Orthogonal Layout with Optimal Face Complexity: NP-hardness and Polynomial-time Algorithms

Md. Jawaherul Alam[1], Stephen G. Kobourov[1] and Debajyoti Mondal[2]

[1]Department of Computer Science, University of Arizona
[2]Department of Computer Science, University of Manitoba
{mjalam,kobourov}@cs.arizona.edu, jyoti@cs.umanitoba.ca

Abstract. Given a biconnected plane graph G and a nonnegative integer k, we examine the problem of deciding whether G admits a strict-orthogonal drawing (i.e., an orthogonal drawing without bends) such that the reflex face complexity (the maximum number of reflex angles in any face) is at most k. We introduce a new technique to solve the problem in $O(n^{1.5} \min\{k^{1.5}, \log n \log k\})$ time, while no such subquadratic-time solution for arbitrary k was known before. In contrast, if the embedding is not fixed, then we prove that it is the NP-complete to decide whether a planar graph admits a strict-orthogonal drawing with reflex face complexity 4.

1 Introduction

A *t-bend orthogonal drawing* of G is an orthogonal drawing of G, where each edge is drawn as an orthogonal polyline with at most t bends. An orthogonal drawing is *strict* if it does not contain any bend. Such a drawing is also referred to as *bendless* or *no-bend* orthogonal drawing [19]. If G is a *plane graph* (i.e., a planar graph with a fixed planar embedding), then an orthogonal drawing of G is additionally constrained to respect the given planar embedding. The *reflex face complexity* of an orthogonal drawing Γ is the smallest integer k such that each inner face of Γ contains at most k reflex angles, and the outer face of Γ contains at most $k + 4$ reflex angles. Thus in an orthogonal drawing of G with reflex face complexity k, each face of G is drawn as an orthogonal polygon with at most $2k + 4$ sides; see Figs. 1(a)-(c).

From technical drawings and wiring schematics to transportation network layouts, orthogonal drawing (or layout) is one of the most common techniques for visualizing planar graphs [1] and is also a popular visualization technique provided by most network layout systems (e.g., yEd [22], graphviz [8], and OGDF [5]). Early work on orthogonal layouts was done by Valiant [21] and Leiserson [15] in the context of VLSI design. The input graphs are assumed to be planar and with maximum-degree four, although models incorporating higher degree graphs were introduced later by Tamassia [20] and Fößmeier and Kaufmann [9].

Optimization Goals and Challenges. The number of reflex corners per face and the number of bends per edge are two important aesthetic criteria in an orthogonal drawing, and a good drawing usually minimizes these two parameters. Minimizing the total
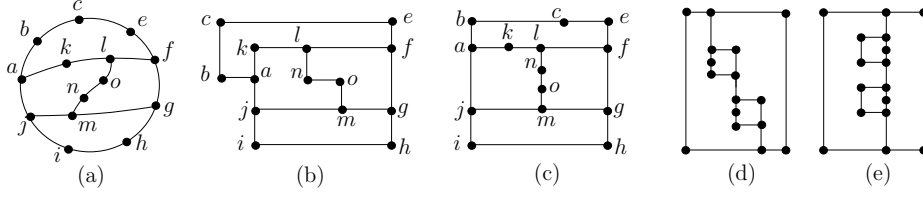
Fig. 1. (a) A plane graph G. (b) A strict-orthogonal drawing of G with reflex face complexity 1. (c) A rectangular drawing of G. (d)–(e) Two strict-orthogonal drawings (0-bend drawings) of the same graph with different reflex face complexities.

number of bends over all possible embeddings of the input planar graph is NP-hard [10]. However, Tamassia [20] introduced a maximum-flow based technique to solve the problem for maximum-degree-4 plane graphs, which takes $O(n^{7/4}\sqrt{\log n})$-time. Later, Cornelsen and Karrenbauer [6] proposed a variation of this maximum-flow based approach that improves the running time to $O(n^{3/2})$.

Note that minimization of the number of total bends, or the number of bends per edge cannot bound the reflex face complexity, see Figs. 1(d)–(e), but a drawing with reflex face complexity k ensures that the number of bends per edge is at most k. Given a plane graph G with four prescribed corner vertices, Miura *et al.* [17] showed how to decide whether G admits a strict-orthogonal drawing with reflex face complexity 0 (also known as *rectangular drawings*, as shown in Fig. 1(c)), that respects the given corners. He reduced the problem to the problem of finding a perfect matching in some graph, which leads to an $O(n^{1.5}/\log n)$-time algorithm. A variation of Tamassia's [20] flow based approach can solve this problem in $O(n \log^2 n)$ time even when the corners are not given in the input (see Appendix A).

An intriguing question in this context is whether one can adapt the maximum-flow based approach [6, 20] or Miura *et al.*'s [17] technique to decide orthogonal drawability with reflex face complexity k in polynomial time, for any nonnegative integer k. While generalizing Miura's technique does not seem simple, careful modifications of the maximum-flow based approach can solve this drawing problem (see Appendix A). The challenge here is that for $k \geq 1$, these modifications reduce the drawing problem to the problem of finding a maximum flow in some nonplanar network with $O(n)$ vertices and edges, and hence takes $O(n^2)$ time [18]. Therefore, it is natural to seek for a faster algorithm to meet the practical needs.

Our Contributions: We study the problem of orthogonal drawing of a biconnected planar graph with a given reflex face complexity k. Note that since every vertex in an orthogonal drawing has degree ≤ 4, we consider only max-degree-4 graphs in this paper. In the fixed embedding setting, we give two different polynomial-time algorithms to compute strict-orthogonal drawings of a biconnected plane graph G with any given face complexity k (if such drawings exist). Furthermore, given the nonnegative integers $k_0, k_1, \ldots, k_r$ for the faces $f_0, f_1, \ldots, f_r$ of G, both our algorithms can compute strict-orthogonal drawings of G, with at most k_i reflex corners in each face f_i,

$i \in \{0, 1, \ldots, r\}$. For example, one can specify $k_i = k$ for each inner face f_i, and $k_0 = 4$ for the outer face f_0 to compute a complexity-k tessellation of a rectangle.

We reduce this drawing problem to two classic graph optimization problems: finding a maximum flow and finding a perfect matching. Although perfect matching problems on bipartite graphs can be solved via maximum flow [16], the two techniques we present here are structurally different, and do not use this relationship. Based on the best known time-complexities for these problems, our matching-based algorithm runs in $O((nk)^{1.5})$ time and the flow-based algorithm runs in $O(n^{1.5} \log n \log k)$ time, where k is the maximum over all k_i's. Both our algorithms can be extended to compute general (non-strict) orthogonal drawings as well as orthogonal drawings with at most t_i bends on each edge e_i, for some nonnegative integer t_i.

