SIAM J. COMPUT.
Vol. 31, No. 2, pp. 601–625

ON THE COMPUTATIONAL COMPLEXITY OF UPWARD AND RECTILINEAR PLANARITY TESTING*

ASHIM GARG† AND ROBERTO TAMASSIA‡

Abstract. A directed graph is upward planar if it can be drawn in the plane such that every edge is a monotonically increasing curve in the vertical direction and no two edges cross. An undirected graph is rectilinear planar if it can be drawn in the plane such that every edge is a horizontal or vertical segment and no two edges cross. Testing upward planarity and rectilinear planarity are fundamental problems in the effective visualization of various graph and network structures. For example, upward planarity is useful for the display of order diagrams and subroutine-call graphs, while rectilinear planarity is useful for the display of circuit schematics and entity-relationship diagrams.

We show that upward planarity testing and rectilinear planarity testing are NP-complete problems. We also show that it is NP-hard to approximate the minimum number of bends in a planar orthogonal drawing of an n-vertex graph with an $O(n^{1-\epsilon})$ error for any $\epsilon > 0$.

Key words. graph drawing, planar drawing, upward drawing, rectilinear drawing, orthogonal drawing, layout, ordered set, planar graph, algorithm, computational complexity, NP-complete problem, approximation algorithm

AMS subject classifications. 05C62, 06A06, 65D18, 68Q17

PII. S0097539794277123

1. Introduction.

Graph drawing addresses the problem of constructing geometric representations of abstract graphs and networks [6, 7]. It is an emerging area of research that combines flavors of topological graph theory and computational geometry. The automatic generation of drawings of graphs has important applications in key computer technologies such as software engineering, database design, visual interfaces, and computer-aided design.

Various graphic standards have been proposed for the representation of graphs in the plane. Usually, each vertex is represented by a point and each edge (u, v) is represented by a simple open Jordan curve joining the points associated with vertices u and v. A *straight-line* drawing maps each edge into a straight-line segment. A drawing is *planar* if no two edges cross. A graph (or digraph) is planar if it admits a planar drawing. A drawing of a digraph is *upward* if every edge is monotonically nondecreasing in the y-direction. A drawing of a digraph is *planar upward* if it is planar and upward. A digraph is *upward planar* if it admits a planar upward drawing. Figure 1.1(a) shows a planar straight-line upward drawing. An *orthogonal* drawing maps each edge into a chain of horizontal and vertical segments. A *rectilinear* drawing is an orthogonal straight-line drawing, i.e., a drawing where every edge is either a horizontal or a vertical segment. A graph is *rectilinear planar* if it admits a planar rectilinear drawing. Figure 1.1(b) shows a planar rectilinear drawing.

*Received by the editors November 16, 1994; accepted for publication (in revised form) July 12, 1999; published electronically October 11, 2001. This research was supported in part by the National Science Foundation under grants CCR-9423847, CCR-9732327, and IIS-9985136 (CAREER Award), and by the Army Research Office under grant DAAH04-96-1-0013. A preliminary version of this paper was presented at *Graph Drawing* '94, Princeton, NJ, 1994.

http://www.siam.org/journals/sicomp/31-2/27712.html

†Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260 (agarg@cse.buffalo.edu). The work of this author was performed in part while he was with the Department of Computer Science at Brown University.

‡Center for Geometric Computing, Department of Computer Science, Brown University, Providence, RI 02912-1910 (rt@cs.brown.edu).

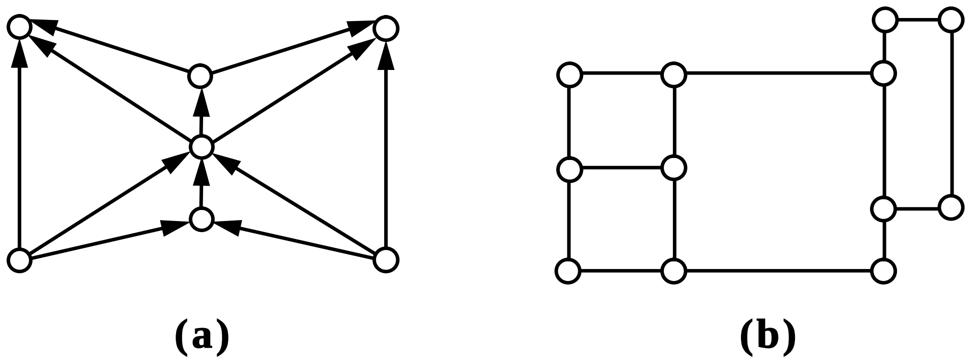

Fig. 1.1. *Examples of* (a) *a planar straight-line upward drawing of a digraph and* (b) *a planar rectilinear drawing of a graph.*

Testing upward planarity and rectilinear planarity are fundamental problems in the effective visualization of various graph and network structures. For example, upward planarity is useful for the display of order diagrams and subroutine-call graphs, while rectilinear planarity is useful for the display of circuit schematics and entity-relationship diagrams. In this paper, we show that the following two problems are NP-complete:

Upward planarity testing. Testing whether a digraph is upward planar.

Rectilinear planarity testing. Testing whether a graph is rectilinear planar.

These problems have challenged researchers in order theory, topological graph theory, computational geometry, and graph drawing for many years. Our intractability results motivate the following observations:

- Testing whether a graph admits a planar drawing or an upward drawing can be done in linear time. Combining the two properties makes the problem NP-hard.
- Every planar graph admits a planar straight-line drawing. Hence, we can say that planarity is equivalent to straight-line planarity, and both properties can be verified in linear time. We can view upward and rectilinear planarity as derived from straight-line planarity by adding further constraints, which apparently make the problem become much more difficult.

We also show that it is NP-hard to approximate the minimum number of bends in a planar orthogonal drawing of an n-vertex graph with an $O(n^{1-\epsilon})$ error for any $\epsilon > 0$.

Previous results on upward and rectilinear planarity testing are summarized below. In the rest of this section, we denote with n the number of vertices of the graph being considered.

Combinatorial results on upward planarity of covering digraphs of lattices were first given in [17, 24]. Further results on the interplay between upward planarity and ordered sets are surveyed by Rival [25, 26, 27]. Lempel, Even, and Cederbaum [18] relate the planarity of biconnected undirected graphs to the upward planarity of *st*-digraphs. A combinatorial characterization of upward planar digraphs is provided in [10, 16]; namely, a digraph is upward planar if and only if it is a spanning subgraph of a planar *st-digraph.* This characterization implies that upward planarity testing is in NP.

Di Battista, Liu, and Rival [9] show that every planar bipartite digraph is upward planar. Papakostas [23] gives a polynomial-time algorithm for upward planarity testing of outerplanar digraphs. Bertolazzi and Di Battista [2] and Bertolazzi et al. [3] give a polynomial-time algorithm for testing upward planarity of triconnected digraphs and of digraphs with a fixed embedding. Concerning single-source digraphs, Thomassen [33] characterizes upward planarity in terms of forbidden circuits. Hutton and Lubiw [14] combine Thomassen's characterization with a decomposition scheme to test upward planarity of a single-source digraph in $O(n^2)$ time. Bertolazzi et al. [4] show that upward planarity testing of a single-source digraph can be done optimally in $O(n)$ time. They also give a parallel algorithm that runs in $O(\log n)$ time on a CRCW PRAM with $n \log\log n / \log n$ processors.

Di Battista and Tamassia [10] and Di Battista, Tamassia, and Tollis [11] give algorithms for constructing upward planar drawings of planar *st*-digraphs and investigate area bounds and symmetry display. Tamassia and Vitter [32] show that the above drawing algorithms can be efficiently parallelized. Upward planar drawings of series-parallel digraphs are studied in [1].

Regarding rectilinear planarity testing, Shiloach [28] and Valiant [34] show that any planar graph of degree at most 4 admits a planar orthogonal drawing. Vijayan and Wigderson [35] study structural properties of rectilinear planar drawings. From their results, the membership of rectilinear planarity testing in NP is easy to establish. Storer [29], Tamassia and Tollis [31], Liu et al. [20], Liu, Morgana, and Simeone [21, 22], Liu, Marchioro, and Petreschi [19], Even and Granot [12], Kant [15], and Biedl and Kant [5] give various techniques for constructing planar orthogonal drawings with $O(n)$ bends. Tamassia [30] gives an $O(n^2 \log n)$-time algorithm that constructs a planar orthogonal drawing with the minimum number of bends for an embedded planar graph. Di Battista, Liotta, and Vargiu [8] give polynomial-time algorithms for minimizing bends in planar orthogonal drawings of series-parallel and cubic graphs. The latter two results show that rectilinear planarity testing can be done in polynomial time for a fixed embedding or for special classes of graphs.

Our proof techniques are based on a two-phase reduction from the known NP-complete problem NOT-ALL-EQUAL-3-SAT. In the first phase, we reduce NOT-ALL-EQUAL-3-SAT to an auxiliary undirected flow problem. In the second phase, we reduce this undirected flow problem to the upward (or rectilinear) planarity testing of a special class of digraphs. The latter reduction is interesting on its own and provides new insights on the characterization by flow networks of the angles formed by the edges of upward planar drawings [2, 3] and orthogonal drawings [8, 30].

The rest of this paper is organized as follows. Preliminary definitions and results are provided in section 2. The reduction from NOT-ALL-EQUAL-3-SAT to the auxiliary flow problem is given in section 3. Sections 4 and 5 describe the reductions from the auxiliary flow problem to upward and rectilinear planarity testing, respectively. Conclusive remarks are given in section 6.

2. Preliminaries. We assume that the reader is familiar with the standard concepts and definitions on NP-completeness [13]. Our results use reductions from the following well-known NP-complete problem:

NOT-ALL-EQUAL-3-SAT. Given a set of clauses with three literals each, is there a truth assignment such that each clause has at least one *true* literal and one *false* literal?

An *embedding* of a planar graph is the collection of circular permutations of the edges incident upon each vertex in a planar drawing of the graph. An *embedded graph* is a planar graph equipped with an embedding. We do not distinguish between a graph and its embedding if the embedding is unique and the meaning is clear from the context. A vertex has *degree* d if it has d edges incident upon it. A *degree-d* graph is one each of whose vertices has degree at most d.

We denote by $G - \{v\}$ the subgraph of graph G obtained by removing vertex v and its incident edges from G.

The *angles* of an embedded *undirected* graph are the pairs of consecutive edges incident on the same vertex. The *angles* of an embedded *directed* graph are the pairs of consecutive incoming and outgoing edges incident upon the same vertex. Such angles are mapped to geometric angles in a straight-line drawing of the graph.

A *labeled embedding* of an *undirected* graph G is an embedding of G in which each angle is assigned a label from the set $\{1, 2, 3, 4\}$. A *labeled embedding* of a *directed* graph G is an embedding of G in which each angle is assigned a label from the set $\{small, large\}$.

A *rectilinear embedding* of a graph G is a labeled embedding of G such that there exists a rectilinear drawing of G in which each angle labeled ℓ in the embedding measures $\ell\pi/2$ in the drawing. Each rectilinear embedding has a unique external face.

The following definitions are from [2, 3]. An *upward embedding* of a directed graph (digraph) $\vec{G}$ is a labeled embedding of $\vec{G}$ such that there exists a planar straight-line upward drawing of $\vec{G}$ where each angle labeled *small* has measure $< \pi$ and each angle labeled *large* has measure $> \pi$. Each upward embedding has a unique external face.

A *source* of a digraph $\vec{G}$ is a vertex with all outgoing edges, and a *sink* of $\vec{G}$ is a vertex with all incoming edges. A *switch* of $\vec{G}$ is a source or sink of $\vec{G}$. A source or sink of a face f of $\vec{G}$ is called a *local* source or sink of f. Note that a local switch of f may or may not be a switch of $\vec{G}$.

2.1. Upward and rectilinear embeddings. In this section, we give some lemmas on upward and rectilinear embeddings that will be used extensively in the proofs.

We first give some definitions. In an embedding of a digraph $\vec{G}$, a vertex is *bimodal* if its incident edges can be partitioned into two (possibly empty) sets of consecutive edges consisting of its incoming and outgoing edges, respectively. An embedding of $\vec{G}$ is *bimodal* if each vertex is bimodal.

Consider an assignment of labels from the set $\{$*small*, *large*$\}$ to the angles of an embedding of $\vec{G}$. For a face f of the embedding, let $L(f)$ and $S(f)$ be the number of angles of f with label *large* and *small*, respectively. Face f is said to be *consistently assigned* if

$$L(f) - S(f) = \begin{cases} -2 & \text{if } f \text{ is an internal face,} \\ +2 & \text{if } f \text{ is the external face.} \end{cases}$$

An assignment of labels to the angles of an embedding of digraph $\vec{G}$ is a *consistent assignment* if

- all the angles at a nonsource or nonsink vertex of $\vec{G}$ are assigned label *small*,
- exactly one angle at a source or sink vertex of $\vec{G}$ is assigned label *large*, and
- each face is consistently assigned.

