Theoretical Computer Science

ELSEVIER Theoretical Computer Science 172 (1997) 175–193

Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems [1]

Goos Kant [a], Xin He [b,*]

[a] *Department of Computer Science, Utrecht University, Padualaan 14, 3584 CH Utrecht, The Netherlands*

[b] *Department of Computer Science, State University of New York at Buffalo, Buffalo, NY 14260, USA*

Received May 1995; revised November 1995
Communicated by O.H. Ibarra

Abstract

In this paper we extend the concept of the *regular edge labeling* for general plane graphs and for triconnected triangulated plane graphs to 4-connected triangulated plane graphs. We present two different linear time algorithms for constructing such a labeling. By using regular edge labeling, we present a new linear time algorithm for constructing *rectangular dual* of planar graphs. Our algorithm is simpler than previously known algorithms. The coordinates of the rectangular dual constructed by our algorithm are integers, while the one constructed by known algorithms are real numbers.

Our second regular edge labeling algorithm is based on *canonical ordering* of 4-connected triangulated plane graphs. By using this technique, we present a new algorithm for constructing *visibility representation* of 4-connected planar graphs. Our algorithm reduces the size of the representation by a factor of 2 for such graphs.

1. Introduction

The problem of "nicely" drawing a graph G on the plane has received increasing attention [4]. Typically, we want to draw the edges and the vertices of G so that certain aesthetic quality conditions and/or optimization measures are met. Such drawings are very useful in visualizing planar graphs and find applications in fields such as computer graphics, VLSI layout, algorithm animation, visual languages and so on.

A powerful method, *regular edge labeling*, was used successfully in solving several planar graph drawing problems. Roughly speaking, for a given plane graph $G = (V, E)$,

* Corresponding author. E-mail: xinhe@cs.buffalo.edu.

[1] The work of the first author was supported by the ESPRIT Basic Research Actions program of the EC under contract No. 7141 (project ALCOM II). The work of the second author was partially supported by National Science Foundation, grant CCR-9205982.

SSDI 0304-3975(95)00257-X

a regular edge labeling of G partitions and orients the edges of G such that the edges around every vertex v show certain regular patterns. The drawing of G is then produced by using the combinatorial structures resulting from the labeling. For an example, it was shown in [17] that the edges of every plane graph G can be oriented so that the edges incident to each vertex v are partitioned into two contiguous nonempty subsets around v: the edges in the first subset are leaving v; the edges in the second subset are entering v. (It is commonly called an *s–t orientation* of G). An elegant algorithm for solving the *visibility representation* problem was obtained by using this labeling [17]. For another example, it was shown in [18] that the edges of every triconnected triangulated plane graph G can be oriented and partitioned into three sets T_1, T_2, T_3 such that the edges incident to each vertex v display the following pattern in counterclockwise order around v: an edge in T_1 leaving v; a set (maybe empty) of edges in T_3 entering v; an edge in T_2 leaving v; a set (maybe empty) of edges in T_1 entering v; an edge in T_3 leaving v; a set (maybe empty) of edges in T_2 entering T_2 entering v. (It is called a *realizer* in [18].) By using this labeling, Schnyder found a beautiful algorithm that computes a *straight line embedding* of G on an $(n-2) \times (n-2)$ grid [19].

In this paper, we extend the regular edge labeling (REL) concept to 4-connected triangulated plane graphs. We present two linear time algorithms for finding an REL for such a graph G. The two algorithms use totally different approaches and both are of independent interests. The first algorithm is based on the *edge contraction* technique. A similar method was used in solving other planar graph problems (for examples, see [6, 19]). The second algorithm is based on the concept of *canonical ordering* for 4-connected triangulated plane graphs. This concept was first defined for triangulated plane graphs [5] and triconnected plane graphs [12]. We extend this concept to 4-connected triangulated plane graphs. We show that the canonical ordering and the REL are closely related and a canonical ordering algorithm leads to an REL algorithm.

As applications, we show that the regular edge labeling can be used to solve the *rectangular dual* problem and the canonical ordering can be used to solve the *visibility representation* problem for 4-connected plane graphs. In the following, we discuss these two problems in more details.

In design of floor planning of electronic chips and in architectural design, it is common to represent a graph G by a *rectangular dual* defined as follows. A *rectangular subdivision system* of a rectangle R is a partition of R into a set $\Gamma = \{R_1, R_2, \dots, R_n\}$ of nonoverlapping rectangles such that no four rectangles in Γ meet at the same point. A *rectangular dual* of a plane graph $G = (V, E)$ is a rectangular subdivision system Γ and a one-to-one correspondence $f : V \to \Gamma$ such that two vertices u and v are adjacent in G if and only if their corresponding rectangles $f(u)$ and $f(v)$ share a common boundary. In applications of this representation, the vertices of G represent circuit modules and the edges represent module adjacencies [10, 16]. A rectangular dual provides a placement of circuit modules that preserves the required adjacencies. Fig. 1 shows an example of a plane graph and its rectangular dual.

The rectangular dual problem has been studied in [1, 2, 14, 15]. Bhasker and Sahni gave an algorithm for solving the problem [2]. Although it runs in linear time, their

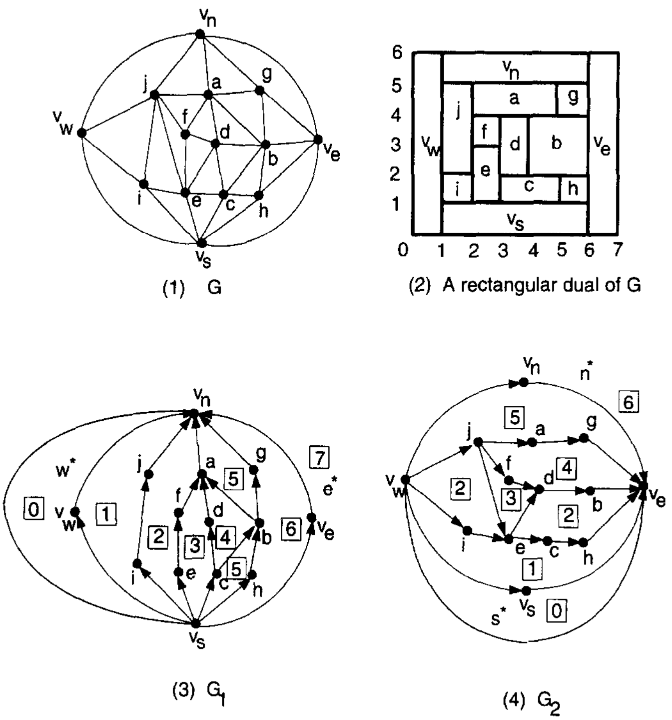

Fig. 1. A 4TP graph, its rectangular dual, and the *st*-graphs G_1 and G_2.

algorithm is fairly complicated. Moreover, the coordinates of the rectangular dual constructed by the algorithm in [2] are real numbers and bear no clear relationship to the structure of the graph. This algorithm consists of two major steps: (1) constructing a so-called *path directed graph* for G; and (2) constructing the rectangular dual using the path directed graph. A simplification of step (2) is given in [7]. (A parallel implementation of this algorithm, working in $O(\log^2 n)$ time with $O(n)$ processors, is given in [8]). However, the step (1) still relies on the complicated algorithm in [2]. We will show that the path directed graph can easily be obtained from an REL of G. Combine our REL algorithms and the result in [7], we obtain a simple rectangular dual algorithm. The coordinates of the rectangular dual constructed by our algorithm are integers and carry clear combinatorial meaning.

The second graph drawing problem we consider is the *visibility representation.* Given a planar graph G, the representation maps each vertex of G into a horizontal line segment and each edge into a vertical line segment that only touches the two horizontal line segments representing its end vertices [17,20]. This gives a nice and readable drawing of G. This representation has been applied in several industrial applications for representing electrical diagrams and schemes [20]. Linear time algorithms for solving this problem has been independently discovered by Rosenstiehl and Tarjan [17] and Tamassia and Tollis [20]. The size of the drawing is $(2n-5) \times (n-2)$ for both algorithms. Our visibility representation algorithm uses the same ideas as in [17,20].