Finally, we show that if the embedding of the planar graph G is not given, deciding whether G has a strict-orthogonal drawing with a given reflex face complexity k is NP-complete, even when $k = 4$.

2 Strict-Orthogonal Drawing Algorithms for Plane Graphs

We begin with a preliminary result showing that to compute a strict-orthogonal drawing it suffices to specify the angles between pairs of consecutive edges around each vertex. We then describe our two algorithms, proving the following main theorem:

Theorem 1. *Let G be an n-vertex biconnected plane graph with the faces $f_0, \ldots, f_r$. Given the nonnegative integers $k_0, \ldots, k_r$ with $k = \max_i\{k_i\}$, one can decide in $O(n^{1.5} \min\{k^{1.5}, \log n \log k\})$ time whether G has a strict-orthogonal drawing, where each face f_i has at most k_i reflex corners, and construct such a drawing if it exists.*

Orthogonal Drawing using Angle Assignment. Tamassia [20] showed that an orthogonal drawing Γ of a biconnected plane graph G can be described by augmenting the embedding of G with the angles at the bends (*bend angles*) and the angles between pairs of consecutive edges around the vertices of G (*vertex angles*). For strict-orthogonal drawings (no bends), we only consider vertex angles. Specifically, an *angle assignment* is a mapping from the set $\{\pi/2, \pi, 3\pi/2\}$ to the angles of G, where each angle is assigned exactly one value. Although an angle assignment of G does not specify edge lengths, it can precisely describe the shape of Γ. Given an angle assignment Φ, one can test if Φ corresponds to a strict-orthogonal drawing by Lemma 1, which is implied from [20]:

Lemma 1. *An angle assignment Φ for a plane graph G corresponds to a strict-orthogonal drawing of G if and only if Φ satisfies the following conditions $(P_1$–$P_2)$:*

(P_1) *The sum of the assigned angles around each vertex v in G is 2π.*
(P_2) *the total assigned angle of every inner (respectively, outer) face f is $(\gamma - 2)\pi$ (respectively, $(\gamma + 2)\pi$), where γ is the number of vertices on the boundary of f.*

Given an angle assignment Φ satisfying $(P_1$–$P_2)$, one can obtain a strict-orthogonal drawing of G (i.e., the exact coordinates for the vertices) in linear time.
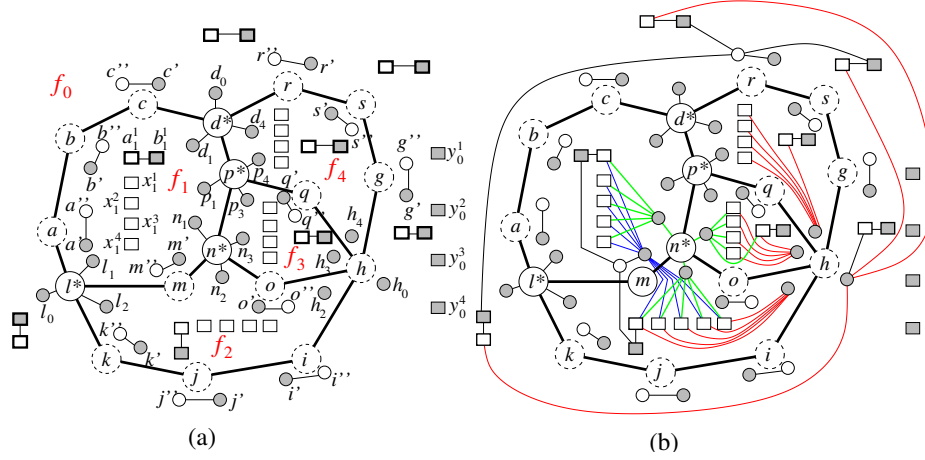
Fig. 2. (a) A plane graph G (induced by the bold edges), and the construction of $B(G)$ with $k_0 = 4$, $k_1 = k_2 = k_3 = k_4 = 1$, where only a few edges of $B(G)$ are shown. (b) The remaining edges in $B(G)$: the edges shown are the ones incident to the convex boundary vertices for a degree-4 (red), a degree-3 (green), a degree-2 (blue) vertices and the ones incident to reflex boundary vertices for two degree-2 vertices (black).

2.1 Bipartite Graph Matching Formulation

Here we prove Theorem 1 by reducing the drawing problem to the problem of finding a perfect matching in a bipartite graph. We construct a bipartite graph $B(G)$ so that one can compute a strict-orthogonal drawing of G with reflex face complexity k from a perfect matching of $B(G)$, and vice versa. Although our result generalizes the rectangular drawing algorithm by Miura *et al.* [17], the bipartite graph we construct is quite different from the one in [17] and it gives the option of having reflex corners in a face.

Let f_0 be the outer face and $f_1, \ldots, f_r$ be the inner faces of G; see Fig. 2(a). For each inner face f_i, $i \in \{1, \ldots, r\}$ of G we have four vertices x_i^1, x_i^2, x_i^3, x_i^4 in $B(G)$. These vertices will correspond to four $\pi/2$ angles in f_i. We also have k_i pairs of vertices $a_i^1, b_i^1, \ldots, a_i^{k_i}, b_i^{k_i}$ associated with f_i, as shown with white and gray squares with bold boundaries. For each $j \in \{1, \ldots, k_i\}$, there is an edge (a_i^j, b_i^j). Later, every a-vertex will correspond to a $\pi/2$ angle, and every b-vertex will correspond to a $3\pi/2$ angle in f_i. In each internal face f_i, there are only k_i pairs of a and b-vertices, which will bound the number of reflex corners of f_i in the final drawing. Observe that by Condition (P_2) of Lemma 1, each internal face of G has exactly four $\pi/2$ angles more than its $3\pi/2$ angles, and hence we have four more white squares than gray squares. Similarly, the outer face f_0 must contain four $3\pi/2$ angles more than its $\pi/2$ angles. Thus for the face f_0, we have four vertices y_0^1, y_0^2, y_0^3 and y_0^4 representing $3\pi/2$ angles, and $p = k_0 - 4$ pairs of vertices $a_0^1, b_0^1, \ldots, a_0^p, b_0^p$. Call the x- and the a-vertices the *convex face-vertices* and the y- and b-vertices the *reflex face-vertices*.