We paraphrase a result of [2, 3] in the following lemma.

LEMMA 2.1. *An embedding of a digraph $\vec{G}$ can be extended to an upward embedding if and only if it is bimodal and admits a consistent assignment of labels to its angles.*

Let G be an undirected graph of degree at most 4. Consider an assignment of labels from the set $\{1, 2, 3, 4\}$ to the angles of an embedding of G. For a face f of the embedding, let $N_i(f)$ be the number of angles of f with label i. Face f is said to be *consistently rectilinearly assigned* if

$$2 \cdot N_4(f) + N_3(f) - N_1(f) = \begin{cases} -4 & \text{if } f \text{ is an internal face,} \\ +4 & \text{if } f \text{ is the external face.} \end{cases}$$

An assignment of labels to the angles of an embedding of a graph G is a *consistent rectilinear assignment* if

- the sum of the labels of the angles around each vertex is 4, and
- each face is consistently rectilinearly assigned.

The following lemma is an immediate consequence of the results in [30, 35].

LEMMA 2.2. *An embedding of a graph G can be extended to a rectilinear embedding if and only if it admits a consistent rectilinear assignment of labels to its angles.*

2.2. Tendrils and wiggles. We now define several graphs that will be used as gadgets in our reductions.

Tendril T_k, for $k \geq 0$, is an acyclic digraph with two designated poles, a *source pole* denoted by s and a *sink pole* denoted by t, and is defined recursively as follows:

- Tendril T_0 consists of a single directed edge (s, t).
- Tendril T_1 is the 10-vertex graph shown in Figure 2.1(a).
- Tendril T_k is constructed from T_{k-1} by adding the graph H_k shown in Figure 2.1(b) to it by identifying edges (and their endpoints) e_{k-1} of T_{k-1} and f_k of graph H_k (Figure 2.1(c)).

It is easy to see that T_k has exactly $k+1$ sources and $k+1$ sinks. Figure 2.2(a) and Figure 2.2(b) show tendrils T_2 and T_3, respectively.

LEMMA 2.3. *Tendril T_k is upward planar and admits a unique upward embedding.*

Proof. We use induction on k. It is straightforward to verify that T_1 has a unique upward embedding by applying Lemma 2.1 to their $O(1)$ planar embeddings. Now suppose that T_{k-1} has a unique upward embedding. Again, it can be easily verified that H_k has a unique upward embedding by applying Lemma 2.1 to its $O(1)$ planar embeddings. Hence, for finding upward embeddings of T_k, we need only to consider the four planar embeddings of T_k obtained by flipping the upward embeddings of T_{k-1} and H_k around their common edge f_k. Applying Lemma 2.1 to the four faces containing both endpoints of f_k shows that only one of these four planar embeddings can be extended to an upward embedding of T_k. ∎

In the upward planar embedding of T_k, the external face consists of two paths, namely, the *outer* and *inner* paths, between s and t. The outer path has $2k$ large angles and no small angles, and the *inner path* has $2k$ small angles and no large angles. The outer and inner paths of T_2 are drawn with shaded and solid thick lines, respectively, in Figure 2.2(a). When T_k replaces an edge of an embedded planar digraph, the outer (inner) path becomes a subpath of a face f, and we say that the *contribution* of the outer (inner) path to f is $+2k$ $(-2k)$.

Figure 2.2(b) shows a *wiggle* W_k, which consists of a chain of $2k+1$ edges whose orientations alternate along the chain. The two extreme vertices of W_k are called its *source* and *sink* poles, respectively, and are denoted by s and t, respectively. Later, in section 4, we will consider transformations where a directed edge (u, v) of an embedded digraph is replaced with W_k such that s is identified with u and t with v, and W_k

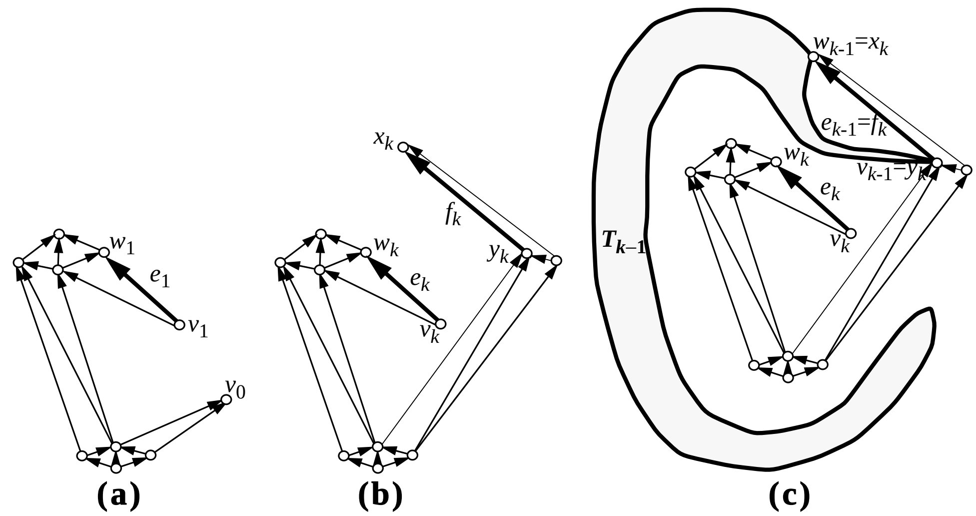

FIG. 2.1. (a) *Tendril* T_1*;* (b) *graph* H_k*;* (c) *constructing* T_k *from* T_{k-1}.

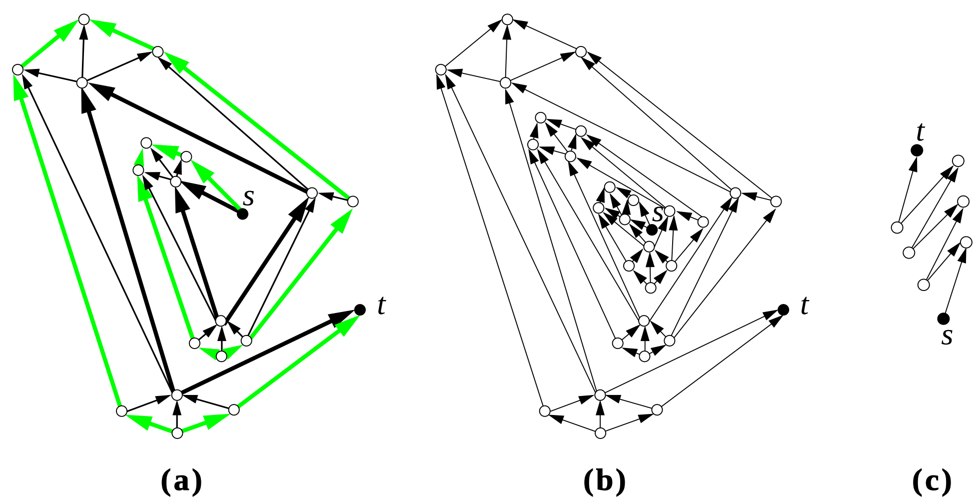

FIG. 2.2. *Examples of tendrils and wiggles:* (a) *tendril* T_2*;* (b) *tendril* T_3*;* (c) *wiggle* W_3*. The outer and inner paths of* T_2 *are drawn using shaded and solid thick lines, respectively.*

becomes a subpath of two faces f_1 and zz_2 in the new digraph. Given an upward embedding of W_k, we say that the *contribution* of W_k to face f_i, where $i = 1$ or 2, is equal to the number of large angles minus the number of small angles of W_k in f_i. Because each angle of W_k in f_i is either large or small, the contribution of W_k to f_i is an even number c, where $-2k \leq c \leq 2k$. Note that if W_k gives contribution c to face f_1 (f_2), it gives contribution $-c$ to face f_2 (f_1).

A *rectilinear tendril* T_k, for $k \geq 0$, is an undirected graph with two designated poles s and t, and is defined recursively as follows:

- T_0 consists of a single edge (s, t).
- T_1 is the 10-vertex graph with poles $s = s_1$ and $t = t_1$ shown in Figure 2.3(a).
- T_k is constructed from T_{k-1} by removing s_{k-1} and its incident edge and

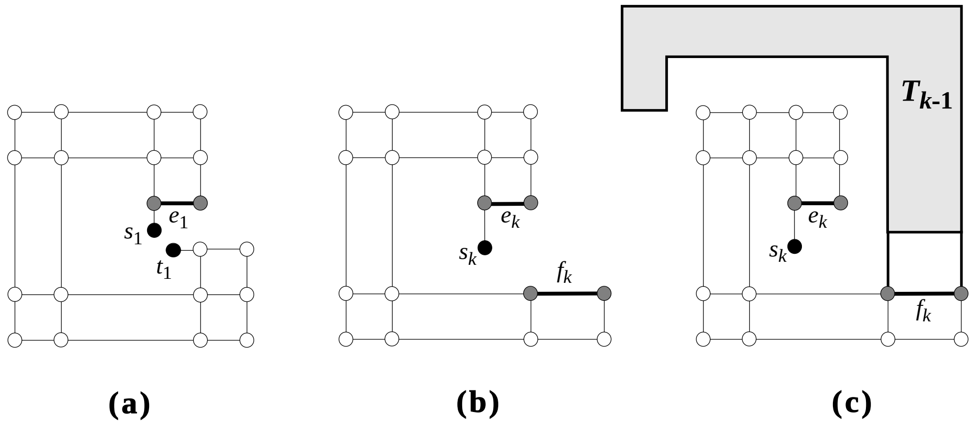

FIG. 2.3. (a) *Rectilinear tendril* T_1*;* (b) *graph* H_k*;* (c) *constructing* T_k *from* T_{k-1}.

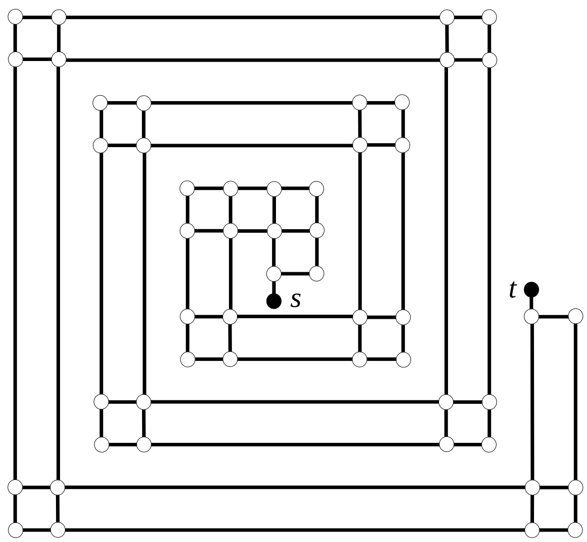

FIG. 2.4. *Rectilinear tendril* T_3.

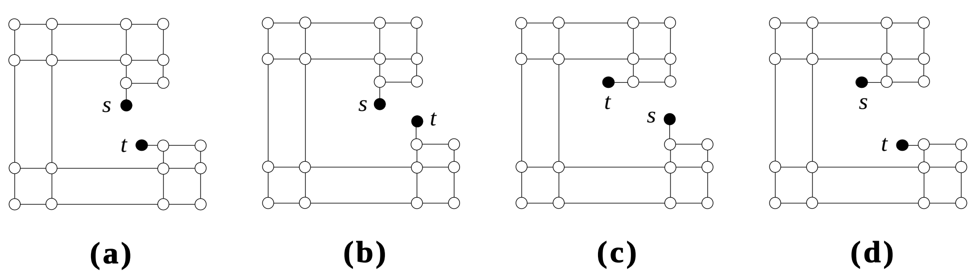

FIG. 2.5. *The four rectilinear embeddings of rectilinear tendril* T_1.

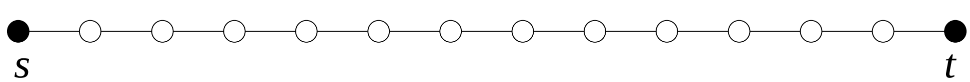

FIG. 2.6. *Rectilinear wiggle* W_3.

adding the graph H_k shown in Figure 2.3(b) to it by identifying its edge e_{k-1} with the edge f_k of H_k (Figure 2.3(c)). The poles of T_k are $s = s_k$ and $t = t_1$.

Figure 2.4 shows rectilinear tendril T_3.

LEMMA 2.4. *Rectilinear tendril T_k is rectilinear planar and admits exactly four rectilinear embeddings.*

Proof. The proof has the same flavor as that of the proof of Lemma 2.3. We use induction on k and apply Lemma 2.2. ∎

Figure 2.5 shows the four rectilinear embeddings of T_1. An *external* path of T_k is a subpath from s to t of its external face in a rectilinear planar embedding of T_k. Notice that T_k has exactly two external paths.