However, we show that, by using the canonical ordering, the size of the drawing can be reduced to $(n-1)\times(n-1)$ for 4-connected planar graphs. Using this result, Kant recently proved that a visibility representation of a general planar graph can be drawn on a grid of size at most $(\lfloor\frac{3}{2}n\rfloor-3)\times(n-1)$ [13].

Although linear time algorithms for solving both problems are previously known, the techniques used in our algorithms are completely different. The simplicity of our approach is a significant factor for applications. The regular edge labeling can also be used to solve other graph drawing problems. For example, it is recently shown in [9], by using this method, that every 4-connected plane graph has a straight line embedding on a $W+H$ grid with $W+H\leqslant n$; $W\leqslant(n+3)/2$ and $H\leqslant 2(n-1)/3$.

The present paper is organized as follows. Section 2 presents the definition of the regular edge labeling of 4-connected plane graphs and describes the algorithm in [7] that computes a rectangular dual from an REL. In Section 3, we present the edge contraction based algorithm for computing an REL. In Section 4, we present the second REL algorithm based on the canonical ordering. Section 5 discusses the algorithm for the visibility representation. Section 6 concludes the paper.

2. Regular edge labeling of 4-connected plane graphs

Let $G=(V,E)$ be a planar graph with n vertices and m edges. We use $N(v)$ to denote the set of neighbors of v. Let $\deg(v)=|N(v)|$. We assume G is equipped with a fixed plane embedding. The unbounded face is the *exterior face*. Other faces are *interior faces*. The vertices and the edges on the boundary of the exterior face are called *exterior vertices* and *exterior edges*. Other vertices and edges are *interior vertices* and *interior edges*. In this paper, the terms *path* and *cycle* always mean *simple path* and *simple cycle* (i.e. the vertices of path and cycle are distinct). A *triangle* is a cycle of 3 edges. A *quadrangle* is a cycle of 4 edges. A cycle C of G divides the plane into its interior and exterior regions. If C contains at least one vertex in its interior and exterior. C is called a *separating cycle*.

We assume the embedding information of G is given by the following data structure. For each $v\in V$, there is a doubly linked circular list $Adj(v)$ containing all vertices of $N(v)$ in counterclockwise order. The two copies of an edge (u,v) are cross-linked to each other. This representation can be constructed as a by-product by using a planarity testing algorithm in linear time (e.g. [11]).

A *triangulated plane graph* is a plane graph all of whose interior faces are triangles. Consider an interior vertex v of such a graph G. If $N(v)=\{u_1,\dots,u_k\}$ are in counterclockwise order around v in the embedding, then the edges $(u_1,u_2),\dots,(u_{k-1},u_k)$, (u_k,u_1) form a cycle, which will be denoted by $Cycle(v)$. The *star* at v, denoted by $Star(v)$, is the set of the edges $\{(v,u_i)\mid 1\leqslant i\leqslant k\}$. Consider a graph G satisfying the following conditions:

(1) every interior face of G is a triangle and the exterior face is a quadrangle;

(2) G is 4-connected.

Remark 1. The first condition can easily be checked in linear time using the embedding information of G. If G satisfies condition 1, condition 2 is equivalent to that G contains no separating triangles. To test condition 2, we first enumerate all triangles of G by using the linear time algorithm in [3]. Then for each triangle C of G listed, we check if C is a separating triangle or not. This takes $O(1)$ time per triangle. So the total time needed is $O(n)$.

A graph G satisfying above two conditions will be called a 4TP graph. Note condition (2) implies that G has no separating triangles. Since G has no separating triangles, the degree of any interior vertex v of G is at least 4. (If $\deg(v) = 3$ for some v, then deleting the three neighbors of v would disconnect G. This contradicts the assumption that G is 4-connected).

Definition 2.1. A regular edge labeling of a 4TP graph G is a partition and an orientation of the interior edges of G into two subsets T_1, T_2 of directed edges such that:

1. For each interior vertex v, the edges incident to v appear in counterclockwise order around v as follows: a set of edges in T_1 leaving v; a set of edges in T_2 entering v; a set of edges in T_1 entering v; a set of edges in T_2 leaving v.
2. Let v_N, u_W, v_S, v_E be the four exterior vertices in counterclockwise order. All interior edges incident to v_N are in T_1 and entering v_N. All interior edges incident to v_W are in T_2 and leaving v_W. All interior edges incident to v_S are in T_1 and leaving v_S. All interior edges incident to v_E are in T_2 and entering v_E.

In the next two sections, we will show that every 4TP graph has an REL and that REL can be constructed in linear time. In the following, we briefly describe the algorithm in [7] that constructs a rectangular dual from an REL.

Consider a plane graph $H = (V,E)$. Let u_0, u_1, u_2, u_3 be four vertices on the exterior face in counterclockwise order. Let P_i $(i = 0,1,2,3)$ be the path on the exterior face consisting of the vertices between u_i and u_{i+1} (addition is mod 4). We seek a rectangular dual R_H of H such that u_0, u_1, u_2, u_3 correspond to the four corner rectangles of R_H and the vertices on P_0 (P_1, P_2, P_3, respectively) correspond to the rectangles located on the north (west, south, east, respectively) boundary of R_H. In order to simplify the problem, we modify H as follows: Add four new vertices v_N, v_W, v_S, v_E. Connect v_N (v_W, v_S, v_E, respectively) to every vertex on P_0 (P_1, P_2, P_3, respectively) and add four new edges $(v_S, v_W), (v_W, v_N), (v_N, v_E), (v_E, v_S)$. Let G be the resulting graph. It is easy to see that H has a rectangular dual R_H if and only if G has a rectangular dual R_G with exactly four rectangles on the boundary of R_G (see Fig. 1(1) and (2)). The following theorem was proved in [1, 14].

Theorem 2.1. *A plane graph G has a rectangular dual R with four rectangles on the boundary of R if and only if G is a 4TP graph.*

The regular edge labeling is closely related to *planar st-graphs*. A planar *st*-graph G is a directed plane graph with exactly one source (in-degree 0) vertex s and exactly

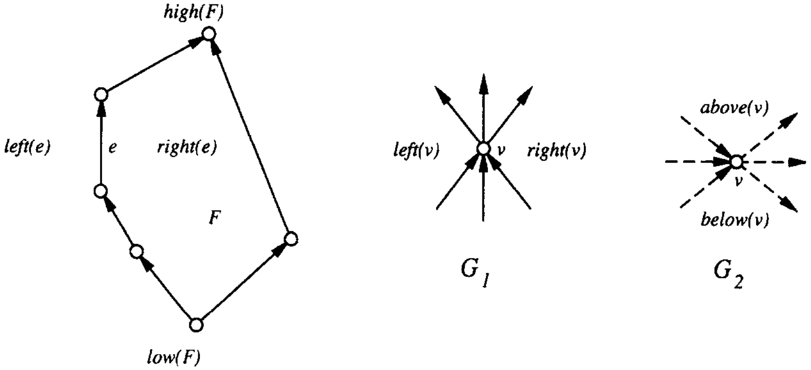

Fig. 2. Properties of planar *st*-graphs.

one sink (out-degree 0) vertex t such that both s and t are on the exterior face and are adjacent. Let G be a planar *st*-graph. For each vertex v, the incoming edges of v appear consecutively around v, and so do the outgoing edges of v. The boundary of every face F of G consists of two directed paths with a common origin, called *low*(F), and a common destination, called *high*(F). The face on the left (right, respectively) side of an edge e is denoted by *left*(e) (*right*(e), respectively) (see Fig. 2).