In addition to the face-vertices above, $B(G)$ also has vertex-vertices that correspond to the vertices of G. For each degree-4 vertex v in G, let f_i, f_j, f_k, f_l be the four faces

4

incident to v. For each $h \in \{i, j, k, l\}$, $B(G)$ has a vertex v_h, which is adjacent to all the convex face-vertices associated with f_h; see vertex h in Fig. 2(a). We refer to these vertices as *convex boundary-vertices*. Each of these convex boundary-vertices will choose a convex face-vertex ensuring four $\pi/2$ angles around v. For each degree-3 vertex v incident to the faces f_i, f_j, f_k, $B(G)$ has three vertices v_i, v_j, v_k, which are adjacent to all the face-vertices of their corresponding faces. We also have an additional vertex v^* in $B(G)$, which is a common neighbor for v_i, v_j, v_k; see vertex n^* in Fig. 2(a). Again we refer to these vertices v_i, v_j, v_k as *convex boundary-vertices*, and the vertex v^* as the *central-vertex*. Intuitively, v^* will match with one of its incident vertices leaving two vertices among $\{v_i, v_j, v_k\}$, which will choose two $\pi/2$ angles around v. Finally, if v is a degree-2 vertex incident to the faces f_i and f_j, then we have two vertices v' and v'' in $B(G)$ that are adjacent to each other. We call v' a *convex boundary-vertex* (shown as gray circle), and v'' a *reflex boundary-vertex* (shown as white circle). The vertex v' is adjacent to all the convex face-vertices associated with f_i and f_j, and the vertex v'' is adjacent to all the reflex vertices associated with f_i and f_j; see vertex m in Fig. 2(a). Note that degree-3 and degree-4 vertices of G do not have any associated reflex boundary-vertices in $B(G)$, since they cannot induce $3\pi/2$ angles in an orthogonal drawing; see Lemma 1, Condition (P_1).

This completes the construction of $B(G)$. It is bipartite, as shown in gray and white in Figs. 2(a)–(b)). We have the following lemma, with the proof details in Appendix A.

Lemma 2. *There is a perfect matching in $B(G)$ if and only if G has a strict-orthogonal drawing, where each face f_i contains at most k_i reflex corners.*

Proof Sketch: If $B(G)$ has a perfect matching M, see Figs. 3(a)–(b), then we compute an angle assignment Φ for G that satisfies Conditions $(P_1–P_2)$ of Lemma 1. For any face f_i of G, assign the angle inside f_i (at some vertex v) the value $\pi/2$ (resp. $3\pi/2$) if the corresponding boundary-vertex in $B(G)$ is matched to some convex (resp. reflex) face-vertex of f_i. Otherwise, the boundary-vertex is matched with some central-vertex, or another boundary vertex. In both cases assign the angle the value π. If the above rule leads to a conflict at some degree-2 vertex, i.e., when it has both convex and reflex boundary-vertices matched to face-vertices of the same face (see vertex q in Fig. 3(b)), we again assign the angle at v a value of π (inside that face). Since M is a perfect matching, the construction of $B(G)$ implies that each inner (resp. the outer) face has exactly four more (resp. fewer) $\pi/2$ angles than $3\pi/2$ angles. Consider now a vertex v of G. If $\deg(v) = 4$, it has exactly four $\pi/2$ angles by the matching of its four convex boundary-vertices. If $\deg(v) = 3$, it has two $\pi/2$ angles and one π angle. For $\deg(v) = 2$, it has either two π angles or one $\pi/2$ and one $3\pi/2$ angles. By Lemma 1, this angle assignment gives a desired orthogonal drawing of G; see Fig. 3(c).

Conversely, if G has a strict-orthogonal drawing Γ, where each face f_i has at most k_i reflex corners, then Γ gives a perfect matching M in G. Inside each face f_i of G, for each $\pi/2$ (resp. $3\pi/2$) angle, match the corresponding boundary-vertex to a convex (resp. reflex) face-vertex of f_i. It is straightforward to match face-vertices with boundary vertices such that the unmatched face vertices remain in pairs and later we take the edges between the unmatched pairs in M. For each degree-2 vertex with two π angles, we take the edge between its boundary-vertices in M. Finally, for each degree-3 vertex v, we match the boundary vertex corresponding to the π angle of v with v^*. □
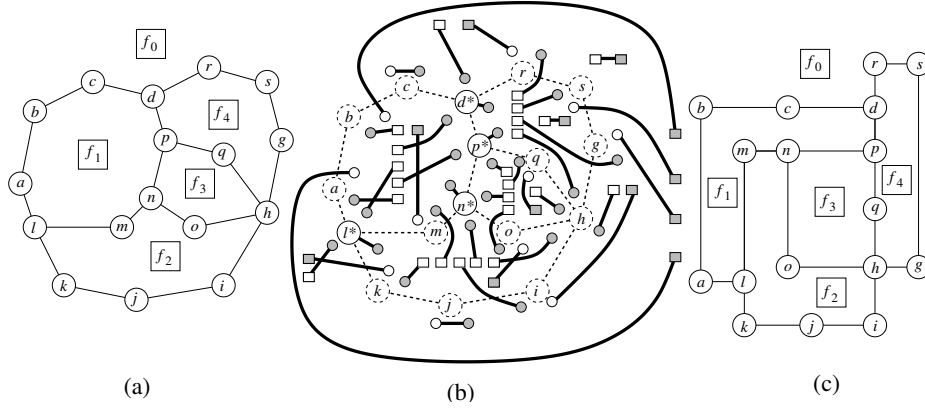
Fig. 3. (a) A biconnected plane graph G with maximum degree four, (b) a perfect matching in $B(G)$, and (c) a strict-orthogonal drawing of G with $k_0 = 4$ and $k_1 = k_2 = k_3 = k_4 = 1$.

The number of vertices $|V|$ in $B(G)$ is $O(nk)$, where $k = \max_i\{k_i\}$. Since there are $O(n)$ boundary-vertices and for each of the $O(n)$ faces, there are $O(k)$ face-vertices, the number of edges $|E|$ in $B(G)$ is again $O(nk)$. Hence the existence of a perfect matching in $B(G)$ can be tested in $O(\sqrt{|V|}|E|) = O(\sqrt{nk} \times nk) = O((nk)^{1.5})$ time using the Hopcroft-Karp algorithm [12].

2.2 Maximum-Network-Flow Formulation

Here we use a maximum-network-flow algorithm to compute strict-orthogonal drawings of a biconnected plane graph G, with at most k_i reflex corners in each face f_i.

Note that network-flow models have been used before in the context of orthogonal drawings [6, 20]. While these network-flow models are also based on the concept of angle assignment, our network-flow model is of independent interest because it is planar for arbitrary choice of k, and thus gives a faster solution to the problem. We refer the reader to Appendix A, where we show how to modify previous network-flow formulations to solve the problem of strict-orthogonal drawings with bounded number of reflex corners for the faces. However, for $k \geq 1$, the modified networks are no longer planar and hence solving the problem takes $O(n^2)$ time [18].

