Algorithmica (2014) 68:859–885
DOI 10.1007/s00453-012-9705-8

Orthogonal Graph Drawing with Flexibility Constraints

Thomas Bläsius · Marcus Krug · Ignaz Rutter · Dorothea Wagner

Received: 3 February 2012 / Accepted: 18 October 2012 / Published online: 7 November 2012
© Springer Science+Business Media New York 2012

Abstract Traditionally, the quality of orthogonal planar drawings is quantified by either the total number of bends, or the maximum number of bends per edge. However, this neglects that in typical applications, edges have varying importance. In this work, we investigate an approach that allows to specify the maximum number of bends for each edge individually, depending on its importance.

We consider a new problem called FLEXDRAW that is defined as follows. Given a planar graph $G = (V, E)$ on n vertices with maximum degree 4 and a function flex : $E \longrightarrow \mathbb{N}_0$ that assigns a *flexibility* to each edge, does G admit a planar embedding on the grid such that each edge e has at most flex(e) bends? Note that in our setting the combinatorial embedding of G is not fixed. FLEXDRAW directly extends the problem β-embeddability asking whether G can be embedded with at most β bends per edge.

We give an algorithm with running-time $O(n^2)$ solving FLEXDRAW when the flexibility of each edge is positive. This includes 1-embeddability as a special case and thus closes the complexity gap between 0-embeddability, which is $\mathcal{NP}$-hard to decide, and 2-embeddability, which is efficiently solvable since every planar graph with maximum degree 4 admits a 2-embedding except for the octahedron. In addition

A preliminary version of this article has appeared as T. Bläsius, M. Krug, I. Rutter and D. Wagner, *Orthogonal Graph Drawing with Flexibility Constraints* in Proc. 18th International Symposium on Graph Drawing (GD'2010).

T. Bläsius · M. Krug · I. Rutter (✉) · D. Wagner
Faculty of Informatics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
e-mail: rutter@kit.edu

T. Bläsius
e-mail: blaesius@kit.edu

M. Krug
e-mail: marcus.krug@kit.edu

D. Wagner
e-mail: dorothea.wagner@kit.edu

to the polynomial-time algorithm we show that FLEXDRAW is $\mathcal{NP}$-hard even if the edges with flexibility 0 induce a tree or a union of disjoint stars.

Keywords Planar graphs · Embeddings · Orthogonal drawing · Bend minimization

1 Introduction

Orthogonal graph drawing is one of the most important techniques for the human-readable visualization of complex data. Its esthetic appeal derives from its simplicity and straightforwardness. Since edges are required to be straight orthogonal lines—which automatically yields good angular resolution and short links—the human eye may easily adapt to the flow of an edge. The readability of orthogonal drawings can be further enhanced in the absence of crossings, that is if the underlying data exhibits a planar structure. Unfortunately, not all planar graphs have an orthogonal drawing in which each edge is represented by a straight horizontal or vertical line. In order to be able to visualize all planar graphs nonetheless, we allow edges to have bends. Since bends obfuscate the readability of orthogonal drawings, however, we are interested in minimizing the number of bends on the edges. Previous approaches to orthogonal graph drawing in the presence of bends focus on the minimization of either the maximum number of bends per edge or the total number of bends in the drawing.

In typical applications, however, edges have varying importance for the readability depending on their semantic and their importance for the application. Thus, it is convenient to allow some edges to have more bends than others. See Fig. 1 for an example.

We consider the following orthogonal graph drawing problem, which we call FLEXDRAW. Given a *4-planar graph* G (i.e., G is planar and has maximum degree 4), and for each edge e a non-negative integer $\text{flex}(e)$, called the *flexibility* of e, does G admit a planar embedding on the grid such that each edge e has at most $\text{flex}(e)$ bends? Such a drawing of G on the grid is called a *flex-drawing*. For a graph with $\text{flex}(e) > 0$ for each edge e in G we say that G itself has *positive flexibility*.

The problem we consider generalizes a well-studied problem in orthogonal graph drawing, namely the problem of deciding whether a given graph is β-embeddable

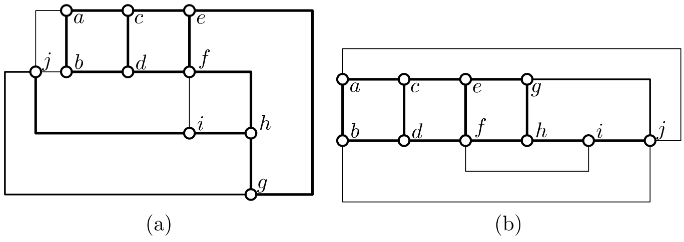


Fig. 1 Two orthogonal drawings of the same graph. The thickness of edges indicates their importance. Although, the drawing in (**a**) has both fewer bends and fewer bends per edge, drawing (**b**) is much clearer since important edges have fewer bends

for some non-negative integer β. A 4-planar graph is *β-embeddable* if it admits an embedding on the grid with at most β bends per edge.

Garg and Tamassia [6] show that it is $\mathcal{NP}$-hard to decide 0-embeddability. The reduction crucially relies on the construction of graphs with rigid embeddings. Later, we show that this is impossible if we allow at least one bend per edge. This is a key observation, which forms the basis for an efficient algorithm for recognizing 1-embeddable graphs. For special cases, namely planar graphs with maximum degree 3 and series-parallel graphs, Di Battista et al. [2] gave an algorithm that minimizes the total number of bends and hence solves 0-embeddability. On the other hand, Biedl and Kant [1] show that every 4-planar graph admits a drawing with at most two bends per edge with the only exception of the octahedron, which requires an edge with three bends. Similar results are obtained by Liu et al. [10].

Liu et al. [9] claim to have found a characterization of the planar graphs with minimum degree 3 and maximum degree 4 that admit an orthogonal embedding with at most one bend per edge. They also claim that this characterization can be tested in polynomial time. Unfortunately, their paper does not include any proofs and to the best of our knowledge a proof of these results did not appear. Morgana et al. [12] characterize the class of *plane graphs* (i.e., planar graphs with a given embedding) that admit a 1-bend embedding on the grid by forbidden configurations. They also present a quadratic algorithm that either detects a forbidden configuration or computes a 1-bend embedding.

If the combinatorial embedding of a 4-planar graph is given, Tamassia's flow network can be used to minimize the total number of bends [13]. Note that this approach may yield drawings with a linear number of bends for some of the edges. Given a combinatorial embedding that admits a 1-bend drawing, however, the flow network can be modified in a straightforward manner to minimize the total number of bends using at most one bend per edge.

The problem we consider involves considering all embeddings of a planar graph. Many problems of this sort are $\mathcal{NP}$-hard. For instance, 0-embeddability is $\mathcal{NP}$-hard [6], even though it can be decided efficiently if we are given an embedding since a graph with fixed embedding is 0-embeddable if and only if Tamassia's flow network yields a drawing without bends.

Contribution and Outline In this work we give an algorithm with running time $O(n^2)$ that solves FLEXDRAW for graphs with positive flexibility. Since FLEXDRAW contains the problem of 1-embeddability as a special case, this closes the complexity gap between the $\mathcal{NP}$-hardness result for 0-embeddability by Garg and Tamassia [6] and the efficient algorithm for computing 2-embeddings by Biedl and Kant [1]. Note that once we have found a feasible planar embedding, a corresponding drawing can be computed in $O(n^2)$ time [13].

We present some preliminaries in Sect. 2. In Sect. 3 we study orthogonal flex-drawings of graphs with a fixed embedding and introduce the maximum rotation of a graph as a measure of how "flexible" it is. In Sect. 4 we show that replacing certain subgraphs with graphs that behave similarly with respect to flexibility does not change the maximum rotation. Based on this fact and the SPQR-tree we give an algorithm that solves FLEXDRAW for biconnected 4-planar graphs with positive flexibility. In Sect. 5 we improve the running time of our algorithm to $O(n^2)$. We extend our algorithm to arbitrary 4-planar graphs with positive flexibility in Sect. 6. Section 7 contains considerations on the complexity of FLEXDRAW if some of the edges are allowed to have flexibility 0. We discuss some extensions and possible directions for future work in Sect. 8.

2 Preliminaries

In this section we introduce some notations and preliminaries.

Orthogonal Representation The *orthogonal representation* introduced by Tamassia [13] describes orthogonal drawings of plane graphs by listing the faces as sequences of bends. As an advantage the orthogonal representation neglects the lengths of segments. Thus, it is possible to manipulate drawings without the need to worry about the exact geometry. Our orthogonal representation is always normalized, that is each edge has only bends in one direction; this slightly differs from the notion introduced by Tamassia. It follows from Tamassia's flow network [13] that an orthogonal representation can be normalized without increasing the number of bends on edges. More precisely, bends in different directions on a single edge translate to a circulation on a 2-cycle in the flow network, which can be eliminated. Thus, assuming orthogonal representations to be normalized is not a restriction.

The orthogonal representation of a plane graph G is defined as a set of lists $\mathcal{R}$ containing a list $\mathcal{R}(f_i)$ for each face f_i of G. For each face f_i the list $\mathcal{R}(f_i)$ is a circular list of *edge descriptions* containing the edges on the boundary of f_i in clockwise order (counter-clockwise if f_i is the external face). Each description $r \in \mathcal{R}(f_i)$ contains the following information: edge(r) denotes the edge represented by r, bends(r) is an integer whose absolute value is the number of 90°-bends of edge(r), where positive numbers represent bends to the right and negative numbers bends to the left; see Fig. 2a. For a given edge description $r \in \mathcal{R}(f_i)$ we denote its successor

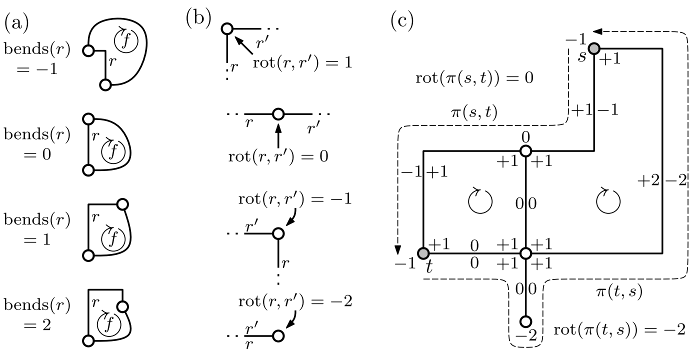


Fig. 2 Illustration of the orthogonal representation. (**a**) An edge with edge description r in the face f for the cases bends(r) $\in \{-1, \dots, 2\}$. (**b**) The possible rotations between the edge description r and its successor r'. (**c**) An orthogonal drawing together with its orthogonal representation. The paths $\pi(s,t)$ and $\pi(t,s)$ are drawn as *dashed lines*

in $\mathcal{R}(f_i)$ by r' and represent the angle α between edge(r) and edge(r') in f_i by their rotation $\mathrm{rot}(r, r') = 2 - \alpha/90°$; see Fig. 2b. The value of $\mathrm{rot}(r, r')$ is also stored in the edge description of r and thus belongs to the orthogonal representation. Every edge has exactly two edge descriptions, if r is one of them, the other is denoted by $\bar{r}$. Since each face forms a rectilinear polygon, every orthogonal representation $\mathcal{R}$ of an orthogonal drawing has the following three properties; compare with the example in Fig. 2c.

(I) Each edge description r is consistent with $\bar{r}$, i.e., $\mathrm{bends}(\bar{r}) = -\mathrm{bends}(r)$.
(II) The interior bends of any face f sum up to 4 and the exterior bends to -4:

$$\sum_{r \in \mathcal{R}(f)} \big(\mathrm{bends}(r) + \mathrm{rot}\big(r, r'\big)\big) = \begin{cases} -4, & \text{if } f \text{ is the external face,} \\ +4, & \text{if } f \text{ is an internal face.} \end{cases}$$

(III) The angles around every node sum up to 360°.