Let G be a 4TP graph and $\{T_1, T_2\}$ be an REL of G. From $\{T_1, T_2\}$, we can construct two planar *st*-graphs as follows. Let G_1 be the graph consisting of the edges of T_1 plus the four exterior edges (directed as $v_S \to v_W, v_W \to v_N, v_S, v_E \to v_N$), and a new edge (v_S, v_N). G_1 inherits a plane embedding from G. Then G_1 is a planar *st*-graph with source v_S and sink v_N. For each vertex v, the face of G_1 that separates the incoming edges of v from the outgoing edges of v in clockwise direction is denoted by *left*(v). The face of G_1 that separates the incoming and the outgoing edges of v is denoted by *right*(v) (see Fig. 2).

Let G_2 be the graph consisting of the edges of T_2 plus the four exterior edges (directed as $v_W \to v_S, v_S \to v_E, v_W \to v_N, v_N \to v_E$), and a new edge (v_W, v_E). G_2 inherits a plane embedding from G. Then G_2 is a planar *st*-graph with source v_W and sink v_E. For each vertex v, the face of G_2 that separates the incoming edges of v from the outgoing edges of v in clockwise direction is denoted by *above*(v). The face of G_2 that separates the incoming and the outgoing edges of v is denoted by *below*(v) (See Fig. 2).

The dual graph G_1^* of G_1 is defined as follows. Every face F_k of G_1 is a node v_{F_k} in G_1^* and there exists an edge (v_{F_i}, v_{F_k}) in G_1^* if and only if F_i and F_k share a common edge in G_1. The edges of G_1^* are directed as follows: if F_l and F_r are the left and the right face of an edge (v, w) of G_1, direct the dual edge from F_l to F_r if $(v, w) \neq (v_S, v_N)$ and from F_r to F_l if $(v, w) = (v_S, v_N)$. G_1^* is a planar *st*-graph whose source and sink are the right face (denoted by w^*) and the left face (denoted by $e^{*}*$) of (v_S, v_N), respectively. For each node F of G_1^* let $d_1(F)$ denote the length of the longest path from w^* to F. Let $D_1 = d_1(e^*)$. For each interior vertex v, define: $x_{\text{left}}(v) = d_1(\text{left}(v))$, and $x_{\text{right}}(v) = d_1(\text{right}(v))$. For the four exterior vertices, define: $x_{\text{left}}(v_W) = 0; x_{\text{right}}(v_W) = 1; x_{\text{left}}(v_E) = D_1 - 1; x_{\text{right}}(v_E) = D_1$; $x_{\text{left}}(v_S) = x_{\text{left}}(v_N) = 1; x_{\text{right}}(v_S) = x_{\text{right}}(v_N) = D_1 - 1$.

The dual graph G_2^* of G_2 is defined similarly. For each node F of G_2^*, let $d_2(F)$ denote the length of the longest path from the source node of G_2^* to F. Let D_2 be the length of the longest path from the source node to the sink node of G_2^*. For each interior vertex v, define: $y_{\text{low}}(v) = d_2(below(v))$, and $y_{\text{high}}(v) = d_2(above(v))$. For the four exterior vertices, define: $y_{\text{low}}(v_W) = y_{\text{low}}(v_E) = 0$; $y_{\text{high}}(v_W) = y_{\text{high}}(v_E) = D_2$; $y_{\text{low}}(v_S) = 0$; $y_{\text{high}}(v_S) = 1$; $y_{\text{low}}(v_N) = D_2 - 1$; $y_{\text{high}}(v_N) = D_2$.

The following theorem was proved in [7].

Theorem 2.2. *Let G be a 4TP graph and $\{T_1, T_2\}$ be an REL of G. For each vertex v of G, assign v the rectangle $f(v)$ bounded by the four lines $x = x_{\text{left}}(v)$, $x = x_{\text{right}}(v)$, $y = y_{\text{low}}(v)$, $y = y_{\text{high}}(v)$. Then the set $\{f(v) \mid v \in V\}$ form a rectangular dual of G. If the REL is given, the rectangular dual can be constructed in linear time.*

Fig. 1 shows an example of this construction. Fig. 1(3) shows the *st*-graph G_1. The small boxes in the figure represent the nodes of G_1^* and the integers in the boxes are their d_1 values. Fig. 1(4) shows the graph G_2. Fig. 1(2) shows the rectangular dual constructed as in Theorem 2.2.

3. REL algorithm based on edge contraction

In this section, we present our first algorithm for computing an REL of a 4TP graph G. The basic technique is *edge contraction* and *edge expansion*. We begin with the definition of edge contraction. Let $e = (v, u)$ be an interior edge of G. Let C_1 and C_2 be the two faces with e as the common boundary. Let e_1 and e_2 be the other two edges and y the third vertex of C_1. Let e_3 and e_4 be the two other edges and z the third vertex of C_2 (see Fig. 3). The operation of *contracting* e deletes e and merges u and v into a new vertex o_e. The edges incident to u and v (except e_1, e_2, e_3, e_4) are incident to the new vertex o_e in the resulting graph. Replace e_1 and e_2 by a new edge (y, o_e). Replace e_3 and e_4 by a new edge (z, o_e). The resulting *contracted graph* is denoted by G/e. The edges e_1, e_2, e_3, e_4 are called the *surrounding edges* of e. The edges (y, o_e) and (z, o_e) are called the *residual edges* of e.

The graph $G' = G/e$ has a plane embedding inherited from the embedding of G. Since G is a 4TP graph, e is not on any separating triangle. Thus G' has no multiple edges. It is easy to see that G' with the inherited embedding is a triangulated plane graph. If e is on a separating quadrangle of G, then G' has a separating triangle. If e is **not** on any separating quadrangle of G, it is called a *contractible edge*. For any contractible edge e, G/e is a 4TP graph.

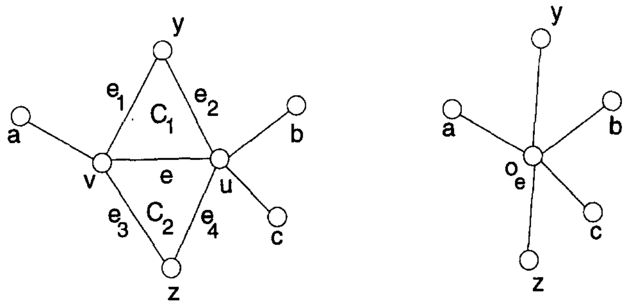

Fig. 3. Edge contraction.

The following equivalent definition of contractible edges is useful in our discussion. Consider a vertex v and a neighbor u of v. Let y and z be the two neighbors of v that are consecutive with u in $N(v)$ (see Fig. 3). The edge (u,v) is contractible if and only if for any neighbor $x (x \neq u,y,z)$ of v, the only common neighbor of u and x is v. In this case, u is called a *contractible neighbor* of v.

Lemma 3.1. *Let G be a* 4TP *graph and v be an interior vertex of G. If $deg(v) = 4$, then v has at least two contractible neighbors. If $deg(v) = 5$, then v has at least one contractible neighbor.*

Proof. Suppose $deg(v) = 4$. Let u_0, u_1, u_2, u_3 be the four neighbors of v in counterclockwise order. If u_0 and u_2 have no common neighbors other than v, then both of them are contractible. Suppose u_0 and u_2 share a common neighbor $w \neq v$. If u_1 and u_3 share no common neighbors other than v, then both of them are contractible. Suppose u_1 and u_3 share a common neighbor $z \neq v$. By the planarity of G, we have $w = z$ and G has a separating triangle (i.e. $\{u_0, u_3, w\}$). This contradicts that G is a 4TP graph. Similarly, we can show each degree-5 vertex has at least one contractible neighbor. □

Let e be a contractible edge of a 4TP graph G. Suppose an REL $\{T_1', T_2'\}$ of the contracted graph $G' = G/e$ has been found. Then we can *expand* e and obtain an REL $\{T_1, T_2\}$ of G from $\{T_1', T_2'\}$ as follows. Let e_1, e_2, e_3, e_4 be the surrounding edges of e. For any edge e' of G that is not e and not a surrounding edge of e, the label of e' with respect to $\{T_1, T_2\}$ is the same as its label with respect to $\{T_1', T_2'\}$. We need to specify proper labels of e, e_1, e_2, e_3, e_4 with respect to $\{T_1, T_2\}$. Depending on the labels of the edges in $Star(o_e)$ with respect to $\{T_1', T_2'\}$, there are six cases (up to rotating the edges around o_e) as shown in Fig. 4. These figures shows the labels of relevant edges before and after the expansion.