Here is an outline of our algorithm. Given a plane graph G, we construct a flow-network H, where the vertices of H are partitioned into two sets: the *boundary-vertices*, V_R, which corresponds to the vertices of G and the *face-vertices*, V_F, which corresponds to the faces of G. Any edge of H that connects a boundary-vertex with a face-vertex corresponds to a vertex-face incidence of G. The vertices of degree more than two in V_R correspond to the sources, and a set of vertices $U_F \subset V_F$ corresponds to the sinks of H. Roughly speaking, the incoming flow to vertices of $(V_F \setminus U_F)$ determines the convex corners, while the outgoing flow determines the reflex corners. We set the edge capacities so that each sink consumes at most four units of flow, as per Condition (P_2) of Lemma 1. The flow constraints also ensure the desired number of reflex corners in each face of the drawing implied by the maximum flow.

6

We now describe the algorithm in detail. Given a plane graph $G = (V, E)$, let F be the faces in G. The graph H is constructed by the following steps; see Figs. 4(a)–(d).

Vertices: For each face f of G, there are two vertices v'_f and v''_f in H. Call v'_f a convex face-vertex and v''_f a reflex face-vertex. Thus $V'_F = \{v'_f | f \in F\}$ and $V''_F = \{v''_f | f \in F\}$ are the sets of all convex and reflex face-vertices, respectively, and are shown in gray-blue and lightgray-red vertices; see Fig. 4. For each vertex v of G with degree three or four, add v as a vertex of H. All these vertices are convex boundary-vertices. For each vertex v of degree two, add two vertices v and v^* to H, where v is a convex boundary-vertex and v^* is a reflex boundary-vertex. The reflex boundary-vertices are denoted by black squares in Fig. 4.

Edges: For each face f in G and for each vertex v on f in G, add an edge from the corresponding convex boundary-vertex in H to the convex face-vertex v'_f; see Fig. 4(b). If v has degree 2, then also add an edge e from the reflex face-vertex v''_f to the reflex boundary-vertex v^*; see Fig. 4(d). Set the capacity upper bound $c_e = 1$ for e. For each face f_i of G, add an edge e from the convex face-vertex v'_{f_i} to the reflex face-vertex v''_{f_i} with $c_e = k_i$; see Fig. 4(c). For each vertex v of degree-2, add an edge from the reflex boundary-vertex v^* to the convex boundary-vertex v with $c_e = 1$; see Fig. 4(c).

Sources and sinks: All the convex boundary-vertices corresponding to the degree-3 and degree-4 vertices of G are sources of H. Each degree-3 vertex has a production of 2 units of flow, and each degree-4 vertex has a production of 4 units of flow. For each inner face f of G, there is a sink (unfilled-green vertices in Fig 4) with an incoming edge e from the convex face-vertex v'_f, where $c_e = 4$; see Fig. 4(c). Finally, there is a source on the outer face f_0 with an outgoing edge e to the convex face-vertex v'_{f_0} with $c_e = 4$. We set production of this source to be four units of flow.

This completes the construction. Before we argue correctness, define a maximum flow in H to be *saturated* if it consumes the productions of all the sources of H, as well as saturates the incoming edges to all the sinks of H. We now show that finding an integral maximum flow in H is equivalent to computing a desired strict-orthogonal drawing of G. We have the following lemma with the proof details in Appendix A.

Lemma 3. *There is a strict-orthogonal drawing of G where each face f_i contains at most k_i reflex corners if and only if the integral maximum flow in H is saturated.*

Proof Sketch: Assume that the maximum flow of H is saturated. We then find an angle assignment for H that corresponds to a desired strict-orthogonal drawing of G. The edges from the convex boundary-vertices to the convex face-vertices carry at most one unit of flow. A non-zero (resp., zero) flow on such an edge corresponds to an angle of $\pi/2$ (resp., π) at the corresponding angle. Similarly the edges from the reflex face-vertices to the reflex boundary-vertices also carry at most one unit of flow. A non-zero (resp., zero) flow on such an edge corresponds to an angle of $3\pi/2$ (resp., π) at the corresponding angle. An exception of the above two rules is the case when there is a degree-2 vertex v and a face f incident to v in G so that the edges (v, v'_f) and (v''_f, v^*) both carry one unit of flow; see the flow through o^* in Figs. 4(e)–(h). In this scenario, assign π to the corresponding angle. In the detailed proof we show that this angle assignment corresponds to a strict-orthogonal drawing of G, in particular, the two properties of Lemma 1 hold; see Figs. 4(e)–(h). $\qquad\square$
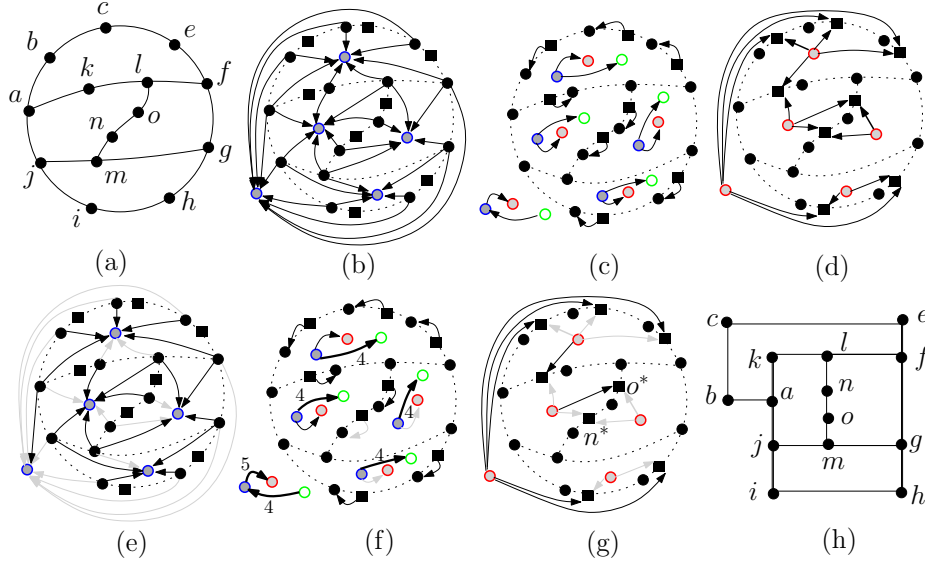
Fig. 4. (a) A plane graph G, (b)–(d) construction of the flow-network H from G, the edges shown in three partitions, (e)–(g) a saturated integral flow on H, where each thin edge in black (resp. gray) carries one (resp. zero) unit of flow, (h) a corresponding strict-orthogonal drawing of G.

Theorem 1 is a direct consequence of Lemma 3. To compute the maximum flow we use the $O(\min(|V|^{2/3}, |E|^{1/2})|E| \log(|V|^2/|E|) \log U)$-time algorithm of Gold-berg and Rao [11]. Since H has $O(n)$ edges and each edge has $O(k)$ capacity upper bound U, the running time is $O(n^{1.5} \log n \log k)$.

