



Unit-length Rectangular Drawings of Graphs

Carlos Alegría , Giordano Da Lozzo ^(✉) , Giuseppe Di Battista ,
Fabrizio Frati , Fabrizio Grosso , and Maurizio Patrignani 

Department of Engineering, Roma Tre University, Rome, Italy
{carlos.alegría,giordano.dalozzo,giuseppe.dibattista,fabrizio.frati,
fabrizio.grosso,maurizio.patrignani}@uniroma3.it

Abstract. A *rectangular drawing* of a planar graph G is a planar drawing of G in which vertices are mapped to grid points, edges are mapped to horizontal and vertical straight-line segments, and faces are drawn as rectangles. Sometimes this latter constraint is relaxed for the outer face. In this paper, we study rectangular drawings in which the edges have unit length. We show a complexity dichotomy for the problem of deciding the existence of a unit-length rectangular drawing, depending on whether the outer face must also be drawn as a rectangle or not. Specifically, we prove that the problem is NP-complete for biconnected graphs when the drawing of the outer face is not required to be a rectangle, even if the sought drawing must respect a given planar embedding, whereas it is polynomial-time solvable, both in the fixed and the variable embedding settings, if the outer face is required to be drawn as a rectangle.

Keywords: Rectangular drawings · Rectilinear drawings · Matchstick graphs · Grid graphs · SPQR-trees · Planarity

1 Introduction

Among the most celebrated aesthetic criteria in Graph Drawing we have: (i) planarity, (ii) orthogonality of the edges, (iii) unit length of the edges, and (iv) convexity of the faces. We focus on drawings in which all the above aesthetics are pursued at once. Namely, we study orthogonal drawings where the edges have length one and the faces are rectangular.

Throughout the paper, any considered graph drawing has the vertices mapped at *distinct* points of the plane. Orthogonal representations are a classic research topic in graph drawing. A rich body of literature is devoted to orthogonal drawings of planar [16, 21, 25, 50] and plane [14, 40, 41, 45, 46] graphs with the minimum number of bends in total or per edge [10, 32, 33]. An orthogonal drawing with no bend is a *rectilinear* drawing. Several papers address rectilinear

This research was partially supported by MIUR Project “AHeAD” under PRIN 20174LF3T8, and by H2020-MSCA-RISE project 734922 – “CONNECT”.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
P. Angelini and R. von Hanxleden (Eds.): GD 2022, LNCS 13764, pp. 127–143, 2023.
https://doi.org/10.1007/978-3-031-22203-0_10

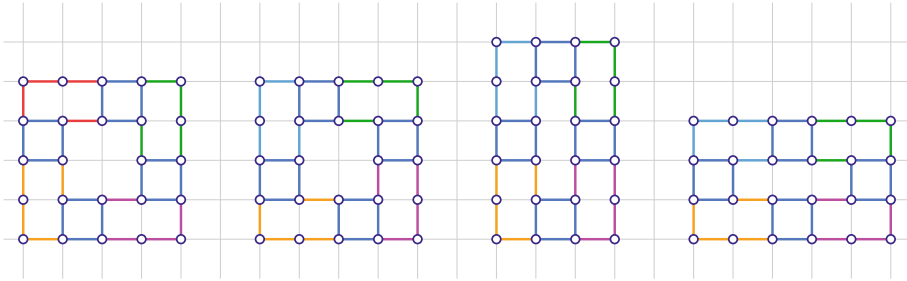


Fig. 1. Unit-length embedding-preserving rectangular drawings of a plane graph.

drawings of planar [13, 24, 26, 29, 37, 38] and plane [20, 24, 43, 49] graphs. When all the faces of a rectilinear drawing have a rectangular shape the drawing is *rectangular*. Maximum degree-3 plane graphs admitting rectangular drawings were first characterized in [47, 48]. A linear-time algorithm to find a rectangular drawing of a maximum degree-3 plane graph, provided it exists, is described in [39] and extended to maximum degree-3 planar graphs in [42]. Surveys on rectangular drawings can be found in [23, 35, 36]. If only the internal faces are constrained to be rectangular, then the drawing is called *inner-rectangular*. In [34] it is shown that a plane graph G has an inner-rectangular drawing Γ if and only if a special bipartite graph constructed from G has a perfect matching. Also, Γ can be found in $O(n^{1.5}/\log n)$ time if G has n vertices and a “sketch” of the outer face is prescribed, i.e., all the convex and concave outer vertices are prescribed.

Computing straight-line drawings whose edges have constrained length is another core topic in graph drawing [1, 2, 4, 7, 12, 22, 44]. The graphs admitting planar straight-line drawings with all edges of the same length are also called *matchstick graphs*. Recognizing matchstick graphs is NP-hard for biconnected [22] and triconnected [12] graphs, and in fact, even strongly $\exists\mathbb{R}$ -complete [1]; see also [44].

A *unit-length grid drawing* maps vertices to grid points and edges to horizontal or vertical segments of unit Euclidean length. A *grid graph* is a graph that admits a unit-length grid drawing¹. Recognizing grid graphs is NP-complete for ternary trees of pathwidth 3 [9], for binary trees [27], and for trees of pathwidth 2 [28], but solvable in polynomial time on graphs of pathwidth 1 [28]. An exponential-time algorithm to compute, for a given weighted planar graph, a rectilinear drawing in which the Euclidean length of each edge is equal to the edge weight has been presented in [7].

Let G be a planar graph. The UNIT-LENGTH INNER-RECTANGULAR DRAWING RECOGNITION (for short, UIR) problem asks whether a unit-length inner-rectangular drawing of G exists. Similarly, the UNIT-LENGTH RECTANGULAR DRAWING RECOGNITION (for short, UR) problem asks whether a unit-length

¹ Note that in some literature the term “grid graph” denotes an “induced” graph, i.e., there is an edge between any two vertices at distance one. See, for example, [31].

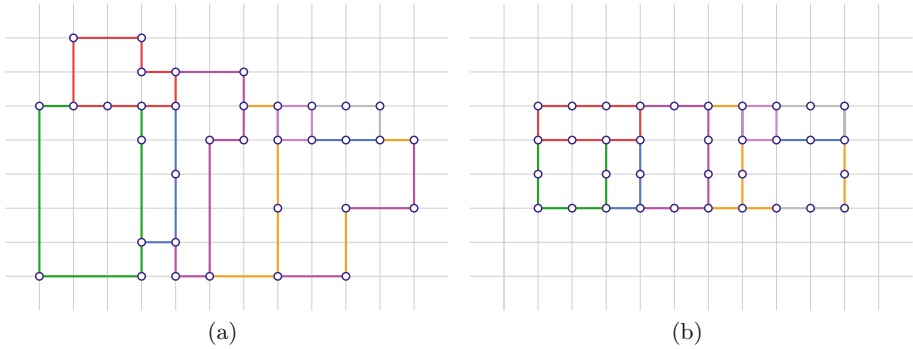


Fig. 2. (a) A planar rectilinear grid drawing of a graph. (b) A unit-length rectangular grid drawing of the same graph.

rectangular drawing of G exists. Let now H be a plane *or* planar embedded (i.e., no outer face specified) graph. The UNIT-LENGTH INNER-RECTANGULAR DRAWING RECOGNITION WITH FIXED EMBEDDING (for short, UIRFE) problem asks whether a unit-length inner-rectangular embedding-preserving drawing of H exists. Similarly, the UNIT-LENGTH RECTANGULAR DRAWING RECOGNITION WITH FIXED EMBEDDING (for short, URFE) problem asks whether a unit-length rectangular embedding-preserving drawing of H exists; see Fig. 1.