A *rectilinear wiggle* W_k consists of a chain of $4k+1$ edges, and its two end vertices are called the *poles*. Figure 2.6 shows rectilinear wiggle W_3.

Later, in section 5, we will consider transformations where an edge (u, v) of an embedded graph is replaced with a rectilinear tendril T_k or wiggle W_k such that u and v are identified with the poles of T_k or W_k. Let f_1 and f_2 be the two faces of the new graph such that W_k or the external paths of T_k are subpaths of f_1 and f_2. The *contribution* of W_k or T_k to f_1 (f_2) is equal to the number of its angles labeled 3 minus the number of its angles labeled 1 that are in f_1 (f_2). Hence, the contribution of T_k to f_1 (f_2) is an integer c, where c is equal to one of $-(4k+2)$, $-(4k+1)$, $-4k$, $4k$, $4k+1$, and $4k+2$, and the contribution to f_2 (f_1) is $-c$. The contribution of W_k to f_1 (f_2) is an integer c, where $-4k \leq c \leq 4k$, and the contribution to f_2 (f_1) is $-c$.

3. An auxiliary flow problem. In this section, we define two auxiliary flow problems and show that they are equivalent to NOT-ALL-EQUAL-3-SAT under polynomial-time reductions.

A *switch-flow network* is an undirected flow network $\mathcal{N}$, where each edge is labeled with a range $[c' \cdots c'']$ of nonnegative integer values, called the *capacity range* of the edge. For simplicity, we denote the capacity range $[c \cdots c]$ with $[c]$. A *flow* for a switch-flow network is an orientation of and an assignment of integer "flow" values to the edges of the network. A *feasible flow* is a flow that satisfies the following two properties:

Range property. The flow assigned to an edge is an integer within the capacity range of the edge.

Conservation property. The total flow entering a vertex from the incoming edges is equal to the total flow exiting the vertex from the outgoing edges.

Starting from an instance $\mathcal{S}$ of NOT-ALL-EQUAL-3-SAT, we construct a switch-flow network $\mathcal{N}$ as follows (see Figures 3.1–3.2). Let the literals of $\mathcal{S}$ be denoted with $x_1, y_1, \ldots, x_n, y_n$, where $y_i = \overline{x_i}$, and let the clauses of $\mathcal{S}$ be denoted with $c_1, \ldots, c_m$. Let θ be a positive integer parameter. We denote with α_i and β_i (where $i = 1, \ldots, n$) the number of occurrences of literals x_i and y_i, respectively, in the clauses of $\mathcal{S}$. Note that $\sum_{i=1}^{n}(\alpha_i + \beta_i) = 3m$. Also, we define $\gamma_i = (2i-1)\theta$ and $\delta_i = 2i\theta$ ($i = 1, \ldots, n$). Network $\mathcal{N}$ has a *literal vertex* for each literal of $\mathcal{S}$ and a *clause vertex* for each clause of $\mathcal{S}$, plus a special dummy vertex z. There are three types of edges in $\mathcal{N}$ (see Figure 3.2):

Literal edges. Joining pairs of literals associated with the same boolean variable; the capacity range of literal edge (x_i, y_i) is $[\alpha_i\gamma_i + \beta_i\delta_i]$.

Clause edges. Joining each literal to each clause; the capacity range of clause edge (x_i, c_j) is $[\gamma_i]$ if $x_i \in c_j$ and $[0]$ otherwise. The capacity range of clause edge (y_i, c_j) is $[\delta_i]$ if $y_i \in c_j$ and $[0]$ otherwise.

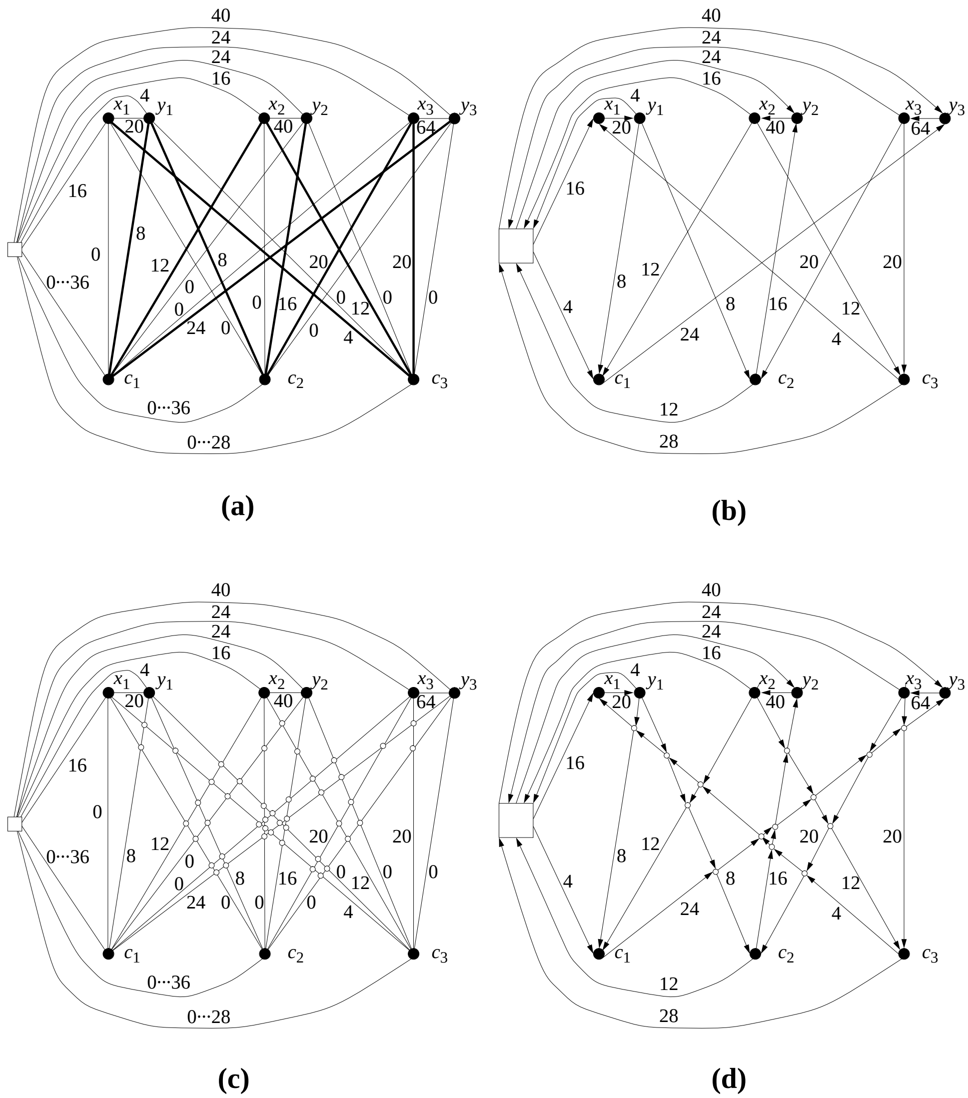

FIG. 3.1. (a) *Switch-flow network $\mathcal{N}$ with parameter $\theta = 4$ associated with the* NOT-ALL-EQUAL-3-SAT *instance $\mathcal{S}$ with clauses $c_1 = y_1x_2y_3$, $c_2 = y_1y_2x_3$, and $c_3 = x_1x_2x_3$. The clause edges with nonzero capacity range are shown with thick lines.* (b) *Feasible flow for $\mathcal{N}$ corresponding to the satisfying truth assignment (y_1, x_2, x_3) for $\mathcal{S}$. Only the edges with nonzero flow are shown.* (c) *Planar switch-flow network $\mathcal{P}$ associated with $\mathcal{S}$.* (d) *Feasible flow for $\mathcal{P}$ corresponding to the satisfying truth assignment (y_1, x_2, x_3) for $\mathcal{S}$. Only the edges with nonzero flow are shown.*

Dummy edges. Joining each literal and each clause to the dummy vertex; the capacity ranges of dummy edges (z, x_i) and (z, y_i) are $[\beta_i\delta_i]$ and $[\alpha_i\gamma_i]$, respectively. The capacity range of dummy edge (z, c_j) is $[0 \cdots \eta_j - 2\theta]$, where η_j is the sum of the capacities of the clause edges incident on c_j.

The construction of network $\mathcal{N}$ from $\mathcal{S}$ is straightforward, and we have the following lemma.

LEMMA 3.1. *Given an instance $\mathcal{S}$ of* NOT-ALL-EQUAL-3-SAT *with n variables and m clauses, the associated switch-flow network $\mathcal{N}$ has $O(n + m)$ vertices and $O(nm)$*

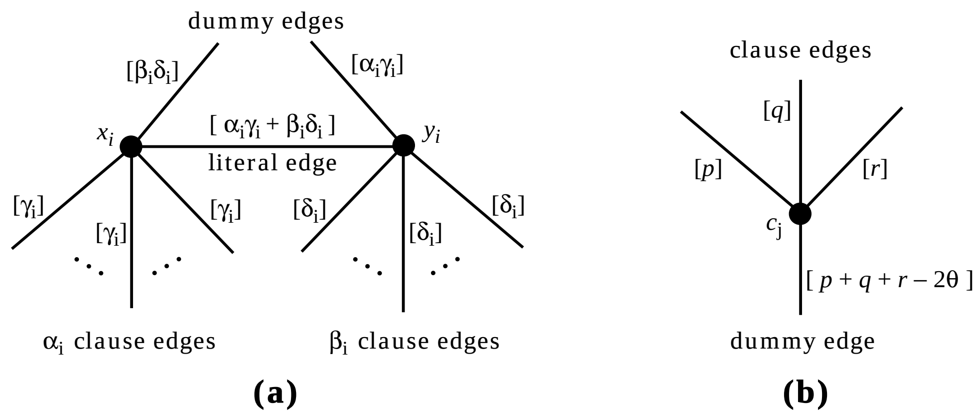

FIG. 3.2. *Schematic illustration of the edges incident on the literal and clause vertices of network N:* (a) *literal vertices x_i and y_i;* (b) *clause vertex c_j.*

edges and can be constructed in $O(nm)$ time.

A feasible flow in network $\mathcal{N}$ corresponds to a satisfying truth assignment for $\mathcal{S}$. Namely, we have that a literal is true whenever its incident literal edge is incoming in the feasible flow (see Figure 3.1(b)) and its incident clause edges with nonzero capacity range are outgoing. We formalize this correspondence in the following lemma.

LEMMA 3.2. *An instance $\mathcal{S}$ of* NOT-ALL-EQUAL-3-SAT *is satisfiable if and only if the associated switch-flow network $\mathcal{N}$ admits a feasible flow. Also, given a feasible flow for $\mathcal{N}$, a satisfying truth assignment for $\mathcal{S}$ can be computed in time $O(nm)$, where n and m are the number of variables and clauses of $\mathcal{S}$, respectively.*

Proof. If. Given a feasible flow in $\mathcal{N}$, we construct a truth assignment A by setting a literal *true* if its incident literal edge is incoming in the flow. We now show that this is a satisfying assignment. Clearly, the two literals x_i and y_i associated with the same boolean variable consistently receive opposite truth assignments.

Because of the conservation property, all the incident clause edges with nonzero capacity range of a *true* literal are outgoing, and the amount of flow in each of them is equal to the capacity. Conversely, if a literal is *false*, then because of the conservation property, all its incident clause edges with nonzero capacity range are incoming, and the amount of flow in each of them is equal to the capacity. The three clause edges with nonzero capacity range incident on a clause vertex c_j cannot be all incoming or all outgoing because of the conservation property at vertex c_j and the choice of the capacity range for the dummy edge incident on c_j. Therefore, three literals in c_j can not be all *true* or all *false*. Hence, A is a satisfying truth assignment for $\mathcal{S}$. Also, A can be constructed in time linear in the number of edges of $\mathcal{N}$, which is $O(nm)$.

Only if. Let A be a satisfying truth assignment of $\mathcal{S}$. We construct a feasible flow f in $\mathcal{N}$ from A. In this flow, we have the following:

- The amount of flow through the literal, clause, and the dummy edges incident on literal vertices is equal to their capacities. The flow, therefore, satisfies the range property in these edges.
- If a literal is true, then its incident literal edge is incoming and its incident clause edges with nonzero capacity range and its dummy edge are outgoing. If a literal is false then its incident literal edge is outgoing and its incident clause edges with nonzero capacity and its dummy edge are incoming. In either case, the amount of flow coming into a literal l_i is $\alpha_i\gamma_i + \beta_i\delta_i$, and

the amount of flow leaving it is $\alpha_i\gamma_i + \beta_i\delta_i$. Therefore, the flow satisfies the conservation property at these vertices.