For the case when $k = 0$, we can delete the reflex face-vertices (except the one in the outer face) to make H planar. Then the maximum-flow problem for a multiple-source and multiple-sink directed planar graph can be solved in $O(n \log^3 n)$ time [4]. However, here the productions and demands for the vertices are known. Thus we need to solve only a feasible flow problem, which can be computed in $O(n \log^2 n)$ time [14].

Corollary 1. *Given a plane graph G with n vertices, one can determine in $O(n \log^2 n)$ time whether G admits a rectangular drawing, and construct such a drawing if it exists.*

2.3 General Orthogonal Drawing with a Given Face-Complexity

Here we extend our algorithms to general (non-strict) orthogonal drawing. Each bend in an orthogonal drawing can be thought of as a degree-2 vertex on some edge in the graph (e.g., a subdivision of an edge). We have the following lemma, whose proof is in Appendix B.

Lemma 4. *Let G be a biconnected plane graph with edges e_1, ..., e_m and faces f_0, f_1, ..., f_r. Consider the set of non-negative integers t_1, ..., t_m and k_0, k_1, ..., k_r. Let G_t be a graph obtained from G by subdividing each edge e_i exactly t_i times. Then G*

has an orthogonal drawing, where each edge e_i has at most t_i bends and each face f_i has at most k_i reflex corners if and only if G_t has a strict-orthogonal drawing where each face f_i has at most k_i reflex corners.

It is straightforward use Lemma 4 to find a polynomial-time algorithm for orthogonal drawing that simultaneously bounds the reflex face complexity and the number of bends per edge. Note that in this paper, our main focus is on bounding the reflex face complexity; thus we leave the task of designing fast algorithms optimizing multiple objectives as a future work. There exists specialized algorithms for bounding the number of bends per edge or for optimizing any convex cost associated with the edges of the input graph, even in the variable embedding setting for some specific cost functions [3, 2]. Adapting these algorithms to bound the reflex-face complexity yields a nonplanar flow network similar to the one described in Appendix A, and hence runs slower than the algorithm corresponding to Theorem 1. Modifying these algorithms in a way that achieve a time complexity better than Theorem 1 would be a compelling research direction.

Observe that we can also assign positive costs to the edges of the flow-network, and then use min-cost max-flow to minimize the number of bends while computing a drawing with a given reflex face complexity. Specifically, if we assign unit cost only to the edges, incoming to the division vertices, then the cost of the maximum flow will directly correspond to the total number of bends in the drawing. This provides an alternative of Tamassia's technique [20] for minimizing the total number of bends in an orthogonal drawing.

Theorem 2. *Given an n-vertex biconnected plane graph G and a positive integer k, we can decide in polynomial time whether G admits an orthogonal drawing with reflex face complexity k, and if such a drawing exists, then we can construct the drawing minimizing the total number of bends in polynomial time.*

3 NP-Hardness for Planar Graphs

In this section we prove that it is NP-complete to decide whether a planar biconnected graph admits a strict-orthogonal drawing with a given reflex face complexity k, even when $k = 4$. Throughout this section we denote this problem by MIN-REFLEX-DRAW. The NP-hardness proof for deciding strict-orthogonal drawability [10] implies that it is NP-hard to determine drawability with k reflex face complexity, but this proof does not hold if we restrict k to be a constant. On the other hand, our NP-hardness proof holds when $k = 4$, even when it is known that the input graph has a strict-orthogonal drawing.

We prove the NP-completeness with a reduction from the rectilinear monotone planar 3-SAT problem (RMP3SAT), which is NP-hard [7]. The input of an RMP3SAT instance I is a collection C of clauses over a set U of variables such that each clause contains at most three variables, and each clause is either positive or negative (i.e., all its variables are either positive or negative). Moreover, the corresponding *SAT-graph G_I* (i.e., a bipartite graph with vertex set $C \cup U$ and edge set $\{(x,y)|x \in C, y \in U, y \in x\}$) admits a planar drawing Γ satisfying the following property: Each vertex in Γ is drawn as an axis-aligned rectangle. All the vertices representing variables lie along a horizontal

line h (known as *backbone*). The vertices representing positive (respectively, negative) clauses lie on the top (respectively, bottom) half-plane of h. Each edge is drawn as a vertical line segment that is incident to the drawings of its end vertices. The RMP3SAT problem asks to decide whether there is a satisfying truth assignment for U satisfying all clauses in C. RMP3SAT remains NP-hard even when each variable appears in at most four clauses [13].

Given an instance $I = (U, C)$ of RMP3SAT, where each variable appears in at least two and at most 4 clauses, we construct a planar graph H so that H has a strict-orthogonal drawing with face complexity 4, if and only if the RMP3SAT instance is satisfiable. details.

We construct H from the drawing Γ of the SAT-graph G_I; see Fig. 5(a). We first draw a polygon with holes (shown in gray) that represents each edge of Γ as a tunnel; see Fig. 5(b). We then place the drawing onto a regular grid H, where each hole is a collection of grid cells, as shown in Fig. 5(c). Then for each variable and clause, we assign a corresponding variable cell and a corresponding clause cell in H. Fig. 5(c) depicts the cells corresponding to the variable x_1 and clauses c_1, c_2, c_4, respectively. In each variable cell, we create a variable-staircase structure of length three (see Fig. 5(f)) such that the base of the staircase is adjacent to the bottom side of the cell. Note that this staircase contributes to four reflex corners in the variable cell, which can be transferred to the cell lying below the variable cell by flipping the staircase vertically. For each edge connecting a variable to a clause, we first find a sequence of cells connecting the variable cell to the clause cell, and then add a staircase of length two and a 4×4 grid structure (see Fig. 5(f)) to each of these cells, as follows:

The staircase is added at a corner of the cell that cannot be flipped and contributes to two reflex corner to the cell; we do not show these staircases in the schematic representations of Fig. 5. The grid is added to one side of the cell such that it contributes to two reflex corners to this cell, which can be transferred to the cell adjacent to it by flipping. Since $k = 4$, none of the cells on the path from the variable to the clause cell can contain more than one grid structure. The grid structures are added exploiting this constraint along the variable to clause path such that the placement of the variable-staircase in the variable cell forces a grid structure to fall into the clause cell if the clause is positive. On the other hand, the placement of the variable-staircase outside of the variable cell forces a grid structure to fall into the clause cell if the clause is negative. Finally, for each clause c, we add a staircase of length $(6 - 2|c|)$ at the corner of its clause cell, where $|c|$ is the number of variables in c. Such a clause-staircase ensures that at least one of the grid structures incident to the clause cell must lie outside of the clause cell; we do not show these staircases in the schematic representations of Fig. 5.