Our Contribution. In Sect. 3 we show NP-completeness for the UIRFE and UIR problems when the input graph is biconnected, which is surprising since a biconnected graph has degrees of freedom that are more restricted than those of a tree. In Sect. 4 we provide a linear-time algorithm for the UIRFE and URFE problems if the drawing of the outer face is given. In Sect. 5 we first show that the URFE problem is cubic-time solvable; the time bound becomes linear if all internal faces of the input graph have maximum degree 6. These results hold both when the outer face is prescribed and when it is not. Second, we show a necessary condition for an instance of the UR problem to be positive in terms of its SPQR-tree. Exploiting the above condition, we show that the UR problem is cubic-time solvable; the running time becomes linear when the SPQR-tree of the input graph satisfies special conditions. Finally, as a by-product of our research, we provide the first polynomial-time algorithm to test whether a planar graph G admits a rectangular drawing, for general instances of maximum degree 4.

Missing details for the proofs of the statements marked with a (\star) are given in [3].

2 Preliminaries

For basic graph drawing terminology and definitions refer, e.g., to [15, 35].

Drawings and Embeddings. Two planar drawings of a connected graph are *planar equivalent* if they induce the same counter-clockwise ordering of the edges

incident to each vertex. Also, they are *plane equivalent* if they are planar equivalent and the clockwise order of the edges along the boundaries of their outer faces is the same. The equivalence classes of planar equivalent drawings are called *planar embeddings*, whereas the equivalence classes of plane equivalent drawings are called *plane embeddings*. A *planar embedded graph* is a planar graph equipped with one of its planar embeddings. Similarly, a *plane graph* is a planar graph equipped with one of its plane embeddings. Given a planar embedded (resp. plane) graph G and a planar (resp. plane) embedding \mathcal{E} of G , a planar drawing Γ of G is *embedding-preserving* if $\Gamma \in \mathcal{E}$.

In a *grid drawing*, vertices are mapped to points with integer coordinates (i.e., *grid points*). A drawing of a graph in which all edges have unit Euclidean length is a *unit-length drawing* (see Fig. 2 for an example).

Observation 1. *A unit-length grid drawing is rectilinear and planar.*

Observation 2. *A unit-length rectangular (or inner-rectangular) drawing is planar and it is a grid drawing, up to a rigid transformation.*

The following simple property has been proved in [6, Lemma 1].

Property 1. Every cycle that admits a unit-length grid drawing has even length.

Since (inner) rectangular drawings exist only for maximum-degree-4 graphs, in the remainder, we assume that all considered graphs satisfy this requirement.

Connectivity. A *biconnected component* (or *block*) of a graph G is a maximal (in terms of vertices and edges) biconnected subgraph of G . A block is *trivial* if it consists of a single edge and *non-trivial* otherwise. A *split pair* of G is either a pair of adjacent vertices or a *separation pair*, i.e., a pair of vertices whose removal disconnects G . The *components* of G with respect to a split pair $\{u, v\}$ are defined as follows. If (u, v) is an edge of G , then it is a component of G with respect to $\{u, v\}$. Also, let G_1, \dots, G_k be the connected components of $G \setminus \{u, v\}$. The subgraphs of G induced by $V(G_i) \cup \{u, v\}$, minus the edge (u, v) , are components of G with respect to $\{u, v\}$, for $i = 1, \dots, k$. Due to space limitations, we refer the reader to [3] and to [17, 18] for the definition of SPQR-tree.

3 NP-Completeness of the UIRFE and UIR Problems

In this section we show NP-completeness for both the UIRFE and UIR problems when the input graph is biconnected. We start with the following theorem.

Theorem 1. *The UIRFE problem is NP-complete, even for biconnected plane graphs whose internal faces have maximum size 6.*

Let ϕ be a Boolean formula in conjunctive normal form with at most three literals in each clause. We denote by G_ϕ the *incidence graph* of ϕ , i.e., the graph that has a vertex for each clause of ϕ , a vertex for each variable of ϕ , and an edge

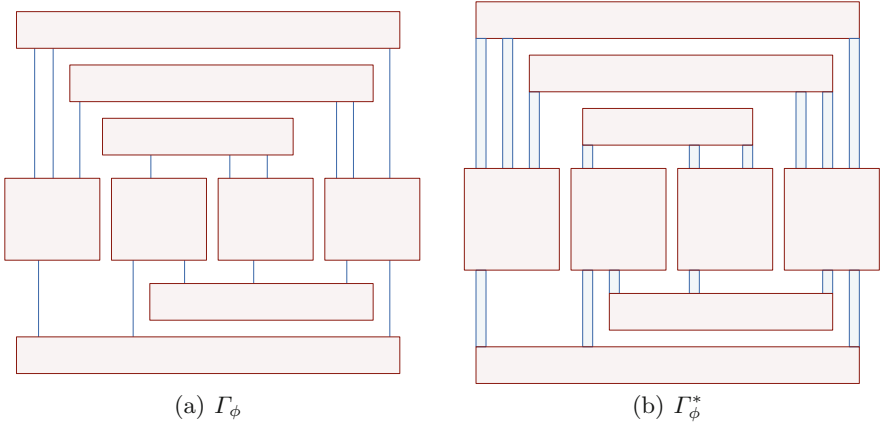


Fig. 3. (a) The monotone rectilinear representations Γ_ϕ of G_ϕ . The rectangles representing variables and clauses are red, whereas the line segments and rectangles representing the edges of ϕ are blue. (b) The auxiliary representation Γ_ϕ^* . (Color figure online)

(c, v) for each clause c that contains the *positive literal* v or the *negated literal* \bar{v} . The formula ϕ is an instance of PLANAR MONOTONE 3-SAT if G_ϕ is planar and each clause of ϕ is either positive or negative. A *positive clause* contains only positive literals, while a *negative clause* contains only negated literals. Hereafter, w.l.o.g., we assume that all the clauses of ϕ contain *exactly* three literals.

A *monotone rectilinear representation* of G_ϕ is a drawing that satisfies the following properties (refer to Fig. 3a). **P1:** Variables and clauses are represented by axis-aligned rectangles with the same height. **P2:** The bottom sides of all rectangles representing variables lie on the same horizontal line. **P3:** The rectangles representing positive (resp. negative) clauses lie above (resp. below) the rectangles representing variables. **P4:** Edges connecting variables and clauses are represented by vertical segments. **P5:** The drawing is crossing-free.