Given an orthogonal representation $\mathcal{R}$ of a graph, a corresponding orthogonal drawing can be computed efficiently [13]. Hence, it is sufficient to work with orthogonal representations. An orthogonal representation is *valid* for a given flexibility function flex if $|\mathrm{bends}(r)| \leq \mathrm{flex}(\mathrm{edge}(r))$ for each edge description r.

For a planar graph $G = (V, E)$ with orthogonal representation $\mathcal{R}$ and two vertices s and t on the outer face f_1, we denote by $\pi_{\mathcal{R}}(s, t)$ the unique shortest path in $\mathcal{R}(f_1)$ that connects s and t in counter-clockwise direction. Note that we define $\pi_{\mathcal{R}}(s, t)$ to be the *shortest* path since the path on the outer face is not unique if s or t are cutvertices. Such a path $\pi = \pi_{\mathcal{R}}(s, t)$ consists of consecutive edge descriptions $r_1, \ldots, r_k$. We define the *rotation* of π as

$$\mathrm{rot}_{\mathcal{R}}(\pi) = \sum_{i=1}^{k} \mathrm{bends}(r_i) + \sum_{i=1}^{k-1} \mathrm{rot}(r_i, r_{i+1}).$$

Moreover, for the vertex s we denote by $\mathrm{rot}_{\mathcal{R}}(s)$ the rotation value of the angle between $\pi_{\mathcal{R}}(s, t)$ and $\pi_{\mathcal{R}}(t, s)$ at s. We define $\mathrm{rot}_{\mathcal{R}}(t)$ analogously. As a single edge description r can be interpreted as a path of length 1, the equation $\mathrm{rot}_{\mathcal{R}}(r) = \mathrm{bends}(r)$ holds. If it is clear from the context which orthogonal representation is meant, we omit the subscripts of $\pi(s, t)$ and rot. The concept of rotation is similar to the spirality defined by Di Battista et al. [2].

The value $\mathrm{rot}(\pi(s, t))$ describes the shape of the path $\pi(s, t)$ in the orthogonal representation in terms of the angle between the first segment of the first edge and the last segment of the last edge of $\pi(s, t)$. Fixing the rotation of $\pi(s, t)$, $\pi(t, s)$ and the outer angles at s and t in a sense determines the shape of the outer face. In Sect. 4, we will exploit this by replacing certain subgraphs of G with simpler graphs whose outer faces have the same shapes.

Flows and Tamassia's Flow Network A *flow network* is a tuple $N = (V, A, \ell, u, q)$ where (V, A) is a directed (multi-)graph, $\ell : A \longrightarrow \mathbb{N}_0$ and $u : A \longrightarrow \mathbb{N}_0 \cup \{\infty\}$ are lower and upper bounds for the amount of flow along the arcs in A with $\ell(a) \leq u(a)$ for all $a \in A$. Finally, $q : V \longrightarrow \mathbb{Z}$ defines a demand for each vertex. Note that demands can be positive or negative.

A *flow* is a function $\phi : A \longrightarrow \mathbb{N}_0$ that maps a certain amount of flow to each arc such that $\ell(a) \le \phi(a) \le u(a)$ holds for all arcs $a \in A$. A flow ϕ is *feasible*, if in addition the difference of incoming and outgoing flow at each vertex equals its demand, that is

$$q(v) = \sum_{(u,v)\in A} \phi(u,v) - \sum_{(v,u)\in A} \phi(v,u) \quad \text{for all } v \in V.$$

The *residual network* N_ϕ of N with respect to the flow ϕ is obtained from N by reducing for each arc $a = (u, v) \in A$ its capacity by $\phi(a)$ and adding an edge (v, u) with capacity $\phi(a)$ in the opposite direction.

The defect of a node v with respect to a flow ϕ is defined as

$$\mathrm{def}_\phi(v) = \sum_{(u,v)\in A} \phi(u,v) - \sum_{(v,u)\in A} \phi(v,u) - q(v).$$

The *defect of a flow* ϕ is defined as $\mathrm{def}(\phi) = \sum_{v\in V} |\mathrm{def}_\phi(v)|$. Clearly, a flow has defect 0 if and only if it is feasible.

Let $G = (V, E)$ be a 4-planar graph together with a planar embedding $\mathcal{E}$ and let $F = \{f_1, \dots, f_k\}$ be the set of faces of G with respect to embedding $\mathcal{E}$, where f_1 is the outer face. Further let n_i be the number of vertices that are incident to f_i.

Tamassia's flow network consists of nodes $V \cup F$ with $q(v) = -4$ for all $v \in V$, $q(f_i) = 2n_i - 4$ for $i \ge 2$ and $q(f_1) = 2n_1 + 4$. The flow network contains the following arcs. For each node v, let F_v be the set of faces incident to v. Then, for each face $f \in F_v$ there is an arc a from v to f with $\ell(a) = 1$ and $u(a) = 4$. Further, for each edge e of G with incident faces g and h there is an arc a_1 from g to h and an arc a_2 from h to g with $\ell(a_1) = \ell(a_2) = 0$ and $u(a_1) = u(a_2) = \infty$. The demands and capacities essentially represent the distribution of 90°-angles around vertices and faces. Tamassia showed that there is a bijection between feasible flows in the network and orthogonal representations of G with embedding $\mathcal{E}$ [13]. FLEXDRAW with fixed embedding can easily be handled by setting the upper bound of the arcs stemming from an edge e to $\mathrm{flex}(e)$ for all edges $e \in E$. Note that this still works if edges are allowed to have flexibility 0.

Let e be an edge of G with incident faces g and h and let a_1 and a_2 be the two arcs stemming from e such that a_1 is directed from g to h and a_2 is directed from h to g. Note that for a flow ϕ, by eliminating 2-cycles with positive flow, we may assume that either $\phi(a_1) = 0$ or $\phi(a_2) = 0$ holds. For ease of notation, we therefore identify these two arcs with e and write $\phi_e(g, h)$ for the amount of flow from g to h via the arcs stemming from e. Note that this value can be negative if there is flow from h to g and we have $\phi_e(g, h) = -\phi_e(h, g)$.

Connectivity, st-Graphs and the SPQR-Tree A graph is *connected* if there exists a path between any pair of vertices. A *separating k-set* is a set of k vertices whose removal disconnects the graph. Separating 1-sets and 2-sets are *cutvertices* and *separation pairs*, respectively. A connected graph is *biconnected* if it does not have a cutvertex and *triconnected* if it does not have a separation pair. The maximal biconnected components of a graph are called *blocks*. The *cut components* with respect to a separation k-set S are the maximal subgraphs that are not disconnected by removing S.

The *block-cutvertex tree* of a connected graph is a tree whose nodes are the blocks and cutvertices of the graph. In the block-cutvertex tree a block B and a cutvertex v are joined by an edge if v belongs to B.

A *weak st-graph* is a 4-planar graph $G = (V, E)$ with two designated vertices s and t such that the graph $G + st$ is planar and has maximum degree 4. An *st-graph* is a weak st-graph such that $G + st$ is biconnected. An orthogonal representation $\mathcal{R}$ of a (weak) st-graph with positive flexibility is *valid* if each edge e has at most flex(e) bends and s and t are embedded on the outer face. A valid orthogonal representation of a (weak) st-graph is *tight* if all the angles at s and t in inner faces are 90°.

An st-graph is of Type (1,1) if $\deg(s) = \deg(t) = 1$, it is of Type (1,2) if one of them has degree 1 and the other one has degree 2 and it is of Type (2,2) if $\deg(s) = \deg(t) = 2$.

We use the SPQR-tree introduced by Di Battista and Tamassia [3, 4] to represent all planar embeddings of a biconnected planar graph G. The SPQR-tree $\mathcal{T}$ of G is a decomposition of G into its triconnected components along its *split pairs* where a split pair is either a separation pair or an edge. We first define the SPQR-tree to be unrooted, representing embeddings on the sphere, that is planar embeddings without a designated outer face. Let $\{s, t\}$ be a split pair and let H_1 and H_2 be two subgraphs of G such that $H_1 \cup H_2 = G$ and $H_1 \cap H_2 = \{s, t\}$. Consider the following tree containing the two nodes μ_1 and μ_2 associated with the graphs $H_1 + st$ and $H_2 + st$, respectively. These graphs are called *skeletons* of the nodes μ_i, denoted by skel(μ_i) and the special edge st is said to be a *virtual edge*. The two nodes μ_1 and μ_2 are connected by an edge, or more precisely, the occurrence of the virtual edges st in both skeletons are linked by this edge. A combinatorial embedding of G uniquely induces a combinatorial embedding of skel(μ_1) and skel(μ_2). Furthermore, arbitrary and independently chosen embeddings for the two skeletons determine an embedding of G, thus the resulting tree can be used to represent all embeddings of G by the combination of all embeddings of two smaller planar graphs. This replacement can of course be applied iteratively to the skeletons yielding a tree with more nodes but smaller skeletons associated with the nodes. Applying this kind of decomposition in a systematic way yields the SPQR-tree as introduced by Di Battista and Tamassia [3, 4]. The SPQR-tree $\mathcal{T}$ of a biconnected planar graph G contains four types of nodes. First, the P-nodes having a bundle of at least three parallel edges as skeleton and a combinatorial embedding is given by any order of these edges. Second, the skeleton of an R-node is triconnected, thus having a unique embedding up to a flip (yielding exactly two embeddings) [14], and third, S-nodes have a simple cycle as skeleton without any choice for the embedding. Finally, every edge in a skeleton representing only a single edge in the original graph G is formally also considered to be a virtual edge linked to a Q-node in $\mathcal{T}$ representing this single edge. Note that all leaves of the SPQR-tree $\mathcal{T}$ are Q-nodes. Besides from being a nice way to represent all embeddings of a biconnected planar graph, the SPQR-tree has only size linear in G and Gutwenger and Mutzel showed how to compute it in linear time [7]. Figure 3 shows a biconnected planar graph together with its SPQR-tree.

In this paper, we consider the rooted version of the SPQR-tree since this determines which face is the outer face (a crucial choice for our algorithm). More precisely,

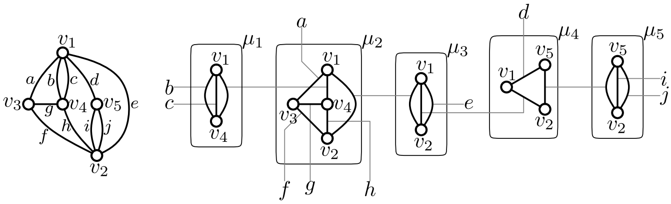


Fig. 3 The unrooted SPQR-tree of a biconnected planar graph. The nodes μ_1, μ_3 and μ_5 are P-nodes, μ_2 is an R-node and μ_4 is an S-node. The Q-nodes are not shown explicitly

we choose one of the Q-nodes to be the root of the SPQR-tree $\mathcal{T}$ of the biconnected planar graph G. We call the edge associated with this Q-node the *reference edge* and denote it by e_{ref}. Then in the skeleton skel(μ) of each node μ there is exactly one virtual edge associated with the parent of μ. Now $\mathcal{T}$ represents all planar embeddings of G with the edge e_{ref} on the outer face by restricting the planar embeddings of the skeletons to those where the virtual edges associated with the parents are on the outer face. The *pertinent graph* of a Q-node is defined to be the edge associated with it. The pertinent graph of an inner node μ, denoted by pert(μ), is recursively defined to be the graph that is obtained from skel(μ) by replacing all virtual edges in skel(μ) by the pertinent graphs of the corresponding children of μ. Note that this definition depends on the root chosen for the SPQR-tree $\mathcal{T}$. A similar term not depending on the root is the so called *expansion graph* of a virtual edge ε in skel(μ). Assume $\mathcal{T}$ to be rooted at μ and let μ' be the child of μ corresponding to the virtual edge ε. Then the expansion graph of ε is defined to be the pertinent graph of μ' with respect to the root μ.