Let the resulting drawing be Γ'. It is straightforward to carry out the above construction in polynomial time, and one can observe that any strict-orthogonal drawing must respect the axis-alignments of the edges of the underlying graph (up to rotation or reflection).

Theorem 3. *It is NP-complete to decide if a planar graph admits a strict-orthogonal drawing with face complexity* 4.

Proof. Given a drawing of H with reflex face complexity 4, we use the above/ below orientations of a variable staircase to find the truth value of the corresponding variable;
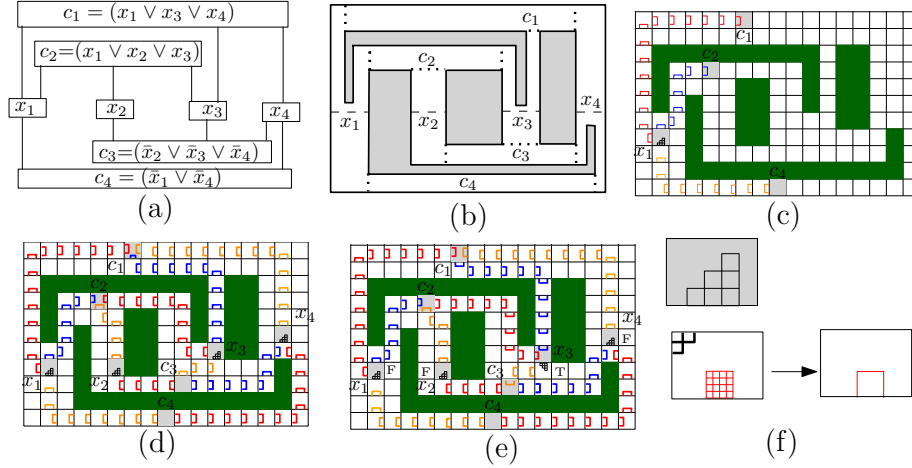
Fig. 5. (a) G_I, (b) Γ', (c)–(d) illustration of the reduction, where the variable and clause cells are shaded, (e) computing truth assignment: $x_1 = x_2 = x_4$ =false, x_3=true, (f) insertion of a staircase and a grid.

see Fig. 5(e). The variable cell that receives a variable staircase obtains at least 4 reflex corners, and hence cannot have any grid structure inside it. Moreover, by construction, no clause cell can have all its adjacent grid structures inside it, otherwise it would have at least $(6 - 2|c|) + 2|c| > 4$ reflex corners. Consequently, every clause cell must have one of its clause-staircases outside of the face, which implies that each clause must be satisfied. On the other hand, given a satisfying truth assignment for I, we orient the variable-staircases above/below depending on whether it is false/true. The placement for the grid structures is then straightforward.

Given a drawing Γ_H of H, it is straightforward to decide in polynomial time if Γ is a strict-orthogonal drawing with reflex face complexity 4. Thus MIN-REFLEX-DRAW is also in NP. □

4 Conclusion

We have proved that one can decide whether a planar graph G admits a strict orthogonal drawing with a given reflex face complexity k, for any given nonnegative integer k, in $O(n^{1.5} \min\{k^{1.5}, \log n \log k\})$ time. Finding faster algorithms for this problem would be a natural direction for future research. We also showed that in the variable-embedding setting the problem of deciding whether a biconnected planar graph admits a strict-orthogonal drawing with a given reflex face complexity 4 is NP-complete. It would be worthwhile to consider the complexity of the problem for smaller values of k.

References

1. G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. The MIT Press, 3rd edition, 2009.

2. T. Bläsius, M. Krug, I. Rutter, and D. Wagner. Orthogonal graph drawing with flexibility constraints. *Algorithmica*, 68(4):859–885, 2014.

3. T. Bläsius, I. Rutter, and D. Wagner. Optimal orthogonal graph drawing with convex bend costs. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 7965 of *LNCS*, pages 184–195. Springer, 2013.

4. G. Borradaile, P. N. Klein, S. Mozes, Y. Nussbaum, and C. Wulff-Nilsen. Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. In *Symposium on Foundations of Computer Science (FOCS)*, pages 170–179, 2011.

5. M. Chimani, C. Gutwenger, M. Jünger, G. Klau, K. Klein, and P. Mutzel. The open graph drawing framework. In *Handbook of Graph Drawing and Visualization*, pages 543–571. 2013.

6. S. Cornelsen and A. Karrenbauer. Acclerated bend minimization. *Journal of Graph Algorithms and Applications*, 16(3):635–650, 2012.

7. M. de Berg and A. Khosravi. Optimal binary space partitions for segments in the plane. *International Journal of Computational Geometry and Applications*, 22(3):187–206, 2012.

8. J. Ellson, E. R. Gansner, E. Koutsofios, S. C. North, and G. Woodhull. Graphviz - open source graph drawing tools. In *Symposium on Graph Drawing (GD)*, pages 483–484, 2001.

9. U. Fößmeier and M. Kaufmann. Drawing high degree graphs with low bend numbers. In *Symposium on Graph Drawing (GD)*, pages 254–266, 1995.

10. A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM Journal on Computing*, 31(2):601–625, 2001.

11. A. V. Goldberg and S. Rao. Beyond the flow decomposition barrier. *Journal of the ACM*, 45(5):783–797, 1998.

12. J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

13. D. Kempe. On the complexity of the "reflections" game, 2003. http://www-bcf.usc.edu/ dkempe/publications/reflections.pdf.

14. P. N. Klein, S. Mozes, and O. Weimann. Shortest paths in directed planar graphs with negative lengths: A linear-space $o(n \log^2 n)$-time algorithm. *ACM Transactions on Algorithms*, 6(2):236–245, 2010.

15. C. E. Leiserson. Area-efficient graph layouts (for VLSI). In *Symposium on Foundations of Computer Science (FOCS)*, pages 270–281, 1980.

16. C. E. Leiserson, T. H. Cormen, C. Stein, and R. Rivest. *Introduction To Algorithms*. Prentice Hall, 1999.

17. K. Miura, H. Haga, and T. Nishizeki. Inner rectangular drawings of plane graphs. *International Journal on Computational Geometry and Applications*, 16(2–3):249–270, 2006.

18. J. B. Orlin. Max flows in $o(nm)$ time, or better. In *Symposium on Theory of Computing Conference (STOC)*, pages 765–774. ACM, 2013.

19. M. S. Rahman, N. Egi, and T. Nishizeki. No-bend orthogonal drawings of subdivisions of planar triconnected cubic graphs. *IEICE Transactions*, 88-D(1):23–30, 2005.