The PLANAR MONOTONE 3-SAT problem is known to be NP-complete, even when the incidence graph G_ϕ of ϕ is provided along with a monotone rectilinear representation Γ_ϕ of G_ϕ [8]. We prove Theorem 1 by showing how to construct a plane graph H_ϕ that is biconnected, has internal faces of maximum size 6, and admits a unit-length inner-rectangular drawing *if and only if* ϕ is satisfiable. Our strategy is to modify Γ_ϕ to create a suitable auxiliary representation Γ_ϕ^* (see Fig. 3) and then to use the geometric information of Γ_ϕ^* as a blueprint to construct H_ϕ . Because of the lack of space, we describe in detail how to obtain Γ_ϕ^* from Γ_ϕ and how to construct H_ϕ in the full version [3]. We provide below a high-level description of the logic behind the reduction.

Overview of the Reduction. The reduction is based on three main types of gadgets. A variable $v \in \phi$ is modeled by means of a *variable gadget*, a clause $c \in \phi$

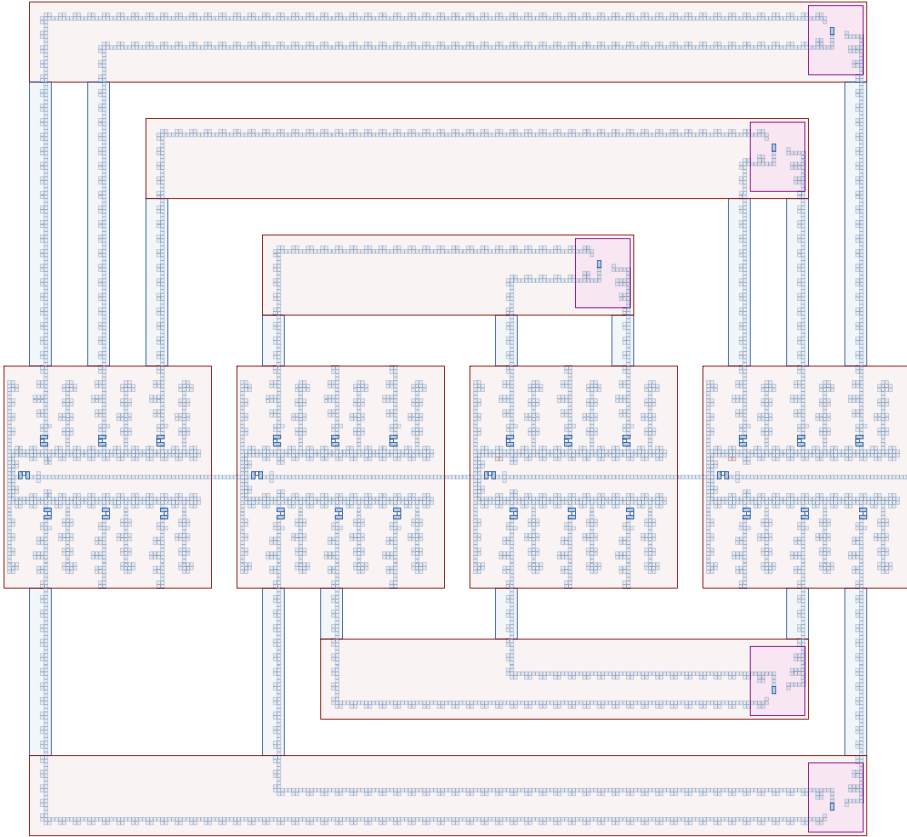


Fig. 4. The graph H_ϕ . Variable and clause gadgets are enclosed in light red boxes, while transmission gadgets are enclosed in light blue boxes. (Color figure online)

by means of an (α, β) -clause gadget, and an edge $(v, c) \in G_\phi$ by means of a λ -transmission gadget. We use the geometric properties of I_ϕ^* to determine the size and structure of each gadget, as well as how to combine the gadgets together to form H_ϕ . The width and height of the rectangles representing variables, clauses, and edges are used to construct variable gadgets and to compute the auxiliary parameters α , β and λ , which in turn are used to construct (α, β) -clause gadgets and λ -transmission gadgets. Finally, the incidences between the rectangles are used to decide how to join the gadgets to construct a single connected graph.

An example of a unit-length inner-rectangular drawing of H_ϕ is shown in Fig. 4; some faces of H_ϕ are omitted. All these missing faces are part of *domino components*, which admit a constant number of unit-length inner-rectangular drawings, see Fig. 5; some of these faces are shown filled in white or blue in Fig. 4.

The logic behind the construction is as follows. A variable gadget admits two unit-length inner-rectangular drawings (see Fig. 6), which differ from each other

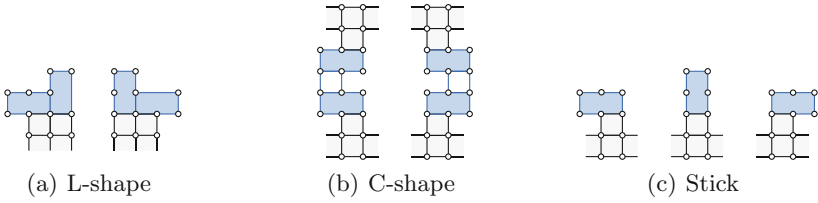
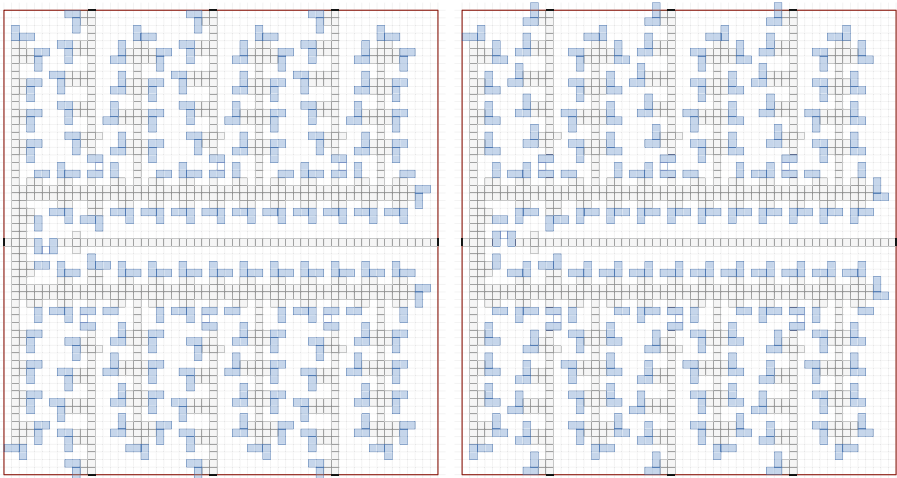


Fig. 5. The unit-length grid drawings of the domino components. Domino component faces are filled blue (size 6) and white (size 4). (Color figure online)



(a) A true configuration. (b) A false configuration

Fig. 6. The variable gadget.