We assume (o_e, y) is in T_1' and directed as $o_e \to y$. Other cases are similar by rotating the edges in $Star(o_e)$. Consider the label of (o_e, z) with respect to $\{T_1', T_2'\}$. If $z \to o_e \in T_1'$, the situation is shown in Fig. 4(1). The case $o_e \to z \in T_1'$ is shown in Fig. 4(2). Suppose $o_e \to z \in T_2'$. Let (o_e, x) be the first edge in $Star(o_e)$ following

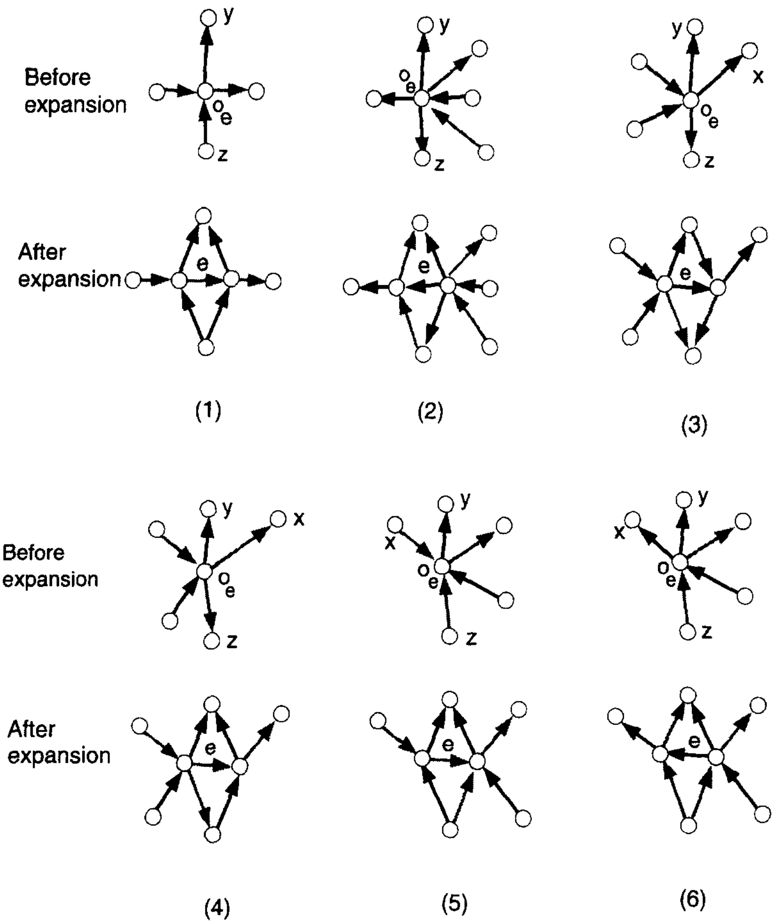

Fig. 4. Edge expansion.

(o_e, y) in clockwise order. Depending on the label of (o_e, x) with respect to $\{T_1', T_2'\}$, there are two cases as shown in Figs. 4(3) and (4). Suppose $z \to o_e \in T_2'$. Let (o_e, x) be the first edge in $Star(o_e)$ following (o_e, y) in counterclockwise order. Depending on the label of (o_e, x) with respect to $\{T_1', T_2'\}$ there are two cases as shown in Figs. 4(5) and (6). Note that the conditions of the six cases are completely determined by the labels of at most six edges in $Star(o_e)$: the two residue edges $(o_e, y), (o_e, z)$ and the four edges that are consecutive with $(o_e, y), (o_e, z)$ in $Star(o_e)$.

The basic idea of our algorithm is as follows. Recall that the 4-connectivity of G implies the minimum degree of G is at least 4. On the other hand, the minimum degree of any planar graph is at most 5. So the minimum degree of G is either 4 or 5. We pick a degree-4 or a degree-5 vertex v and select a contractible neighbor u of v. Then contract $e = (v, u)$ and recursively find an REL for the graph $G' = G/e$. Finally, expand e to obtain an REL for G. In order to find the contractible neighbors of v, however, we need to check, for each pair u and w of v's neighbors, if u and w share a common neighbor (other than v) or not. Since the degree of u and w can be large, this checking might be too expensive. In order to achieve linear time, we will only consider special *good* degree-4 and degree-5 vertices defined as follows. Let $V_i = \{v \in V \mid deg(v) = i\}$ and $V_{[i,j]} = \{v \in V \mid i \leqslant deg(v) \leqslant j\}$. Define $n_i = |V_i|$ and $n_{[i,j]} = |V_{[i,j]}|$. The vertices in $V_{[4,19]}$ are called *light* vertices. The vertices in $V_{[20,\infty)}$ are called *heavy* vertices. A degree-5 vertex v is *good* if v has at most one heavy neighbor. A degree-4 vertex v is *good* if either v has at most one heavy neighbor, or v has two heavy neighbors which are not consecutive in $N(v)$.

Lemma 3.2. *Any* 4TP *graph* $G = (V,E)$ *with at least one heavy vertex has at least* 7 *good vertices.*

Proof. Since the exterior face of G is a quadrangle and all interior faces of G are triangles, we have $|E| = 3n - 7$ by Euler's formula. Hence, $4n_4 + 5n_5 + 6n_{[6,19]} + 20n_{[20,\infty)} \leqslant \sum_{4 \leqslant i} in_i = \sum_{v \in V} deg(v) = 2|E| = 6n - 14 = 6(n_4 + n_5 + n_{[6,19]} + n_{[20,\infty)}) - 14$.
This gives: $14n_{[20,\infty)} + 2n_6 \leqslant 2n_4 + n_5 + 2n_6 - 14 \leqslant 2n - 14$. Hence,

$$7n_{[20,\infty)} + n_6 \leqslant n - 7. \tag{1}$$

Let p_4 (p_5, respectively) be the number of good degree-4 (degree-5, respectively) vertices. There are $n_4 - p_4$ bad degree-4 vertices and $n_5 - p_5$ bad degree-5 vertices. Define $S = \sum_{v \in V_{[20,\infty)}} deg(v)$. Since each bad degree-5 vertex v has at least two heavy neighbors, it contributes at least 2 to S. Consider a bad degree-4 vertex v. If v has at least three heavy neighbors, then v contributes at least 3 to S. Suppose v has two heavy neighbors u and w which are consecutive in $N(v)$. The edges (v,u) and (v,w) contribute 2 to S. The edge (u,w) also contributes 2 to S. But since (u,w) is shared with one other face, just half of the contribution can be apportioned to v. It follows that the contribution of v to S is at least 3. Thus $3(n_4 - p_4) + 2(n_5 - p_5) \leqslant S$, which gives $3n_4 + 2n_5 - (3p_4 + 2p_5) \leqslant \sum_{v \in V_{[20,\infty)}} deg(v)$. This in turn implies:

$$\begin{aligned}
&3n_4 + 2n_5 + (4n_4 + 5n_5) + 6n_6 + 7n_{[7,19]} - (3p_4 + 2p_5)\\
&\quad \leqslant 3n_4 + 2n_5 + \sum_{v \in V_{[4,5]}} deg(v) + \sum_{v \in V_{[6,19]}} deg(v) - (3p_4 + 2p_5)\\
&\quad \leqslant \sum_{v \in V} deg(v) = 2|E| = 6n - 14 = 6(n_4 + n_5 + n_6 + n_{[7,19]} + n_{[20,\infty)}) - 14.
\end{aligned}$$

Simplifying this inequality yields: $n_4 + n_5 + n_{[7,19]} - (3p_4 + 2p_5) \leqslant 6n_{[20,\infty)} - 14$. Hence,

$$3p_4 + 2p_5 \geqslant n - (n_6 + 7n_{[20,\infty)}) + 14. \tag{2}$$

From (1) and (2): $3(p_4 + p_5) \geqslant 3p_4 + 2p_5 \geqslant n - (n - 7) + 14 = 21$. This proves the lemma. $\square$

We are now ready to present our first REL construction algorithm.