Our Approach We now briefly sketch how our algorithm deciding FLEXDRAW for graphs with positive flexibility works. We start out with an observation. Let G be a 4-planar graph with positive flexibility and let $\{s, t\}$ be a split pair of G that splits G into two subgraphs G_1, G_2 and let e_{ref} be an edge of G_1. Let ρ be the maximum rotation of $\pi(s,t)$ over all embeddings of G_2 where s and t are on the outer face.

If G_2 is of Type (1,1), then obviously the following holds. If G admits a valid orthogonal drawing with the given flexibility such that e_{ref} is embedded on the outer face, then also the graph G' that is obtained from G by replacing G_2 by the single edge st with flexibility ρ admits such a drawing. Thus, we can substitute graphs of Type (1,1) with single edges to obtain a new graph G' with the property that G' has a valid drawing if G has one. We show that the converse is also true, that is if the graph G' admits such an embedding then also G does. Graphs of Type (1,2) and (2,2) allow for similar substitutions.

We then exploit this characterization algorithmically, using the SPQR-tree of G to successively replace subgraphs of G by simpler graphs. This substantially reduces the number of planar embeddings we need to consider, yielding a polynomial-time algorithm.

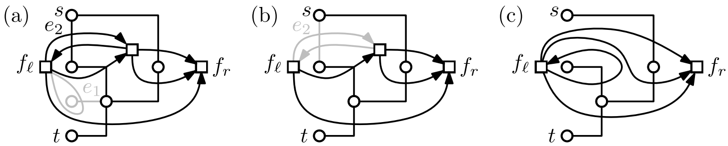


Fig. 4 An st-graph with flexibility 1 for all edges, $\text{rot}(\pi(s,t)) = 1$, and its flex graph $G^\times$ (**a**), after removal of bridge e_1 (**b**), and removal of edge e_2 (**c**)

3 The Maximum Rotation with a Fixed Embedding

The goal of this section is to derive a description of the valid orthogonal representations of a given (weak) st-graph with positive flexibility and a fixed embedding. To this end, we prove that the values that can be obtained for $\text{rot}(\pi(s,t))$ form an interval for these graphs, namely, we show that if there exists a valid orthogonal representation $\mathcal{R}$ with $\text{rot}_\mathcal{R}(\pi(s,t)) \geq 0$ then there exists an orthogonal representation $\mathcal{R}'$ with $\text{rot}_{\mathcal{R}'}(\pi(s,t)) = \text{rot}_\mathcal{R}(\pi(s,t)) - 1$, which can be obtained from $\mathcal{R}$ by only altering the number of bends on certain edges.

To model the possible changes of an orthogonal representation $\mathcal{R}$ of a (weak) st-graph G that can be performed by only varying the number of bends on edges (i.e., without changing the angles at vertices), we introduce the *flex graph* $G^\times$ of G with respect to $\mathcal{R}$, which is based on the bidirected dual graph of G. Thus, the flex graph is a directed multigraph; see Fig. 4a for an illustration. We start out by adding to G the edge st and embed it into the outer face of G, thus splitting the outer face into two faces f_ℓ and f_r, where f_ℓ is bounded by $\pi(s,t)$ and the new edge st and f_r is bounded by $\pi(t,s)$ and st. We denote this graph by $\bar{G}$ and its dual graph by $\bar{G}^*$. We set $V^\times = V(\bar{G}^*)$ and we define $E^\times$ as follows. For each edge e of G denote its incident faces in $\bar{G}$ by f_u and f_v and let r_u and r_v be the edge descriptions of e in $\mathcal{R}(f_u)$ and $\mathcal{R}(f_v)$, respectively. We add the edge (f_u, f_v) if $\text{flex}(e) > \text{bends}(r_v)$ and, analogously, we add (f_v, f_u) if $\text{flex}(e) > \text{bends}(r_u)$. Consider an edge (f_u, f_v) of $G^\times$ and let r_u and r_v be the edge descriptions of the corresponding edge e in G. The fact that $(f_u, f_v) \in E^\times$ indicates that it is possible to decrease $\text{bends}(r_u)$ (and thus increase $\text{bends}(r_v)$) by at least 1 without violating the flexibility of e.

Assume that there exists a simple directed path from f_ℓ to f_r in $G^\times$. Let $f_\ell = f_1, f_2, \dots, f_k = f_r$ be this path. We construct a new orthogonal representation $\mathcal{R}'$ from $\mathcal{R}$ as follows. For each edge $f_i f_{i+1}$, $i = 1, \dots, k-1$, let e_i be the corresponding edge of G and let $r_i \in \mathcal{R}(f_i), \bar{r}_i \in \mathcal{R}(f_{i+1})$ be its edge descriptions. We obtain $\mathcal{R}'$ from $\mathcal{R}$ by decreasing $\text{bends}(r_i)$ by 1 and increasing $\text{bends}(\bar{r}_i)$ by 1, for $i = 1, \dots, k-1$. First, it is clear that $\mathcal{R}'$ satisfies Properties I and III since we increase and decrease the number of bends consistently and we do not change any angles at vertices. Property II holds since each face of G has either none of its edge descriptions changed or exactly one of them is increased by 1 and exactly one of them is decreased by 1. Moreover, since the path starts at f_ℓ and ends at f_r we have that $\text{rot}_{\mathcal{R}'}(\pi(s,t)) = \text{rot}_\mathcal{R}(\pi(s,t)) - 1$. We now show that such a path exists if $\text{rot}(\pi(s,t)) \geq 0$.

Lemma 1 *Let G be a weak st-graph with positive flexibility and let $\mathcal{R}$ be a valid orthogonal representation of G with* $\mathrm{rot}_{\mathcal{R}}(\pi(s,t)) \geq 0$. *Then the flex graph $G^\times$ contains a directed path from f_ℓ to f_r.*

Proof First, we show that in $G^\times$ there exists at least one edge starting from f_ℓ. Let $\pi(s,t)$ be composed of the edge descriptions $r_1, \dots, r_k$ in $\mathcal{R}(f)$, where f is the outer face of G. Then, by assumption we have $\mathrm{rot}(\pi(s,t)) = \sum_{i=1}^{k} \mathrm{bends}(r_i) + \sum_{i=1}^{k-1} \mathrm{rot}(r_i, r_{i+1}) \geq 0$. Since $\mathrm{rot}(r_i, r_{i+1}) \leq 1$ for $i = 1, \dots, k-1$ we have that $\sum_{i=1}^{k} \mathrm{bends}(r_i) \geq -k+1$ and hence there is at least one r_j with $\mathrm{bends}(r_j) \geq 0$. Hence, $G^\times$ contains an edge corresponding to $\mathrm{edge}(r_j)$ that starts at f_ℓ. This shows that there always exists an edge (f_ℓ, f_u) in $G^\times$. For the case that G is a path, this immediately implies the claim of the lemma, as in this case $f_u = f_r$. This will serve as the base case for an induction on the size of G.

Let G and $\mathcal{R}$ be as in the statement of the lemma, and assume that the lemma holds for all smaller graphs. As argued above, the flex graph $G^\times$ contains an edge (f_ℓ, f_u) and we distinguish three types of edges (f_ℓ, f_u). If $f_u = f_r$ then (f_ℓ, f_u) is the desired path in $G^\times$.

If $f_u = f_\ell$, the corresponding edge e of G is a bridge whose removal does not disconnect s and t, see the changes between Fig. 4a and Fig. 4b. Then, let H be the connected component of $G - e$ containing s and t and let $\mathcal{S}$ be the restriction of $\mathcal{R}$ to H. For the outer face of H we have that $\mathrm{rot}_{\mathcal{S}}(\pi(s,t)) + \mathrm{rot}_{\mathcal{S}}(s) + \mathrm{rot}_{\mathcal{S}}(\pi(t,s)) + \mathrm{rot}_{\mathcal{S}}(t) = -4$. Since $\pi_{\mathcal{R}}(t,s) = \pi_{\mathcal{S}}(t,s)$ we have that $\mathrm{rot}_{\mathcal{S}}(\pi(t,s)) = \mathrm{rot}_{\mathcal{R}}(\pi(t,s))$. Moreover, since we only remove edges, the angles at s and t (and thus their rotations) do not decrease, that is we have $\mathrm{rot}_{\mathcal{S}}(t) \leq \mathrm{rot}_{\mathcal{R}}(t)$ and $\mathrm{rot}_{\mathcal{S}}(s) \leq \mathrm{rot}_{\mathcal{R}}(s)$. Hence, we have that $\mathrm{rot}_{\mathcal{S}}(\pi(s,t)) \geq -4 - \mathrm{rot}_{\mathcal{R}}(\pi(t,s)) - \mathrm{rot}_{\mathcal{R}}(s) - \mathrm{rot}_{\mathcal{R}}(t) = \mathrm{rot}_{\mathcal{R}}(\pi(s,t)) \geq 0$. Since H has fewer edges than G its flex graph $H^\times$ contains a path from f_ℓ to f_r. The claim follows since $H^\times$ is a subgraph of $G^\times$.

Otherwise, f_u is an internal face of G; see Fig. 4b and Fig. 4c. Let e be the corresponding edge of G. Let $H = G - e$ and let $\mathcal{S}$ be the orthogonal representation $\mathcal{R}$ restricted to H. Note that the flex graph $H^\times$ of H can be obtained from $G^\times$ by removing all edges between f_ℓ and f_u and merging f_ℓ and f_u into a single node f'_ℓ. As above we obtain that $\mathrm{rot}_{\mathcal{S}}(\pi(s,t)) \geq 0$ and hence by induction there exists a path from f'_ℓ to f_r in $H^\times$. The corresponding path in $G^\times$ (after undoing the contraction of f_ℓ and f_u) either starts at f_ℓ or at f_u and ends at f_r. In the former case we have found our path, in the latter case the path together with the edge (f_ℓ, f_u) forms the desired path. □

Recall that a valid orthogonal representation of a (weak) st-graph is tight if the inner angles at s and t are 90°. We show that a valid orthogonal representation can be made tight without decreasing $\mathrm{rot}(\pi(s,t))$ or $\mathrm{rot}(\pi(t,s))$. The proof is illustrated in Fig. 5.

Lemma 2 *Let G be a weak st-graph with positive flexibility and let $\mathcal{R}$ be a valid orthogonal representation. Then there exists a valid orthogonal representation $\mathcal{R}'$ of G with the same planar embedding such that $\mathcal{R}'$ is tight,* $\mathrm{rot}_{\mathcal{R}'}(\pi(s,t)) \geq \mathrm{rot}_{\mathcal{R}}(\pi(s,t))$ *and* $\mathrm{rot}_{\mathcal{R}'}(\pi(t,s)) \geq \mathrm{rot}_{\mathcal{R}}(\pi(t,s))$.

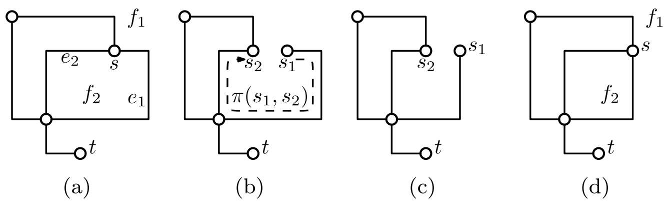


Fig. 5 Orthogonal representation that is not tight since s has an angle of 180° in f_2 (**a**). Splitting s into s_1 and s_2 yields the path $\pi(s_1, s_2)$ with rotation at least 4 (**b**), hence the rotation can be reduced (**c**). Merging s_1 and s_2 back into s yields a tight orthogonal representation (**d**)

Proof Let f_1 be the outer face and assume that f_2 is an inner face incident to s whose inner angle at s is larger than 90°. We show how to decrease this angle by 90° by only changing the number of bends on certain edges. Hence, by applying the described operation iteratively, we can reduce all internal angles at inner faces incident to s and t to 90°.

