the rectangles*



(a) A nonsliceable layout, a windmill, where rays $\ell_1, \ldots, \ell_4$ define quadrants.

(b) An equivalent layout, where four paths define rectangular arrangements.

Fig. 5: A rays $\ell_1, \ldots, \ell_4$ deform into monotone paths in an equivalent layout.

$r_1, \ldots, r_k$ in this order. Suppose that in an equivalent layout $\mathcal{L}'$, the corresponding rectangles $r'_1, \ldots, r'_k$ are sliced by horizontal segments $s_1, \ldots, s_k$. Then there exists an x-monotone path comprised of horizontal edges $s_1, \ldots, s_k$, and vertical edges along vertical segment of the layout $\mathcal{L}'$.

Proof. Assume w.l.o.g. that ℓ points to the right. Since ℓ does not contain any other segment and it intersects the rectangles $r_1, \ldots, r_k$ in this order, then r_i and r_{i+1} are on opposite sides of a vertical segment for $i = 1, \ldots, k-1$. The same holds for r'_i and r'_{i+1} as $\mathcal{L}'$ is equivalent to $\mathcal{L}$. In particular, the right endpoint of s_i and the left endpoint of s_{i+1} are on the same vertical segment in $\mathcal{L}'$, for all $i = 1, \ldots k-1$. □

The next lemma allows us to bound the aspect ratio of the bounding box of a rectangular arrangement in terms of the aspect ratios of individual rectangles.

Lemma 6. *If every rectangle in a rectangular arrangement has aspect ratio αm, where m is the number of rectangles in the arrangement, then the aspect ratio of the bounding box of the arrangement is at least α and at most αm^2.*

Proof. Consider an arrangement A with m rectangles and a bounding box R. Let w be the maximum width of a rectangle in A. Then, $\text{width}(R) \leq mw$. A rectangle of width w has height $\alpha m w$, and so $\text{height}(R) \geq \alpha m w$. The aspect ratio of R is $\text{height}(R)/\text{width}(R) \geq (\alpha m w)/(mw) = \alpha$.

Similarly, let h be the maximum height of rectangle in A. Then $\text{height}(R) \leq mh$. A rectangle of height h has width $\frac{h}{\alpha m}$, and so $\text{width}(R) \geq \frac{h}{\alpha m}$. The aspect ratio of R is $\text{height}(R)/\text{width}(R) \leq mh/(\frac{h}{\alpha m}) = \alpha m^2$, as claimed. □

We can now complete the characterization of aspect ratio universal layouts.

Lemma 7. *If a layout $\mathcal{L}$ is not sliceable, it is not aspect ratio universal.*

Proof. Let R be a nonslicable layout of n rectangles in a bounding box of $\mathcal{L}$. We may assume that $\mathcal{L}$ is irreducible, otherwise we can choose a minimal nonsliceable sublayout $\mathcal{L}^*$ from $\mathcal{L}$, and replace each maximal sublayout of $\mathcal{L}^*$ with a rectangle to obtain an irreducible layout. By Lemma 2, a suitable aspect ratio assignment to each sliceable sublayout of $\mathcal{L}^*$ can generate any aspect ratio for the replacement rectangle.

In particular, $\mathcal{L}$ thus contains no slices, as any slice would create two smaller sublayouts. Every nonsliceable layout contains a windmill, which may be assumed to be clockwise. Consider an arbitrary windmill in $\mathcal{L}$, and let c be its central rectangle. By extending the arms of the windmill into rays, $\ell_1, \ldots, \ell_4$, we subdivide $R \setminus c$ into four *quadrants*, denoted by $Q_1, \ldots, Q_4$ in counterclockwise order starting with the top-right quadrant.

Note that at most one ray intersects the interior of a rectangle in $\mathcal{L}$. Indeed, any two points in two different rays, $p_i \in \ell_i$ and $p_j \in \ell_j$, span an axis-parallel rectangle that intersects the interior of c. Consequently, p_i and p_j cannot be in the same rectangle in $R \setminus c$. It follows that every rectangle of $\mathcal{L}$ in $R \setminus c$ lies in one quadrant or in the union of two consecutive quadrants.

We define an aspect ratio assignment α as follows: Let $\alpha(c) = 1$. If $r \subseteq Q_1$ or $r \subseteq Q_3$, let $\alpha(r) = 6n$; and if $r \subseteq Q_2$ or $r \subseteq Q_4$, let $\alpha(r) = (6n^2)^{-1}$. For a rectangle r split by a ray, we set $\alpha(r) = 6n + (6n^2)^{-1}$ if r is split by a horizontal ray ℓ_1 or ℓ_3; and $\alpha(r) = ((6n)^{-1} + (6n^2))^{-1}$ if split by a vertical ray ℓ_2 or ℓ_4.

Suppose that a layout $\mathcal{L}'$ equivalent to $\mathcal{L}$ realizes α. Split every rectangle of aspect ratio $6n + (6n^2)^{-1}$ in $\mathcal{L}'$ horizontally into two rectangles of aspect ratios $6n$ and $(6n^2)^{-1}$. Similarly, split every rectangle of aspect ratio $((6n)^{-1}+(6n^2))^{-1}$ vertically into two rectangles of aspect ratios $6n$ and $(6n^2)^{-1}$; see Fig. 5b. By Lemma 5, there are four x- or y-monotone paths $P_1, \ldots, P_4$ from the four arms of the windwill to four distinct sides of the bounding box that pass through the slitting segments. The paths $P_1, \ldots, P_4$ subdivide the exterior of the windmill into four arrangements of rectangles, $A_1, \ldots, A_4$ that each contain a unique corner of the bounding box. By construction, every rectangle in A_1 and A_3 has aspect ratio $6n$, and every rectangle in A_2 and A_4 has aspect ratio $(6n^2)^{-1}$.