- If the net flow entering a clause vertex through the clause edges is ν_j, then the flow leaving it through its dummy edge is ν_j. Conversely, if the net flow leaving a clause through the clause edges is ν_j, then the flow entering it through its dummy edge is ν_j. Since A is a satisfying assignment, the three clause edges with nonzero capacity range incident on a clause vertex cannot be all incoming or all outgoing. The magnitude of flow in a clause edge is at least θ. Therefore, the net flow coming into or leaving a clause vertex c_j through its clause edges is at most $\eta_j - 2\theta$. Hence, the conservation property at c_j and the range property in its dummy edge are both satisfied by the flow.
- We show now that the conservation property holds at the dummy vertex by using the following argument. By successive mergers of two nondummy vertices of $\mathcal{N}$ and elimination of the resultant self loops and the flow through them, we can reduce $\mathcal{N}$ into a graph with multiple edges between two vertices: the dummy vertex and another vertex u for which the conservation property holds. The net flow between these two vertices is same as in f. Since the conservation property holds for u and the dummy vertex is neither a source nor a sink, the conservation property holds for the dummy vertex also. ∎

Now, starting from $\mathcal{S}$, we construct a planar switch-flow network $\mathcal{P}$ (see Figure 3.1). We first construct a *layered drawing* $\psi_{\mathcal{N}}$ of $\mathcal{N}$ as follows (see Figure 3.1(a)):

- Each literal and clause edge is drawn as a straight line. The dummy edges are drawn as continuous curves.
- The clause vertices are horizontally aligned and ordered $c_1, c_2, \ldots, c_m$ from left to right.
- The literal vertices are horizontally aligned above the clause vertices and ordered $x_1, x_2, y_1, y_2, \ldots, x_m, y_m$ from left to right.
- There are crossings only between the clause edges. However, no more than two clause edges cross at the same point.

We next replace the crossings of $\psi_{\mathcal{N}}$ with vertices called the *crossing* vertices, thus splitting the clause edges at the crossing vertices. We call *fragment edges*, or simply *fragments*, the edges originated by the splitting of the clause edges. Each fragment edge inherits the capacity range of the originating clause edge. We define the *facial degree* of a vertex as the total number of edges in its incident faces.

Lemma 3.3. *Given network $\mathcal{N}$ representing an instance $\mathcal{S}$ of* not-all-equal-3-sat *with $n \geq 3$ variables and $m \geq 3$ clauses, the associated planar switch-flow network $\mathcal{P}$ is triconnected, has $O(n^2m^2)$ vertices and edges, and can be constructed in $O(n^2m^2)$ time. Also, in the unique embedding of $\mathcal{P}$, the facial degree of each vertex is at most $7nm$.*

Proof. Let $\psi_{\mathcal{N}}$ be the layered drawing of $\mathcal{N}$ that is used to construct $\mathcal{P}$ (see Figure 3.1(a)). There are $O(n^2m^2)$ crossings in $\psi_{\mathcal{N}}$. Therefore, $\mathcal{P}$ has $O(n^2m^2)$ crossing vertices and fragment edges. Consequently, $\mathcal{P}$ has $O(n^2m^2)$ vertices and edges.

Now we show that $\mathcal{P}$ is triconnected. We denote by u_k the crossing vertex that corresponds to the crossing between the line joining c_k and y_n and the line joining c_{k+1} and x_1. We denote by v_k the crossing vertex that corresponds to the crossing between the line joining y_k and c_m and the line joining x_{k+1} and c_1. We call the vertices of type u_k and v_k the *bounding* crossing vertices of $\mathcal{P}$. No two bounding vertices are identical because they correspond to crossings between different pairs of lines. Hence there are two vertex disjoint paths, $p = x_1y_1v_1 \dots x_ky_kv_kx_{k+1} \dots y_n$ and $q = c_1u_1c_2 \dots c_ku_kc_{k+1} \dots c_m$, in $\mathcal{P}$. Let a and b be two nondummy vertices of $\mathcal{P}$. From a and b, there are four vertex-disjoint paths p_a, q_a, p_b, and q_b such that p_a and q_a consist of the fragment edges of a clause edge incident upon a and connect a with a vertex of the paths p and q, respectively, and p_b and q_b consist of the fragment edges of a clause edge incident upon b and connect b with a vertex of the paths p and q, respectively. Notice that if a (b) is a literal vertex, then p_a (p_b) is empty; and if a (b) is a clause vertex, then q_a (q_b) is empty. Hence, between a and b there are two vertex-disjoint paths: one consisting of p_a, a subpath of p, and p_b, and the other consisting of q_a, a subpath of q, and q_b. Hence, graph $\mathcal{P} - \{z\}$ is biconnected. Now suppose, for contradiction, that $\mathcal{P}$ is not triconnected. Since the graph $\mathcal{P} - \{z\}$ is biconnected, z is not a member of a separating pair of $\mathcal{P}$. Let C_1 and C_2 be two connected components of $\mathcal{P}$ obtained after removing a separating pair. Suppose z is in C_1. Hence there are at most two vertex disjoint paths between z and the vertices in C_2. However, between z and a vertex w of $\mathcal{P} - \{z\}$, there are at least four vertex disjoint paths as follows:

- if w is a crossing vertex, then the four paths go through the endpoints of the two clause edges of $\mathcal{N}$ whose crossing is associated with v;
- if w is a clause vertex, then one path consists of the dummy edge incident upon w, and the other three paths go through three literal vertices; and
- if w is a literal vertex, then one path consists of the dummy edge incident upon w, and the other three paths go through three clause vertices.

Thus, we get a contradiction. Therefore $\mathcal{P}$ is triconnected.

Now we show that the facial degree of any vertex is at most $7nm$. All the faces incident on the dummy vertex z have at most four vertices: one is z, two of them are clause and/or literal vertices, and the fourth (if present) is a bounding crossing vertex. There are $n - 1 + m - 1$ bounding crossing vertices in $\mathcal{P}$. A simple counting argument, therefore, shows that the facial degree of z is equal to $3n + m + 2(n - 1 + m - 1) + 2 = 5n + 3m - 2$. Let f be a face that does not contain z. Face f contains at most two fragments of clause edges incident on the same clause or literal vertex. Hence, the number of edges in face f is at most $\min\{2 \cdot 2n, 2m\} = \min\{4n, 2m\}$. The degree of a nondummy vertex u is at most $\max\{2n + 1, m + 2, 4\}$, which is at most $\max\{2n + 1, m + 2\}$ for $n, m \geq 3$. Hence, the facial degree of u is at most $\max\{2n+1, m+2\} \cdot \min\{4n, 2m\} \leq (\max\{2n, m\}+2) \cdot \min\{4n, 2m\} = (\max\{2n, m\}+2) \cdot 2\min\{2n, m\} = 2(\max\{2n, m\} + 2)\min\{2n, m\} = 2\max\{2n, m\}\min\{2n, m\} + 4\min\{2n, m\} = 2(2nm) + 4\min\{2n, m\} = 4nm + \min\{8n, 4m\}$. Since $\min\{8n, 4m\}$ is at most $3nm$ for $n, m \geq 3$, we have that the facial degree of u is at most $4nm + 3nm = 7nm$. Since the facial degree of the dummy vertex is equal to $5n + 3m - 2$, we have that the facial degree of a vertex of $\mathcal{P}$ is at most $\max\{7nm, 5n + 3m - 2\}$, which is equal to $7nm$ for $n, m \geq 3$.

Finally, we show how to construct $\mathcal{P}$ from $\mathcal{N}$ in $O(n^2m^2)$ time. Suppose the literals are numbered from 1 to $2n$ so that literal $l_{2k-1} = x_k$ and $l_{2k} = y_k$. We inductively construct a layered drawing $\psi(k, m)$ of the subgraph of $\mathcal{N}$ induced by literals $l_1, \dots, l_k$ and clauses $c_1, \dots, c_m$ (see Figure 3.3). Drawing $\psi(2, m)$ is shown in Figure 3.3(a). Suppose we have already constructed drawing $\psi(k - 1, m)$ (see Figure 3.3(b)). We place literal l_k at the same height as l_{k-1} and sufficiently far to its right so that edge (l_k, c_1) intersects each clause edge of $\psi(k - 1, m)$ below its lowest crossing. Replacing the crossings of $\psi(2n, m)$ by crossing vertices gives us the planar network $\mathcal{P}$.

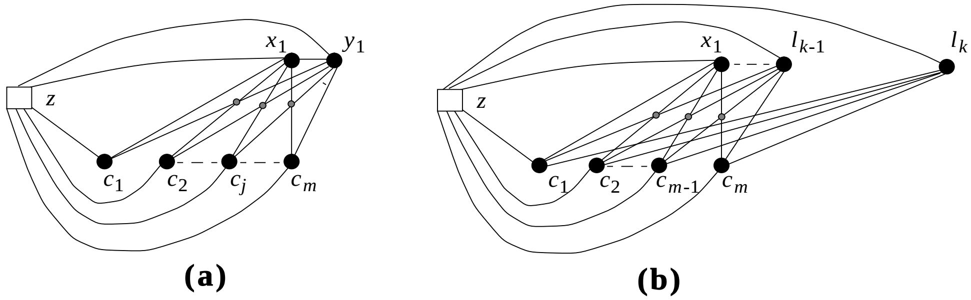

FIG. 3.3. *Proof of Lemma* 3.3: (a) *drawing* $\psi(2, m)$*;* (b) *drawing* $\psi(k, m)$*. The small circles show the lowest crossings of the clause edges in the drawing of* $\psi(k-1, m)$.

The construction of the network $\mathcal{P}$ does not require computing the exact coordinates of the vertices and crossings of $\psi(2n, m)$. In an actual implementation, $\mathcal{P}$ can be constructed directly by maintaining and updating at each inductive step a list of the lowest crossing vertices of the clause edges. The manipulation of this list takes constant time per crossing vertex and hence can be done in total $O(n^2m^2)$ time. The rest of the construction can also be carried out in $O(n^2m^2)$ time. ∎

LEMMA 3.4. *Network* $\mathcal{N}$ *admits a feasible flow if and only if network* $\mathcal{P}$ *admits a feasible flow, and a feasible flow for* $\mathcal{N}$ *can be computed from a feasible flow for* $\mathcal{P}$ *in* $O(n^2m^2)$ *time.*

Proof. Only if. Given a feasible flow in $\mathcal{N}$, we can get a feasible flow in $\mathcal{P}$ in which the flow through each fragment edge is the same as in the corresponding clause edge in $\mathcal{N}$, and the flow through the dummy and literal edges is same as in the corresponding edges in $\mathcal{N}$.

If. Suppose that $\mathcal{P}$ admits a feasible flow f. In f, at any crossing vertex, of the two fragment edges of a clause edge incident upon it, one is incoming and the other is outgoing. This is so because the fragment edges of different clause edges have different capacities and the conservation property is satisfied at the vertex. Consequently, all the fragment edges of a clause edge have the same flow. We can get a feasible flow in $\mathcal{N}$ in which the flow through a clause edge is same the as the flow in its fragment edges in f, and the flow through the dummy edges and the literal edges is the same as the flow in the corresponding edges in $\mathcal{P}$. ∎

By combining Lemmas 3.2, 3.3, and 3.4, we obtain the main result of this section.

THEOREM 3.5. *Given an instance* $\mathcal{S}$ *of* NOT-ALL-EQUAL-3-SAT *with* $n \geq 3$ *variables and* $m \geq 3$ *clauses, the associated planar switch-flow network* $\mathcal{P}$ *is triconnected, has* $O(n^2m^2)$ *vertices and edges, has facial degree at most* $7nm$*, and can be constructed in* $O(n^2m^2)$ *time. Instance* $\mathcal{S}$ *is satisfiable if and only if network* $\mathcal{P}$ *admits a feasible flow. Also, given a feasible flow for* $\mathcal{P}$*, a satisfying truth assignment for* $\mathcal{S}$ *can be computed in time* $O(n^2m^2)$.

4. Upward planarity testing. In this section, we show how to reduce the problem of computing a feasible flow in the planar switch-flow network associated with a NOT-ALL-EQUAL-3-SAT instance to the problem of testing the upward planarity of a suitable digraph.

Let $\mathcal{P}$ be the planar switch-flow network with parameter $\theta = 4$ associated with a NOT-ALL-EQUAL-3-SAT instance $\mathcal{S}$. Now we construct an orientation $\vec{\mathcal{P}}$ of $\mathcal{P}$ as follows (see Figure 4.1):

- Every literal edge (x_i, y_i) is oriented from x_i to y_i.