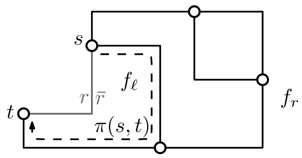
Let e_1 and e_2 be the two edges incident to s such that e_1 occurs before e_2 when traversing the boundary of f_2 clockwise starting from s. The degree of s is at most 3 since the angle at s in f_2 is at least 180°. Thus, one of the two edges e_1 and e_2 is incident to the outer face f_1. We assume that e_1 is incident to f_1 (the case in which only e_2 is incident to f_1 can be treated analogously).

We split s into two vertices s_1 and s_2. We attach e_1 to s_1 and we attach to s_2 the remaining edges incident to s. Let H be the resulting graph and let $\mathcal{S}$ be the orthogonal representation of H induced by $\mathcal{R}$. Since f_2 is an internal face its total rotation in $\mathcal{R}$ is 4 and since the angle at s was either 180° or 270°, we have that $\mathrm{rot}_{\mathcal{S}}(\pi(s_1, s_2))$ is either 4 or 5. By Lemma 1 the flex graph $H^{\times}$ of H contains a simple path that can be used to reduce the rotation along $\pi(s_1, s_2)$ by 1. This path either contains an edge stemming from an edge of $\pi(s_2, t)$ or an edge of $\pi(t, s_1)$. It hence either increases $\mathrm{rot}_{\mathcal{S}}(\pi(s_2, t))$ or $\mathrm{rot}_{\mathcal{S}}(\pi(t, s_1))$ by 1, whereas the other one remains unchanged. We obtain $\mathcal{R}'$ by merging s_1 and s_2 back into s. Since $\mathrm{rot}_{\mathcal{S}}(\pi(s_1, s_2))$ was decreased we increase the rotation at s in f_2 by 1 without decreasing $\mathrm{rot}_{\mathcal{R}}(\pi(s, t)) = \mathrm{rot}_{\mathcal{R}}(\pi(s_2, t))$ or $\mathrm{rot}_{\mathcal{R}}(\pi(t, s)) = \mathrm{rot}_{\mathcal{R}}(\pi(t, s_1))$. Note that aside from changing the number of bends on certain edges we did only change angles incident to s. □

Let G be an st-graph with positive flexibility and a fixed planar embedding $\mathcal{E}$. Lemma 1 shows that the attainable values of $\mathrm{rot}(\pi(s, t))$ for a given st-graph with a fixed embedding form an interval. Hence, the set of possible rotations can be described by the boundaries of this interval and we define the *maximum rotation* of G with respect to $\mathcal{E}$ as $\mathrm{maxrot}_{\mathcal{E}}(G) = \max_{\mathcal{R} \in \Omega} \mathrm{rot}_{\mathcal{R}}(\pi(s, t))$, where Ω contains all valid orthogonal representations of G whose embedding is $\mathcal{E}$. For the case that G does not admit any valid orthogonal representation respecting the embedding $\mathcal{E}$, that is $\Omega = \emptyset$, we formally set $\mathrm{maxrot}_{\mathcal{E}}(G) = -\infty$.

The following theorem states that indeed the maximum rotation describes the orthogonal representations of st-graphs with fixed embedding and positive flexibility.

Fig. 6 If $\mathrm{rot}(r)$ is maximized, $\mathrm{rot}(\pi(s,t))$ is also maximized and the angles at s and t in f_ℓ are both 90°



Theorem 1 *Let G be an st-graph with positive flexibility and fixed embedding* $\mathcal{E}$. *Then for each* $\rho \in \{-1, \dots, \mathrm{maxrot}_{\mathcal{E}}(G)\}$ *there exists a valid and tight orthogonal representation* $\mathcal{R}$ *of G with planar embedding* $\mathcal{E}$ *such that* $\mathrm{rot}_{\mathcal{R}}(\pi(s,t)) = \rho$.

Proof Let $\rho \in \{-1, \dots, \mathrm{maxrot}_{\mathcal{E}}(G)\}$. We show how to construct an orthogonal representation $\mathcal{R}$ with $\mathrm{rot}(\pi(s,t)) = \rho$. Let $\mathcal{S}$ be an orthogonal representation of G with embedding $\mathcal{E}$ such that $\mathrm{rot}_{\mathcal{S}}(\pi(s,t)) = \mathrm{maxrot}_{\mathcal{E}}(G)$. By Lemma 2 we can make $\mathcal{S}$ tight while preserving its embedding and $\mathrm{rot}(\pi(s,t))$. We then apply Lemma 1 to reduce $\mathrm{rot}(\pi(s,t))$ to ρ. Note that the representation remains tight as the angles around vertices are not changed by this operation. □

Before we show how to compute the maximum rotation for st-graphs with a fixed embedding, we need the following technical lemma.

Lemma 3 *Let G be an st-graph with orthogonal representation* $\mathcal{R}$ *and let* $\mathcal{S}$ *be an orthogonal representation of* $G + st$ *such that* $\mathcal{S}$ *induces* $\mathcal{R}$ *on G and the outer face of* $G + st$ *is bounded by st and* $\pi(t,s)$. *Let r be the edge description of st in the outer face. Then* $\mathrm{rot}(\pi(s,t)) \geq \mathrm{rot}(r) + 2$ *holds. Moreover, equality holds if* $\mathcal{S}$ *is tight.*

Proof Let f_r and f_ℓ be the external and internal face incident to st, respectively, and let $\bar{r}$ be the edge description of st in f_ℓ; see Fig. 6. We first consider the case where $\mathcal{S}$ is tight. Then, the vertices s and t form 90° angles in f_ℓ, yielding a rotation of 1 between st and the path $\pi(s,t)$ for both vertices. Since the total rotation around the face f_ℓ is 4, this yields $\mathrm{rot}(\pi(s,t)) + \mathrm{rot}(\bar{r}) + 2 = 4$, which is equivalent to $\mathrm{rot}(\pi(s,t)) = \mathrm{rot}(r) + 2$. Increasing the angles at s or t decreases the rotation between st and $\pi(s,t)$ yielding the inequality $\mathrm{rot}(\pi(s,t)) + \mathrm{rot}(\bar{r}) + 2 \geq 4$, which concludes the proof. □

Using a variant of Tamassia's flow network [13] the maximum rotation can be computed efficiently for st-graphs with a fixed embedding.

Theorem 2 *Given an st-graph* $G = (V, E)$ *with fixed embedding* $\mathcal{E}$ *and with s and t on the outer face, one can either compute* $\mathrm{maxrot}_{\mathcal{E}}(G)$ *or decide that G does not admit a valid orthogonal representation with embedding* $\mathcal{E}$ *in* $O(n^{3/2})$ *time.*

Proof We use the flow network of Tamassia [13] to check whether G admits a valid orthogonal representation with its given embedding. Since this flow network is planar and the in- and out-flow of each sink and source is fixed this can be done in $O(n^{3/2})$ time [11].

We add to G the edge st and embed it into the outer face such that we split the outer face of G into two parts f_ℓ and f_r where f_ℓ is bounded by $\pi(s,t)$ and st and f_r is the outer face of $G+st$. Let r be the edge description of st in f_r; see Fig. 6. We claim that $\text{maxrot}_{\mathcal{E}}(G)$ can be obtained by finding an orthogonal representation of $G+st$ that maximizes $\text{rot}(r)$. More precisely, we claim that $\text{maxrot}_{\mathcal{E}}(G)=\text{rot}(r)+2$ holds.

The equation $\text{maxrot}_{\mathcal{E}}(G)\geq \text{rot}(r)+2$ follows directly from Lemma 3. Conversely, by Lemma 2 there exists a tight orthogonal representation $\mathcal{R}$ of G with embedding $\mathcal{E}$ such that $\text{rot}(\pi(s,t))=\text{maxrot}_{\mathcal{E}}(G)$. Since $\mathcal{R}$ is tight, both vertices s and t have a free incidence in the outer face. Thus, the edge st can be added to $\mathcal{R}$ such that the angles in the new internal face are 90°, yielding a tight orthogonal representation of $G+st$. Due to Lemma 3, the edge st has rotation $\text{maxrot}_{\mathcal{E}}(G)-2$ in the outer face. Thus, in an orthogonal representation maximizing $\text{rot}(r)$ the inequality $\text{maxrot}_{\mathcal{E}}(G)-2\leq\text{rot}(r)$ holds. This shows the claim.

Now it remains to show that we can maximize $\text{rot}(r)$ efficiently. We first use the flow network of Tamassia [13] to compute an arbitrary valid orthogonal representation of $G+st$. To maximize $\text{rot}(r)$ we wish to modify the corresponding flow ϕ in the flow network of Tamassia such that the flow on the edge (f_r, f_ℓ) is maximized while the flow on (f_ℓ, f_r) is 0, which corresponds to maximizing $\text{bends}(r)$. This can be done by computing a maximum flow from f_ℓ to f_r in the residual graph of Tamassia's flow network with respect to ϕ after removing the edges dual to st. Since this network is planar and the source and the sink lie on the same face a maximum flow can be computed in linear time [8]. □

4 Biconnected Graphs

Until now the planar embedding of our input graph was fixed. Now, we assume that the embedding is variable. Following the approach of the previous section, we define the maximum rotation of a (weak) st-graph G as $\text{maxrot}(G)=\max_{\mathcal{E}\in\Psi}\text{maxrot}_{\mathcal{E}}(G)$ where Ψ contains all planar embeddings of G such that s and t are embedded on the outer face.

In this section we show that $\text{maxrot}(G)$ essentially describes all valid orthogonal representations of G in the sense that substituting a subgraph H of G with a different graph H' with $\text{maxrot}(H)=\text{maxrot}(H')$ does not change $\text{maxrot}(G)$. We use this substitution to give an algorithm that computes $\text{maxrot}(G)$ by successively reducing the size of the graph. To handle the different possible planar embeddings of G we use the SPQR-tree and we substitute subgraphs of G with smaller graphs that have only one embedding. We need the following technical lemma.

Lemma 4 *Let G be an st-graph with* $\deg(s), \deg(t)\leq 2$ *and let $\mathcal{R}$ be a tight orthogonal representation of G. Then* $\text{rot}(\pi(s,t))+\text{rot}(\pi(t,s))=-x$ *where x is* 0,1 *and* 2 *for graphs of Type* (1,1), (1,2) *and* (2,2), *respectively.*

Proof By Property II we have $\text{rot}(\pi(s,t))+\text{rot}(t)+\text{rot}(\pi(t,s))+\text{rot}(s)=-4$. If s has degree 1, we have $\text{rot}(s)=-2$. If $\deg(s)=2$ holds, then s is incident to exactly one inner face and by assumption it has an angle of 90° in this face. Hence, in the outer face there is an angle of 270° and thus $\mathrm{rot}(s) = -1$. As the same analysis holds for t, the claim follows. □

The following theorem shows that indeed the maximum rotation describes all possible rotation values of an st-graph.

Theorem 3 *Let G be an st-graph with* $\deg(s), \deg(t) \le 2$ *and positive flexibility and let ρ be an integer. Then there exists a tight orthogonal representation $\mathcal{R}$ of G with* $\mathrm{rot}(\pi(s,t)) = \rho$ *if and only if* $-\mathrm{maxrot}(G) - x \le \rho \le \mathrm{maxrot}(G)$, *where* $x = 0, 1, 2$ *for G being of Type* (1,1), (1,2) *and* (2,2), *respectively.*

Proof We first show the only if part. Let $\mathcal{R}$ be any orthogonal representation of G. By the definition of $\mathrm{maxrot}(G)$ we clearly have that $\mathrm{rot}_{\mathcal{R}}(\pi(s,t)) \le \mathrm{maxrot}(G)$. By definition we also have that $\mathrm{rot}_{\mathcal{R}}(\pi(t,s)) \le \mathrm{maxrot}(G)$ (otherwise by mirroring we could obtain an orthogonal representation $\mathcal{R}'$ with $\mathrm{rot}_{\mathcal{R}'}(\pi(s,t)) > \mathrm{maxrot}(G)$) and hence with Lemma 4 we obtain $-\mathrm{rot}(\pi(s,t)) - x \le \mathrm{maxrot}(G)$.