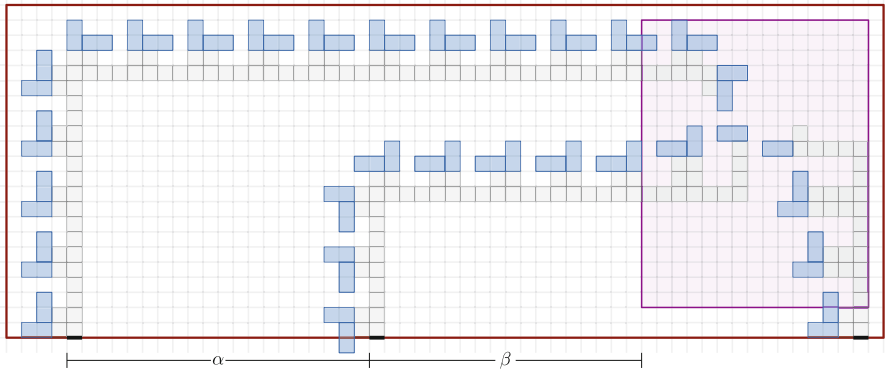


Fig. 7. In every unit-length inner-rectangular drawing of an (α, β) -clause gadget, at least one L-shape domino component crosses the red rectangle. (Color figure online)

on whether the domino components of the gadget stick out of the bottom or top side of the red enclosing rectangle, and correspond to a **true/false** assignment for the associated variable, respectively. The truth assignments are propagated from variable to clause gadgets via λ -transmission gadgets. A domino component sticking out of a variable gadget invades a transmission gadget, which causes a domino component at the other end of the transmission gadget to be directed towards the incident (α, β) -clause gadget. The clause gadget is designed so that it admits a unit-length inner-rectangular drawing if and only if at least one of the extremal domino components of its three incident transmission gadgets is not directed towards it; this allows a domino component of the clause gadget to invade the transmission gadget and save space inside the clause gadget; see Fig. 7.

By showing that all the unit-length inner-rectangular drawings of H_ϕ respect the same plane embedding, we prove the following theorem.

Theorem 2 (\star). *The UIR problem is NP-complete, even for biconnected plane graphs whose internal faces have maximum size 6.*

4 An Algorithm for the UIRFE and URFE Problems with a Prescribed Drawing of the Outer Face

Consider a connected instance of the UIRFE problem, i.e., an n -vertex connected plane graph G ; let \mathcal{E} be the plane embedding prescribed for G . Let Γ_o be a unit-length grid drawing of the walk bounding the outer face f_o of \mathcal{E} . W.l.o.g., assume that the smallest x - and y - coordinates of the vertices of Γ_o are equal to 0. Next, we describe an $O(n)$ -time algorithm, called RECTANGULAR-HOLES ALGORITHM, to decide whether G admits a unit-length inner-rectangular drawing that respects \mathcal{E} and in which the walk bounding f_o is represented by Γ_o .

We first check whether each internal face of \mathcal{E} is bounded by a simple cycle of even length, as otherwise the instance is negative by Property 1. This can be trivially done in $O(n)$ time. Consider the plane graph obtained from G by removing the bridges incident to the outer face and the resulting isolated vertices. A necessary condition for G to admit an inner-rectangular drawing is that the resulting graph contains no trivial block. This can be tested in $O(n)$ time [30].

The algorithm processes the internal faces of G one at a time. When a face f is considered, the algorithm either detects that G is a negative instance or assigns x - and y - coordinates to all the vertices of f . In the latter case, we say that f is *processed* and its vertices are *placed*. Since the drawing of f_o is prescribed, at the beginning each vertex incident to f_o is placed, while the remaining vertices are not. Also, every internal face of \mathcal{E} is not processed. The algorithm concludes that the instance is negative if one of the following conditions holds: **(C1)** there is a placed vertex to which the algorithm tries to assign coordinates different from those already assigned to it, or **(C2)** there are two placed vertices with the same x -coordinate and the same y -coordinate. If neither Condition C1 nor C2 occurs, after processing all the internal faces the vertex placement provides a unit-length inner-rectangular drawing of the input instance.

To process faces, the algorithm maintains some auxiliary data structures:

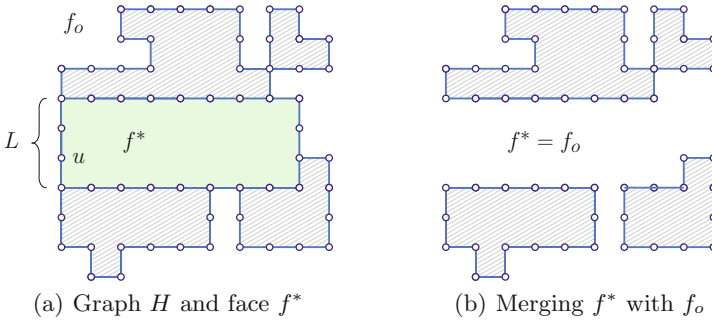


Fig. 8. A step of the RECTANGULAR-HOLES ALGORITHM.

- A **graph** H , called the CURRENT GRAPH, which is the subgraph of G composed of the vertices and of the edges incident to non-processed (internal) faces. Initially, we have $H = G$. In particular, we will maintain the invariant that each biconnected component of H is non-trivial. We will also maintain the outer face of the restriction \mathcal{E}_H of \mathcal{E} to H , which we will still denote by f_o .
- An **array** A , called the CURRENT OUTER-SORTER, that contains $M_x + 1$ buckets, each implemented as a double-linked list, where M_x is the largest x -coordinate of a vertex in Γ_o . The bucket $A[i]$ contains the placed vertices of H (i.e., those incident to the outer face of H) whose x -coordinate is equal to i . Moreover, A is equipped with the index x_{\min} of the first non-empty bucket. To allow removals of vertices in $O(1)$ time, we enrich each placed vertex with x -coordinate i with a pointer to the corresponding list-item in the list $A[i]$.
- A **set of pointers** for the edges of H : Each edge (u, v) is equipped with two pointers ℓ_{uv} and ℓ_{vu} , that reference the faces of \mathcal{E} lying to the left of (u, v) , when traversing such an edge from u to v and from v to u , respectively.

At each iteration the algorithm performs the following steps; see Fig. 8. **Retrieve:** It retrieves an internal face f^* with at least one vertex u with minimum x -coordinate (i.e., x_{\min}) among the placed vertices of H ; such a vertex is incident to the outer face of H . **Draw:** It assigns coordinates to all the vertices incident to f^* in such a way that f^* is drawn as a rectangle R^* . Note that such a drawing is unique as the left side of R^* in any unit-length grid drawing of H with the given drawing of f_o coincides with the maximal path L containing u that is induced by all the placed vertices of f^* with x -coordinate equal to x_{\min} . **Merge:** It merges f^* with f_o by suitably changing the pointers of every edge incident to f^* , and by removing each edge (u, v) with pointers $\ell_{uv} = \ell_{vu} = f_o$, as well as any resulting isolated vertex. Further, it updates A consequently. Note that, after the merge step, the outer face f_o of the new current graph H is completely drawn. This invariant is maintained through each iteration of the algorithm. In [3], we describe each step in detail.