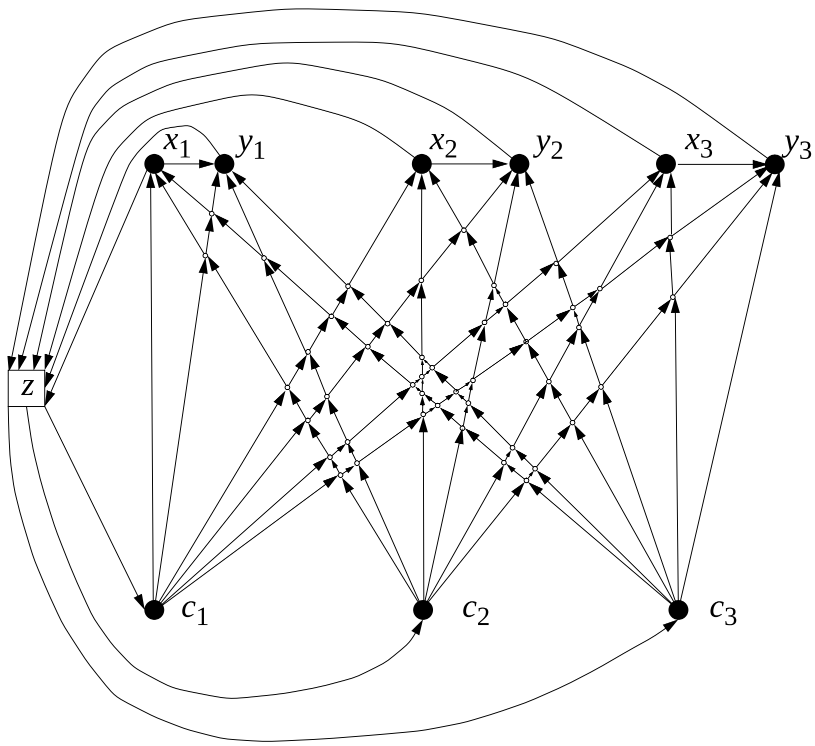

FIG. 4.1. *Orientation $\vec{\mathcal{P}}$ of the network $\mathcal{P}$ shown in Figure* 3.1(c).

- Every fragment edge is oriented "away from" the clause vertex and "towards" the literal vertex.
- Every dummy edge incident upon a literal vertex is oriented towards the dummy vertex, and every dummy edge incident upon a clause vertex is oriented towards the clause vertex.

LEMMA 4.1. *In digraph $\vec{\mathcal{P}}$, every vertex has at least one incoming and one outgoing edge, every directed cycle contains the dummy vertex, and there are exactly two faces that are directed cycles. Also, $\vec{\mathcal{P}}$ is bimodal and each face of $\vec{\mathcal{P}}$ consists of at most two directed paths.*

Proof. Let m be the number of clauses and n be the number of variables in $\mathcal{S}$. Let z be the dummy vertex of $\vec{\mathcal{P}}$.

Each crossing vertex has two incoming and two outgoing fragment edges. Each literal vertex y_i has $m+1$ incoming edges and one outgoing edge (to z). Each literal vertex x_i has m incoming edges and two outgoing edges (to z and y_i). Each clause vertex has $2n$ outgoing edges and one incoming edge (from z). The dummy vertex has $2n$ incoming edges and m outgoing edges. Therefore, each vertex has at least one incoming and one outgoing edge. It can easily be verified that each vertex of $\vec{\mathcal{P}}$ is bimodal, and hence $\vec{\mathcal{P}}$ is bimodal.

The digraph $\vec{\mathcal{P}} - \{z\}$ is upward planar with m sources, each being a clause vertex, and n sinks, each being a literal vertex y_i. Therefore, every directed cycle in $\vec{\mathcal{P}}$ contains z. There are exactly two faces in $\vec{\mathcal{P}}$ that that are directed cycles: one consisting of vertices x_1, z, and c_1, and the other consisting of vertices y_n, z, and c_m. It can be easily verified that the other faces of $\vec{\mathcal{P}}$ each consist of exactly two directed

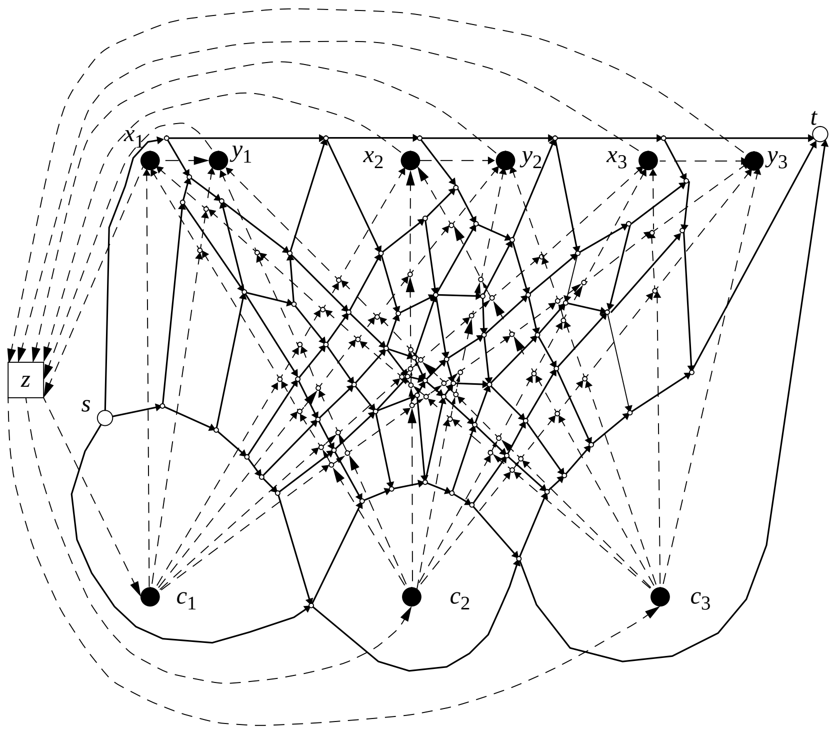

FIG. 4.2. *Dual digraph $\vec{\mathcal{D}}$ of the network $\vec{\mathcal{P}}$ shown in Figure* 4.1.

paths. ∎

Since $\mathcal{P}$ is triconnected (see Theorem 3.5), the planar embedding of $\mathcal{P}$ and the dual graph of $\mathcal{P}$ are unique. We construct the dual digraph $\vec{\mathcal{D}}$ of $\vec{\mathcal{P}}$ by taking the dual graph $\mathcal{D}$ of $\mathcal{P}$ and orienting every dual edge from the face on the left to the face on the right of the primal edge (see Figure 4.2).

LEMMA 4.2. *The dual digraph $\vec{\mathcal{D}}$ of $\vec{\mathcal{P}}$ is upward planar, triconnected, acyclic, and has exactly one source and one sink, denoted with s and t, both of which are on the same face. Also, each face of $\vec{\mathcal{D}}$ has exactly one source and one sink.*

Proof. Because $\vec{\mathcal{P}}$ is planar and triconnected (Theorem 3.5), so is its dual $\vec{\mathcal{D}}$. By Lemma 4.1, exactly two faces of $\vec{\mathcal{P}}$ are directed cycles. Hence, $\vec{\mathcal{D}}$ has exactly two switches, denoted by s and t (see Figure 4.2), respectively, corresponding to these two cycles. Switch s is a source vertex and corresponds to the face of $\vec{\mathcal{P}}$ that consists of the literal vertex x_1, dummy vertex z, and the clause vertex c_1. Switch t is a sink vertex and corresponds to the face of $\vec{\mathcal{P}}$ that consists of the vertices y_n, z, and c_m. Also notice that both s and t are on the face that is the dual of z. From Lemma 4.1, each face of $\vec{\mathcal{P}}$ consists of at most two directed paths, and hence each vertex of $\vec{\mathcal{D}}$ is bimodal. Therefore, $\vec{\mathcal{D}}$ is also bimodal. Again from Lemma 4.1, each vertex of $\vec{\mathcal{P}}$ is bimodal and none of them is a source or a sink. Hence, each face of $\vec{\mathcal{D}}$ has exactly one source and one sink. Since s and t are the only switches of $\vec{\mathcal{D}}$ and both of them are on the same face, it follows that there is a consistent assignment of labels to the angles of $\vec{\mathcal{D}}$ in which exactly two angles are labeled large, namely, the angles at s and t in their common face. Therefore, by Lemma 2.1, $\vec{\mathcal{D}}$ has an upward embedding and hence is upward planar. ∎

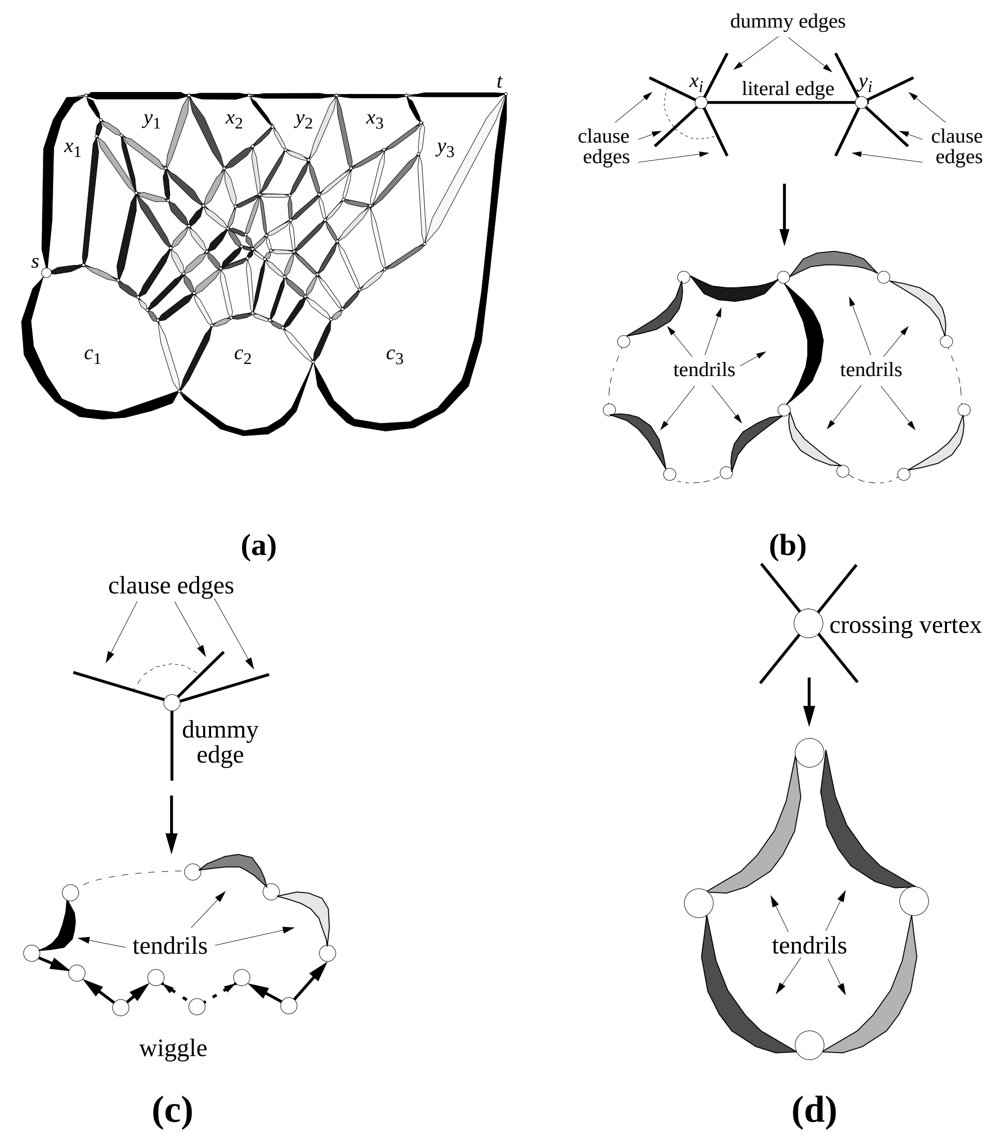

FIG. 4.3. (a) *Schematic illustration of digraph $\vec{\mathcal{G}}$ obtained from $\vec{\mathcal{D}}$ by replacing edges with tendrils and wiggles.* (b) *The two faces of $\vec{\mathcal{G}}$ associated with literal vertices x_i and y_i of $\mathcal{P}$.* (c) *The face of $\vec{\mathcal{G}}$ associated with a clause vertex of $\mathcal{P}$.* (d) *The face of $\vec{\mathcal{G}}$ associated with a crossing vertex of $\mathcal{P}$.*

Starting from digraph $\vec{\mathcal{D}}$, we construct a new digraph $\vec{\mathcal{G}}$ by replacing the edges of $\vec{\mathcal{D}}$ with subgraphs (tendrils or wiggles) as follows (see Figure 4.3):

- Every edge of $\vec{\mathcal{D}}$ that is the dual of a literal edge, fragment edge, or dummy edge incident on a literal vertex is replaced with tendril T_c, where $[c]$ is the capacity range of the dual edge. Notice that c is a multiple of parameter θ.
- Every edge of $\vec{\mathcal{D}}$ that is the dual of a dummy edge incident on a clause vertex is replaced with wiggle W_c, where $[0 \cdots c]$ is the capacity range of the dual edge.