It remains to show that, for any given ρ in the range, we can find a valid orthogonal representation such that $\mathrm{rot}(\pi(s,t)) = \rho$. If $-1 \le \rho \le \mathrm{maxrot}(G)$, let $\mathcal{E}$ be a planar embedding of G with $\mathrm{maxrot}_{\mathcal{E}}(G) = \mathrm{maxrot}(G)$. Then the desired orthogonal representation exists due to Theorem 1.

If $\rho \le -2$ holds, by Lemma 4 we need to find a valid orthogonal representation $\mathcal{R}$ with $\mathrm{rot}_{\mathcal{R}}(\pi(t,s)) = -\rho - x =: \rho'$. Note that, by the definitions of ρ and x, we have that $0 \le \rho' \le \mathrm{maxrot}(G)$. A valid orthogonal embedding $\mathcal{R}'$ of G with $\mathrm{rot}_{\mathcal{R}'}(\pi(s,t)) = \rho'$ can be found as above. We obtain $\mathcal{R}$ by mirroring $\mathcal{R}'$. □

Note that if s (or t) has degree 1, then its incident edge allows for three different rotations and hence the range of valid rotations contains at least three integers. This observation together with the theorem yields the following.

Corollary 1 *Let G be an st-graph with positive flexibility. If G admits a valid orthogonal representation, then* $\mathrm{maxrot}(G) \ge 1$ *if G is of Type* (1,1) *or* (1,2) *and* $\mathrm{maxrot}(G) \ge -1$ *if G is of Type* (2,2).

Theorem 3 shows that an st-graph G with $\deg(s) = \deg(t) = 1$ essentially behaves like a single edge st with flexibility $\mathrm{maxrot}(G)$. The following lemma shows that we can replace any st-graph with $\deg(s), \deg(t) \le 2$ in a graph G by a different st-graph of the same type and with the same maximum rotation without changing $\mathrm{maxrot}(G)$. Figure 7 illustrates the lemma and its proof.

Lemma 5 *Let $G = (V, E)$ be an st-graph with positive flexibility and let $\{u, v\}$ be a split pair of G that splits G into two components G^- and H such that G^- contains s and t and H is an st-graph of Type* (1,1), *Type* (1,2) *or Type* (2,2) (*with respect to vertices u and v*). *Let H' be an st-graph with designated vertices u', v' of the same type as H with* $\mathrm{maxrot}(H') = \mathrm{maxrot}(H)$, $\deg(u) = \deg(u')$ *and* $\deg(v) = \deg(v')$.

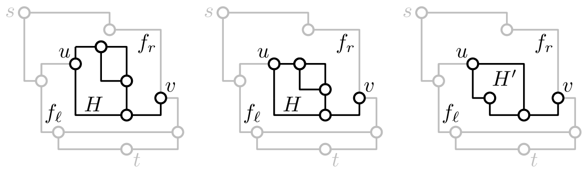


Fig. 7 Illustration of Lemma 5, st-graph G with split pair $\{u, v\}$ splitting off H (*left*), replacement of H with a tight orthogonal representation (*middle*) and replacement of H with a graph H' with $\mathrm{maxrot}(H) = \mathrm{maxrot}(H') = 3$ (*right*)

Then G admits a valid orthogonal representation $\mathcal{R}$ with $\mathrm{rot}_{\mathcal{R}}(\pi(s,t)) = \rho$ if and only if the graph G', which is obtained from G by replacing H with H', admits a valid orthogonal representation $\mathcal{R}'$ with $\mathrm{rot}_{\mathcal{R}'}(\pi(s,t)) = \rho$.

Proof Given a valid orthogonal representation $\mathcal{R}$ of G we wish to find a valid orthogonal representation $\mathcal{R}'$ of G' such that $\mathrm{rot}_{\mathcal{R}}(\pi(s,t)) = \mathrm{rot}_{\mathcal{R}'}(\pi(s,t))$. The other direction is symmetric.

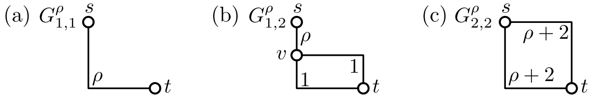
We first treat the case that H is of Type (1,1). Let $\mathcal{S}$ be the restriction of $\mathcal{R}$ to H. By Theorem 3 we have that $-\mathrm{maxrot}(H) \le \mathrm{rot}_{\mathcal{S}}(\pi(u,v)) \le \mathrm{maxrot}(H)$ and hence, again by Theorem 3, there exists a valid orthogonal representation $\mathcal{S}'$ of H' with $\mathrm{rot}(\pi(u',v')) = \mathrm{rot}(\pi(u,v))$. Since H is of Type (1,1) we have that $\mathrm{rot}_{\mathcal{S}'}(u') = \mathrm{rot}_{\mathcal{S}}(u)$, $\mathrm{rot}_{\mathcal{S}'}(v') = \mathrm{rot}_{\mathcal{S}}(v)$, $\mathrm{rot}_{\mathcal{S}'}(\pi(u',v')) = \mathrm{rot}_{\mathcal{S}}(\pi(u,v))$ and $\mathrm{rot}_{\mathcal{S}'}(\pi(v',u')) = \mathrm{rot}_{\mathcal{S}}(\pi(v,u))$. Hence by plugging $\mathcal{S}'$ into the restriction of the orthogonal representation $\mathcal{R}$ to G^- we obtain the desired representation $\mathcal{R}'$ of G'.

In case H is of Type (1,2), we can assume that u has degree 2 and $\deg(v) = 1$. Then the angle at u in f_i is 90° or 180° where f_i is the inner face of H incident to u. If this angle is 90°, that is $\mathcal{S}$ is tight, we replace it by a corresponding tight embedding of H' with the same rotation, which exists by Theorem 3. For the case that we have an angle of 180° at u in f_i, we show how to construct an orthogonal representation $\mathcal{R}''$ of G having the same planar embedding as $\mathcal{R}$ such that $\mathrm{rot}_{\mathcal{R}''}(\pi(s,t)) = \mathrm{rot}_{\mathcal{R}}(\pi(s,t))$ and the angle at u in f_i is 90°. Then $\mathcal{R}'$ can be constructed from $\mathcal{R}''$ as above.

By Theorem 3 there exists a valid and tight orthogonal representation $\mathcal{S}''$ of H with either $\mathrm{rot}_{\mathcal{S}''}(\pi(u,v)) = \mathrm{rot}_{\mathcal{S}}(\pi(u,v))$ or $\mathrm{rot}_{\mathcal{S}''}(\pi(v,u)) = \mathrm{rot}_{\mathcal{S}}(\pi(v,u))$. Without loss of generality assume the former, the other case is symmetric. Since we have increased the outer angle at u we have that $\mathrm{rot}_{\mathcal{S}''}(u) = \mathrm{rot}_{\mathcal{S}}(u) - 1$ and hence $\mathrm{rot}_{\mathcal{S}''}(\pi(v,u)) = \mathrm{rot}_{\mathcal{S}}(\pi(v,u)) + 1$. Let f_ℓ and f_r be the faces in G whose boundaries contain $\pi(u,v)$ and $\pi(v,u)$, respectively. Then we obtain $\mathcal{R}''$ by plugging $\mathcal{S}''$ into the restriction of $\mathcal{R}$ to G^- such that the angle at u in f_r is increased by 90° to 180°. Since the angle at u in f_i was decreased by 90° the sum of angles around u remains 360°. Additionally, by increasing the angle at u in f_r, its rotation is decreased by 1, which compensates the increased rotation along $\pi(v,u)$. Hence $\mathcal{R}''$ is the claimed orthogonal representation. This finishes the treatment of graphs of Type (1,2). Graphs of Type (2,2) can be treated analogously. □

We now present three especially simple families of replacement graphs, called *gadgets*, for st-graphs of Types (1,1), (1,2) and (2,2), respectively; see Fig. 8. Let ρ

Fig. 8 Gadgets for st-graphs with maximum rotation ρ



be a positive integer. The graph $G^{\rho}_{1,1}$ is an edge st with $\text{flex}(st) = \rho$. The graph $G^{\rho}_{1,2}$ has three vertices s, v, t and two edges between t and v, both with flexibility 1, and the edge vs with flexibility ρ. The gadget $G^{\rho}_{2,2}$ consists of two parallel edges between s and t, both with flexibility $\rho + 2$. Note that by Corollary 1 all edges of our gadgets have positive flexibility and that $\text{maxrot}(G^{\rho}_{1,1}) = \text{maxrot}(G^{\rho}_{1,2}) = \text{maxrot}(G^{\rho}_{2,2}) = \rho$. Moreover, each of these graphs has a unique embedding with s and t on the outer face.

We now describe an algorithm that computes $\text{maxrot}(G)$ for a given st-graph G with positive flexibility or decides that G does not admit any valid orthogonal representation. We use the SPQR-tree $\mathcal{T}$ of $G + st$, rooted at the Q-node corresponding to st to represent all planar embeddings of G with s and t on the outer face. Our algorithm processes the nodes of $\mathcal{T}$ in a bottom-up fashion and computes the maximum rotation of each pertinent graph from the maximum rotations of the pertinent graphs of its children. For each node μ we maintain a variable $\text{maxrot}(\mu)$. We will prove later that, after processing a node μ, we have that $\text{maxrot}(\mu) = \text{maxrot}(\text{pert}(\mu))$. For each Q-node μ we initialize $\text{maxrot}(\mu)$ to be the flexibility of the corresponding edge. We now show how to compute $\text{maxrot}(\mu)$ from the maximum rotations of its children. We make a case distinction based on the type of μ.

If $\boldsymbol{\mu}$ **is an R-node**, then let $\mu_1, \dots, \mu_k$ be the children of μ. Each virtual edge in $\text{skel}(\mu)$ represents at least one incidence of an edge of G to its poles. Since $\text{skel}(\mu)$ is 3-connected, each node has degree at least 3 and hence no virtual edge can represent more than two incidences, that is the poles of μ_i have degree at most 2 in the pertinent graph of μ_i, for $1 \le i \le k$. As we already know the maximum rotations of these pertinent graphs, we can simply replace each of them by a corresponding gadget; we call the resulting graph G_μ. As the only possible embedding choice for each of the gadgets consists of reordering a pair of parallel edges with the same flexibility, the embeddings of the gadgets can be assumed to be fixed. Thus, it is sufficient to compute the maximum rotations of G_μ for the only two embeddings $\mathcal{E}_1$ and $\mathcal{E}_2$ induced by the embeddings of $\text{skel}(\mu)$. We set $\text{maxrot}(\mu) = \max\{\text{maxrot}_{\mathcal{E}_1}(G_\mu), \text{maxrot}_{\mathcal{E}_2}(G_\mu)\}$ if one of them admits a valid representation. Otherwise we stop and return “infeasible”.

If $\boldsymbol{\mu}$ **is a P-node**, then we treat μ similar as in the case where μ is an R-node. Again, each pole has degree at least 3 in $\text{skel}(\mu)$ and hence no virtual edge can represent more than two edge incidences. We replace each virtual edge with the corresponding gadget and try all possible embeddings of $\text{skel}(\mu)$, which are at most six, and store the maximum rotation or stop if none of the embeddings admits a valid representation.

If $\boldsymbol{\mu}$ **is an S-node**, then let $\mu_1, \dots, \mu_k$ be the children of μ. We set $\text{maxrot}(\mu) = \sum_{i=1}^{k} \text{maxrot}(\mu_i) + k - 1$.