The proof of the next theorem exploits the RECTANGULAR-HOLES ALGORITHM.

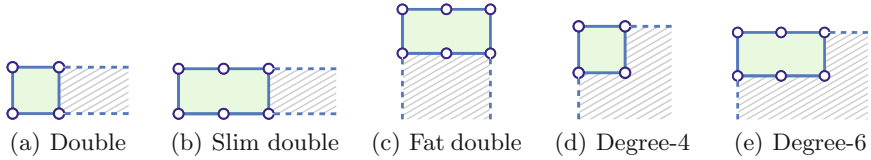


Fig. 9. Corner faces for the proof of Theorem 4.

Theorem 3 (*). *The UIRFE and URFE problems are $O(n)$ -time solvable for an n -vertex connected plane graph, if the drawing of the outer face is prescribed.*

Since any unit-length grid drawing of a cycle with 4 or 6 vertices is a rectangle, the previous theorem implies the following result, which contrasts with the NP-hardness of Theorem 1, where the drawing of the outer face is not prescribed.

Corollary 1. *The UIRFE problem is linear-time solvable if the drawing of the outer face is prescribed and all internal faces have maximum degree 6.*

5 Algorithms for the URFE and UR Problems

In this section we study the UR problem. Since rectangular drawings are convex, the input graphs for the UR problem must be biconnected [19].

Fixed Embedding. We start by considering instances with either a prescribed plane embedding (Theorem 4) or a prescribed planar embedding (Theorem 5).

Theorem 4 (*). *The URFE problem is cubic-time solvable for a plane graph G and it is linear-time solvable if all internal faces of G have maximum degree 6.*

Proof (sketch). To solve the problem in cubic time, we examine the quadratically-many drawings of the outer face f_o , and invoke Theorem 3 for each of them.

Assume now that all internal faces have maximum degree 6. We efficiently determine $O(1)$ possible rectangular drawings of f_o and then invoke Theorem 3 for each of them. If G is a 4-cycle or a 6-cycle, then the instance is trivially positive. Refer to Fig. 9. A *double corner face* is a degree-4 face with three edges incident to f_o . A *slim double corner face* is a degree-6 face with five edges incident to f_o . A *fat double corner face* is a degree-6 face with four edges incident to f_o . Note that each of such faces must provide two consecutive 270° angles incident to f_o . Hence, if G has at least one of the above faces, the drawing of f_o is prescribed, and hence RECTANGULAR-HOLES ALGORITHM can be invoked.

Suppose now that none of the above cases holds. A *corner face* is a degree-4 (degree-6) face that has two (resp. three) edges incident to f_o . Each corner face provides a 270° angle incident to any realization of f_o as a rectangle. Hence, there must be exactly four corner faces in order for G to be a positive instance. These faces can be computed in linear time, and determine $O(1)$ possible drawings of the outer face on which we invoke RECTANGULAR-HOLES ALGORITHM. \square

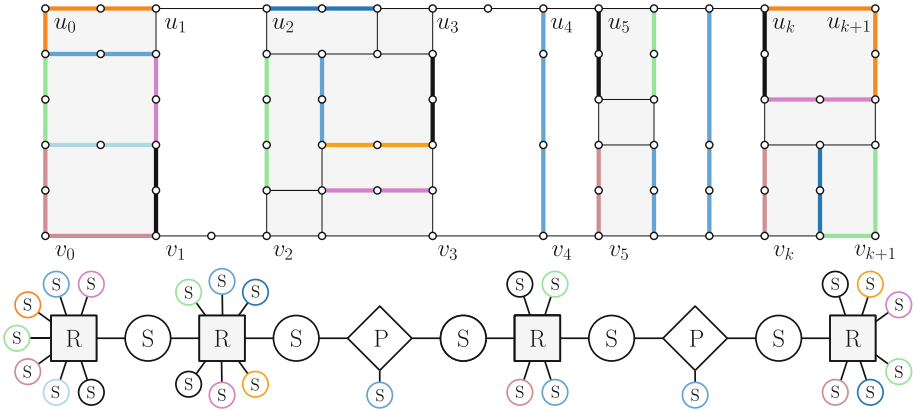


Fig. 10. A rectangular unit-length grid drawing of a planar graph and its pruned SPQR-tree T^* . S -, P -, and R -nodes are circles, rhombuses and squares, respectively. The subgraphs corresponding to S -nodes that are leaves of T^* are thick.

By showing that any planar embedding has a unique candidate outer face supporting a unit-length rectangular drawing, we get the following.

Theorem 5 (\star). *The URFE problem is cubic-time solvable for a planar embedded graph G , and it is linear-time solvable if all but at most one face of G have maximum degree 6.*

Variable Embedding. Now, we turn our attention to instances with a variable embedding. We start by providing some relevant properties of the graphs that admit a rectangular (not necessarily unit-length or grid) drawing. Let G be one such graph. To avoid degenerate cases, in what follows, we assume that G is not a cycle (cfr. Property 1). Let Γ be a rectangular drawing of G and let Γ_o be the rectangle delimiting the outer face of Γ . Refer to Fig. 10. Consider the plane graph G_Γ corresponding to Γ . Since Γ is convex, then G_Γ is a subdivision of an *internally triconnected* plane graph [5, Theorem 1]. That is, every separation pair $\{u, v\}$ of G_Γ is such that u and v are incident to the outer face and each connected component of $G_\Gamma \setminus \{u, v\}$ contains a vertex incident to the outer face.

A *caterpillar* is a tree such that removing its leaves results in a path, called *spine*. The *pruned SPQR-tree* of a biconnected planar graph G , denoted by T^* , is the tree obtained from the SPQR-tree T of G , after removing the Q -nodes of T .

Lemma 1 (\star). *Let G be a graph that admits a rectangular drawing. Then the pruned SPQR-tree T^* of G is a caterpillar with the following properties: (i) All its leaves are S -nodes; (ii) its spine contains no two adjacent R -nodes; (iii) its spine contains no two adjacent nodes μ and ν , such that μ is a P -node and ν is an R -node; (iv) each P -node μ has exactly 3 neighbors; and (v) the skeleton of each S -node of the spine of T^* contains two chains of virtual edges corresponding to Q -nodes, separated by two virtual edges each corresponding to either a P - or an R -node.*

Proof (sketch). Let Γ be a rectangular drawing of G and let Γ_o be the rectangle bounding the outer face of Γ . By inspecting Γ “from left to right”, we argue about the structure of T^* , which ultimately leads to prove the statement of the lemma; refer to Fig. 10. At each point of the inspection, T^* will be a caterpillar whose spine does not have a P -node as an end-point. Also, a leaf will be denoted as *active* and will be used as an attachment endpoint to extend T^* .