Let $R_1, \ldots, R_4$ be the bounding boxes of $A_1, \ldots, A_4$, respectively. By Lemma 6, both R_1 and R_3 have aspect ratios at least 6, and both R_2 and R_4 have aspect ratios at most $\frac{1}{6}$. By construction, the arrangements $A_1, \ldots, A_4$ each contain an arm of the windmill. This implies that $\mathrm{width}(c) < \min\{\mathrm{width}(R_1), \mathrm{width}(R_3)\}$ and $\mathrm{height}(c) < \min\{\mathrm{height}(R_2), \mathrm{height}(R_4)\}$. Consider the arrangement comprised of A_1, c, and A_3. It contains two opposite corners of R, and so its bounding box is R. Furthermore, $\mathrm{height}(R) \geq \max\{\mathrm{height}(R_1), \mathrm{height}(R_3)\}$, and

$$\begin{aligned}\mathrm{width}(R) &\leq \mathrm{width}(R_1) + \mathrm{width}(c) + \mathrm{width}(R_3) < 3\max\{\mathrm{width}(R_1), \mathrm{width}(R_3)\}\\ &\leq 3\max\left\{\frac{\mathrm{height}(R_1)}{6}, \frac{\mathrm{height}(R_3)}{6}\right\} = \frac{\max\{\mathrm{height}(R_1), \mathrm{height}(R_3)\}}{2},\end{aligned}$$

and so the aspect ratio of R is at least 2. Similarly, the bounding box of the arrangement comprised of A_2, c, and A_3 is also R, and an analogous argument implies that its aspect ratio must be at most $\frac{1}{2}$. We have shown that the aspect ratio of R is at least 2 and at most $\frac{1}{2}$, a contradiction. Thus the aspect ratio assignment α is not realizable, and so $\mathcal{L}$ is not aspect ratio universal. $\square$

We can now combine Corollary 1, Lemma 4, and Lemma 7.

Theorem 2. *A layout is aspect ratio universal iff it is one-sided and sliceable.*

Together with Lemma 1, this completes the proof of Theorem 1.

2.2 Unique Transversal Structure

Subdividing a square into squares has fascinated humanity for ages [4,12,28]. For example, a *perfect square tiling* is a tiling with squares with distinct integer side length. Schramm [26] (see also [16, Chap. 6]) proved that every near triangulation with an outer 4-cycle is the extended dual of a (possibly degenerate or nongeneric) subdivision of a rectangle into squares. The result generalizes to faces of arbitrary aspect ratios (rather than squares):

Theorem 3. *(Schramm [26, Thm. 8.1]) Let $T = (V, E)$ be near triangulation with an outer 4-cycle, and $\alpha : V^* \to \mathbb{R}^+$ a function on the set V^* of the inner vertices of T. Then there exists a unique (but possibly degenerate or nongeneric) layout $\mathcal{L}$ such that $G^*(\mathcal{L}) = T$, and for every $v \in V^*$, the aspect ratio of the rectangle corresponding to v is $\alpha(v)$.*

The caveat in Schramm's result is that all rectangles in the interior of every separating 3-cycle must degenerate to a point, rectangles in the interior of some separating 4-cycles may also degenerate to a point, and four rectangles may have a point in common. While this severely limits the scope of Schramm's result, we only use the *uniqueness* claim under the assumption that a nondegenerate and generic realization exists for a given aspect ratio assignment.

Lemma 8. *If a layout $\mathcal{L}$ is aspect ratio universal, then it admits a unique transversal structure.*

Proof. Consider the extended dual graph $T = G^*(\mathcal{L})$ of an aspect ratio universal layout $\mathcal{L}$. As noted above, T is a 4-connected inner triangulation of a 4-cycle. If T admits two different transversal structures, then there are two nonequivalent layouts, $\mathcal{L}$ and $\mathcal{L}'$, such that $T = G^*(\mathcal{L}) = G^*(\mathcal{L}')$, which in turn yield two aspect ratio assignments, α and α', on the inner vertices of T. By Theorem 3, the (nondegenerate) layouts $\mathcal{L}$ and $\mathcal{L}'$, that realize α and α', are unique. Consequently, neither of them can be aspect ratio universal. □

Lemma 8 readily shows that Theorem 1(i) implies Theorem 1(iii), and provides an alternative proof for the geometric arguments in Lemmata 4 and 7.

3 Recognizing Duals of Aspect Ratio Universal Layouts

In this section, we describe an algorithm that, for a given graph G, either finds a one-sided sliceable layout $\mathcal{L}$ whose dual graph is G, or reports that no such layout exists (Theorem 4).

Assume that we are given a near-triangulation G (that is, a plane graph where every bounded face is a triangle). A slice of a layout corresponds to an edge cut in the dual graph that contains at most two edges of the outer face. If a slice is one-sided, the edges in the edge cut form a star; and the edge cut is determined by its edges on the boundary of the outer face. A brute force algorithm would guess a slice (i.e., edge cut), and recurse on the two subproblems; it would run in exponential time. We obtain a polynomial-time algorithm using a key insight: If we already know a slice, then in each subproblem, we know two corner rectangles. Our algorithm will utilize partial information about the corner rectangles.

Problem formulation. The input of our recursive algorithm will be an *instance* $I = (G, C, P)$, where $G = (V, E)$ is a near-triangulation, $C : V(G) \to \mathbb{N}_0$ is a *corner count*, and P is a set of ordered pairs (u, v) of vertices on the outer face of G. An instance $I = (G, C, P)$ is *realizable* if there exists a one-sided sliceable layout $\mathcal{L}$ such that G is the dual graph of $\mathcal{L}$, every vertex $v \in V$ corresponds to a rectangle in $\mathcal{L}$ incident to *at least* $C(v)$ corners of $\mathcal{L}$, and every pair $(a, b) \in P$ corresponds to a pair of rectangles in $\mathcal{L}$ incident to two ccw *consecutive* corners. When we have no information about corners, then $C(v) = 0$ for all $v \in V$, and $P = \emptyset$. For convenience, we also maintain the total count $C(V) = \sum_{v \in V} C(v)$, and the set $K = \{v \in V(G) : C(v) > 0\}$ of vertices with positive corner count.

In the remainder of this section, we prove structural results for one-sided sliceable layouts (Section 3.1), and then present our algorithm (Section A.2).

3.1 Structural Properties of One-Sided Sliceable Layouts

In this section, we use a default notation: If an instance (G, C, P) is realizable by a one-sided sliceable layout $\mathcal{L}$, then R denotes the bounding box of $\mathcal{L}$, and r_v the rectangle in $\mathcal{L}$ corresponding to v. Note that any sublayout of a one-sided sliceable layout is also one-sided and sliceable (both properties are hierarchical).

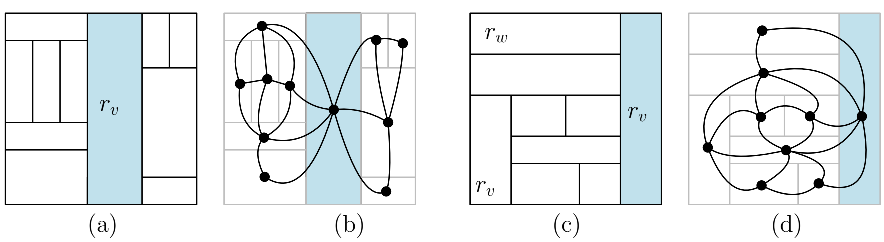


Fig. 6: (a-b) If v is a cut vertex of G, then r_v is bounded by two slices. (c-d) If there is no cut vertex, some rectangle r_v is incident to two corners.