Algorithm 1. REL (Input: A 4TP graph $G = (V, E)$).

1. Compute the degrees of the vertices of G.
2. Collect all good degree-4 and degree-5 interior vertices into a list L.
3. $i \leftarrow n$.
4. While G has more than one interior vertex do:
   4.1. Remove a vertex v from L. Mark v as w_i. Decrease i by 1. Record the neighborhood structure of v.
   4.2. Find a contractible neighbor u of v. Contract the edge (v, u). (The new vertex is still denoted by u). Modify the adjacency lists and the degrees of the vertices affected by the contraction. If any of the affected vertices becomes a good vertex, put it into L.

   End While (the last marked vertex is w_6).
5. G has only one interior vertex now. Construct the trivial REL for G.
6. For $i = 6$ to n do:

   Expand the edge e associated with w_i. Compute the REL of the graph with e expanded from the REL of the graph with e contracted.

Theorem 3.3. *Algorithm* 1 *computes an* REL *of a* 4TP *graph in* $O(n)$ *time.*

Proof. The correctness of the algorithm follows from the above discussion. We only need to analyze its complexity. Step 1 clearly takes $O(n+m) = O(n)$ time. Since good vertices have degree at most 5, each of them can be determined and put into L in $O(1)$ time. By Lemma 3.2, L will never be empty during the execution of the while loop.

Since the degree of a good vertex v is at most 5, the neighborhood structure of v can be recorded in $O(1)$ time. Other operations of Step 4.1 can be easily done in $O(1)$ time also.

The only nontrivial part is Step 4.2. We need to find a contractile neighbor of v in $O(1)$ time. Suppose $deg(v) = 5$ and u_i $(0 \leqslant i \leqslant 4)$ are v's neighbors. If v has no heavy neighbor or has one heavy neighbor (say u_0), we can check, for each pair u_i, u_j $(1 \leqslant i, j \leqslant 4)$, if they share a common neighbor. Since the degrees of u_i and u_j are bounded by 19, this takes $O(1)$ time. If none of the u_i $(1 \leqslant i \leqslant 4)$ is contractible, then u_0 is contractible by Lemma 3.1. Now suppose $deg(v) = 4$ with neighbors u_0, u_1, u_2, u_3. If v has at most one heavy neighbor, the situation is the same as the degree-5 case. If v has two heavy neighbors, then they are not consecutive in $N(v)$. Suppose they are u_0 and u_2. We can check if u_1 and u_3 share a common neighbor in $O(1)$ time. If u_1 and u_3 have no common neighbors, then both of them are contractible. Otherwise u_0 and u_2 are contractible.

After selecting a contractible neighbor u for v, the operation of contracting (v, u) affects the vertices in $N(v)$. The adjacency lists and the degrees of these vertices are modified. Since $deg(v) \leqslant 5$, this can be done in $O(1)$ time by using the cross-linked adjacency lists data structure. New good vertices can be detected and inserted into L in O(1) time.

Finally, the edge expansion only involves 5 edges adjacent to the corresponding contracted edge. This can be done in O(1) time by using the neighborhood structure recorded at Step 4.1. □

4. REL algorithm based on canonical ordering

In this section we consider 4-connected triangulated plane graphs (all of whose faces, including the exterior face, are triangles). We introduce the *canonical ordering* for such graphs, which is the basis of our second algorithm for finding an REL of a 4TP graph G. Note that adding an edge connecting two non-adjacent exterior vertices of G leads to a 4-connected triangulated plane graph. The applications of the canonical ordering to other classes of planar graphs have been studied in [5, 12].

4.1. Canonical ordering of 4-connected triangulated plane graphs

Definition 4.1. Let $G = (V, E)$ be a 4-connected triangulated plane graph with three exterior vertices u, v, w. A canonical ordering of G is a labeling of V by $\{v_1, v_2, \ldots, v_n\}$ such that $v_1 = u$, $v_2 = v$, $v_n = w$ and the following requirements are met for every $4 \leqslant k \leqslant n$:

1. The subgraph G_{k-1} of G induced by $v_1, v_2, \ldots, v_{k-1}$ is biconnected and the boundary of its exterior face is a cycle C_{k-1} containing the edge (u, v).
2. v_k is in the exterior face of G_{k-1}, and its neighbours in G_{k-1} form a subinterval of the path $C_{k-1} - \{(u, v)\}$ consisting of at least two vertices. If $k \leqslant n-2$, v_k has at least two neighbours in $G - G_{k-1}$.

Remark 2. For a general triangulated plane graph G (not necessarily 4-connected), a similar definition of canonical ordering of G was given in [5] where, in condition (2), the vertex v_k is required to have at least *one* neighbor in $G - G_{k-1}$. Our definition is stronger since we require that the vertex v_k has at least *two* neighbors in $G - G_{k-1}$.

Theorem 4.1. *Every 4-connected triangulated plane graph G has a canonical ordering.*

Proof. The ordering $v_n, v_{n-1}, \ldots, v_3$ are defined by reverse induction. The three exterior vertices u, v, w are numbered by v_1, v_2 and v_n. Let G_{n-1} be the subgraph of G after deleting v_n. By the 4-connectivity of G, the exterior face C_{n-1} of G_{n-1} is a cycle and G_{n-1} satisfies the conditions in Definition 4.1. Let v_{n-1} be the third vertex of the interior triangular face of G adjacent to the edge (v_2, v_n). Then $v_{n-1} \neq v_1$ is a vertex of C_{n-1} adjacent to both v_2 and v_n in G. By 4-connectivity, $G - \{v_n, v_{n-1}\}$ is biconnected and, hence, G_{n-2} satisfies the conditions in Definition 4.1.

Let $k < n-1$ be fixed and assume that v_i has been defined for every $i > k$ such that the subgraph G_i induced by $V - \{v_{i+1}, \ldots, v_n\}$ satisfies the conditions in Definition 4.1.

Let C_k denote the boundary of the exterior face of G_k. We will pick a vertex on C_k to be the next vertex v_k in the ordering. Note that v_k cannot be adjacent to a cord of C_k. (Otherwise the exterior face C_{k-1} of G_{k-1} would not be biconnected and this would violate condition 1 of Definition 4.1.)

Assume first that C_k has no interior chords. Suppose $v_1, c_1, \ldots, c_p, v_2$ are the vertices of C_k in this order between v_1 and v_2. Then it follows by the 4-connectivity of G that $p \geqslant 2$. If all vertices $c_1, \ldots, c_p$ have only one edge incident with the vertices in $G - G_k$, then since G is a triangulated graph, they are adjacent to the same vertex v_j for some j with $k < j < n$. In this case we also have $(v_1, v_j), (v_2, v_j) \in G$. But then $\{(v_1, v_j), (v_j, v_2), (v_2, v_1)\}$ would be a separating triangle. Hence at least one vertex, say c_α, has at least 2 neighbors in $G - G_k$. Define c_α to be the next vertex v_k in our ordering.