20. R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM Journal on Computing*, 16(3):421–444, 1987.

21. L. G. Valiant. Universality considerations in VLSI circuits. *IEEE Transaction on Computers*, 30(2):135–140, 1981.

22. R. Wiese, M. Eiglsperger, and M. Kaufmann. yFiles visualization and automatic layout of graphs. In *Graph Drawing Software*, pages 173–191. Springer, 2004.

Appendix A

Proof of Lemma 2: Assume that $B(G)$ has a perfect matching M; see Figs. 3(a)–(b). From this matching, we compute an angle assignment Φ for G from the set $\{\pi/2, \pi, 3\pi/2\}$ so that Φ satisfies Conditions (P_1–P_2) of Lemma 1.

Consider an arbitrary face f_i of G. We assign an angle inside f_i (at some vertex v) the value $\pi/2$ if the corresponding boundary-vertex in $B(G)$ is matched to some convex face-vertex of f_i. For example, the convex boundary-vertices associated with the vertices b and h in Fig. 3(b) are determining $\pi/2$ angles around b and h in Fig. 3(c). Similarly, a $3\pi/2$ angle is assigned to v when its corresponding boundary-vertex in $B(G)$ is matched with a reflex face-vertex for f_i, e.g., see vertex m in Fig. 3(b). Otherwise, the boundary-vertex is either matched with some central-vertex, or another boundary vertex (e.g., see vertex c). In both cases we assign the corresponding angle the value π.

Note that the above rules may lead to a conflict at some degree-2 vertex, when it has both convex and reflex boundary-vertices matched to the convex and reflex face-vertices of the same face. For example, the vertex q in Fig. 3(b) has its boundary vertices matched with the face-vertices in the same face f_3. In such a case we assign the angle at v a value of π (inside the corresponding face). Since M is a perfect matching, the construction of $B(G)$ implies that each inner face has exactly four more $\pi/2$ angles than $3\pi/2$ angles. Similarly, the outer face f_0 contains exactly four more $3\pi/2$ angles than $\pi/2$ angles. Thus Condition (P_2) of Lemma 1 is satisfied for each face of G.

Consider now the assignment of angles around each vertex v of G. If $\deg(v) = 4$, then all its four convex boundary-vertices are matched to some convex face-vertices, and hence it has exactly four $\pi/2$ angles. If $\deg(v) = 3$, then exactly one of its three convex boundary-vertices is matched with v^*, and hence it has two $\pi/2$ angles and one π angle. Finally, if $\deg(v) = 2$, then it either has two π angles (because v' and v'' are either matched to each other or to the face-vertices in the same face); or it receives exactly one $\pi/2$ angle and exactly one $3\pi/2$ angle. Thus the sum of angles around each vertex is 2π, satisfying Condition (P_1) of Lemma 1. By Lemma 1, this angle assignment gives an orthogonal drawing of G. Since each face f_i can have at most k_i reflex boundary-vertices matched to its k_i reflex face-vertices, the number of reflex corners in the drawing of f_i is at most k_i; see Fig. 3(c).

Conversely, if G has a strict-orthogonal drawing Γ, where each face f_i of G has at most k_i reflex corners, then Γ gives a perfect matching M in G, as follows. For each face f_i of G, traverse around its drawing in Γ, and for each $\pi/2$ (respectively, $3\pi/2$) angle, match the corresponding boundary-vertex to a convex (respectively, reflex) face-vertex of f_i. There are always sufficiently many face-vertices, since each inner face f_i is associated with k_i pairs of convex and reflex face-vertices, and the outer face f_0 has exactly $p = k_0 - 4$ such pairs. It is straightforward to match face-vertices with boundary vertices such that the unmatched face vertices remain in pairs. Hence we can afterwards choose the edges between the unmatched pairs of face-vertices in M. For each degree-2 vertex with two π angles, we take the edge between its boundary-vertices in M. Finally, for each degree-3 vertex v, we match the boundary vertex corresponding to the π angle of v with v^*. □

Proof of Lemma 3: Assume that the maximum flow of H is saturated. We then find an angle assignment for H that corresponds to a desired strict-orthogonal drawing of

G. The edges from the convex boundary-vertices to the convex face-vertices carry at most one unit of flow. A non-zero (resp., zero) flow on such an edge corresponds to an angle of $\pi/2$ (resp., π) at the corresponding angle. Similarly the edges from the reflex face-vertices to the reflex boundary-vertices also carry at most one unit of flow. A non-zero (resp., zero) flow on such an edge corresponds to an angle of $3\pi/2$ (resp., π) at the corresponding angle. An exception of the above two rules is the case when there is a degree-2 vertex v and a face f incident to v in G so that the edges (v, v'_f) and (v''_f, v^*) both carry one unit of flow; see the flow through o^* in Figs. 4(e)–(h). In this scenario[1], assign π to the corresponding angle. We now show that this angle assignment corresponds to a strict-orthogonal drawing of G, in particular, the two properties of Lemma 1 hold; see Figs. 4(e)–(h).

Property P1: To show that the sum of assigned angles around each vertex v in G is π, we first consider the case when $\deg(v) = 4$. Since this vertex corresponds to a source s in H with production 4, each outgoing edge from s must have one unit of flow, implying four $\pi/2$ angles around v. If $\deg(v) = 3$, then the production of the corresponding source s in H is 2. Therefore, two of the outgoing edges from s will have one unit of flow, implying two $\pi/2$ angles around v. The remaining outgoing edge from s will have zero flow determining a π angle at v. Finally, if $\deg(v) = 2$, then let the two faces incident to v in G be f and f'. According to the capacity constraints, there are three possibilities: (1) The path $v''_f, v^*, v, v'_{f'}$ (or, $v''_{f'}, v^*, v, v'_f$) carries one unit of flow implying a $\pi/2$ (resp., $3\pi/2$) angle at f and a $3\pi/2$ (resp., $\pi/2$) angle at f'. (2) The path $v''_{f'}, v^*, v, v'_{f'}$ (or, v''_f, v^*, v, v'_f) carries one unit of flow, which implies a π angle at f and a π angle at f'. (3) The amount of outgoing flow from v and hence the incoming flow to v^* is zero, implying a π angle at f and a π angle at f'. In all the above three cases, the sum of assigned angles around v is π.

Property P2: Every inner (resp., outer) face contains four more $\pi/2$ (resp., $3\pi/2$) angles than $3\pi/2$ (resp., $\pi/2$) angles. Since we assumed the maximum flow is saturated, the difference between incoming flows between the convex and reflex face-vertices of f is exactly four. Then f has exactly four more convex corners than reflex corners. Similarly, for the outer face, the source ensures that the number of reflex vertices is exactly four more than the number of convex corners. Both cases imply the condition P_2 of Lemma 1.