Lemma 9. *Assume that (G, C, P) admits a realization $\mathcal{L}$ and $|V(G)| \geq 2$. Then G contains a vertex v with one of the following (mutually exclusive) properties.*

(I) Vertex v is a cut vertex in G. Then r_v is bounded by two parallel sides of R and by two parallel slices; and $C(v) = 0$. (See Fig. 6 (a-b).)

(II) Rectangle r_v is bounded by three sides of R and a slice; and $0 \le C(v) \le 2$. (See Fig. 6 (c-d).)

Proof. Let v be a cut vertex in G. Then r_v intersects the boundary of R in at least two disjoint arcs. Since both r_v and R are axis-parallel rectangles and $r_v \subset R$, their boundaries can intersect in at most two disjoint arcs, which are two parallel sides of r_v. The other two parallel sides of r_v form slices. In particular, r_v is bounded by two parallel sides of R and two slices, and so it is not incident to any corner of R. In this case, v has property (I).

Assume that G does not have cut vertices. Since $\mathcal{L}$ is sliceable, it is subdivided by a slice s which is a line segment between two opposite sides of R. Since $\mathcal{L}$ is one-sided, s must be the side of a rectangle r_v for some $v \in V(G)$. If both sides of r_v parallel to s are in the interior of R, then r_v is bounded by two sides of R and by two slices. Since the sublayouts of $\mathcal{L}$ on the opposite sides of these slices are disjoint, then v is a cut vertex in G, contrarily to our assumption. Consequently, the other side of r_v parallel to s must be a side of R. Then r_v is bounded by three sides of R and by s. Clearly, r_v is incident to precisely two corners of R, and so v has property (II). $\square$

Based on property (II), a vertex v of G is a *pivot* if there exists a one-sided sliceable layout $\mathcal{L}$ with $G \simeq G(\mathcal{L})$ in which r_v is bounded three sides of R and a slice. If we find a cut vertex or a pivot v in G, then at least one side of r_v is a slice, so we can remove v and recurse on the connected components of $G - v$. We define the subproblems created by $G - v$ in both cases:

Recursive calls. **(I)** For a cut vertex v of G in an instance, we define the operation $\textsc{Split}(G, C, P; v)$. The graph $G - v$ must have precisely two components, G_1 and G_2. Let $(u_1, \dots, w_1)$ and $(u_2, \dots, w_2)$ be the sequence of neighbors of v in G_1 and G_2, resp., in cw order. Initialize C_1 and C_2 as the restriction of C to $V(G_1)$ and $V(G_2)$, respectively. Set $C_i(u_i) \leftarrow C_i(u_i) + 1$ and $C_i(w_i) \leftarrow C_i(w_i) + 1$ for $i = 1, 2$ (if $u_i = w_i$, we increment $C_i(u_i)$ by 2). For each pair $(a, b) \in P$, if both a and b are in $V(G_i)$ for some $i \in \{1, 2\}$, then add (a, b) to P_i. Otherwise, w.l.o.g., $a \in V(G_1)$ and $b \in V(G_2)$, and the ccw path (a, b) contains either u_1, v, w_2 or w_1, v, u_2. The removal of v splits the path into two subpaths, that we add into P_1 and P_2, accordingly. Finally we add (u_1, w_1) to P_1 and (u_2, w_2) to P_2. Return the instances (G_1, C_1, P_1) and (G_2, C_2, P_2).

Lemma 10. *Let v be a cut vertex of G. An instance (G, C, P) is realizable iff both instances in $\textsc{Split}(G, C, P; v)$ are realizable. (See Appendix A.1 for the proof.)*

(II) Let v be a pivot of G. We define the operation $\textsc{Remove}(G, C, P; v)$. Since v is not a cut vertex, $G - v$ has precisely one component, denoted G'. Let $(u, \dots, w)$ be the sequence of neighbors of v in G' in cw order. Initialize C' as the restriction of C to $V(G')$, and then set $C'(u) \leftarrow C'(u) + 1$ and $C'(w) \leftarrow C'(w) + 1$. If for any $(a, b) \in P$, the ccw path from a to b in G contains v, then return $\textsc{False}$. Otherwise, set $P' = P$, and add (u, w) to P'. Return the instance (G', C', P').

Lemma 11. *Let v be a vertex of the outer face of G, but not a cut vertex. Then an instance (G, C, P) is realizable with pivot v if and only if instance* Remove$(G, C, P; v)$ *is realizable. (See Appendix A.1 for the proof.)*

How to find a pivot. It is easy to find cut vertices in G, since G is internally triangulated, then every cut vertex is incident to the outer face. In the absence of cut vertices, however, any vertex of the outer face of G might be a pivot. We use partial information on the corners to narrow down the search for a pivot.

Lemma 12. *Assume that an instance (G, C, P) admits a realization $\mathcal{L}$ and $|V(G)| \geq 2$. If $C(v) \geq 2$ for some vertex $v \in V(G)$, then v is a pivot.*

Proof. The rectangle r_v is incident to at least two corners of R. If r_v is incident to two opposite corners of R, then $r_v = R$, contradicting the assumption that G has two or more vertices. Hence r_v is incident to two consecutive corners of R, and so it contains some side s of R. The other side of r_v parallel to s is a maximal segment between two opposite sides of R, so it must be a slice. □

Lemma 13. *Assume that an instance (G, C, P) is realizable; G is 2-connected, it has 4 or more vertices; there exist two distinct vertices, u and v, such that $C(u) = C(v) = 1$, and $C(w) = 0$ for all other vertices; and $P = \{(u, v)\}$. Then u or v is a pivot; or else G has a 2-cut and a vertex of an arbitrary 2-cut can be taken to be a pivot.*

Lemma 14. *Assume that (G, C, P) admits a realization $\mathcal{L}$ and $|V(G)| \geq 2$.*

1. *If $|K| = 4$, then G has a cut vertex.*
2. *If $|K| = 3$, then G has a cut vertex or some vertex $v \in K(G)$ is a pivot.*

Proof. If G has a cut vertex, the proof is complete. Assume otherwise. By Lemma 9, for every realization $\mathcal{L}$ of the instance (G, C, P), there exists a pivot vertex v, and so r_v is incident to two corners of R. As R has only four corners, each of which is incident to a unique rectangle in $\mathcal{L}$, then at most two additional rectangles in $\mathcal{L}$ are incident to corners, hence $|K| \leq 3$.