Next assume C_k has interior chords. Let (c_a, c_b) $(b > a + 1)$ be a chord such that $b - a$ is minimal. Note that $v_1, v_2 \notin \{c_{a+1}, \ldots, c_{b-1}\}$. If all vertices $c_{a+1}, \ldots, c_{b-1}$ have only one edge to the vertices in $G - G_k$, then since G is a triangulated graph, they are adjacent to the same vertex v_j with $k < j$, and we also have $(c_a, v_j), (c_b, v_j) \in G$. But then $\{(c_a, v_j), (v_j, c_b), (c_b, c_a)\}$ would be a separating triangle. Hence there is at least one vertex $c_\alpha, a < \alpha < b$, having at least two neighbors in $G - G_k$ and no incident chords. Define c_α to be the next vertex v_k in our ordering. □

Guided by the proof of Theorem 4.1, the following algorithm computes the canonical ordering of G. For each vertex v of G, we keep the following variables:

- *Mark*(v) = *true*, if v has been added to the ordering, and *false* otherwise.
- *Visited*(v) = the number of v's neighbors u with *Mark*(u) = *true*.
- *Chords*(v) = the number of chords of the exterior face of the subgraph induced by $V - \{u \in V \mid Mark(u) = true\}$ that is incident to v.

The algorithm is as follows.

Algorithm 2. Canonical Ordering (Input: A 4-connected triangulated plane graph $G = (V, E)$.)

1. Let $u = v_1, v = v_2$ and $w = v_n$ be the vertices on the exterior face.
2. Set *Chords*(x) and *Visited*(x) to 0, and *mark*(x) = *false* for all $x \in V$. Set *Visited*$(v_n) = 2$.
3. For $k \leftarrow n$ down to 3 do:
   - 3.1 Pick a vertex x with *Mark*(x) = *false*, *Visited*$(x) \geqslant 2$ and *Chords*$(x) = 0$, and $x \neq u, v$. Let $v_k = x$ and set *Mark*(x) := *true*.
   - 3.2 Let $c_i, \ldots, c_j$ be the neighbors of v_k (in this order around v_k) with *Mark*(c_l) = *false*.
   - 3.3 For each c_l $(i \leqslant l \leqslant j)$, increase *Visited*$(c_l)$ by one; update the variable *Chords* for c_l and the neighbors of c_l.

Theorem 4.2. *Algorithm 2 computes a canonical ordering of a 4-connected triangulated plane graph in* $O(n)$ *time.*

Proof. Let $G_k = G - \{v_{k+1}, \ldots, v_n\}$. By the proof of Theorem 4.1, there always exists a vertex x satisfying the conditions in Step 3.1. This proves the correctness of Algorithm 2.

To find the vertex x in Step 3.1, we maintain a list L containing all vertices satisfying the conditions in Step 3.1. Testing these conditions and adding or deleting a vertex from L requires O(1) time.

Step 3.3 is implemented as follows: If $j = i + 1$, then since G is a triangulated plane graph, there was a chord (c_i, c_j) of the exterior face of G_k, hence we decrease *Chords*(c_i) and *Chords*(c_j) by one, since (c_i, c_j) becomes part of the exterior face C_{k-1} of G_{k-1}. If $j > i+1$, then we compute *Chords*(c_l) for each $i < l < j$. Notice that if *Visited*$(z) > 0$, then z has a neighbor y with *Mark*(y) = *true*, hence by planarity, $z \in C_k$. For every vertex c_l $(i < l < j)$ we inspect its neighbors z. If *Visited*$(z) > 0$, and $z \neq c_{l-1}, c_{l+1}$, then (c_l, z) is a chord of the exterior face of G_{k-1}, and we increase *Chords*(c_l) and *Chords*(z) by one. This requires $O(deg(c_l))$ time in total. But since this is done only once for every vertex c_l, the total running time is $O(n)$. □

4.2. From a canonical ordering to an REL

To compute an REL of a 4TP graph G, we first add an edge connecting two non-adjacent exterior vertices of G. This gives a 4-connected triangulated plane graph G'. We compute a canonical ordering of G' and then delete the added edge. The four exterior vertices of G are now numbered as v_1, v_2, v_{n-1}, v_n, respectively. Next we show that an REL of G can be derived from the canonical ordering.

First, for each edge (v_i, v_j) of G, we direct it from v_i to v_j if $i < j$. Define the *basis-edge* of a vertex v_k to be the edge (v_l, v_k) for which $l < k$ is minimal. The vertex v_k has incoming edges from $c_i, \ldots, c_j$ belonging to the exterior face C_{k-1} of G_{k-1}, assuming in this order from left to right. We call c_i the *leftpoint* of v_k and c_j the *rightpoint* of v_k. Let $v_{k_1}, \ldots, v_{k_l}$ be the higher-numbered neighbors of v_k, in this order from left to right. We call (v_k, v_{k_1}) the *leftedge* and (v_k, v_{k_l}) the *rightedge*.

Lemma 4.3. *A basis-edge cannot be a leftedge or a rightedge.*

Proof. Assume the lemma is false. Suppose the leftedge (v_k, v_{k_1}) of v_k is the basis-edge of v_{k_1}. Thus v_k is the lowest-numbered neighbor of v_{k_1}. Since G is triangulated, there is an edge between the leftpoint of v_k, say v_i, with $i < k$, and v_{k_1}. But this contradicts the fact that (v_k, v_{k_1}) is the basis-edge of v_{k_1}. Analog follows for the rightedge. □

Lemma 4.4. *An edge is either a leftedge, a rightedge or a basis-edge.*

Proof. Fix a vertex v_k $(3 \leqslant i \leqslant n-2)$, we will show that each incoming edge of v_k is either a leftedge, or a rightedge, or a basis-edge. Suppose that the incoming edges of v_k are from $c_i, \ldots, c_j$, in this order from left to right. Let (c_α, v_k) be the basis-edge of v_k. All vertices c_l $(i < l < j)$ have at least two higher-numbered neighbors, one of them is v_k, the other one is adjacent to v_k in c_l's neighbor set $N(c_l)$. Hence it is either

c_{l-1} or c_{l+1}. Thus for each l ($i \leqslant l < \alpha$), c_{l+1} is the rightpoint of c_l. Similarly, for each l ($\alpha \leqslant l < j$), c_l is the leftpoint of c_{l+1}. Hence the edges (c_l, v_k) are rightedges for $i \leqslant l < \alpha$ and leftedges for $\alpha < l \leqslant j$. The edge (c_α, v_k) is a basis-edge. Similarly, we can show the lemma holds for the incoming edges of v_{n-1} and v_n. □

We construct an REL for G as follows: all leftedges belong to T_1, all rightedges belong to T_2. The basis-edge (c_α, v_k) of v_k belongs to: (a) T_1, if $\alpha = j$; (b) T_2 if $\alpha = i$; or (c) either T_1 or T_2, otherwise. (The four exterior edges belong to neither T_1 nor T_2).

Lemma 4.5. *$\{T_1, T_2\}$ forms a regular edge labeling for G.*

Proof. Let $v_{k_1}, \ldots, v_{k_d}$ be the outgoing edges of the vertex v_k ($3 \leqslant k \leqslant n-2$). It follows from Definition 4.1 that $d \geqslant 2$. Then (v_k, v_{k_1}) is the leftedge of v_k and is in T_1. (v_k, v_{k_d}) is the rightedge of v_k and is in T_2. The edges $(v_k, v_{k_2}), \ldots, (v_k, v_{k_{d-1}})$ are the basis-edges of $v_{k_2}, \ldots, v_{k_{d-1}}$, respectively. Let the vertex v_{k_β} ($1 \leqslant \beta \leqslant d$) be the highest-numbered neighbor of v_k. Then all vertices from v_{k_1} to v_{k_β} have a monotone increasing number, as well as the vertices from v_{k_d} to v_{k_β}. (Otherwise there was a vertex v_{k_l} such that $v_{k_{l-1}}$ and $v_{k_{l+1}}$ are numbered higher than v_{k_l}. But this implies that v_k is the only lower-numbered neighbor of v_{k_l}, which is a contraction with the canonical ordering of G.) Hence for every v_{k_l}, we have $k_{l-1} < k_l < k_{l+1}$ if $1 < l < \beta$; or $k_{l-1} > k_l > k_{l+1}$ if $\beta < l < d$. Thus, by the construction of T_1 and T_2, the edges (v_k, v_{k_l}) are in T_1 for $1 \leqslant l < \beta$, and are in T_2 for $\beta < l \leqslant d$. The edge (v_k, v_{k_β}) is arbitrarily added to either T_1 or T_2. This completes the proof that the edges appear in counterclockwise order around v_k as follows: a set of edges in T_2 entering v_k; a set of edges in T_1 entering v_k; a set of edges in T_2 leaving v_k; a set of edges in T_1 leaving v_k.