Thus by Lemma 1, this angle assignment corresponds to a strict-orthogonal drawing of G. Furthermore, since the edge between the convex and the reflex face-vertices v'_f and v''_f in each face f_i can carry at most k_i units of flow, and since this is the only incoming edge to v''_f, the face f_i has at most k_i reflex corners.

Conversely, if G has a strict-orthogonal drawing where each face f_i has at most k_i reflex corners, then we can find a saturated integral flow in H, as follows. For every $\pi/2$ angle inside some face f, we assign one unit of flow from the corresponding convex boundary-vertex v to the convex face-vertex v'_f. Similarly, for each $3\pi/2$ angle inside f, we assign one unit of flow from the reflex face-vertex v''_f to the corresponding reflex boundary-vertex. Since each inner face has exactly four $\pi/2$ angles more than $3\pi/2$ angles, and the outer face has exactly four $3\pi/2$ angles more than $\pi/2$ angles, the incoming sink edges at each inner face and the outgoing source edges at the outer face

---

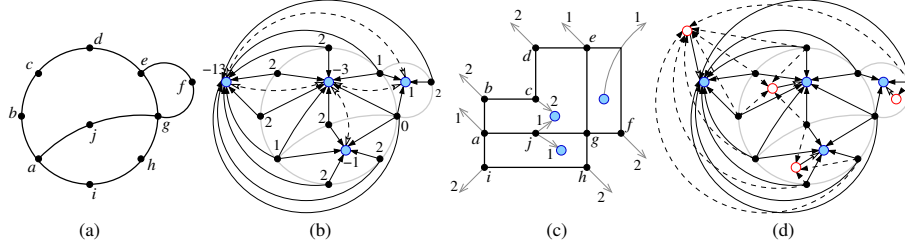[1] Alternatively, we can use a min-cost max-flow network with positive costs for edges.

Fig. 6. (a) A plane graph G, (b) construction of the flow-network H from G by Tamassia, (c) an orthogonal drawing of G and the corresponding flow, (d) modification of the network by Tamassia to solve the problem of orthogonal drawing with bounded reflex complexity for the faces.

are saturated. Since each degree-3 vertex has exactly two $\pi/2$ angles and each degree-4 vertex has exactly four $\pi/2$ angles, the production from all the sources is consumed. Finally, since the number of reflex corners at each face f_i is at most k_i, the flow on the edge between the convex and the reflex face-vertices for a face is at most the capacity upper bound k_i. Hence this flow assignment gives a saturated integral flow. □

Previous Flow-Networks: Here we briefly review the network-flow formulations by Tamassia [20] and by Cornelsen and Karrenbauer [6] for computing minimum-bend orthogonal drawings of plane graphs. We then describe how these algorithms can be modified in order to compute drawings with bounded reflex face complexities.

In Tamassia's network H there are *boundary-vertices*, V_R, and *face-vertices*, V_F; see Fig. 6(a–b). The edges of H are the bidirectional edges of the dual graph of G (dashed edges in Fig. 6(b), called *dual edges*) and the edges from each boundary-vertex to its incident face-vertices (solid edges). Each vertex v in V_R is a source with a production of $4 - deg(v)$ units; while the production or consumption of each face-vertex is either $4 - \deg(f)$ units (for inner faces) or $-4 - \deg(f)$ (for the outer face). The cost of an edge is 1 unit if it connects two face-vertices, and 0 otherwise. A min-cost max-flow in this network corresponds to an orthogonal drawing of G, as follows. A flow of $t \in \{0, 1, 2, 3\}$ units from a boundary-vertex to a face-vertex determines a $(t+1)\pi/2$ assignment to the corresponding angle in G. A flow of t units through some dual edge (dashed edge) corresponds to t bends in the corresponding edge of G; see Fig. 6(c). Using this network, Tamassia [20] gave an $O(n^2 \log n)$-time algorithm for orthogonal drawing with minimum number of bends. Cornelsen and Karrenbauer [6] used the same network but improved the running time to $O(n^{1.5})$ with a faster min-cost max-flow algorithm for this planar network.

One can modify the above network to solve the problem of orthogonal drawings with bounded reflex face complexities as follows; see Fig. 6(d). Delete the dual edges, i.e., dashed edges of H. For each face-vertex v_f in H, add a new vertex v'_f (unfilled red vertices) in H. For each edge (v_b, v_f) in H, with a degree-2 boundary vertex v_b, add the edge (v_b, v'_f). Add the edges (v'_f, v_f) and call the resulting network H'; see Fig. 6(d). Note that only degree-two vertices can contribute to $3\pi/2$ angles in the drawing. Place a capacity upper bound of 1 unit on each edge that is incident to some degree-two boundary-vertex v_b. Consequently, a $3\pi/2$ angle at v_b inside some face f corresponds

to one unit of flow from v_b to v_f and one unit of flow through v_b, v'_f, v_f. Finally, add a capacity upper bound of k_f on (v'_f, v_f), where k_f is the given reflex face complexity for f. Note that this network is no longer planar and one cannot use the primal-dual algorithm from [6] to solve the min-cost max-flow problem on this network. Furthermore, unlike the original network in [20, 6], this modified network is not "uncapacitated", as it has capacity upper bound on some edges.

Our network-flow formulation is different from the above network. For example, this network-flow formulation gives a planar network. Besides, it directly uses the geometric property that the sum of the angles inside (respectively, outside) an orthogonal face with t vertices is $2t - 4$ (respectively, $2t + 4$). In our network-flow formulation we use the property that the number of $\pi/2$ angles in an inner (respectively, outer) face is four more (respectively, less) than the number of $3\pi/2$ angles, which results in a different set of vertices, edges, edge capacities and different interpretation of the flows in the network.

Although the modified network described above is nonplanar for $k \geq 1$, for the case when $k = 0$, we can find a planar network by deleting the unfilled red vertices, i.e., v'_f, along with the incident edges. Thus the problem reduces to finding a maximum flow in a planar network with multiple sources and sinks, which can be computed in $O(n \log^2 n)$ time [14] since the productions and demands of all the vertices of the network are known.

Appendix B

Proof of Lemma 4: Assume that G has a desired orthogonal drawing. For each bend point p, subdivide the corresponding edge at p. In this way each edge e_i is subdivided at most t_i times. For each edge e_i that has not been subdivided t_i times in this process, further subdivide itso that the total number of subdivisions is exactly t_i. Then this corresponds to a strict-orthogonal drawing of G_t, where each face f_i has at most k_i reflex corners. Conversely, if Γ is a strict-orthogonal drawing of G_t, where each face f_i has at most k_i reflex corners, then a desired orthogonal drawing of G can be obtained from Γ by considering the degree two vertices (with angles $\pi/2$ and $3\pi/2$) of Γ as the bends of the corresponding edges in G. □