Assume that $|K| = 3$. Since R has only four corners, each of which is incident to a unique rectangle in $\mathcal{L}$, one of the vertices in K must be v. □

Theorem 4. *We can decide in $O(n^2)$ time whether a given graph G with n vertices is the dual of a one-sided sliceable layout.*

Proof. Given a graph G, we can decide in $O(n)$ time whether G is a proper graph [10,19,21,22]. If G is proper, then it is a connected plane graph in which all bounded faces are triangles. Let an initial instance be $I = (G, C, P)$, where $C(v) = 0$ for all vertices v, and $P = \emptyset$. We describe an analyze our recursive algorithm for an instance I in Appendix **??**. □

4 Conclusions

We have shown that a rectangular layout $\mathcal{L}$ is aspect ratio universal if and only if $\mathcal{L}$ is one-sided and sliceable; and we can decide in $O(n^2)$-time whether a given graph G on n vertices is the dual of a one-sided sliceable layout. An immediate open problem is whether the runtime can be improved. Cut vertices and 2-cuts play a crucial role in our algorithm. We can show (Proposition 1) that the duals of one-sided sliceable layouts have vertex cuts of size at most 3. Perhaps 3-cuts can be utilized to find a pivot efficiently. Recall that no polynomial-time algorithm is currently known for recognizing the duals of one-sided layouts [5].

Felsner [8] distinguished between strong, weak, and dual equivalence relations over layouts. The *strong equivalence* is the same as the equivalence considered in this paper; two layouts are *dual equivalent* if their dual graphs are isomorphic; and two layouts are *weakly equivalent* if there is a bijection between their horizontal and vertical segments, resp., such that the contact graphs of the segments are isomorphic plane graphs. Felsner [8, Thm. 3] showed that every area assignment to the faces of a layout can be realized by a weakly equivalent layout, that is, every weak equivalence class is area-universal. It is an open problem to characterize weak equivalence classes of layouts that are aspect ratio universal.

References

1. Ackerman, E., Barequet, G., Pinter, R.Y.: A bijection between permutations and floorplans, and its applications. Discret. Appl. Math. **154**(12), 1674–1684 (2006)
2. Asinowski, A., Mansour, T.: Separable d-permutations and guillotine partitions. Annals of Combinatorics **14**, 17–43 (2010)
3. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: Binary space partitions. In: Computational Geometry: Algorithms and Applications, chap. 12, pp. 259–281. Springer (2008)
4. Brooks, R.L., Smith, C.A.B., Stone, A.H., Tutte, W.T.: The dissection of rectangles into squares. Duke Math. J. **7**(1), 312–340 (1940)
5. Eppstein, D., Mumford, E., Speckmann, B., Verbeek, K.: Area-universal and constrained rectangular layouts. SIAM J. Comput. **41**(3), 537–564 (2012)
6. Felsner, S.: Lattice structures from planar graphs. Electron. J. Comb. **11**(1) (2004)
7. Felsner, S.: Rectangle and square representations of planar graphs. In: Pach, J. (ed.) Thirty Essays on Geometric Graph Theory, pp. 213–248. Springer (2013)
8. Felsner, S.: Exploiting air-pressure to map floorplans on point sets. J. Graph Algorithms Appl. **18**(2), 233–252 (2014)
9. Fusy, É.: Transversal structures on triangulations: A combinatorial study and straight-line drawings. Discret. Math. **309**(7), 1870–1894 (2009)
10. Hasan, M.M., Rahman, M.S., Karim, M.R.: Box-rectangular drawings of planar graphs. J. Graph Algorithms Appl. **17**(6), 629–646 (2013)
11. He, X.: On finding the rectangular duals of planar triangular graphs. SIAM J. Comput. **22**(6), 1218–1226 (1993)
12. Henle, F.V., Henle, J.M.: Squaring the plane. Am. Math. Mon. **115**(1), 3–12 (2008)
13. Kant, G., He, X.: Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems. Theor. Comput. Sci. **172**(1-2), 175–193 (1997)

14. Koźmiński, K., Kinnen, E.: Rectangular duals of planar graphs. Networks **15**(2), 145–157 (1985)
15. Kreveld, M.J., Speckmann, B.: On rectangular cartograms. Comput. Geom. **37**(3), 175–187 (2007)
16. Lovász, L.: Graphs and Geometry, Colloquium Publications, vol. 65. AMS, Providence, RI (2019)
17. Merino, A.I., Mütze, T.: Efficient generation of rectangulations via permutation languages. In: 37th International Symposium on Computational Geometry (SoCG). LIPIcs, vol. 189, pp. 54:1–54:18. Schloss Dagstuhl (2021)
18. Mitchell, W.J., Steadman, J.P., Liggett, R.S.: Synthesis and optimization of small rectangular floor plans. Environ. Plann. B Plann. Des. **3**(1), 37–70 (1976)
19. Nishizeki, T., Rahman, M.S.: Rectangular drawing algorithms. In: Handbook on Graph Drawing and Visualization, pp. 317–348. Chapman and Hall/CRC (2013)
20. Otten, R.H.J.M.: Automatic floorplan design. In: Proc. 19th Design Automation Conference (DAC). pp. 261–267. ACM/IEEE (1982)
21. Rahman, M.S., Nakano, S., Nishizeki, T.: Rectangular grid drawings of plane graphs. Comput. Geom. **10**(3), 203–220 (1998)
22. Rahman, M.S., Nakano, S., Nishizeki, T.: Rectangular drawings of plane graphs without designated corners. Comput. Geom. **21**(3), 121–138 (2002)
23. Raisz, E.: The rectangular statistical cartogram. Geogr. Review **24**(2), 292–296 (1934)
24. Rinsma, I.: Existence theorems for floorplans. Ph.D. thesis, University of Canterbury, Christchurch, New Zealand (1987)
25. Rinsma, I.: Nonexistence of a certain rectangular floorplan with specified areas and adjacency. Environ. Plann. B Plann. Des. **14**(2), 163–166 (1987)
26. Schramm, O.: Square tilings with prescribed combinatorics. Israel Journal of Mathematics **84**, 97–118 (1993)
27. Thomassen, C.: Plane representations of graphs. In: Progress in Graph Theory, pp. 43–69. Academic Press Canada (1984)
28. Tutte, W.T.: Squaring the square. Scientific American **199**, 136–142 (1958), in Gardner's 'Mathematical Games' column. Reprinted with addendum and bibliography in the US in M. Gardner, *The 2nd Scientific American Book of Mathematical Puzzles & Diversions*, Simon and Schuster, New York (1961), pp. 186—209
29. Ungar, P.: On diagrams representing maps. Journal of the London Mathematical Society **s1-28**(3), 336–342 (1953)
30. Wimer, S., Koren, I., Cederbaum, I.: Floorplans, planar graphs, and layouts. IEEE Transactions on Circuits and Systems **35**(3), 267–278 (1988)
31. Yao, B., Chen, H., Cheng, C., Graham, R.L.: Floorplan representations: Complexity and connections. ACM Trans. Design Autom. Electr. Syst. **8**(1), 55–80 (2003)

A Appendix

A.1 Omitted Proofs

For convenience, we restate each lemma that we prove in this section.