Let $v_{1_1}, \ldots, v_{1_d}$ be the higher numbered neighbors of v_1 from left to right. Then $v_{1_1} = v_n$ and $v_{1_d} = v_2$. By the argument described above, $(v_1, v_{1_2}), \ldots, (v_2, v_{1_{d-1}})$ belong to T_2. Similarly, all outgoing edges of v_2 belong to T_1. All incoming edges of v_{n-1} belong to T_2, and all incoming edges of v_n belong to T_1. This completes the proof. □

Since the construction of $\{T_1, T_2\}$ from the canonical numbering can be easily done in $O(n)$ time. Theorem 4.2 and Lemma 4.5 constitute our second linear time REL algorithm.

5. Algorithm for visibility representation

The *visibility representation* of a planar graph G maps the vertices of G to horizontal line segments and edges of G to vertical line segments such that, for each edge $e = (u, v)$, the vertical line segment corresponding to e touches only the two horizontal line segments corresponding to u and v (see Figure 5 for an example). A visibility representation of G with size $(2n-5)\times(n-2)$ can be constructed in linear time [17, 20]. In this section, we show that the canonical ordering can be used to construct a visibility

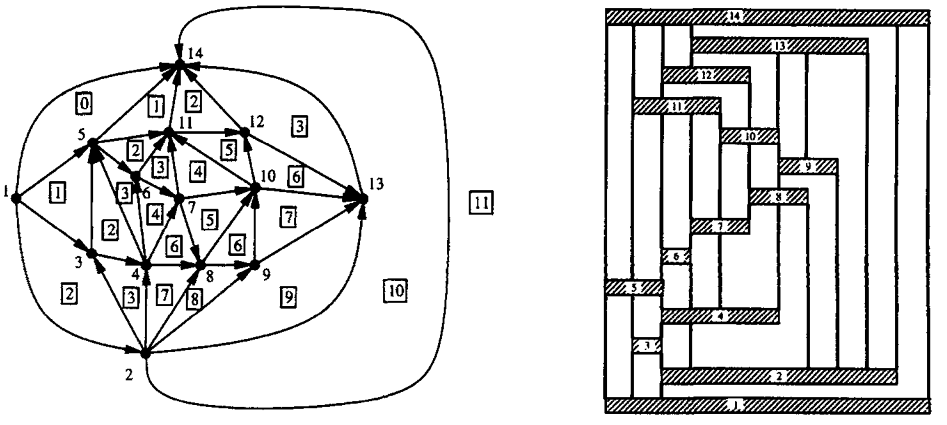

Fig. 5. The canonical ordering leads to a compact visibility representation.

representation of size $(n-1)\times(n-1)$ for 4-connected planar graphs. Thus, our algorithm reduces the width of the visibility representation by a factor of 2 for 4-connected planar graphs.

Let G be a 4-connected plane graph. Let G' be a plane triangulation of G. Clearly, G' is still 4-connected. After we obtain a visibility representation of G', we can delete the line segments corresponding to the added edges and obtain a visibility representation of G. From now on we assume, without loss of generality, that G is a 4-connected triangulated plane graph.

Suppose that the canonical ordering of G is $v_1, v_2, \ldots, v_n$. We direct an edge (v_i, v_j) of G as $v_i \to v_j$, if $i < j$. G is a planar st-graph and every vertex (except v_1, v_2, v_{n-1} and v_n) has at least 2 incoming and 2 outgoing edges. Let $d(v)$ denote the length of the longest path from the source v_1 of G to v. We construct the dual graph G^* of G and direct the edges of G^* as follows: if F_l and F_r are the left and the right face of some edge (v,w) of G, direct the dual edge from F_l to F_r if $(v,w) \neq (v_1, v_n)$ and from F_r to F_l if $(v,w) = (v_1, v_n)$. G^* is a planar st-graph. For each node F of G^*, let $d^*(F)$ denote the length of the longest path from the source node of G^* to F. Our visibility representation algorithm is very similar to the rectangular dual algorithm.

Algorithm 3. Visibility Representation (Input: A 4-connected triangulated plane graph G.)

1. Compute a canonical ordering of G.
2. Construct the planar st-graphs G and its dual G^*.
3. Compute $d(v)$ for the vertices of G and $d^*(F)$ for the nodes of G^*.
4. For each vertex v of G do:
   If $v \neq v_1, v_n$, draw horizontal line between $(d^*(\mathit{left}(v)), d(v))$ and $(d^*(\mathit{right}(v))-1, d(v))$.
   If $v = v_1$ or v_n, draw horizontal line between $(0, d(v))$ and $(D, d(v))$, where D is the length of the longest path between the source node and the sink node in G^*.
5. For each edge (u,v) of G do:

Draw vertical line between $(d^*(left(u,v)), d(u))$ and $(d^*(left(u,v)), d(v))$.
Fig. 5 shows an example of this algorithm.

Theorem 5.1. *Algorithm 3 constructs, in linear time, a visibility representation of G on a grid of size at most $(n-1) \times (n-1)$.*

Proof. Rosenstiehl and Tarjan [17] and Tamassia and Tollis [20] proved that whenever we construct in step 1 an ordering of the vertices of G that indices a planar *st*-graph, Algorithm 3 always constructs a visibility representation. Since the canonical ordering satisfies this requirement, our algorithm is correct. Since the construction of the canonical ordering talkes $O(n)$ time, and all other steps takes $O(n)$ time [17,20], Algorithm 3 runs in linear time. We need to show the size of the grid is at most $(n-1) \times (n-1)$. Since the length of the longest path from v_1 to v_n is at most $n-1$, the height of the representation is clearly bounded by $n-1$.

Let s^* be the source node of G^* and t^* be the sink node of G^*. Every vertex v of G corresponds to a face F_v of G^*. If $v \neq v_1, v_2, v_{n-1}, v_n$, then v has at least 2 incoming and at least 2 outgoing edges, hence the two directed paths from $low(F_v)$ to $high(F_v)$ both have length at least 2. Let $G^{*\prime}$ be the graph obtained from G^* by removing the sink node t^* and its incident edges. (In Fig. 5, s^* is the node represented by the square labeled by 0. t^* is the node represented by the square labeled by 11.) This merges the faces F_{v_1}, F_{v_2} and F_{v_n} of G^* into one face F'. Note that for any face $F \neq F_{v_{n-1}}$ of $G^{*\prime}$, the two directed paths of F between $low(F)$ and $high(F)$ in $G^{*\prime}$ have length $\geqslant 2$.

Let $s^{*\prime}$ be the source of $G^{*\prime}$ and let $t^{*\prime}$ be the sink of $G^{*\prime}$. Notice that $s^{*\prime} = s^* = low(F')$ and $t^{*\prime} = left((v_2, v_n)) = high(F')$. (In Fig. 5, $t^{*\prime}$ is the node represented by the square labeled by 10.) Clearly, there are at least two edges e in $G^{*\prime}$ with $left(e) = F_{v_{n-1}}$ and the only edge e with $right(e) = F_{v_{n-1}}$ has endpoint $t^{*\prime}$. Let P_{long} be any longest path in $G^{*\prime}$ from $s^{*\prime}$ to $t^{*\prime}$. Then the length of any longest path from s^* to t^* in G^* is 1 plus the length of P_{long}.