A vertex of $\vec{\mathcal{G}}$ is a *primary* vertex if it is also a vertex of $\vec{\mathcal{D}}$, and is a *secondary* vertex otherwise. A face of an embedding of $\vec{\mathcal{G}}$ is a *secondary* face if it is bounded by the edges of the same tendril, and is a *primary* face otherwise. Figure 4.3(a) shows the primary faces of an embedding ψ of $\vec{\mathcal{G}}$ and the primary vertices of $\vec{\mathcal{G}}$; in this figure, the secondary faces of ψ and the secondary vertices of $\vec{\mathcal{G}}$ are hidden inside the shaded regions denoting the tendrils and the wiggles. We establish the following correspondence between the primary faces of an embedding ψ of $\vec{\mathcal{G}}$, the faces of $\vec{\mathcal{D}}$, and the vertices of $\vec{\mathcal{P}}$. Let f be a primary face of ψ; f corresponds to a face f' of $\vec{\mathcal{D}}$, namely, the one whose boundary edges were replaced by tendrils and wiggles to give f. Recall that $\vec{\mathcal{D}}$ is the dual digraph of $\vec{\mathcal{P}}$, and its faces correspond to the vertices of $\vec{\mathcal{P}}$. Therefore, f also corresponds to a vertex v of $\vec{\mathcal{P}}$, namely, the one that corresponds to f'. f is the *dummy face* of $\vec{\mathcal{G}}$ if v is the dummy vertex of $\vec{\mathcal{P}}$, and is a *nondummy* face otherwise. A *primary* angle a of f is its angle at a primary vertex u; a corresponds to an angle of f', namely, the one at u. A primary vertex of f is a *primary source* of f if it is also a source of f, and is a *primary sink* of f if it is also a sink of f.

There is no directed path from the sink pole to the source pole of a tendril or a wiggle. Therefore, $\vec{\mathcal{G}}$ is also acyclic. From Lemma 4.2 and the construction of digraph $\vec{\mathcal{G}}$, the following lemma is immediate.

LEMMA 4.3. *Digraph $\vec{\mathcal{G}}$ is planar and acyclic. All its embeddings can be obtained by choosing one of the two possible flips for each tendril and have the same set of secondary faces. Also, for every face f in an embedding of $\vec{\mathcal{G}}$, the following holds:*

- *if f is a secondary face, then it has exactly one source and exactly one sink;*
- *or else (f is a primary face), it has exactly one primary source and exactly one primary sink.*

We need the following technical lemma to prove Lemma 4.5.

LEMMA 4.4. *Let f be a primary face of an upward embedding ψ of $\vec{\mathcal{G}}$. Let $\tau(f)$ and $\omega(f)$ be the total contribution to f of its tendrils and wiggles, respectively. Then,*

$$|\tau(f) + \omega(f)| \leq \theta.$$

Proof. From Lemma 4.3, f has exactly one primary source s and exactly one primary sink t. In other words, f has exactly two primary angles. Let $\nu(f)$ be equal to the number of *large* primary angles minus the number of *small* primary angles of f. Clearly, $|\nu(f)| \leq 2$. Let us denote with $L(f)$ and $S(f)$, respectively, the number of *large* and *small* angles of f. Because ψ is an upward embedding, from Lemma 2.1 it follows that $|L(f) - S(f)| = 2$. An angle of f is either a primary angle of f or an angle of one of its tendrils and wiggles. Therefore, $L(f) - S(f) = \nu(f) + \tau(f) + \omega(f)$. Therefore, $\tau(f) + \omega(f) = L(f) - S(f) - \nu(f)$. Hence, $|\tau(f) + \omega(f)| = |L(f) - S(f) - \nu(f)| \leq |L(f) - S(f)| + |\nu(f)|$. Since $|\nu(f)| \leq 2$ and $|L(f) - S(f)| = 2$, we have that $|\tau(f) + \omega(f)| \leq 4$. Recall from the beginning of this section that $\theta = 4$. Hence, $|\tau(f) + \omega(f)| \leq \theta$. ◻

We are now ready to present Lemma 4.5.

LEMMA 4.5. *Digraph $\vec{\mathcal{G}}$ is upward planar if and only if its tendrils can be flipped and labels can be assigned to the angles of its wiggles such that for every primary face the total contribution to it of its tendrils and wiggles is zero.*

Proof. If. From Lemma 4.2, $\vec{\mathcal{D}}$ has an upward embedding $\psi_{\vec{\mathcal{D}}}$. Let g' be the external face of $\psi_{\vec{\mathcal{D}}}$. Let ψ be a labeled embedding of $\vec{\mathcal{G}}$ such that, for every face f of ψ,

- if f is a secondary face, then the angles of f are labeled *small*;
- or else (f is a primary face)

– the total contribution to it of its tendrils and wiggles is zero, and
– its primary angles have the same label as the corresponding angles of $\psi_{\vec{\mathcal{D}}}$.

Notice that such a ψ exists because the tendrils of $\vec{\mathcal{G}}$ can be flipped and labels can be assigned to the angles of the wiggles of $\vec{\mathcal{G}}$ such that for every primary face the total contribution to it of its tendrils and wiggles is zero.

Let g be the primary face of ψ that corresponds to g'. We now show that ψ is an upward embedding with g as its external face. From Lemma 2.1, this is equivalent to showing that each face of ψ is consistently assigned with g as the external face (see the definition of consistently assigned faces in section 2.1).

Let f be a face of ψ. If f is a secondary face, then, from Lemma 4.3, f has exactly one source and exactly one sink. Because both of them are labeled *small* and f is an internal face, f is consistently assigned.

If f is a primary face, then it corresponds to a face f' of $\psi_{\vec{\mathcal{D}}}$. Let d be an integer equal to the number of large angles minus the number of small angles of f. Define d' likewise for f'. Because the total contribution to f of its tendril and wiggles is zero, and the angles at its primary vertices have the same label as the corresponding angles of f', it follows that $d = d'$. Thus, since f' is consistently assigned, f is consistently assigned too.

Only if. Suppose $\vec{\mathcal{G}}$ has an upward embedding ψ. Let f be a face of ψ. Since ψ is an upward embedding, f is consistently assigned, and therefore, from Lemma 2.1, the difference of its large and small angles is 2. However, because f, in general, also has primary vertices, each one of which contributes a *large* or *small* angle to it, it may be possible that the total contribution to it of its tendrils and wiggles is not zero. However, using Lemma 4.5, we show now that $\vec{\mathcal{G}}$ admits a labeled embedding ψ' such that[1]

- ψ' has the same faces as ψ,
- the primary angles of ψ' are the same as those of ψ,
- the contribution of a tendril to a primary face f of ψ' is the same as its contribution to f in ψ,
- the contribution of a wiggle to a primary face f of ψ' *may be different* from its contribution to f in ψ, and
- for every primary face f of ψ', the total contribution to f of its tendrils and wiggles is zero.

(Thus, ψ' and ψ are the same except for a possible difference in the assignment of labels to the angles of their wiggles.)

Let f be a primary face of ψ (and therefore also of ψ'). Let $\tau(f)$ and $\omega(f)$ be the total contributions to f of its tendrils and wiggles, respectively, in ψ. Let $\tau'(f)$ and $\omega'(f)$ be the total contributions to f of its tendrils and wiggles, respectively, in ψ'. Our goal is to show that $\tau'(f) + \omega'(f) = 0$.

First of all, we notice that because ψ and ψ' have the same faces, $\tau(f) = \tau'(f)$.

We now describe the assignment of labels to the angles of the wiggles of ψ' and show that $\tau'(f) + \omega'(f) = 0$ with this assignment. We have the following three cases:

Case 1. f is a nondummy face and it corresponds to a literal vertex or a crossing vertex of $\vec{\mathcal{P}}$. Clearly, f has no wiggles. Therefore, $\omega'(f) = \omega(f) = 0$. Since $\omega(f) = 0$,

[1]By appropriately reassigning labels to the angles of those primary vertices that are endpoints of the wiggles, from ψ we can obtain an upward embedding in which for every primary face f the total contribution to f of its tendrils and wiggles is zero. However, for our purposes, it is sufficient to show the existence of a labeled embedding ψ', as described here.

from Lemma 4.4 it follows that $|\tau(f)| \leq \theta$. From the construction of graph $\vec{\mathcal{G}}$ and the fact that a tendril T_k gives a contribution equal to either $2k$ or $-2k$ to a face (see section 2.2), it follows that $|\tau(f)|$ is a multiple of 2θ. Therefore, $|\tau(f)| \leq \theta$ implies that $\tau(f)$ is equal to 0. Since $\tau'(f) = \tau(f)$, we have that $\tau'(f)$ is also equal to 0. Because $\omega'(f) = 0$, it follows that $\tau'(f) + \omega'(f) = 0$.

Case 2. f is a nondummy face and it corresponds to a clause vertex c_j of $\vec{\mathcal{P}}$. Clearly, f has exactly one wiggle w. In ψ', we assign labels to the angles of w such that $\omega'(f) = -\tau(f)$. As noted earlier, $\tau'(f) = \tau(f)$. Hence, $\tau'(f)+\omega'(f) = \tau(f)-\tau(f) = 0$ with this assignment of labels to w. Therefore, if we can show that it is possible to assign labels to the angles of w such that $\omega'(f) = -\tau(f)$, we are done.

Recall from the construction of network $\mathcal{N}$ from section 3 that the capacity range of the dummy edge incident on c_j is $[0 \cdots \eta_j - 2\theta]$, where η_j is the sum of the capacities of the clause edges incident on c_j. Also it follows from the construction of $\vec{\mathcal{G}}$ that w is a copy of the wiggle $w_{\eta_j - 2\theta}$. Because the magnitude of the contribution of a wiggle w_k to a face is at most $2k$ (see section 2.2), we have that $|\omega(f)| \leq 2(\eta_j - 2\theta)$. Therefore, if we are able to show that $|\tau(f)|$ is at most $2(\eta_j - 2\theta)$, then because $\tau(f)$ is an even number, and w can give any even valued contribution in the range $-2(\eta_j - 2\theta)$ to $2(\eta_j - 2\theta)$, we will be able to show that it is possible to assign labels to the angles of w such that the contribution of w to f is equal to $-\tau(f)$.

We now prove that $|\tau(f)|$ is at most $2(\eta_j - 2\theta)$. From the construction of $\vec{\mathcal{G}}$ and the fact that a tendril T_k gives a contribution equal to either $2k$ or $-2k$ to a face (see section 2.2), it follows that the maximum value of $|\tau(f)|$ is $2\eta_j$, and $|\tau(f)|$ is a multiple of 2θ. Therefore, either $|\tau(f)| = 2\eta_j$, or $|\tau(f)| = 2\eta_j - 2\theta$, or $|\tau(f)| \leq 2\eta_j - 4\theta$. However, because $|\omega(f)| \leq 2(\eta_j - 2\theta)$, from Lemma 4.4 it follows that $|\tau(f)| \leq 2(\eta_j - 2\theta) + \theta = 2\eta_j - 3\theta$. Hence, $|\tau(f)|$ cannot be equal to $2\eta_j$ or $2\eta_j - 2\theta$. Consequently, $|\tau(f)| \leq 2\eta_j - 4\theta = 2(\eta_j - 2\theta)$.

Case 3. f is the dummy face of ψ'. Let T be a tendril of $\vec{\mathcal{G}}$. If T contributes k to a primary face of ψ', it also contributes $-k$ to another primary face of ψ'. Therefore, the total contribution of the tendrils of ψ', when summed over all its primary faces, is 0. Similarly, the total contribution of the wiggles of ψ', when summed over all its primary faces, is 0. We have already shown by considering Cases 1 and 2 that if h is a nondummy face, then $\tau'(h) + \omega'(h) = 0$. Therefore, it follows that $\tau'(f) + \omega'(f) = 0$. ◻

THEOREM 4.6. *Given an instance $\mathcal{S}$ of* NOT-ALL-EQUAL-3-SAT *with n variables and m clauses and the associated planar switch-flow network $\mathcal{P}$, digraph $\vec{\mathcal{G}}$ associated with $\mathcal{S}$ and $\mathcal{P}$ has $O(n^3m^2)$ vertices and edges and can be constructed in $O(n^3m^2)$ time. Instance $\mathcal{S}$ is satisfiable and network $\mathcal{P}$ admits a feasible flow if and only if digraph $\vec{\mathcal{G}}$ is upward planar. Also, given an upward planar embedding for $\vec{\mathcal{G}}$, a feasible flow for $\mathcal{P}$ and a satisfying truth assignment for $\mathcal{S}$ can be computed in time $O(n^3m^2)$.*

Proof. Since $\theta = 4 = O(1)$, from the construction of $\vec{\mathcal{G}}$ we have that the number of vertices and edges in a tendril or a wiggle is $O(n)$. Since $\mathcal{P}$ has $O(n^2m^2)$ vertices and edges (see Theorem 3.5), $\vec{\mathcal{G}}$ has $O(n^3m^2)$ vertices and edges. $\mathcal{P}$ can be constructed from $\mathcal{S}$ in $O(n^2m^2)$ time (see Theorem 3.5), and $\vec{\mathcal{G}}$ can be constructed from $\mathcal{P}$ in $O(n^3m^2)$ time. Thus, we can construct $\vec{\mathcal{G}}$ from $\mathcal{S}$ in $O(n^3m^2)$ time.