Lemma 10. *Let v be a cut vertex of G. An instance (G, C, P) is realizable iff both instances in* $\textsc{Split}(G, C, P; v)$ *are realizable. (See Appendix A.1 for the proof.)*

Proof. First assume that $\mathcal{L}$ is a realization of instance (G, C, P). The removal of rectangle r_v splits $\mathcal{L}$ into two one-sided and sliceable sublayouts, $\mathcal{L}_1$ and $\mathcal{L}_1$. It is easily checked that they realize (G_1, C_1, P_1) and (G_2, C_2, P_2), respectively.

Conversely, if both (G_1, C_1, P_1) and (G_2, C_2, P_2) are realizable, then they are realized by some one-sided sliceable layouts $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively. The union of a square r_v and scaled copies of $\mathcal{L}_1$ and $\mathcal{L}_2$ attached to two opposite sides r_v yields a one-sided sliceable layout $\mathcal{L}$ that realizes (G, C, P). ◻

Lemma 11. *Let v be a vertex of the outer face of G, but not a cut vertex. Then an instance (G, C, P) is realizable with pivot v if and only if instance* $\textsc{Remove}(G, C, P; v)$ *is realizable. (See Appendix A.1 for the proof.)*

Proof. Assume that (G, C, P) is realized by a one-sided scliceable layout $\mathcal{L}$, and v has property (II). The removal of rectangle r_v from $\mathcal{L}$ creates a one-sided sliceable sublayouts $\mathcal{L}'$. It is easily checked that $\mathcal{L}'$ realizes the instance (G', C', P').

Conversely, assume that (G', C', P') is realized by a layout $\mathcal{L}'$. Then we can attach a single rectangle r_v to the bounding box of $\mathcal{L}'$ between two consecutive corners incident to r_u and r_w, and obtain a layout $\mathcal{L}$ that realizes (G, C, P). ◻

A.2 Algorithm and its Analysis for Theorem 4

Theorem 4. *We can decide in $O(n^2)$ time whether a given graph G with n vertices is the dual of a one-sided sliceable layout.*

Proof. Given a graph G, we can decide in $O(n)$ time whether G is a proper graph [10,19,21,22]. If G is proper, then it is a connected plane graph in which all bounded faces are triangles. Let an initial instance be $I = (G, C, P)$, where $C(v) = 0$ for all vertices v, and $P = \emptyset$. We run the following recursive algorithm.

```
1  Main(G, C, P)
2  begin
3      if |V(G)| = 1 then
4          return True
5      else if ∃v ∈ V(G) : C(v) > 2 then
6          return False
7      else if G has a cut vertex v then
8          Split(G, C, P; v) yields (G_1, C_1, P_1) and (G_2, C_2, P_2)
9          return Main(G_1, C_1, P_1) ∧ Main(G_2, C_2, P_2)
10     else if G has a vertex v with C(v) = 2 then
11         return Main(Remove(G, C, P; v))
12     else if P = {(u, v)} with C(u) = C(v) = 1 and K = 2 then
13         for all w ∈ {u, v} do
14             if Main(Remove(G, C, P; w)) then
15                 return True
16         for all vertices w ∈ {w_1, w_2} of an arbitrary 2-cut of G do
17             if Main(Remove(G, C, P; w)) then
18                 return True
19     else if |K| = 3 then
20         for all vertices v ∈ K do
21             if Main(Remove(G, C, P; v)) then
22                 return True
23     else if |K| = 0 then
24         for all vertices v in the outer face of G do
25             if Main(Remove(G, C, P; v)) then
26                 return True
27     return False
```

Correctness. We argue that algorithm $\mathsf{Main}(G, C, P)$ correctly reports whether an instance (G, C, P) is realizable.

Lines 3–4. A graph with only one vertex corresponds to a layout containing a single rectangle, which is clearly aspect ratio universal.

Lines 5–6. A rectangle that contains 3 or more corners of a layout must be the only rectangle in the layout. However, the algorithm only reaches this step if there are multiple vertices in the graph, so a vertex with a corner count of 3 or more is a contradiction.

Lines 7–9. The correctness of this step directly follows from Lemma 10.

Lines 10–26. In the absence of a cut vertex, we try to find a pivot. By Lemma 11, the instance (G, C, P) is realizable with a pivot v if and only if the instance $\textsc{Remove}(G, C, P; v)$ is realizable.

(A) **Lines 10–11.** The correctness of this step follows from Lemma 12.

(B) **Lines 12–18.** The correctness of this step follows from Lemma 13.
(D) **Lines 19–22.** By Lemma 14, when $|K(G)| = 3$, the pivot must be a vertex in K, or else the instance is not realizable.
(E) **Lines 23–26.** If we have no information about the corners and there is no cut vertex, one of the vertices in the outer face must correspond to a pivot by Lemma 9, or else (G, C, P) is not realizable.

Line 27. If we find neither a cut vertex nor a pivot, then the instance is not realizable by Lemma 11.

Runtime analysis. Let T be the recursion tree of the algorithm for some initial instance (G, C, P). The number of vertices in G strictly decreases along each descending path of T, and so the depth of the tree is $O(n)$.

We distinguish between two types of nodes in T: If a step in Lines 8–9 is executed, then the vertex set $V(G)$ is partitioned among the recursive subproblems; we call these *partition nodes* of T. In the steps in (B) Lines 12–18, (D) Lines 19–22, and (E) Lines 23–26, however, $|V(G)| - 1$ vertices appear in all four, three, or $O(|V(G)|)$ recursive subproblems; we call these *duplication nodes* of T.

We first analyze the special case that T does not have duplication nodes. Then T is a binary tree with $O(n)$ nodes. The algorithm maintains the property that G is a connected plane graph and all bounded faces are triangles; this in turn implies that any cut vertex of G is incident to the outer face. Indeed, both operations $\textsc{Split}(G, C, P; v)$ and $\textsc{Remove}(G, C, P; v)$ remove a vertex from the outer face. Any new cut vertex is incident to a vertex that has been removed. Overall, the total time taken by maintaining the set of cut vertices and the annotation C and P is $O(n)$ over the entire algorithm.