Claim. *P_{long} has at most one consecutive sequence of edges in common with any face F of $G^{*\prime}$.*

Toward a contradiction assume the claim is false. Suppose that P_{long} visits some nodes of F, assume that w_1 is the last one, then $l \geqslant 1$ nodes $u_1, \ldots, u_l \notin F$, then some nodes of F again, let w_d be the first one. Let $w_2, \ldots, w_{d-1}$ be the nodes, in this order, of F, which are not visited by P_{long} (see Fig. 6). Suppose $F = right((w_1, w_2))$. (If $F = left((w_2, w_2))$, the proof is similar.) Let $F_1 = left((w_1, w_2))$. Notice that $w_2 = low(F_1)$. The directed path of F_1, starting with edge (w_1, w_2), has length $\geqslant 2$. Hence ω_2 has an outgoing edge to a node of F_1, and an outgoing edge to w_3. Thus $w_2 = low(F_2)$, with $F_2 = left((w_2, w_3))$. Repeating this argument it follows that $w_{d-1} = low(F_{d-1})$, with $F_{d-1} = left((w_{d-1}, w_d))$. However it is easy to see that $w_d = high(F_{d-1})$. This means that one of the two directed paths of F_{d-1} has length 1. This contradiction proves the claim.

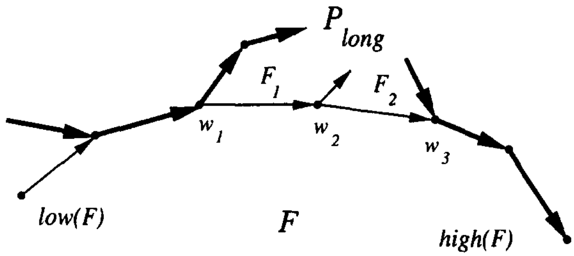

Fig. 6. Example of the proof of Theorem 5.1.

When traversing an edges e of P_{long}, we visit either *left*(e) or *right*(e) (or both) for the first time. If the face *left*(e) is visited for the first time, we *assign* e to *left*(e). If the face *right*(e) is visited for the first time, we *assign* e to *right*(e). If both *left*(e) and *right*(e) are visited for the first time, we arbitrarily assign e to either of the two faces. Consider any F of $G^{*\prime}$. By the claim, P_{long} has at most one consecutive sequence P_F of edges in common with F and only the first edge of P_F can be assigned to F. Thus at most one edge $e \in P_{\mathrm{long}}$ is assigned to F.

In summary, each edge e of P_{long} is assigned to a face of $G^{*\prime}$ and each face of $G^{*\prime}$ is assigned at most one edge $e \in P_{\mathrm{long}}$. Since $G^{*\prime}$ has $n-2$ faces, the length of P_{long} is at most $n-2$. Hence, the width of the visibility representation is at most $n-1$. $\square$

6. Conclusion

We presented two linear time algorithms for constructing an REL of 4TP graphs. We also use the REL algorithms to solve the rectangular dual problem. Our algorithm is much simpler than the previously known ones. More importantly, the coordinates of the rectangular dual constructed by our algorithm are integers and carry clear combinatorial meanings. This allows us to consider the related optimization problems: Let $w(R)$ and $h(R)$ denote the width and the height of the rectangular dual R. How to find a rectangular dual R such that $w(R)$ is minimized? $w(R)+h(R)$ is minimized? or $w(R)h(R)$ is minimized? These problems deserve further study.

For the visibility representation problem, our algorithm reduces the size of the representation by a factor of 2 for 4-connected planar graphs. Recently, using this result, Kant proved that every general planar graph has a visibility representation on a grid of size at most $(\lfloor \frac{3}{2}n \rfloor - 2) \times (n-1)$ [13].

An important tool used in our paper is the canonical ordering. Such an ordering implies an acyclic orientation of the graph, in which every vertex (except v_1, v_2, v_{n-1}, v_n) has at least 2 incoming and at least 2 outgoing edges. This extends the results for the *st*-ordering for biconnected plane graphs [17] (in which every vertex $v, v \neq v_1, v_n$, has at least 1 incoming and at least 1 outgoing edge in the acyclic orientation); and the canonical ordering for triangulated plane graphs [12] (in which every vertex $v, v \neq v_1, v_2, v_n$, has at least 2 incoming and at least 1 outgoing edge in the acyclic orientation). It would be nice to obtain more applications of this canonical ordering.

References

[1] J. Bhasker and S. Sahni, A linear algorithm to check for the existence of a rectangular dual of a planar triangulated graph, *Networks* **17** (1987) 307–317.
[2] J. Bhaskar and S. Sahni, A linear algorithm to find a rectanular dual of a planar triangulated graph, *Algorithmica* **3** (1988) 247–278.
[3] N. Chiba and N. Takao, Arbority and subgraph listing algorithms, *SIAM J. Comput.* **14** (1985) 210–223.
[4] G. Di Battista, P. Eades, R. Tamassia and I.G. Tollis, Algorithms for drawing graphs: an annotated bibliography, *Comput. Geom. Theory Appl.* **4** (1994) 235–282.
[5] H. de Fraysseix, J. Pach and R. Pollack, How to draw a planar graph on a grid, *Combinatorica* **10** (1990) 41–51.
[6] T. Hagerup, M. Chrobak and K. Diks. Optimal parallel 5-coloring of planar graphs, *SIAM J. Comput.* **18** (2) (1989) 288–300.
[7] X. He, On finding the rectangular duals of planar triangulated graphs, *SIAM J. Comput.* **22** (1993) 1218–1226.
[8] X. He, An efficient parallel algorithm for finding rectangular dual duals of planar triangular graphs, *Algorithmica* **13** (6) (1995) 553–572.
[9] X. He, Grid embedding of 4-connected plane graphs, in: *Proc. Graph Drawing (GD'95)*, Lecture Notes in Comp. Science, Vol. 1027 (1995) 287–299.
[10] W.R. Heller, G. Sorkin and K. Mailing, The planar package planner for system designers. in: *Proc. 19th Ann. IEEE Design Automation Conf.*, New York (1982) 253–260.
[11] J. Hopcroft and R.E. Tarjan, Efficient planarity testing, *J. ACM* **21** (4) (1974) 549–568.
[12] G. Kant, Drawing planar graphs using the *lmc*-ordering, in *Proc. 33th Ann. IEEE Symp. on Found. of Comp. Science*, Pittsburgh (1992) 101–110.
[13] G. Kant, A more compact visibility representation, in: *Proc. 19th Intern. Workshop on Graph-Theoretic Concepts in Comp. Science (WG'93)*, Lecture Notes in Comp. Science, Vol. 790 (Springer, Berlin, 1993) 411–424.
[14] K. Koźmiński and E. Kinnen, Rectangular dual of planar graphs, *Networks* **15** (1985) 145–157.
[15] Y.-T. Lai and S.M. Leinwand, A theory of rectangular dual graphs, *Algorithmica* **5** (1990) 467–483.
[16] K. Mailing, S.H. Meuller and W.R. Heller, On finding most optimal rectangular package plans. *Proc. 19th Ann. IEEE Design Automation Conf.*, New York (1982) 263–270.
[17] P. Rosenstiehl and R.E. Tarjan, Rectilinear planar layouts and bipolar orientations of planar graphs, *Discrete Comp. Geometry* **1** (1986) 343–353.
[18] W. Schnyder, Planar graphs and poset dimension, *Orders* **5** (1989) 323–343.
[19] W. Schnyder, Embedding planar graphs on the grid, in: *Proc. 1st Ann. ACM-SIAM Symp. on Discr. Alg.*, San Francisco (1990) 138–147.
[20] R. Tamassia and I.G. Tollis, A unified approach to visibility representations of planar graphs, *Discrete Comp. Geometry* **1** (1986) 321–341.