Let $S = [\{u_1, v_1\}, \{u_2, v_2\}, \dots, \{u_k, v_k\}]$ be the separation pairs of G such that both u_i and v_i lie on opposite sides of Γ_o , have degree 3, and share the same x -coordinate, for $i = 1, \dots, k$, sorted in increasing order of their x -coordinate. In [3], we provide properties of rectangular drawings that show that these pairs are the only ones that correspond to poles of P - and R -nodes of T^* . We set $L = \{u_0, v_0\} \circ S \circ \{u_{k+1}, v_{k+1}\}$, where u_0, u_{k+1}, v_{k+1} , and v_0 are the vertices on the top-left, top-right, bottom-right, and bottom-left corner of Γ_o .

Consider any two consecutive pairs $\{u_i, v_i\}$ and $\{u_{i+1}, v_{i+1}\}$, for $i = 0, \dots, k$. We can define a cycle C_i in G that contains u_i, u_{i+1}, v_{i+1} , and v_i , and that is drawn as a rectangle in Γ . Moreover, any two cycles C_i and C_{i+1} share a path P_{i+1} that is drawn as a straight-line segment in Γ . We denote by G_i the subgraph of G induced by the vertices in the interior and along the boundary of C_i .

We skip the discussion for the consecutive pairs $\{u_0, v_0\}$ and $\{u_1, v_1\}$. For $i = 1, \dots, k$, consider the separation pair $\{u_i, v_i\}$. Let ξ be the active endpoint of the spine. In the following, we denote by $\text{sk}(\mu)$ the skeleton of a node μ of T^* . Two cases are possible: ξ is either an S - or an R -node.

Suppose that $G_i = C_i$. If ξ is an S -node, then we introduce a P -node $\mu_{i,1}$ in T^* adjacent to ξ and to two new S -nodes $\mu_{i,2}$ and $\mu_{i,3}$. We have that $\text{sk}(\mu_{i,1})$ is a bundle of three parallel edges (u_i, v_i) , $\text{sk}(\mu_{i,2})$ is a cycle containing one virtual edge for each edge of the path P_i plus a virtual edge (u_i, v_i) , and $\text{sk}(\mu_{i,3})$ is a cycle consisting of a virtual edge (u_i, v_i) , followed by one virtual edge for each horizontal edge in the top side of C_i , followed by one virtual edge (u_{i+1}, v_{i+1}) , followed by one virtual edge for each horizontal edge in the bottom side of C_i . We set S -node $\mu_{i,3}$ as the active node of T^* . If ξ is an R -node, then we introduce an S -node μ_i in T^* adjacent to ξ whose skeleton is a cycle consisting of a virtual edge (u_i, v_i) , followed by one virtual edge for each horizontal edge in the top side of C_i , followed by a path P^* of virtual edges defined below, followed by one virtual edge for each horizontal edge in the bottom side of C_i . If $i < k$, then P^* consists of the single virtual edge (u_{i+1}, v_{i+1}) ; otherwise, if $i = k$, then P^* contains a virtual edge for each real edge incident to the right side of Γ_o . We set the S -node μ_i as the active endpoint of T^* , unless $i = k$.

Suppose now that $G_i \neq C_i$. In this case, G_i is the subdivision of a triconnected planar graph. We introduce an R -node μ_i in T^* adjacent to ξ and to the S -nodes corresponding to the components of G_i , with respect to its split pairs, that are simple paths. We add to $\text{sk}(\mu_i)$ a virtual edge for each of such paths, as well as (u_i, v_i) and (u_{i+1}, v_{i+1}) , unless $i = k$. We set the R -node μ_i as the active endpoint of T^* , unless $i = k$. \square

Consider a graph G that satisfies the conditions of Lemma 1. If the spine of the pruned SPQR-tree of G contains at least two nodes or at least one P -node,

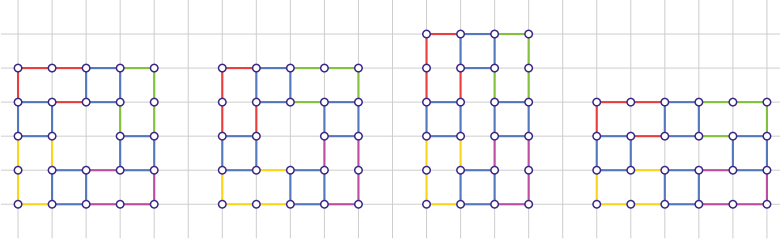


Fig. 11. Four plane embeddings of a graph G that support a rectangular drawing of G , obtained by selecting one of the plane embeddings \mathcal{E}_1 and \mathcal{E}_4 of the subgraph G_0 of G and one of the the plane embeddings \mathcal{E}_2 and \mathcal{E}_3 of the subgraph G_4 of G . Only the embeddings \mathcal{E}_1 and \mathcal{E}_2 support a unit-length rectangular drawing.

we say that G is *flat*; otherwise, G is the subdivision of a triconnected planar graph. Exploiting Lemma 1, we can prove the following; refer to Fig. 11.

Lemma 2 (\star). *Let G be an n -vertex graph. The following hold:*

- All the unit-length rectangular drawings of G , if any, have the same plane embedding \mathcal{E} (up to a reflection), which can be computed in $O(n)$ time.
- If G is flat, all the rectangular drawings of G , if any, have at most four possible plane embeddings (up to a reflection), which can be computed in $O(n)$ time.

The next theorem shows that the UR problem is polynomial-time solvable. Surprisingly, the problem seems to be harder for non-flat instances.

Theorem 6 (\star). *Let G be a planar graph. The UR problem is cubic-time solvable for G . Also, if G is flat, then the UR problem is linear-time solvable.*

Proof (sketch). First, we test whether G satisfies the conditions of Lemma 1, which can clearly be done in $O(n)$ time by computing and visiting T^* , and reject the instance if this test fails. Then, by Lemma 2, we compute in $O(n)$ time the unique candidate plane embedding \mathcal{E} of G that may support a unit-length rectangular drawing of G , if any, and reject the instance if such an embedding does not exist. Let f_o be the outer face of \mathcal{E} . If the spine of T^* consists of a single R -node, then \mathcal{E} coincides with the unique planar embedding of G , and we test for the existence of such a drawing using Lemma 5 in $O(n^3)$ time. If G is flat, then we can show that there exists a unique candidate drawing Γ_o of f_o . Then, we use Theorem 3 to test in $O(n)$ time whether a unit-length rectangular drawing of G exists that respects \mathcal{E} and such that f_o is drawn as Γ_o . \square

Theorem 7 (\star). *Let G be an n -vertex planar graph. The problem of testing for the existence of a rectangular drawing of G is solvable in $O(n^2 \log^3 n)$ time. Also, if G is flat, then this problem is solvable in $O(n \log^3 n)$ time.*

Proof (sketch). Assume that G satisfies the conditions of Lemma 1. If G is flat, then Lemma 2 guarantees the existence of only up to four plane embeddings

of G that are candidates for a rectangular drawing of G that respects them. Otherwise, G is the subdivision of a triconnected planar graph, and there exists $O(n)$ candidate plane embeddings. For each of them, we test for the existence of a rectangular drawing respecting it by solving a max-flow problem on a linear-size planar network with multiple sources and sinks in $O(n \log^3 n)$ time [11]. Such a network can be defined following Tamassia's [15] classic approach to test for the existence of rectilinear drawings of plane graphs. \square

6 Conclusions and Open Problems

We studied the recognition of graphs admitting the beautiful drawings that require unit-length and orthogonality of the edges, planarity, and convexity of the faces. We show that, if the outer face is drawn as a rectangle, the problem is polynomial-time solvable, while it is NP-hard if the outer face is an arbitrary polygon (even if the input is biconnected), unless such a polygon is specified in advance. These results hold both in the fixed-embedding and in the variable-embedding settings. A byproduct of our results is a polynomial-time algorithm to recognize graphs admitting a rectangular (non-necessarily unit-length) drawing.