Theorem 4 *Given an st-graph $G = (V, E)$ with positive flexibility it can be checked in $O(n^{3/2})$ time whether G admits a valid orthogonal representation. In the positive case* $\text{maxrot}(G)$ *can be computed within the same time complexity.*

Proof We prove that after the algorithm has processed node μ the invariant $\text{maxrot}(\mu) = \text{maxrot}(\text{pert}(\mu))$ holds. The proof is by induction on the height h of the SPQR-tree $\mathcal{T}$ of $G + st$. Let μ be the node of $\mathcal{T}$ whose parent corresponds to st.

If $h = 1$, then $G - st$ is a single edge e and μ its corresponding Q-node. Since $\text{maxrot}(G)$ equals $\text{flex}(e)$ the claim holds. For $h > 1$, let $\mu_1, \dots, \mu_k$ be the children of μ. By induction we have that $\text{maxrot}(\mu_i) = \text{maxrot}(\text{pert}(\mu_i))$ for $i = 1, \dots, k$. We make a case distinction based on the type of μ.

If μ is an R- or a P-node, then by Lemma 5 we have that $\text{maxrot}(G_\mu) = \text{maxrot}(\text{pert}(\mu))$ and since the gadgets have a unique embedding we consider all relevant embeddings of G_μ. If none of the embeddings admits a valid orthogonal representation, then obviously also $\text{pert}(\mu)$ and thus G do not admit valid orthogonal representations.

If μ is an S-node and the pertinent graphs of its children admit a valid orthogonal representation, then there always exists a valid orthogonal representation of $\text{pert}(\mu)$. Let $H_1, \dots, H_k$ be the pertinent graphs of the children of μ and let $v_1, \dots, v_{k+1}$ be the vertices in $\text{skel}(\mu)$ such that v_i and v_{i+1} are the poles of H_i. By Theorem 3 there exist tight orthogonal representations $\mathcal{R}_1, \dots, \mathcal{R}_k$ of $H_1, \dots, H_k$ with $\text{rot}(\pi(v_i, v_{i+1})) = \text{maxrot}(\mu_i)$. We put these orthogonal representations together such that the angles at the nodes $v_2, \dots, v_k$ on $\pi(v_1, v_{k+1})$ are 90°. Hence we get an orthogonal representation of $\text{pert}(\mu)$ with $\text{rot}(\pi(v_1, v_{k+1})) = \sum_{i=1}^{k} \text{maxrot}(\mu_i) + k - 1$. On the other hand if we had an orthogonal representation of $\text{pert}(\mu)$ with a higher rotation, then at least one of its children μ_i would need to have a rotation that is bigger than $\text{maxrot}(\mu_i)$, because the rotation at vertices can be at most 1.

This proves the correctness of the algorithm. For the running time note that the SPQR-tree can be computed in linear time [7]. Computing $\text{maxrot}(\mu)$ for a given node μ from the maximum rotations of its children takes $O(|\text{skel}(\mu)|^{3/2})$ time by Theorem 2 since $\text{skel}(\mu)$ has only a constant number of embeddings. The overall running-time follows from the fact that the total size of all skeletons is linear. □

This theorem can be used to solve FLEXDRAW for biconnected 4-planar graphs with positive flexibility. Each such graph G admits a valid orthogonal representation if and only if one of the graphs $G - e$, $e \in E(G)$ (which is an st-graph with respect to the endpoints of e) admits a valid orthogonal representation such that e can be added to this representation. We claim that this is possible if and only if $\text{maxrot}(G - e) + \text{flex}(e) \geq 2$. Let s and t be the endpoints of e. To show necessity, assume G admits a valid orthogonal representation $\mathcal{R}$ with e on the outer face. Possibly after mirroring $\mathcal{S}$, we can assume without loss of generality that the outer face of $\mathcal{R}$ is bounded by e and $\pi(t, s)$. Let r denote the edge description of e in the outer face. Due to Theorem 1 we can reduce $\text{rot}(r)$ to be at most 0, then e has $-\text{rot}(r)$ bends. As the rotation along the path $\pi(s, t)$ in $G - e$ is at most $\text{maxrot}(G - e)$ and $-\text{rot}(r) \leq \text{flex}(e)$ since $\mathcal{R}$ is a valid orthogonal representation, the necessity of the inequality $\text{maxrot}(G - e) + \text{flex}(e) \geq 2$ follows from Lemma 3.

On the other hand, let $\mathcal{S}$ be a tight orthogonal representation of $G - e$ such that $\text{rot}(\pi(s, t)) = \min\{\text{maxrot}(G - e), 2\}$, which exists due to the definition of $\text{maxrot}(G - e)$ and Theorem 1. Since $\mathcal{S}$ is tight, we can add e to $\mathcal{S}$ such that e and $\pi(t, s)$ form the new outer face and the resulting orthogonal representation is tight. Let r be the edge description of e in the outer face. Due to Lemma 3 we have $\mathrm{rot}(r) = \mathrm{rot}(\pi(s,t)) - 2$. Since this is at most 0, we have that e has $-\mathrm{rot}(r) = -\mathrm{rot}(\pi(s,t)) + 2$ bends. If the inequality $\mathrm{maxrot}(G - e) + \mathrm{flex}(e) \geq 2$ holds, then this yields that e has at most $\mathrm{flex}(e)$ bends, which shows sufficiency.

If $\mathrm{flex}(e) \leq 3$, we get $\mathrm{maxrot}(G) \geq 2 - \mathrm{flex}(e) \geq -1$, and thus using Theorem 1 we can find a tight orthogonal representation with $\mathrm{rot}(\pi(s,t)) = 2 - \mathrm{flex}(e)$. We add e to this orthogonal representation in such a way that the new internal face bounded by e and $\pi(s,t)$ has a total rotation of 4, and thus forms a valid orthogonal representation of G. Then e has exactly $\mathrm{flex}(e)$ bends. We obtain the following theorem; the running time is due to $O(n)$ applications of the algorithm for st-graphs.

Theorem 5 FLEXDRAW *can be solved in time* $O(n^{5/2})$ *for biconnected* 4*-planar graphs with positive flexibility.*

5 Quadratic-Time Implementation

In this section we improve the running time of the algorithm for the biconnected case to $O(n^2)$. In the previous section we have shown that checking whether a biconnected 4-planar graph G admits a valid orthogonal representation with a given edge e on the external face can be done in $O(n^{3/2})$ time. Recall that the running time stems from the fact that, for each embedding $\mathcal{E}$ of the skeleton of each node μ of the SPQR-tree, we have to compute the maximum rotation of its pertinent graph with respect to this embedding. For a graph with a fixed embedding, this is done by a two-step process, as in Theorem 2. We first compute in $O(|\mathrm{skel}(\mu)|^{3/2})$ time an arbitrary feasible flow in an instance of Tamassia's flow network for the skeleton where some of the edges are replaced by gadgets. We call this a *base flow*. In a second step, we compute a maximum flow in the residual network with respect to the base flow in $O(|\mathrm{skel}(\mu)|)$ time. The running time is hence dominated by the computation of base flows.

To obtain an algorithm for the case of biconnected graphs, we simply try every edge as the reference edge that has to lie on the external face, resulting in $O(n)$ applications of the above algorithm. However, when we choose a new root of the SPQR-tree and perform the traversal of the SPQR-tree, a lot of information that was already acquired in previous iterations is recomputed. In this section we show that the information computed in different traversals of the SPQR-tree can be reused to improve the time that is required to compute base flows to $O(n^2)$ total time, which improves the running time of the algorithm for the biconnected case to $O(n^2)$.

Let G be a biconnected 4-planar graph with a positive flexibility function flex and let μ be a node of the SPQR-tree of G with an arbitrary embedding $\mathcal{E}$ of $\mathrm{skel}(\mu)$. Note that in this section we consider two embeddings to be equal if they differ only by the choice of the external face. Let further e be the reference edge of $\mathrm{skel}(\mu)$, that is the edge that μ shares with its parent. Recall that the expansion graph of a virtual edge ε in $\mathrm{skel}(\mu)$ is the subgraph of G represented by this edge. Let $G(\mu,\mathcal{E},e)$ be the skeleton of μ with embedding $\mathcal{E}$, where all edges except for e are replaced by the corresponding gadget according to the maximum rotation of their expansion graphs, as used by the algorithm of the previous section. Our goal is to reuse flow information that was computed for $G(\mu, \mathcal{E}, e)$ when processing $G(\mu, \mathcal{E}, e')$, where e' is a different edge of skel(μ) serving as the reference edge. To this end, we define a set of operations on such graphs that allow us to transform $G(\mu, \mathcal{E}, e)$ into $G(\mu, \mathcal{E}, e')$ in $O(|\mathrm{skel}(\mu)|)$ time. We show that, while performing these operations, a flow ϕ in the flow network of $G(\mu, \mathcal{E}, e)$ can be updated to a flow ϕ' in the flow network of $G(\mu, \mathcal{E}, e')$ such that the defects of ϕ and ϕ' differ only by a constant. We will then use ϕ' as a starting point to quickly check feasibility of the flow network of $G(\mu, \mathcal{E}, e')$.

We first show that given a flow network, the knowledge of a flow with small defect in the network allows to quickly check whether a feasible flow exists and that a flow with minimum defect in an instance of Tamassia's flow network can be computed in time that is quadratic in the size of the network.

Lemma 6 *Given a flow network $N = (V, A, \ell, u, q)$ together with a flow ϕ, a flow ϕ' of N with minimum defect can be computed in $O(|N|\,\mathrm{def}(\phi))$ time.*

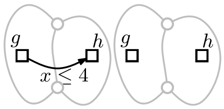
Proof We apply the algorithm of Ford and Fulkerson [5]. We iteratively augment the flow with augmenting paths from a vertex with positive defect to a vertex with negative defect. A single path of this type can be computed in $O(|N|)$ time. The algorithm stops when no such path exists. Since each such path decreases the defect of the current flow by 2 the algorithm takes at most def(ϕ) iterations. □

Corollary 2 *Let $G = (V, E)$ be a 4-planar graph with n vertices and a fixed embedding. A flow with minimum defect in the corresponding flow network of Tamassia can be computed in $O(n^2)$ time.*

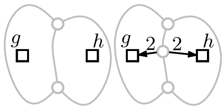
Proof For each vertex v, denote the set of incident faces by F_v. We define an initial flow ϕ by setting $\phi(v, f) = 1$ for each v and each $f \in F_v$. All other arcs receive a flow of 0. As the total amount of demands for Tamassia's flow network is in $O(n)$, the claim follows from Lemma 6. □

Next we define the operations that allow us to transform Tamassia's flow network for $G(\mu, \mathcal{E}, e)$ into Tamassia's flow network for $G(\mu, \mathcal{E}, e')$ where e and e' are different virtual edges in skel(μ). The operations allow to preserve a flow with small defect. Let G be a planar graph, let $e = uv$ be an edge of G with incident faces g and h and let ϕ be a flow in Tamassia's flow network of G. We introduce the following basic operations on G, respectively on ϕ. For an illustration see Fig. 9.

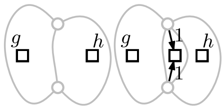
1. Setting the flow along an edge e with flow at most 4 to 0.
2. Subdividing an edge uv with flow 0 into uw and vw.
3. Doubling an edge with flow 0.
4. Removing a subdivision vertex and inserting an edge with capacity ∞ between its two neighbors.
5. Removing one edge of a pair of double edges and setting the capacity of the remaining edge to ∞.



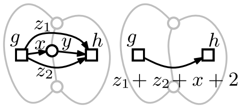
(a) Reducing flow along an edge with $\phi_e(g,h) \leq 4$ creates a defect of at most 8.