We now show that instance $\mathcal{S}$ is satisfiable and network $\mathcal{P}$ admits a feasible flow if and only if digraph $\vec{\mathcal{G}}$ is upward planar.

We establish the following correspondences between digraph $\vec{\mathcal{G}}$ and network $\mathcal{P}$ (see Figure 4.3):

- The faces of $\vec{\mathcal{G}}$ correspond to the vertices of $\mathcal{P}$.
- The tendrils and wiggles of $\vec{\mathcal{G}}$ correspond to the the edges of $\mathcal{P}$.
- Flipping a tendril T_k of $\vec{\mathcal{G}}$ corresponds to orienting an edge e of $\mathcal{P}$, where e is the dual of the edge replaced by T_k in constructing $\vec{\mathcal{G}}$ from $\vec{\mathcal{D}}$. Edge e is oriented towards an endpoint v (which is a vertex of $\mathcal{P}$) if and only if v corresponds to f and T_k contributes $2k$ to f.
- The contribution of a tendril or wiggle U of $\vec{\mathcal{G}}$ corresponds to the flow in an edge e of $\mathcal{P}$. Here e is the dual of the edge which is replaced by U in constructing $\vec{\mathcal{G}}$ from $\vec{\mathcal{D}}$. The contribution of U to a face f is equal to the amount of the flow coming through e into the endpoint (of e) that corresponds to f.
- The balance of the contributions of the tendrils and wiggles to the faces of $\vec{\mathcal{G}}$ corresponds to the conservation of flow at the corresponding vertices of $\mathcal{P}$; i.e., the total contribution of the tendrils and wiggles to a face is zero if and only if there is a conservation of flow at its corresponding vertex in $\mathcal{P}$.

From Theorem 3.5, Lemma 4.5, and the correspondence established above between a feasible flow in $\mathcal{P}$ and the upward planarity of $\vec{\mathcal{G}}$, it follows that $\mathcal{S}$ is satisfiable and $\mathcal{P}$ admits a feasible flow if and only if $\vec{\mathcal{G}}$ is upward planar. It also follows that given an upward planar embedding for $\vec{\mathcal{G}}$, a feasible flow for $\mathcal{P}$ and a satisfying truth assignment for $\mathcal{S}$ can be computed in $O(n^3m^2)$ time. ∎

From Theorem 4.6, we conclude the following corollary.

COROLLARY 4.7. *Upward planarity testing is NP-complete.*

5. Rectilinear planarity testing. In this section, we show that rectilinear planarity testing is NP-complete by reducing the problem of computing a feasible flow in the planar switch-flow network associated with an instance of NOT-ALL-EQUAL-3-SAT to the problem of testing the rectilinear planarity of a suitable graph $\mathcal{G}$. The construction of $\mathcal{G}$ is similar to the construction of $\vec{\mathcal{G}}$ in section 4 and is carried out in several stages, where at each stage an intermediate graph is produced.

Let $\mathcal{S}$ be an instance of NOT-ALL-EQUAL-3-SAT with n variables and m clauses; let $\mathcal{P}$ be the associated planar switch-flow network of $\mathcal{S}$ with parameter $\theta = 4+37nm$ (see section 3).

Let $\mathcal{D}$ be the dual graph of $\mathcal{P}$. Starting from $\mathcal{D}$, we construct a degree-3 planar graph $\mathcal{F}$ using the following two-step process:

1. First, replace each vertex of $\mathcal{D}$ by a binary tree with d leaves. Let $\mathcal{E}$ be the resultant graph.
2. Next, replace each edge e of $\mathcal{E}$ with a chain c_e consisting of five edges. The middle edge of c_e is called the *representative* of e in $\mathcal{F}$.

Since $\mathcal{P}$ has $O(n^2m^2)$ vertices and edges, it follows that $\mathcal{D}$ and $\mathcal{F}$ also have $O(n^2m^2)$ vertices and edges each.

LEMMA 5.1. *Graph $\mathcal{F}$ has a unique planar embedding and admits a rectilinear embedding, which can be constructed in linear time.*

Proof. Since $\mathcal{D}$ has a unique planar embedding, it follows that $\mathcal{F}$ also has a unique planar embedding.

It is known that every degree-4 planar graph admits an orthogonal drawing with at most four bends per edge, which can be constructed in linear time (see, e.g., [31]). Hence, $\mathcal{E}$ also admits a planar orthogonal drawing R with at most four bends per edge. Because each edge e of $\mathcal{E}$ is replaced by a chain c_e in $\mathcal{F}$, from R we can construct a rectilinear embedding of $\mathcal{F}$ by replacing each bend of e by an intermediate vertex of c_e. ∎

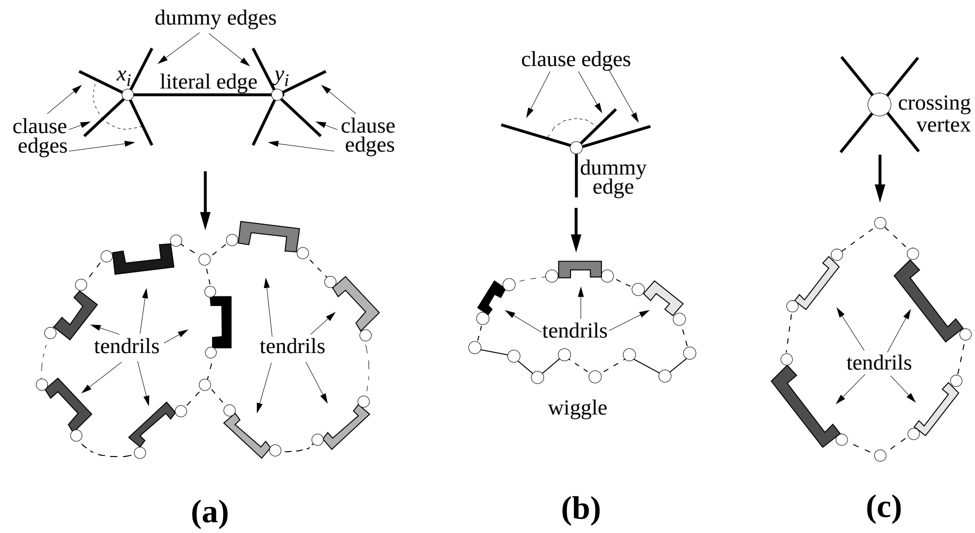

FIG. 5.1. *Schematic illustration of graph $\mathcal{G}$ obtained from $\mathcal{F}$ by replacing edges with rectilinear tendrils and wiggles:* (a) *the two faces of $\mathcal{G}$ associated with literal vertices x_i and y_i of $\mathcal{P}$;* (b) *the face of $\mathcal{G}$ associated with a clause vertex of $\mathcal{P}$;* (c) *the face of $\mathcal{G}$ associated with a crossing vertex of $\mathcal{P}$.*

We construct $\mathcal{G}$ from $\mathcal{F}$ by replacing the edges of $\mathcal{F}$ with subgraphs (rectilinear tendrils or wiggles) as follows (see Figure 5.1). Let e be an edge of $\mathcal{D}$ and *rep(e)* be the representative of e in $\mathcal{G}$.

- If e is the dual of a literal edge, fragment edge, or dummy edge incident on a literal vertex, then *rep(e)* is replaced with a rectilinear tendril T_c, where $[c]$ is the capacity range of the dual edge of e. Note that c is a multiple of parameter θ.
- If e is the dual of a dummy edge e' incident on a clause vertex, then *rep(e)* is replaced with a rectilinear wiggle W_c, where $[0 \cdots c]$ is the capacity range of the dual edge of e.

The vertices of $\mathcal{G}$ that are also vertices of $\mathcal{F}$ are called its *primary* vertices. We define the *primary*, *secondary*, *dummy*, and *nondummy* faces of $\mathcal{G}$ similar to their definition for $\vec{\mathcal{G}}$ in section 4 and also establish similar correspondences between the primary faces of an embedding ψ of $\mathcal{G}$, the faces of $\mathcal{F}$, and the vertices of $\mathcal{P}$.

By Lemma 5.1 and the construction of graph $\mathcal{G}$, all the embeddings of $\mathcal{G}$ are obtained by choosing one of the two possible flips for each rectilinear tendril.

Let ψ be a rectilinear embedding of $\mathcal{G}$. Recall that a rectilinear tendril T_k contributes one of $4k$, $4k+1$, $4k+2$, $-4k$, $-(4k+1)$, and $-(4k+2)$ to a face of $\mathcal{G}$. In ψ, the *significant* contribution of a rectilinear tendril T_k to a face is $4k$ if its contribution is one of $4k$, $4k+1$, and $4k+2$, and is $-4k$ otherwise. In ψ, the *significant* contribution of a rectilinear wiggle to a face is equal to its contribution to the face. Hence, the difference in the contribution and significant contribution of a tendril (wiggle) to a face of ψ is at most 2 (0). Also, the total significant contribution of the tendrils to a face of ψ is a multiple of 4θ. The *contribution* of a primary vertex to a face f of ψ is 1 if its angle in f is labeled 3, is -1 if its angle in f is labeled 1, and is 0 otherwise.

LEMMA 5.2. *If $n \geq 3$ and $m \geq 3$, then in a rectilinear embedding of $\mathcal{G}$ the magnitude of the total contribution of primary vertices to a nondummy face is at* *most* $35nm$.

Proof. We show that, in a rectilinear embedding of $\mathcal{G}$, a nondummy face has at most $35nm$ primary vertices, and hence the magnitude of the total contribution of primary vertices to it is at most $35nm$.

Let f be a face of $\mathcal{D}$ with k vertices. Let f be the dual of vertex u of $\mathcal{P}$. Expanding a vertex of f with degree d by a binary tree adds at most $d-1$ vertices to f. Hence, expanding each vertex of f by a binary tree adds at most $r-k$ vertices to f, where r is the facial degree of u. Therefore, after step 1, f has at most $k+r-k=r$ vertices. Hence, after replacing each edge of f by a chain of five edges in step 2, f has at most $5r$ vertices. From Theorem 3.5, r is at most $7nm$ for $n, m \geq 3$. Hence, f has at most $35nm$ vertices in $\mathcal{G}$. ☐

LEMMA 5.3. *Let f be a primary face of a rectilinear embedding ψ of $\mathcal{G}$. Let $\tau(f)$ and $\omega(f)$ be the total significant contributions to f of its tendrils and wiggles, respectively. Then*

$$|\tau(f)+\omega(f)| \leq \theta.$$

Proof. Let $\nu(f)$ be equal to the number of primary angles labeled 3 minus the number of primary angles labeled 1 of f. Let $\tau'(f)$ and $\omega'(f)$ be the total contribution to f of its tendrils and wiggles, respectively. Let us denote with $N_3(f)$ and $N_1(f)$ the number of angles of f labeled 3 and 1, respectively. An angle of f is either a primary angle of f or an angle of one of its tendrils and wiggles. Therefore, $N_3(f) - N_1(f) = \nu(f)+\tau'(f)+\omega'(f)$. Hence, $\tau'(f)+\omega'(f) = N_3(f)-N_1(f)-\nu(f)$. Therefore, $|\tau'(f)+\omega'(f)| = |N_3(f)-N_1(f)-\nu(f)| \leq |N_3(f)-N_1(f)|+|\nu(f)|$. Because ψ is a rectilinear embedding, the sum of the labels of the angles around each vertex of ψ is equal to 4. Hence, because each angle of $\mathcal{G}$ has label at least 1, and each vertex of $\mathcal{G}$ has degree at least 2, no angle of $\mathcal{G}$ has label 4. Therefore, $N_4(f) = 0$. Hence, from Lemma 2.2, it follows that $|N_3(f) - N_1(f)| = 4$. Therefore, $|\tau'(f)+\omega'(f)| \leq 4+|\nu(f)|$. Since the difference in the contribution and significant contribution of a tendril to a face of ψ is at most 2, and f has at most nm tendrils, we have that $|\tau(f)| \leq |\tau'(f)|+2nm$. Since $\omega(f) = \omega'(f)$, it follows that $|\tau(f)+\omega(f)| \leq |\tau'(f)+\omega'(f)|+2nm \leq 4+|\nu(f)|+2nm$. Since, from Lemma 5.2, $|\nu(f)| \leq 35nm$, we have that $|\tau(f)+\omega(f)| \leq 4+37nm$. Recall from the beginning of this section that $\theta = 4+37nm$. Hence, $|\tau(f)+\omega(f)| \leq \theta$. ☐

We are now ready to present Lemma 5.4.