Next, we analyze the impact of duplication nodes. We claim that steps (B), (D) and (E) are reached at most once. Note first that the total corner count $C(V) = \sum_{v \in V} C(v)$ monotonically increases along any descending path of T: In the initial instance, we have $C(V) = 0$. All Split and Remove operations produce subproblems with $C(V) \geq 2$. Furthermore, if $C(V) = 4$, then we will never reach (A) or (B): Indeed, $C(V) = 4$ implies $K \neq \emptyset$. If $|K| = 2$ or 3, there would be a vertex v with $C(v) = 2$; and $\textsc{Remove}(G, C, P; v)$ in Lines 13-14 produces an instance with $C(V') = 4$, as the corner counts are incremented by two. It follows that we can reach step (D) only if $C(V) = 3$. Thus step (D) can only be reached once, when $C(V) = 3$ (after which point, $C(V)$ will be 4).

Overall, the duplication steps increase the upper bound on the runtime by a factor of $12n$, hence it is $O(n^2)$. $\square$

Lemma 15. *Let $\mathcal{L}$ be a one-sided sliceable layout such that $G(\mathcal{L})$ is 2-connected, and has a 2-cut $\{u, v\}$. Then there exists a one-sided sliceable layout $\mathcal{L}'$ with the same dual graph such that the first slice separates the rectangles corresponding to u and v. In particular, u or v is a pivot.*

Furthermore, if there are two rectangles in $\mathcal{L}$ that are each incident to a single corner of $\mathcal{L}$, then the corresponding rectangles in $\mathcal{L}'$ are also incident to some corners in $\mathcal{L}'$, or there exists a one-sided sliceable layout $\mathcal{L}''$ in which one of these rectangles is a pivot.

Proof. Let R be the bounding box of $\mathcal{L}$; let r_u and r_v denote the rectangles corresponding to u and v, and let s_0 be the segment separating r_u and r_v. If s_0 connects two opposite sides of R, the proof is complete with $\mathcal{L}' = \mathcal{L}$, so we may assume otherwise. Because $\mathcal{L}$ is one-sided, whenever two rectangles are in contact, a side of one rectangle fully contains a side of the other. We distinguish between two cases:

Case 1: r_u and r_v contact opposite sides of R. We may assume w.l.o.g. that r_u and $r - v$ contact the bottom and top side of R, respectively, and the bottom side of r_v is contained in the top side of r_u, as in Fig. 7a. Since $\mathcal{L}$ is one-sided, s_0 is a side of some rectangle in $\mathcal{L}$, and we may assume that s_0 is the bottom side of r_v. The left (resp., right) side of r_v lies either along ∂R or in a vertical segments s_1 (resp., s_2). Since s_0 does not reach both left and right sides of R, then at least one of s_1 and s_2 exists. Assume w.l.o.g. that s_1 exists. Since the bottom-left corner of r_v is the endpoint of s_0, it lies in the interior of segment s_1. The bottom endpoint of s_1 must be on the bottom side of R, or else the clockwise winding path starting with s_1 would create windmill (as it can cross neither r_u nor r_v), contradicting the assumption that $\mathcal{L}$ is sliceable. As $\mathcal{L}$ is one-sided, s_1 is the side of a rectangle r_1, which is necessarily to the left of s_1. Rectangle r_1 contacts the top and bottom sides of R. Since $G(\mathcal{L})$ is 2-connected, is does not have a cut vertex, and r_1 is the only rectangle in $\mathcal{L}$ to the left of s_1.

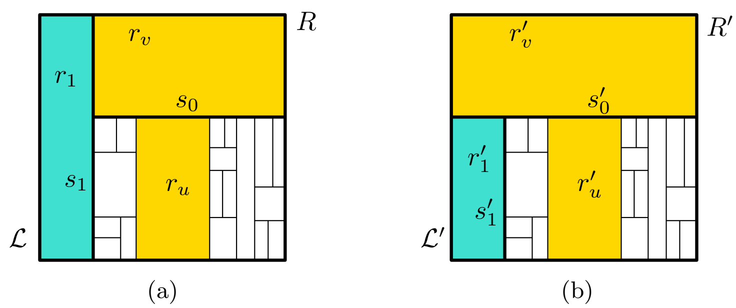
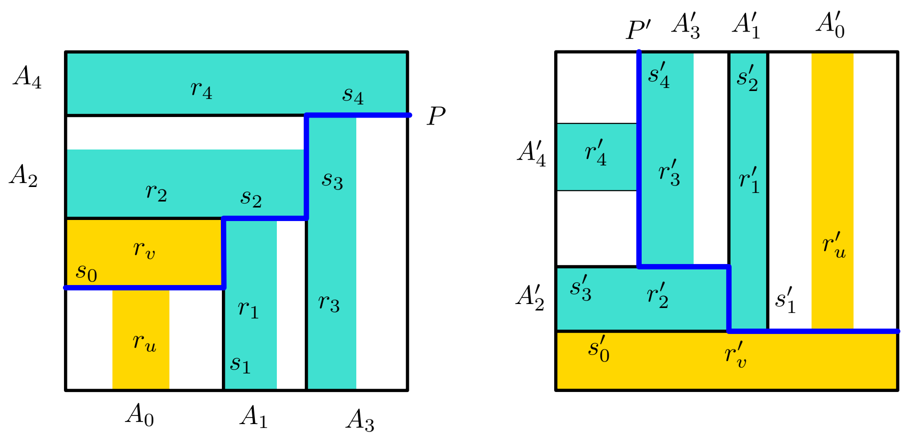


Fig. 7: (a) A one-sided sliceable layout $\mathcal{L}$ where r_u and r_v touch two opposite sides of the bounding box. (b) The modified layout $\mathcal{L}'$.

We can now modify $\mathcal{L}$ by extending s_0 and r_v horizontally to the right side of R, and clip both s_1 and r_1 to s_0, as in Fig. 7b. This modification changes the contacts between r_v and r_1 from vertical to horizontal, but does not change any other contacts in the layout, so it does not change the dual graph. If segment s_2 exists, we can similarly extend s_0 to the right side of R and clip s_2. We obtain a layout $\mathcal{L}'$ with $G(\mathcal{L}) \simeq G(\mathcal{L}')$ in which the first slice s'_0 separates r'_u and r'_v, and rectangle r'_v is a pivot. Furthermore, every rectangle incident to a corner in $\mathcal{L}$ remains incident to some corner in $\mathcal{L}'$.