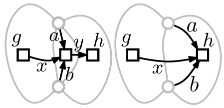
(b) Subdividing an edge e with $\phi_e(g,h) = 0$ does not increase the defect.



(c) Doubling an edge without flow increases the defect by at most 4.



(d) Removing a subdivision vertex and inserting an edge with capacity ∞ between its neighbors does not increase the defect.



(e) Replacing a double edge by a single edge with capacity ∞ does not increase the defect.

Fig. 9 Manipulation of a graph, the corresponding flow network of Tamassia and its flow. Edges of the flow network whose flow changes are shown in *black*. The *labels* indicate the amount of flow on these edges

Lemma 7 *Let G be a planar graph with fixed embedding $\mathcal{E}$ and let ϕ be a flow in Tamassia's flow network of G with respect to $\mathcal{E}$. Let G' be the graph resulting from G by applying Operation i. Then a flow ϕ' in Tamassia's flow network of G' with* $\operatorname{def}(\phi') \leq \operatorname{def}(\phi) + \operatorname{def}_i$ *can be computed in linear time*, *where* $\operatorname{def}_i = 8, 0, 4, 0, 0$ *for Operations* 1–5.

Proof The operations are illustrated in Fig. 9. We consider the operations one by one. For Operation 1, assume that e is incident to faces g and h and that $|\phi_e(g,h)| \leq 4$. To obtain ϕ' from ϕ we simply set $\phi'_e(g,h) = 0$. Clearly, ϕ' is a flow as it satisfies the lower and upper bounds on all edges. We have $\operatorname{def}_{\phi'}(g) \leq \operatorname{def}_{\phi}(g) + 4$ and $\operatorname{def}_{\phi'}(h) \leq \operatorname{def}_{\phi}(h) + 4$. As the operation does not change the defect of any other node we have $\operatorname{def}(\phi') \leq \operatorname{def}(\phi) + 8$. See Fig. 9a for an illustration.

For Operation 2, we replace an edge $e = uv$ with incident faces g and h by two edges uw and wv. We set $\phi' = \phi$. Note that inserting the vertex increases the demands of g and h by 2. On the other hand, the new vertex w has four flow units. We route two units of flow from w to h and two units from w to g. Hence, the resulting flow ϕ' has the same defect as ϕ, see Fig. 9b.

For Operation 3, see Fig. 9c, we double the edge $e = uv$, whose corresponding dual arcs (g,h) and (h,g) have flow 0. Let k be the new face that is bounded by the two parallel edges between u and v. We route one unit of flow from each of u and v to k. This increases the defects of u and v by at most 1, each. The resulting function is a flow, as all capacity restrictions hold. Moreover, since the demand of k is 0, its defect is 2. All other nodes keep their defects and we have $\operatorname{def}(\phi') \leq \operatorname{def}(\phi) + 4$.

For Operation 4, we show how to remove a subdivision vertex w and connect its neighbors u and v by an edge with capacity ∞, see Fig. 9d. Let $z_1 = \phi_{uw}(g,h)$ and let $z_2 = \phi_{wv}(g,h)$ be the amounts of flow from g to h via uw and wv, respectively. Let further $x = \phi(g,w)$ and $y = \phi(w,h)$. Note that by definition of the flow network it is $\operatorname{def}(w) = x - y + 4$. We replace the path uwv by the single edge e and set $\phi'_e(g,h) = z_1 + z_2 + x + 2$. For g, the difference d_g of incoming and outgoing flow with respect to ϕ can be written as $d_g = F_g - z_1 - z_2 - x$, where F_g is the difference of the incoming and outgoing flow along the remaining edges. With respect to ϕ' the same difference d'_g is $d'_g = F_g - z_1 - z_2 - x - 2$. Hence $d_g - d'_g = 2$. Since the removal of w also reduces the demand of g by 2, its defect is preserved. Similarly, for h, we have $d_h = F_h + z_1 + z_2 + y$ and $d'_h = F_h + z_1 + z_2 + x + 2$. Then $d_h - d'_h = y - x - 2 = 2 - \mathrm{def}(w)$. Again the removal of w decreases the demand of h by 2, hence $|\mathrm{def}_{\phi'}(h)| \leq |\mathrm{def}\,\phi(h)| + |\mathrm{def}_{\phi}(w)|$. Since w is removed, the overall defect is not increased, that is $\mathrm{def}(\phi') = \mathrm{def}(\phi)$.

Finally, we show how to remove a double edge, that is how to implement Operation 5; see Fig. 9e. Let e_1 and e_2 be the two edges between u and v, let k be the face incident to both of them, and let g and h be the other faces incident to e_1 and e_2, respectively. Let $x = \phi_{e_1}(g,k)$, $y = \phi_{e_2}(k,h)$, $a = \phi(u,k)$ and $b = \phi(v,k)$. Note that $\mathrm{def}(k) = a + b + x - y$ since the demand of k is 0 as it has only two incident vertices.

We replace the double edges e_1, e_2 by a single edge e with capacity ∞. We set $\phi'_e(g,h) = x$, $\phi'(u,h) = \phi(u,h) + a$ and $\phi'(v,h) = \phi(v,h) + b$. Clearly, the defects of g, u and v do not change. For h, observe that the change of incoming flow is $y - (x + a + b) = -\mathrm{def}_{\phi}(k)$. Hence, $|\mathrm{def}_{\phi'}(h)| \leq |\mathrm{def}_{\phi}(h)| + |\mathrm{def}_{\phi}(k)|$ and since k is removed, the total defect does not increase. □

Note that a sequence of Operations 1–5 can be used to replace a gadget by a single edge and vice versa. The following lemma states that this replacement is always possible while increasing the defect of a given flow only by a certain constant.

Lemma 8 *Given $G(\mu, \mathcal{E}, e)$ together with a flow ϕ in its corresponding flow network in which e has at most 4 units of flow, a flow ϕ' in the flow network of $G(\mu, \mathcal{E}, e')$ with $\mathrm{def}(\phi') \leq \mathrm{def}(\phi) + 28$ can be computed in linear time.*

Proof To transform $G(\mu, \mathcal{E}, e)$ into $G(\mu, \mathcal{E}, e')$ we first replace e by the corresponding gadget representing its expansion graph. Then we transform the gadget that represents e' in $G(\mu, \mathcal{E}, e)$ into the single edge e'. Finally, we may have to change the outer face. We now show that the desired flow ϕ' can be computed while performing all these operations, starting with $\phi = \phi'$.

The cost of the first step depends on the Type of the expansion graph of e. If the expansion graph of e is of Type (1,1), no change is necessary. For Type (1,2) we reduce the flow along e (Operation 1), subdivide e (Operation 2) and double one of the resulting edges (Operation 3). For Type (2,2) we first reduce the flow of e (Operation 1) and then double it (Operation 3). In all cases we apply Lemma 7 for each operation to obtain the flow ϕ'. In the worst case the total increase of defect is 12.

In the second step, we replace a gadget by the edge e'. If the expansion graph of e' is of Type (1,1), no change is necessary. For Type (1,2), we replace the double edge of the gadget by a single edge (Operation 5) and then remove the subdivision vertex (Operation 4). For Type (2,2) we replace the double edge by a single edge (Operation 5). By Lemma 7, this yields the flow ϕ' with $\mathrm{def}(\phi') \leq \mathrm{def}(\phi)$.

Both steps together thus yield a flow ϕ' with $\mathrm{def}(\phi') \leq \mathrm{def}(\phi) + 12$. Finally, by the definition of Tamassia's flow network, to change the outer face to an interior face we decrease its demand by 8 and we increase the demand of an inner face by 8 to make it the outer face. The defect of ϕ' in the resulting flow network is therefore at most 28. □

We apply this lemma to save computing time when processing the graph $G(\mu, \mathcal{E}, e)$. If we process μ for the first time with embedding $\mathcal{E}$ (embeddings are considered equal if they only differ by the choice of the external face), using Lemma 2, we compute in $O(|\operatorname{skel}(\mu)|^2)$ time a flow ϕ with minimum defect in the corresponding flow network, where the flexibility of e is set to 4. We now distinguish several cases.

If ϕ has defect 0, it is a valid solution and we have found our base flow. Moreover, we store ϕ along with μ and $\mathcal{E}$ for future reuse. If the defect of ϕ is greater than 0 and at most 28, we know that μ does not admit a valid orthogonal representation with embedding $\mathcal{E}$. Again, we store ϕ for future reuse. If the defect of ϕ is greater than 28, we store the information that μ does not admit a valid orthogonal representation with embedding $\mathcal{E}$, independently of the choice of the reference edge and of the outer face. This is true since otherwise we could apply Lemma 7 to obtain a flow ϕ' for $G(\mu, \mathcal{E}, e)$ with defect at most 28, contradicting the optimality of ϕ.

Whenever we encounter μ with embedding $\mathcal{E}$ again, but this time with reference edge e', we can either conclude that it does not admit a valid orthogonal representation (if the embedding is marked as invalid for μ), or check in $O(|\operatorname{skel}(\mu)|)$ time whether it admits a feasible flow as follows. Let ϕ be the stored flow. By applying Lemma 7 we can construct in $O(|\operatorname{skel}(\mu)|)$ time a flow ϕ' in the flow network of $G(\mu, \mathcal{E}, e')$ with $\operatorname{def}(\phi') \leq \operatorname{def}(\phi) + 28 \leq 56$. We then apply Lemma 6 to compute a feasible base flow in $O(|\operatorname{skel}(\mu)|)$ time, if it exists.

Since each node μ has only a constant number of embeddings, the total time for all base flow computations is bounded by $O(n^2)$. After that, checking a node μ with a given embedding can be done in $O(|\operatorname{skel}(\mu)|)$ time. Since each node is checked at most linearly often for each embedding of its skeleton, the total running time is in $O(n^2)$. We have proved the following theorem.

Theorem 6 FLEXDRAW *can be solved in $O(n^2)$ time for biconnected 4-planar graphs with positive flexibility.*

6 Connected Graphs

In this section we generalize our results to connected 4-planar graphs that are not necessarily biconnected. We analyze the conditions under which orthogonal representations sharing a cutvertex can be combined and use the block-cutvertex tree to derive an algorithm that decides whether a connected 4-planar graph with positive flexibility admits a valid orthogonal representation.

Lemma 9 *Let G be a connected 4-planar graph with positive flexibility and let v be a cutvertex with corresponding cut components $H_1, \dots, H_k$. Then G admits a valid orthogonal representation if and only if all cut components H_i have valid orthogonal representations such that at most one of them has v not on the outer face.*

Fig. 10 A degree-2 cutvertex with 180° angles in both faces

Proof The only if part is clear since a valid orthogonal representation of G induces valid orthogonal representations of all cut components H_i such that at most one of them does not have v on its outer face. Now let $\mathcal{S}_i$ be valid orthogonal representations of the cut components H_i for $i = 1, \ldots, k$ such that at most one of them does not have v on its outer face.

If all of them have v on their outer face, then by Lemma 2 we can assume that these representations are tight. Then it is clear that the components $H_1, \ldots, H_k$ can be merged together in v while maintaining their representations $\mathcal{S}_i$.

Otherwise, one of the representations, without loss of generality $\mathcal{S}_1$, does not have v on the outer face. If v has degree 1 in a cut component H_i, then the incident edge is a bridge. Clearly, all free incidences in v with respect to the orthogonal representation $\mathcal{S}_i$ of H_i lie in the same face. Thus, for the case that v has degree at least 2 in at most one of the cut components, this allows to merge their orthogonal representations.

The only problem that can arise is that there are exactly two components H_1 and H_2, v has degree 2 in both of them, and the angles incident to v in H_1 are 180°. Figure 10 illustrates how we resolve this situation. If at least one of the two edges incident to v has a bend, we decrease its number of bends and change the angles at v appropriately, yielding angles of 90° and 270° in the faces incident to v. If none of the edges incident to v has a bend, we achieve the same result by increasing the number of bends on one of the two edges. Note that the orthogonal representation remains valid since the flexibility of every edge is a least 1. □

Now let G be a connected 4-planar graph with positive flexibility and $\mathcal{B}$ its block-cutvertex tree. Let further B be a block of G that is a leaf in $\mathcal{B}$ and let v be the unique cutvertex of B.