It is worth remarking that if the input is a subdivision of a triconnected planar graph, then our algorithms pay an extra time to handle the outer face. Specifically, for the rectangular unit-length setting, an extra quadratic time is used to guess a rectangular drawing of the unique candidate outer face, while, for the general rectangular setting, an extra linear time is used to determine the actual candidate outer face. Hence, it is appealing to study efficient algorithms for this specific case. Further, it is interesting to determine the complexity of the grid graph recognition problem for trees with a given embedding, even for the case of trees that are as simple as caterpillars. Observe that the NP-hardness results on trees in [9, 27] heavily rely on the variable embedding setting.

References

1. Abel, Z., Demaine, E.D., Demaine, M.L., Eisenstat, S., Lynch, J., Schardl, T.B.: Who needs crossings? Hardness of plane graph rigidity. In: Fekete, S.P., Lubiw, A. (eds.) 32nd International Symposium on Computational Geometry (SoCG 2016). LIPIcs, vol. 51, pp. 3:1–3:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016). <https://doi.org/10.4230/LIPIcs.SoCG.2016.3>
2. Alegría, C., Borrazzo, M., Da Lozzo, G., Di Battista, G., Frati, F., Patrignani, M.: Planar straight-line realizations of 2-trees with prescribed edge lengths. In: Purchase, H.C., Rutter, I. (eds.) GD 2021. LNCS, vol. 12868, pp. 166–183. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92931-2_12
3. Alegría, C., Da Lozzo, G., Di Battista, G., Frati, F., Grosso, F., Patrignani, M.: Unit-length rectangular drawings of graphs. CoRR abs/2208.14142 (2022). <https://arxiv.org/abs/2208.14142>
4. Angelini, P., et al.: Anchored drawings of planar graphs. In: Duncan, C., Symvonis, A. (eds.) GD 2014. LNCS, vol. 8871, pp. 404–415. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45803-7_34

5. Angelini, P., Da Lozzo, G., Frati, F., Lubiw, A., Patrignani, M., Roselli, V.: Optimal morphs of convex drawings. In: Arge, L., Pach, J. (eds.) 31st International Symposium on Computational Geometry, SoCG 2015, 22–25 June 2015, Eindhoven, The Netherlands. LIPIcs, vol. 34, pp. 126–140. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2015). <https://doi.org/10.4230/LIPIcs.SOCG.2015.126>
6. Asgharian Sardroud, A., Bagheri, A.: Embedding cycles and paths on solid grid graphs. *J. Supercomputing* **73**(4), 1322–1336 (2016). <https://doi.org/10.1007/s11227-016-1811-y>
7. Beck, M., Storandt, S.: Puzzling grid embeddings. In: Blelloch, G.E., Finocchi, I. (eds.) Proceedings of the Symposium on Algorithm Engineering and Experiments, ALENEX 2020, Salt Lake City, UT, USA, 5–6 January 2020, pp. 94–105. SIAM (2020)
8. de Berg, M., Khosravi, A.: Optimal binary space partitions for segments in the plane. *Int. J. Comput. Geom. Appl.* **22**(03), 187–205 (2012). <https://doi.org/10.1142/S0218195912500045>
9. Bhatt, S.N., Cosmadakis, S.S.: The complexity of minimizing wire lengths in VLSI layouts. *Inf. Process. Lett.* **25**(4), 263–267 (1987). [https://doi.org/10.1016/0020-0190\(87\)90173-6](https://doi.org/10.1016/0020-0190(87)90173-6)
10. Biedl, T.C., Kant, G.: A better heuristic for orthogonal graph drawings. *Comput. Geom.* **9**(3), 159–180 (1998). [https://doi.org/10.1016/S0925-7721\(97\)00026-6](https://doi.org/10.1016/S0925-7721(97)00026-6)
11. Borradaile, G., Klein, P.N., Mozes, S., Nussbaum, Y., Wulff-Nilsen, C.: Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. *SIAM J. Comput.* **46**(4), 1280–1303 (2017). <https://doi.org/10.1137/15M1042929>
12. Cabello, S., Demaine, E.D., Rote, G.: Planar embeddings of graphs with specified edge lengths. *J. Graph Algorithms Appl.* **11**(1), 259–276 (2007). <https://doi.org/10.7155/jgaa.00145>
13. Chang, Y., Yen, H.: On bend-minimized orthogonal drawings of planar 3-graphs. In: 33rd International Symposium on Computational Geometry, SoCG 2017, 4–7 July 2017, Brisbane, Australia, pp. 29:1–29:15 (2017). <https://doi.org/10.4230/LIPIcs.SoCG.2017.29>
14. Cornelsen, S., Karrenbauer, A.: Accelerated bend minimization. *J. Graph Algorithms Appl.* **16**(3), 635–650 (2012). <https://doi.org/10.7155/jgaa.00265>
15. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall (1999)
16. Di Battista, G., Liotta, G., Vargiu, F.: Spirality and optimal orthogonal drawings. *SIAM J. Comput.* **27**(6), 1764–1811 (1998). <https://doi.org/10.1137/S0097539794262847>
17. Di Battista, G., Tamassia, R.: On-line maintenance of triconnected components with SPQR-trees. *Algorithmica* **15**(4), 302–318 (1996). <https://doi.org/10.1007/BF01961541>
18. Di Battista, G., Tamassia, R.: On-line planarity testing. *SIAM J. Comput.* **25**(5), 956–997 (1996)
19. Di Battista, G., Tamassia, R., Vismara, L.: Incremental convex planarity testing. *Inf. Comput.* **169**(1), 94–126 (2001). <https://doi.org/10.1006/inco.2001.3031>
20. Didimo, W., Kaufmann, M., Liotta, G., Ortali, G.: Rectilinear planarity testing of plane series-parallel graphs in linear time. In: GD 2020. LNCS, vol. 12590, pp. 436–449. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-68766-3_34
21. Didimo, W., Liotta, G., Ortali, G., Patrignani, M.: Optimal orthogonal drawings of planar 3-graphs in linear time. In: Chawla, S. (ed.) Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA 2020), pp. 806–825. ACM-SIAM (2020). <https://doi.org/10.1137/1.9781611975994.49>