Case 2: r_u and r_v do not contact opposite sides of R. Then they each contact a single side of R, and these sides are adjacent. We may assume w.l.o.g.

that r_u contacts the bottom side of $\mathcal{L}$, r_v contacts the left side of $\mathcal{L}$, and the bottom side of r_u contains the top side of r_v; refer to Fig. 8a. Because $\mathcal{L}$ is one-sided and s_0 is not the first slice, s_0 equals the bottom side of r_v.



(a) $\{u, v\}$ is a 2-cut of $G(\mathcal{L})$.

(b) Layout $\mathcal{L}'$, where r'_v is a pivot.

Fig. 8: The construction of $\mathcal{L}'$ from $\mathcal{L}$.

We incrementally construct an x- and y-monotone increasing directed path P (staircase) starting with edge s_0, directed to its right endpoint p_0. Initially, let $P = \{s_0\}$ and $i := 0$. While p_i is not in the top or right side of R, let p_{i+1} be the top or right endpoint of segment s_i, append the edge $p_i p_{i+1}$ to P, and let s_{i+1} be the segment orthogonal to s_i that contains p_{i+1}. Since the path P is x- and y-monotonically increases, it does not revisit any segment. Thus the recursion terminates, and P reaches the top or right side of R.

Assume that P is formed by the segments $s_0, s_1, \ldots, s_k$ of $\mathcal{L}$ for some $k \geq 1$. We claim that if s_i is vertical, its bottom endpoint is on the bottom side of R, and if s_i is horizontal, its right endpoint is on the right side of R. for $i = 0$ since s_0 is the bottom side of r_v, which contacts the left side of R. Suppose for contradiction that the claim holds for s_{i-1} but not for s_i. Then the the clockwise or counterclockwise winding path starting with s_i would create windmill (as it can cross neither s_{i-1} nor s_{i-2}, where s_{-1} is the right side of r_u), contradicting the assumption that $\mathcal{L}$ is sliceable.

The segments $s_0, s_1, \ldots, s_k$ jointly form a one-sided sliceable layout, that is, they subdivide R into $k + 2$ rectangular regions, each of which contains a sublayout of $\mathcal{L}$. One of these regions is r_v. Label the remaining k regions by $A_0, A_1, \ldots, A_k$ in the order in which they occur along P (see Fig. 8a). In particular, we have $r_u \subset A_0$. For $i = 1, \ldots, k$, region A_i is bounded by ∂R and segments s_i, s_{i+1}, and s_{i+2} (if they exist); and A_k is adjacent to the top-right corner of R. Because $\mathcal{L}$ is one-sided, segment s_i is a side of a rectangle that we denote by r_i, for $i = 1, \ldots, k$; and $r_i \subseteq A_i$ as the opposite side of s_i is subdivided by segment s_{i-1}.

Furthermore, we claim that $A_k = r_k$. Indeed, A_k is bounded by segment s_k and three sides of B. If $A_k \neq r_k$, then r_k separates the subarrangement $A \setminus r_k$ from r. This means that v_{r_k} would be a cut vertex in $G(\mathcal{L})$, contradicting the assumption that $G(\mathcal{L})$ is 2-connected.

We recursively construct a one-sided sliceable $\mathcal{L}'$ by placing rectangles and subarrangement corresponding to those in $\mathcal{L}$ such that $G(\mathcal{L}) \simeq G(\mathcal{L}')$; refer to Fig. 8b. Let R' be the bounding box of $\mathcal{L}'$. First subdivide R' by a horizontal segment s'_0; and let r'_v be the rectangle below s'_0. This ensures that s'_0 is the first slice and r'_v is a pivot. Subdivide the region above s'_0 by a vertical segment s'_1 into two rectangular regions. Denote the right region by A'_0, and subdivide the left region as follows: For $i = 2, \ldots, k$, recursively subdivide the rectangle incident to the top-left corner of R' by a segment s'_i orthogonal to s_i.

Segments $s'_0, \ldots, s'_k$ jointly subdivide R' into $k+2$ rectangular regions: r'_v and $A'_0, A'_1, \ldots, A'_k$ in the order in which they are created, where A'_k is incident to the top-left corner of R'; and all other regions contact either the left or the top side of R'. We insert a sublayout in each region A'_i. First insert a 180°-rotated affine copy of A_0 into A'_0. For $i = 1, \ldots, k-1$, insert r'_i into A'_i such that its top or left side is s'_{i+1}; and if $A_{i-2} \setminus r_{i-2}$ is nonempty, insert an affine copy of the sublayout $A_{i-1} \setminus r_{i-1}$ into A'_k, as well. Finally, for $i = k$, subdivide A'_k into three rectangles by slices orthogonal to s'_k: If $A_{k-2} \setminus r_{k-2}$ or $A_{k-1} \setminus r_{k-1}$ is nonempty, insert an affine copy in the first and third rectangle in A'_k; and fill all remaining space by r'_k. This completes the construction of layout $\mathcal{L}'$ (see Fig. 8b). By construction, we have $G(\mathcal{L}) \simeq G(\mathcal{L}')$.

It remains to track the rectangles incident to the corners of $\mathcal{L}$ and $\mathcal{L}'$. In the original layout $\mathcal{L}$, rectangle r_k is incident to two corners of R. Assume w.l.o.g. that r_k is incident to the two top corners of R (as in Fig. 8a), and two distinct rectangles $r_{\text{left}} \subset A_0$ and $r_{\text{right}} \subset A_{k-1}$ are incident to the bottom-left and bottom-right corners of R, respectively. The sublayout A_0 was inserted into A'_0 after a 180° rotation, and so $r'_{\text{left}} \subset A'_0$ is incident to the top-right corner in $\mathcal{L}'$. If $r_{\text{right}} \subset A_{k-1} \setminus r_{k-1}$, then $A_{k-1} \setminus r_{k-1}$ is nonempty and it was inserted into the top third of A'_k after a 180° rotation, and so r'_{right} is incident to the top-left corner in $\mathcal{L}'$. Otherwise $A_{k-1} = r_{k-1}$, and then $r_{\text{right}} = r_{k-1}$. In this case r'_k is incident to the top-left corner in $\mathcal{L}'$. However, we can modify $\mathcal{L}$ by extending r_{k-1} and s_{k-1} to the top side of R, and obtain a one-sided sliceable layout $\mathcal{L}''$ in which $r''_{\text{right}} = r''_{k-1}$ is a pivot. This completes the proof in Case 2. □

Lemma 13. *Assume that an instance (G, C, P) is realizable; G is 2-connected, it has 4 or more vertices; there exist two distinct vertices, u and v, such that $C(u) = C(v) = 1$, and $C(w) = 0$ for all other vertices; and $P = \{(u, v)\}$. Then u or v is a pivot; or else G has a 2-cut and a vertex of an arbitrary 2-cut can be taken to be a pivot.*

Proof. Assume that (G, C, P) is realized by a one-sided sliceable layout $\mathcal{L}$. The graph G is 2-connected, so it has no cut vertices. Therefore, the pivot must correspond to a rectangle in $\mathcal{L}$ that contains two corners. So, if u and v correspond to rectangles containing opposite corners of $\mathcal{L}$, then one of them must also contain another corner, and thus be a pivot.