If B is the whole graph G, then we return "true" if and only if G admits any valid orthogonal representation. This can be checked with the algorithm from the previous section.

If B is not the whole graph G we check whether B admits a valid orthogonal representation having v on its outer face. This can be done with the algorithm from the previous section by rooting the SPQR-tree of B at all edges incident to v. If it does admit such a representation, then by Lemma 9 G admits a valid orthogonal representation if and only if the graph G', which is obtained from G by removing the block B, admits a valid orthogonal representation. We check G' recursively. If B does not admit such a representation, then we mark B and proceed with another unmarked leaf. If we ever encounter another block B' that has to be marked we return "infeasible", as in this case B should be embedded in the interior of B' and vice versa, which is obviously impossible. Checking a single block B requires $O(|B|^2)$ time by Theorem 6. Since the total size of all blocks is linear, the total running time is $O(n^2)$. This proves the following theorem, which is the main result of this paper.

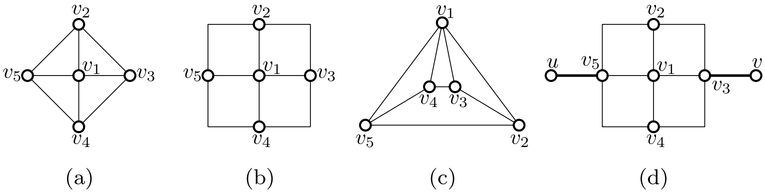


Fig. 11 A graph that is almost rigid graph, even if every edge has flexibility 1

Theorem 7 FLEXDRAW *can be solved in* $O(n^2)$ *time for* 4-*planar graphs with positive flexibility.*

7 Complexity

In this section, we consider the complexity of FLEXDRAW for cases that lie between 0-embeddability and the case of positive flexibility. For example, it is an interesting question, whether FLEXDRAW can still be solved efficiently, if the subgraph consisting only of the edges with flexibility 0 has a special structure. Since 0-embeddability can be solved in polynomial time for series-parallel graphs and graphs with maximum degree 3 [2], one might hope for an algorithm that solves FLEXDRAW efficiently if the edges with flexibility 0 form a graph in one of these classes. However, this is unlikely; we show $\mathcal{NP}$-hardness for the case where the subgraph with flexibility 0 is a collection of disjoint stars. We in fact show that the problem is $\mathcal{NP}$-hard even if the stars are required to be spanning or if the subgraph with flexibility 0 forms a spanning tree. The construction relies on a basic building block, which is described next.

Consider the *wheel* on five vertices, which consists of a cycle on vertices $v_2, \ldots, v_5$ and the center vertex v_1 that is connected to all other vertices; see Fig. 11a. In the following we assume that the flexibility of each edge of the wheel is 1. A corresponding flex-drawing is shown in Fig. 11b. We claim that every valid orthogonal representation of the wheel has the same outer face with the same shape, i.e., with the same list of edge descriptions associated with it. To see this, consider Tamassia's flow network for the wheel in the embedding shown in Fig. 11a. Since the outer face is incident to four vertices, it has a demand of 12. On the other hand, it can receive at most 8 units of flow from $v_2, \ldots, v_5$ and at most 4 units of flow via the incident edges. Hence, in any feasible flow the outer face must receive two units of flow from each of its incident vertices, and one unit of flow from each incident edge. This completely describes the outer face. The only degree of freedom is that the center vertex can be rotated by 90 degrees to the left or to the right. Moreover, all other embeddings of the wheel do not allow for a 1-bend embedding. Since the wheel is 3-connected, the only embedding choice is the outer face. Up to renaming the vertices the only embedding that is different from the embedding in Fig. 11a is shown in Fig. 11c. In Tamassia's flow network the outer face has a demand of 10. However, it can receive at most one unit of flow from v_1, at most two units of flow from v_2 and v_5, and at most three units of flow via its incident edges, which adds up to a total of 8. Hence, the wheel does not admit a 1-bend drawing with this embedding. With these considerations it is easy to prove the following theorem.

Theorem 8 FLEXDRAW *is $\mathcal{NP}$-hard, even if the subgraph with flexibility* 0 *is a spanning tree or a spanning union of disjoint stars.*

Proof We reduce from 0-embeddability, which is known to be $\mathcal{NP}$-hard [6]. Let $G = (V, E)$ be an instance of 0-embeddability and let G' be the graph that is obtained from G by replacing each edge $uv \in E$ by the gadget shown in Fig. 11d, where the two bold edges have flexibility 0 and all other edges in the gadget have flexibility 1. As in each flex-drawing the rotation between the two vertices of degree 1 is 0, it follows that G' admits a flex-drawing if and only if G admits a 0-embedding. Obviously the edges with flexibility 0 form a collection of disjoint stars, each having a vertex of the original graph as its center. The only vertices that are not contained in one of the stars are the vertices v_1, v_2 and v_4. Obviously, we can ensure that the union of disjoint stars is spanning by simply assigning flexibility 0 to the edges v_1v_2 and v_1v_4 in each of the gadgets.

To show $\mathcal{NP}$-hardness for the case that the subgraph with flexibility 0 is a spanning tree, we first choose a spanning tree T in G. Then we again replace the edges in G by the gadgets shown in Fig. 11d except for the edges contained in T. This yields an equivalent instance G' of FLEXDRAW where the edges with flexibility 0 form a tree containing all vertices of G' except for v_1, v_2 and v_4 of each gadget. However, we can easily ensure that the tree is a spanning tree by adding the edges v_1v_2, v_1v_4 and v_1v_5 for each gadget. □

8 Concluding Remarks

The main result of this work is that FLEXDRAW can be solved efficiently for graphs with positive flexibility. To prove this, we first showed that the set of possible drawings of a graph with positive flexibility can be described by a single number, its maximum rotation. The fact that subgraphs can be substituted by graphs with the same maximum rotation without affecting the overall maximum rotation enabled us to gradually reduce the size of the graph that needed to be considered. This directly led to a polynomial-time algorithm, which together with a specialized out-of-the-box flow algorithm for planar graphs resulted in a running time of $O(n^{5/2})$. We then introduced flows with defects to share more information between different phases of the algorithm. Using these concepts and a careful implementation we were able to reduce the running time to $O(n^2)$.

The efficient algorithm for FLEXDRAW with positive flexibility closes the long-standing complexity gap between 0-embeddability and 2-embeddability. However, the result is much more general, as it enables us to specify the maximum number of bends for each edge, individually. This may have interesting applications in domains such as the layout of UML diagrams, which are typically drawn with orthogonal edges, and where certainly some of the edges are much more important than others, and thus should have few bends, possibly at the cost of more bends at unimportant

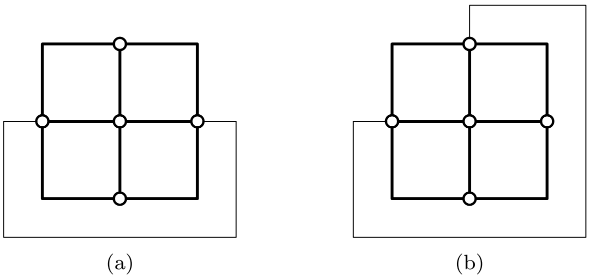


Fig. 12 Examples of graphs that require an edge with several bends in a flex-drawing. The wheel with an additional edge shown in (**a**) requires an edge with four bends in any drawing, if the bold edges have a flexibility of 1. If the embedding is fixed, five bends are necessary for the thin edge in (**b**)

edges. To obtain a nice drawing, it may even be desirable to specify no upper bound at all on the number of bends for unimportant edges. It is straightforward to generalize the results presented in this work to positive flexibility functions flex : $E \longrightarrow \mathbb{N} \cup \{\infty\}$, where some edges may be bent arbitrarily often.

We further explored the complexity gap between 0-embeddability and FLEXDRAW with positive flexibility. We have shown that FLEXDRAW is $\mathcal{NP}$-hard, even if the subgraph consisting of edges with flexibility 0 forms a tree or a union of disjoint stars.

Open Questions We leave open two questions, which we believe to be most interesting. As we have seen, FLEXDRAW remains $\mathcal{NP}$-hard, even if the edges with flexibility 0 form a tree, or a collection of stars. However, the complexity remains open for other graph classes such as matchings. We conjecture that FLEXDRAW remains $\mathcal{NP}$-hard for the case that the edges with flexibility 0 form a matching.

Another interesting question stems from the problem of computing a *nice* flex-drawing. In general, 4-planar graphs admit drawings with at most two bends per edge, with the only exception of the octahedron, which requires an edge with three bends [1]. Thus, in the absence of flexibility constraints a linear number of bends is sufficient. Figure 12 shows that flex-drawings of graphs with positive flexibility may require edges with four bends. How many bends may be required for a flex-drawing of a graph with positive flexibility? Is there a constant C such that for each graph G with positive flexibility that admits a flex-drawing there is one with at most C bends per edge? This also has implications on the time required for actually computing a flex-drawing since computing a drawing from an orthogonal representation requires time that is quadratic in the number of bends [13]. If it was possible that a non-constant number of edges would need to have a non-constant number of bends, this would result in a total running time that is super-quadratic, and thus slower than the algorithm for finding the embedding. We suspect that this is not the case; we conjecture that generally $C = 4$, and that $C = 5$ if the embedding is fixed.

Acknowledgements We thank the anonymous reviewers for their comments, which helped us to improve the presentation of our paper.

References

1. Biedl, T., Kant, G.: A better heuristic for orthogonal graph drawings. Comput. Geom. **9**(3), 159–180 (1998)

2. Di Battista, G., Liotta, G., Vargiu, F.: Spirality and optimal orthogonal drawings. SIAM J. Comput. **27**(6), 1764–1811 (1998)
3. Di Battista, G., Tamassia, R.: On-line maintenance of triconnected components with SPQR-trees. Algorithmica **15**, 302–318 (1996)
4. Di Battista, G., Tamassia, R.: On-line planarity testing. SIAM Journal on Computing **25**(5), 956–997 (1996)
5. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. Can. J. Math. **8**, 399–404 (1956)
6. Garg, A., Tamassia, R.: On the computational complexity of upward and rectilinear planarity testing. SIAM J. Comput. **31**(2), 601–625 (2001)
7. Gutwenger, C., Mutzel, P.: A linear time implementation of SPQR-trees. In: Graph Drawing. Lecture Notes in Computer Science, vol. 1984, pp. 77–90. Springer, Berlin/Heidelberg (2001)
8. Henzinger, M.R., Klein, P.N., Rao, S., Subramanian, S.: Faster shortest-path algorithms for planar graphs. J. Comput. Syst. Sci. **55**(1), 3–23 (1997)
9. Liu, Y., Marchioro, P., Petreschi, R., Simeone, B.: Theoretical results on at most 1-bend embeddability of graphs. Acta Math. Appl. Sin. **8**, 188–192 (1992)
10. 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)
11. Miller, G.L., Naor, J.: Flow in planar graphs with multiple sources and sinks. SIAM J. Comput. **24**(5), 1002–1017 (1995)
12. Morgana, A., de Mello, C.P., Sontacchi, G.: An algorithm for 1-bend embeddings of plane graphs in the two-dimensional grid. Discrete Appl. Math. **141**(1–3), 225–241 (2004)
13. Tamassia, R.: On embedding a graph in the grid with the minimum number of bends. SIAM J. Comput. **16**(3), 421–444 (1987)
14. Whitney, H.: Congruent graphs and the connectivity of graphs. Am. J. Math. **54**(1), 150–168 (1932)