22. Eades, P., Wormald, N.C.: Fixed edge-length graph drawing is NP-hard. *Disc. Appl. Math.* **28**(2), 111–134 (1990). [https://doi.org/10.1016/0166-218X\(90\)90110-X](https://doi.org/10.1016/0166-218X(90)90110-X)
23. Felsner, S.: Rectangle and square representations of planar graphs. In: Pach, J. (ed.) *Thirty Essays on Geometric Graph Theory*, pp. 213–248. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-0110-0_12
24. Frati, F.: Planar rectilinear drawings of outerplanar graphs in linear time. In: Auber, D., Valtr, P. (eds.) *GD 2020. LNCS*, vol. 12590, pp. 423–435. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-68766-3_33
25. Garg, A., Liotta, G.: Almost bend-optimal planar orthogonal drawings of biconnected degree-3 planar graphs in quadratic time. In: Kratochvíl, J. (ed.) *GD 1999. LNCS*, vol. 1731, pp. 38–48. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-46648-7_4
26. Garg, A., Tamassia, R.: On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.* **31**(2), 601–625 (2001). <https://doi.org/10.1137/S0097539794277123>
27. Gregori, A.: Unit-length embedding of binary trees on a square grid. *Inf. Process. Lett.* **31**(4), 167–173 (1989). [https://doi.org/10.1016/0020-0190\(89\)90118-X](https://doi.org/10.1016/0020-0190(89)90118-X)
28. Gupta, S., Sa’ar, G., Zehavi, M.: Grid recognition: classical and parameterized computational perspectives (2021). <https://doi.org/10.48550/ARXIV.2106.16180>
29. Hasan, M.M., Rahman, M.S.: No-bend orthogonal drawings and no-bend orthogonally convex drawings of planar graphs (extended abstract). In: Du, D.-Z., Duan, Z., Tian, C. (eds.) *COCOON 2019. LNCS*, vol. 11653, pp. 254–265. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26176-4_21
30. Hopcroft, J., Tarjan, R.: Algorithm 447: efficient algorithms for graph manipulation. *Commun. ACM* **16**(6), 372–378 (1973). <https://doi.org/10.1145/362248.362272>
31. Itai, A., Papadimitriou, C.H., Szwarcfiter, J.L.: Hamilton paths in grid graphs. *SIAM J. Comput.* **11**(4), 676–686 (1982). <https://doi.org/10.1137/0211056>
32. Kant, G.: Drawing planar graphs using the canonical ordering. *Algorithmica* **16**(1), 4–32 (1996). <https://doi.org/10.1007/BF02086606>
33. Liu, Y., Morgana, A., Simeone, B.: A linear algorithm for 2-bend embeddings of planar graphs in the two-dimensional grid. *Discrete Appl. Math.* **81**(1–3), 69–91 (1998). [https://doi.org/10.1016/S0166-218X\(97\)00076-0](https://doi.org/10.1016/S0166-218X(97)00076-0)
34. Miura, K., Haga, H., Nishizeki, T.: Inner rectangular drawings of plane graphs. *Int. J. Comput. Geom. Appl.* **16**(2–3), 249–270 (2006). <https://doi.org/10.1142/S0218195906002026>
35. Nishizeki, T., Rahman, M.S.: *Planar Graph Drawing, Lecture Notes Series on Computing*, vol. 12. World Scientific (2004). <https://doi.org/10.1142/5648>
36. Nishizeki, T., Rahman, M.S.: Rectangular drawing algorithms. In: Tamassia, R. (ed.) *Handbook on Graph Drawing and Visualization*, pp. 317–348. Chapman and Hall/CRC (2013)
37. Rahman, M.S., Egi, N., Nishizeki, T.: No-bend orthogonal drawings of series-parallel graphs. In: Healy, P., Nikolov, N.S. (eds.) *GD 2005. LNCS*, vol. 3843, pp. 409–420. Springer, Heidelberg (2006). https://doi.org/10.1007/11618058_37
38. Rahman, M.S., Egi, N., Nishizeki, T.: No-bend orthogonal drawings of subdivisions of planar triconnected cubic graphs. *IEICE Trans.* **88-D**(1), 23–30 (2005)
39. Rahman, M.S., Nakano, S., Nishizeki, T.: Rectangular grid drawings of plane graphs. *Comput. Geom.* **10**(3), 203–220 (1998). [https://doi.org/10.1016/S0925-7721\(98\)00003-0](https://doi.org/10.1016/S0925-7721(98)00003-0)

40. Rahman, M.S., Nakano, S., Nishizeki, T.: A linear algorithm for bend-optimal orthogonal drawings of triconnected cubic plane graphs. *J. Graph Algorithms Appl.* **3**(4), 31–62 (1999). <http://www.cs.brown.edu/publications/jgaa/accepted/99/SaidurNakanoNishizeki99.3.4.pdf>
41. Rahman, M.S., Nishizeki, T.: Bend-minimum orthogonal drawings of plane 3-graphs. In: *Graph-Theoretic Concepts in Computer Science, 28th International Workshop, WG 2002, Cesky Krumlov, Czech Republic, 13–15 June 2002, Revised Papers*, pp. 367–378 (2002). https://doi.org/10.1007/3-540-36379-3_32
42. Rahman, M.S., Nishizeki, T., Ghosh, S.: Rectangular drawings of planar graphs. *J. Algorithms* **50**(1), 62–78 (2004). [https://doi.org/10.1016/S0196-6774\(03\)00126-3](https://doi.org/10.1016/S0196-6774(03)00126-3)
43. Rahman, M.S., Nishizeki, T., Naznin, M.: Orthogonal drawings of plane graphs without bends. *J. Graph Algorithms Appl.* **7**(4), 335–362 (2003). <http://jgaa.info/accepted/2003/Rahman+2003.7.4.pdf>
44. Schaefer, M.: Realizability of graphs and linkages. In: Pach, J. (ed.) *Thirty Essays on Geometric Graph Theory*, pp. 461–482. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-0110-0_24
45. Storer, J.A.: The node cost measure for embedding graphs on the planar grid (extended abstract). In: Miller, R.E., Ginsburg, S., Burkhard, W.A., Lipton, R.J. (eds.) *Proceedings of the 12th Annual ACM Symposium on Theory of Computing*, 28–30 April 1980, Los Angeles, California, USA, pp. 201–210. ACM (1980). <https://doi.org/10.1145/800141.804667>
46. Tamassia, R.: On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.* **16**(3), 421–444 (1987). <https://doi.org/10.1137/0216030>
47. Thomassen, C.: Plane representations of graphs. In: Bondy, J., Murty, U. (eds.) *Progress in Graph Theory*, pp. 43–69. Academic Press, Toronto, Orlando (1987)
48. Ungar, P.: On Diagrams Representing Maps. *J. London Math. Soc.* **s1-28**(3), 336–342 (1953). <https://doi.org/10.1112/jlms/s1-28.3.336>
49. Vijayan, G., Wigderson, A.: Rectilinear graphs and their embeddings. *SIAM J. Comput.* **14**(2), 355–372 (1985)
50. Zhou, X., Nishizeki, T.: Orthogonal drawings of series-parallel graphs with minimum bends. *SIAM J. Discrete Math.* **22**(4), 1570–1604 (2008). <https://doi.org/10.1137/060667621>