LEMMA 5.4. *Graph $\mathcal{G}$ is rectilinear planar if and only if its tendrils can be flipped and labels can be assigned to the angles of its wiggles such that for every primary face the total significant contribution to it of its tendrils and wiggles is zero.*

Proof. If. From Lemma 5.1, $\mathcal{F}$ has a rectilinear embedding $\psi_{\mathcal{F}}$. Let g' be the external face of $\psi_{\mathcal{F}}$. Let ψ be a labeled embedding of $\mathcal{F}$ such that, for every face f of ψ,

- if f is a secondary face, then the angles of f are labeled 1,
- or else (f is a primary face)
    - the contribution of a tendril or a wiggle to f is equal to its significant contribution, and the total contribution to f of its tendrils and wiggles is zero, and
    - its primary angles have the same label as the corresponding angles of $\psi_{\mathcal{F}}$.

The rest of the proof uses the same arguments as those in the *if* part of the proof for Lemma 4.5 with the contribution of a tendril or wiggle replaced by the significant contribution of a rectilinear tendril or wiggle.

Only if. The proof uses Lemma 5.3 and the same arguments as those by the *only if* part of the proof for Lemma 4.5 with the contribution of a tendril or wiggle replaced by the significant contribution of a rectilinear tendril or wiggle. ∎

THEOREM 5.5. *Given an instance $\mathcal{S}$ of* NOT-ALL-EQUAL-3-SAT *with n variables and m clauses, graph $\mathcal{G}$ associated with $\mathcal{S}$ has $O(n^4m^3)$ vertices and edges and can be constructed in $O(n^4m^3)$ time. Instance $\mathcal{S}$ is satisfiable if and only if graph $\mathcal{G}$ is rectilinear planar. Also, given a rectilinear planar embedding for $\mathcal{G}$, a satisfying truth assignment for $\mathcal{S}$ can be computed in time $O(n^4m^3)$.*

Proof. Since $\theta = 4+37nm = O(nm)$, from the construction of $\mathcal{G}$ we have that the number of vertices and edges in a tendril or a wiggle is $O(n^2m)$. Since $\mathcal{P}$ has $O(n^2m^2)$ vertices and edges (see Theorem 3.5), $\mathcal{G}$ has $O(n^4m^3)$ vertices and edges. $\mathcal{P}$ can be constructed from $\mathcal{S}$ in $O(n^2m^2)$ time (see Theorem 3.5), and $\mathcal{G}$ can be constructed from $\mathcal{P}$ in $O(n^4m^3)$ time. Hence, we can construct $\mathcal{G}$ from $\mathcal{S}$ in $O(n^4m^3)$ time.

We now show that instance $\mathcal{S}$ is satisfiable and network $\mathcal{P}$ admits a feasible flow if and only if graph $\mathcal{G}$ is rectilinear planar.

We establish the following correspondences between graph $\mathcal{G}$ and network $\mathcal{P}$ (see Figure 5.1):

- The faces of $\mathcal{G}$ correspond to the vertices of $\mathcal{P}$.
- The rectilinear tendrils and wiggles of $\mathcal{G}$ correspond to the the edges of $\mathcal{P}$.
- Flipping a rectilinear tendril T_k of $\mathcal{G}$ corresponds to orienting an edge e of $\mathcal{P}$. Edge e is oriented towards an endpoint v (which is a vertex of $\mathcal{P}$) if and only if v corresponds to f and the significant contribution of T_k to f is $4k$.
- The significant contribution of a tendril or wiggle U of $\mathcal{G}$ corresponds to the flow in an edge e of $\mathcal{P}$. Edge e is the dual of the edge whose representative is replaced by U in constructing $\mathcal{G}$ from $\mathcal{F}$. The significant contribution of U to a face f is equal to the amount of the flow coming through e into the end point (of e) that corresponds to f.
- The balance of the significant contributions of the tendrils and wiggles to the faces of $\mathcal{G}$ corresponds to the conservation of flow at the corresponding vertices of $\mathcal{P}$, i.e., the total significant contribution of the tendrils and wiggles to a face f is zero if and only if there is a conservation of flow at its corresponding vertex in $\mathcal{P}$.

From Theorem 3.5, Lemma 5.4, and the correspondence established above between a feasible flow in $\mathcal{P}$ and the rectilinear planarity of $\mathcal{G}$, it follows that instance $\mathcal{S}$ is satisfiable and $\mathcal{P}$ admits a feasible flow if and only if graph $\mathcal{G}$ is rectilinear planar. It also follows that given a rectilinear planar embedding for $\mathcal{G}$, a satisfying truth assignment for $\mathcal{S}$ can be computed in time $O(n^4m^3)$. ∎

From Theorem 5.5 we conclude the following corollaries.

COROLLARY 5.6. *Rectilinear planarity testing is NP-complete.*

COROLLARY 5.7. *Computing a planar orthogonal drawing with the minimum number of bends is NP-hard.*

We can strengthen Corollary 5.7 as follows.

COROLLARY 5.8. *Let G be an n-vertex planar graph whose minimum number of bends in any planar orthogonal drawing is b^*. Computing a planar orthogonal drawing of G with $O(b^* + n^{1-\epsilon})$ bends is NP-hard for $\epsilon > 0$.*

Proof. Suppose there is a polynomial-time algorithm A that computes a planar orthogonal drawing of G with at most $c(b^* + n^{1-\epsilon})$ bends, where c is some constant. We can then use algorithm A to test in polynomial time whether graph G is rectilinear planar as follows. Construct a graph G' consisting of $K = \lceil (cn^{1-\epsilon})^{1/\epsilon} \rceil + 1$ copies of

G and give G' as an input to algorithm A. Clearly, G is rectilinear planar if and only if G' has a planar orthogonal drawing with fewer than K bends. Since G' has Kn vertices and $K > c(Kn)^{1-\epsilon}$, algorithm A computes a drawing of G' with fewer than K bends if and only if G is rectilinear planar. $\square$

6. Conclusions. Finding efficient algorithms for upward and rectilinear planarity testing had been an open problem for many years. In this paper we have shown that a polynomial-time algorithm for either of these problems is unlikely to exist by proving that both problems are NP-complete. NP-completeness of rectilinear planarity testing also implies that the bend-minimization problem for planar orthogonal drawings is NP-hard.

Acknowledgments. We would like to thank Giuseppe Di Battista for useful discussions and Ivan Rival for comments on a preliminary version of this paper.

REFERENCES

[1] P. Bertolazzi, R. F. Cohen, G. Di Battista, R. Tamassia, and I. G. Tollis, *How to draw a series-parallel digraph*, Internat. J. Comput. Geom. Appl., 4 (1994), pp. 385–402.

[2] P. Bertolazzi and G. Di Battista, *On upward drawing testing of triconnected digraphs*, in *Proceedings of the 7th Annual Symposium on Computational Geometry*, North Conway, NH, 1991, pp. 272–280.

[3] P. Bertolazzi, G. Di Battista, G. Liotta, and C. Mannino, *Upward drawings of triconnected digraphs*, Algorithmica, 6 (1994), pp. 476–497.

[4] P. Bertolazzi, G. Di Battista, C. Mannino, and R. Tamassia, *Optimal upward planarity testing of single-source digraphs*, SIAM J. Comput., 27 (1998), pp. 132–169.

[5] T. Biedl and G. Kant, *A better heuristic for orthogonal graph drawings*, Comput. Geom., 9 (1998), pp. 159–180.

[6] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Algorithms for drawing graphs: An annotated bibliography*, Comput. Geom., 4 (1994), pp. 235–282.

[7] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph Drawing*, Prentice Hall, Upper Saddle River, NJ, 1999.

[8] G. Di Battista, G. Liotta, and F. Vargiu, *Spirality and optimal orthogonal drawings*, SIAM J. Comput., 27 (1998), pp. 1764–1811.

[9] G. Di Battista, W. P. Liu, and I. Rival, *Bipartite graphs upward drawings and planarity*, Inform. Process. Lett., 36 (1990), pp. 317–322.

[10] G. Di Battista and R. Tamassia, *Algorithms for plane representations of acyclic digraphs*, Theoret. Comput. Sci., 61 (1988), pp. 175–198.

[11] G. Di Battista, R. Tamassia, and I. G. Tollis, *Area requirement and symmetry display of planar upward drawings*, Discrete Comput. Geom., 7 (1992), pp. 381–401.

[12] S. Even and G. Granot, *Rectilinear Planar Drawings with Few Bends in Each Edge*, Technical report 797, Computer Science Department, Technion, Haifa, Israel, 1994.

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

[14] M. D. Hutton and A. Lubiw, *Upward planar drawing of single-source acyclic digraphs*, SIAM J. Comput., 25 (1996), pp. 291–311.

[15] G. Kant, *Drawing planar graphs using the canonical ordering*, Algorithmica, 16 (1996), pp. 4–32.

[16] D. Kelly, *Fundamentals of planar ordered sets*, Discrete Math., 63 (1987), pp. 197–216.

[17] D. Kelly and I. Rival, *Planar lattices*, Canad. J. Math., 27 (1975), pp. 636–665.

[18] A. Lempel, S. Even, and I. Cederbaum, *An algorithm for planarity testing of graphs*, in Theory of Graphs, Gordon and Breach, New York, 1967, pp. 215–232.

[19] Y. Liu, P. Marchioro, and R. Petreschi, *At most single-bend embeddings of cubic graphs*, Appl. Math. J. Chinese Univ. Ser. B, 9 (1994), pp. 127–142.

[20] Y. Liu, P. Marchioro, R. Petreschi, and B. Simeone, *Theoretical Results on at Most 1-Bend Embeddability of Graphs*, Technical report, Dipartimento di Statistica, Università di Roma "La Sapienza," Rome, Italy, 1990.

[21] Y. Liu, A. Morgana, and B. Simeone, *General theoretical results on rectilinear embeddability of graphs*, Acta Math. Appl. Sinica, 7 (1991), pp. 187–192.

[22] Y. Liu, A. Morgana, and B. Simeone, *A Linear Algorithm for* 3*-Bend Embeddings of Planar Graphs in the Grid*, manuscript, 1993.

[23] A. Papakostas, *Upward planarity testing of outerplanar dags*, in Graph Drawing, R. Tamassia and I. G. Tollis, eds., Lecture Notes in Comput. Sci. 894, Springer-Verlag, Berlin, 1995, pp. 298–306.

[24] C. Platt, *Planar lattices and planar graphs*, J. Combin. Theory Ser. B, 21 (1976), pp. 30–39.

[25] I. Rival, *The diagram*, in Graphs and Order, I. Rival, ed., Reidel, Dordrecht, the Netherlands, 1985, pp. 103–133.

[26] I. Rival, *Graphical data structures for ordered sets*, in Algorithms and Order, I. Rival, ed., Kluwer Academic Publishers, Dordrecht, the Netherlands, 1989, pp. 3–31.

[27] I. Rival, *Reading, drawing, and order*, in Algebras and Orders, I. G. Rosenberg and G. Sabidussi, eds., Kluwer Academic Publishers, Dordrecht, the Netherlands, 1993, pp. 359–404.

[28] Y. Shiloach, *Arrangements of Planar Graphs on the Planar Lattice*, Ph.D. thesis, Weizmann Institute of Science, Rehovot, Israel, 1976.

[29] J. A. Storer, *On minimal node-cost planar embeddings*, Networks, 14 (1984), pp. 181–212.

[30] R. Tamassia, *On embedding a graph in the grid with the minimum number of bends*, SIAM J. Comput., 16 (1987), pp. 421–444.

[31] R. Tamassia and I. G. Tollis, *Planar grid embedding in linear time*, IEEE Trans. Circuits Systems, 36 (1989), pp. 1230–1234.

[32] R. Tamassia and J. S. Vitter, *Parallel transitive closure and point location in planar structures*, SIAM J. Comput., 20 (1991), pp. 708–725.

[33] C. Thomassen, *Planar acyclic oriented graphs*, Order, 5 (1989), pp. 349–361.

[34] L. Valiant, *Universality considerations in VLSI circuits*, IEEE Trans. Comput., 30 (1981), pp. 135–140.

[35] G. Vijayan and A. Wigderson, *Rectilinear graphs and their embeddings*, SIAM J. Comput., 14 (1985), pp. 355–372.