We will assume, then, that u and v correspond to rectangles r_u and r_v which contain adjacent corners of $\mathcal{L}$, which we may assume w.l.o.g. to be the top-left and bottom-left corners, respectively. If either spans the width of $\mathcal{L}$ and contains another corner, then it corresponds to a pivot and we are done.

If r_u and r_v each contain only one corner of $\mathcal{L}$, there must be some rectangle r_p which contains the top-right and bottom-right corners of $\mathcal{L}$, or else there would be no pivot, contradicting Lemma 9. If r_u contacts r_p, then we may reverse the contact between r_u and r_p by extending the bottom side of r_u to the right side of R, and removing the segment of the left side of r_p above the extended side. This reverses the contact between the two, but yields a layout with an equivalent contact graph in which r_u contains two corners, and thus u can be taken as a pivot. The same argument can be made for v as a pivot if r_v contacts r_p.

If neither r_u nor r_v contacts r_p, then they do not contact one another either, or else the line segment separating them would not be one-sided. The layout $\mathcal{L}$ is sliceable and G is 2-connected, so there must be at least one horizontal slice from the left side of R to the left side of r_p. Let s_1 be the topmost such slice. As $\mathcal{L}$ is one-sided, then s_1 must be the side of some rectangle r_1. The rectangle r_1 can be neither r_u nor r_v, since they do not contact r_p. The vertices in G corresponding to r_1 and r_p form a 2-cut, so G has a 2-cut. Because rectangles r_u and r_v are each incident to a single corner of $\mathcal{L}$, Lemma 15 guarantees that u or v is a pivot; or for any 2-cut, there exists a one-sided sliceable layout $\mathcal{L}'$ that realizes (G, C, P) and has one of the vertices in the 2-cut as a pivot. □

B Dual Graphs of One-Sided Sliceable Layouts

In this section, we prove a remarkable property of one-sided sliceable layouts: Their dual graphs have cuts of size at most three. In contrast, the minimum vertex cut in the duals of one-sided layouts (resp., sliceable layouts) is unbounded.

Proposition 1. *Let G be the dual graph of a one-sided sliceable layout. If G has 4 or more vertices, then it contains a vertex cut of size at most 3.*

Proof. Let $\mathcal{L}$ be a one-sided sliceable layout with $n \geq 4$ rectangles in a bounding box B, and with dual graph $G = G(\mathcal{L})$. For a rectangle r in $\mathcal{L}$, let $v(r)$ denote the corresponding vertex in G. If G is outerplanar, then either G has a cut vertex, or G is a triangulated n-cycle, hence any diagonal forms a 2-cut. We may assume that G has an interior vertices.

Consider a sequence of segments that incrementally slice B into the layout $\mathcal{L}$; and let us focus on the first step that created a rectangle R_0 that lies in the interior of B. We may assume w.l.o.g. that R_0 is bounded by the segments $s_1, \ldots, s_4$ in counterclockwise order, s_4 sliced a rectangle R into R_0 and R_1, and s_4 is the bottom side of R_0. Since $\mathcal{L}$ is one-sided, then s_1 is the right side of some rectangle r_1, and then s_3 is the left side of some rectangle r_3 of $\mathcal{L}$; see Fig. 9a.

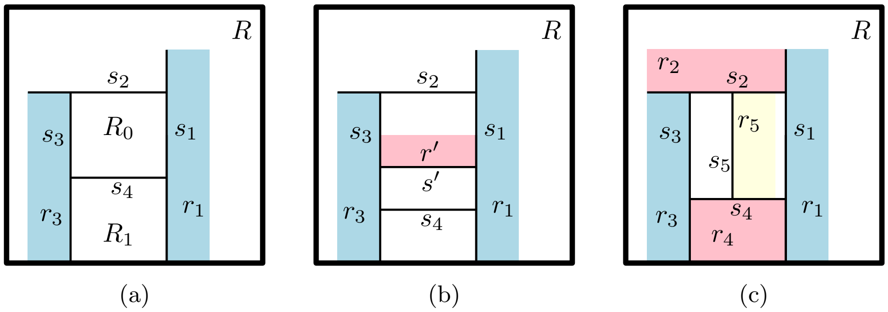


Fig. 9: Schematic views of the arrangements in the proof of Lemma **??**.

We claim that at most two horizontal segments in $\mathcal{L}$ intersect both s_1 and s_3. Indeed, if there are three or more such segments (Fig. 9b), let s' be one of them other than the lowest or highest. Then s' is a side of some rectangle r' in $\mathcal{L}$, which is adjacent to both r_1 and r_3, but not adjacent to the boundary of B, hence $\{v(r_1), v(r'), v(r_3)\}$ is a 3-cut in G. It follows that s_2 (s_4) is the highest (lowest) horizontal segment that intersects both s_1 and s_3. Since $\mathcal{L}$ is one-sided, then s_4 is the side of some rectangle r_4 in $\mathcal{L}$. If s_4 is the bottom side of a rectangle r_4 in $\mathcal{L}$, then $\{v(r_1), v(r_2), v(r_3)\}$ is a 3-cut in G. We may assume that s_4 is the top side of a rectangle r_4 in $\mathcal{L}$. Since no segment intersects both s_1 and s_3 below s_4, then $r_4 = R_1$ (as in Fig. 9c).

If R_0 is not sliced further recursively, then $\{v(r_1), v(R_0), v(r_3),\}$ is a 3-cut in G. Let s_5 be the first segment that slices R_0. Segment s_5 cannot be horizontal, as it would intersect both s_1 and s_3. So s_5 is vertical (Fig. 9c), and it is a side of some rectangle r_5 of $\mathcal{L}$, which is adjacent to s_2 and s_4. This further implies that s_2 is the bottom side of some rectangle r_2 of $\mathcal{L}$. If r_2 is adjacent to the boundary of B, then $\{v(v_4), v(r_5), v(r_2),\}$ is a 3-cut; else r_2 lies in the interior of B and $\{v(r_1), v(r_2), v(r_3)\}$ is a 3-cut in G. $\square$