Two Algorithms for Finding Rectangular Duals of Planar Graphs*

Goos Kant[1] and Xin He[2]

[1] Department of Computer Science, Utrecht University,
Padualaan 14, 3584 CH Utrecht, the Netherlands. goos@cs.ruu.nl
[2] Department of Computer Science, State University of New York at Buffalo, Buffalo, NY 14260, USA. xinhe@cs.buffalo.edu

Abstract. We present two linear-time algorithms for computing a regular edge labeling of 4-connected planar triangular graphs. This labeling is used to compute in linear time a rectangular dual of this class of planar graphs. The two algorithms are based on totally different frameworks, and both are conceptually simpler than the previous known algorithm and are of independent interests. The first algorithm is based on edge contraction. The second algorithm is based on the canonical ordering. This ordering can also be used to compute more compact visibility representations for this class of planar graphs.

1 Introduction

The problem of drawing a graph on the plane has received increasing attention due to a large number of applications [3]. Examples include VLSI layout, algorithm animation, visual languages and CASE tools. Vertices are usually represented by points and edges by curves. In the design of floor planning of electronic chips and in architectural design, it is also 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 $\Gamma = \{R_1, R_2, \ldots, R_n\}$ of non-overlapping rectangles such that no four rectangles in Γ meet at the same point. A *rectangular dual* of a planar 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 the application of this representation, the vertices of G represent circuit modules and the edges represent module adjacencies. A rectangular dual provides a placement of the circuit modules that preserves the required adjacencies. Figure 1 shows an example of a planar graph and its rectangular dual.

This problem was studied in [1, 2, 8]. Bhasker and Sahni gave a linear time algorithm to construct rectangular duals [2]. The algorithm is fairly complicated and requires many intriguing procedures. The coordinates of the rectangular dual constructed by it are real numbers and bear no meaningful relationship with the

* 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 supported by National Science Foundation, grant number CCR-9011214.

structure of the graph. This algorithm consists of two major steps: (1) constructing a so-called *regular edge labeling* (REL) of G; and (2) constructing the rectangular dual using this labeling. A simplification of step (2) is given in [5]. The coordinates of the rectangular dual constructed by the algorithm in [5] are integers and carry clear combinatorial meaning. However, the step (1) still relies on the complicated algorithm in [2]. (A parallel implementation of this algorithm, working in $O(\log n \log^* n)$ time with $O(n)$ processors, is given in [6].)

In this paper we present two linear time algorithms for finding a regular edge labeling. The two algorithms use totally different approaches and both are of independent interests. The first algorithm is based on the *edge contraction* technique, which was also used for drawing triangular planar graphs on a grid [10]. The second algorithm is based on the *canonical ordering* for 4-connected planar triangular graphs. This technique extends the canonical ordering, which was defined for triangular planar graphs [4] and triconnected planar graphs [7], to this class of graphs. Another interesting representation of planar graphs is the *visibility representation*, which maps vertices into horizontal segments and edges into vertical segments [9, 11]. It turns out that the canonical ordering also gives a reduction of a factor 2 in the width of the visibility representation of 4-connected planar graphs.

The present paper is organized as follows. Section 2 presents the definition of the regular edge labeling and reviews the algorithm in [5] that computes a rectangular dual from a REL. In section 3, we present the edge contraction based algorithm for computing a REL. In section, 4 we present the second REL algorithm based on the canonical ordering. Section 5 discusses the algorithm for the visibility representation and some final remarks.

2 The rectangular dual algorithm

Let $G = (V, E)$ be a planar graph with n vertices and m edges. If $(u, v) \in E$, u is a *neighbor* of v. $deg(u)$ denotes the number of neighbors of u. We assume G is equipped with a fixed plane embedding. The embedding divides the plane into a number of *faces*. 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*. An interior edge between two exterior vertices is called a *chord*. A path (or a cycle) of G consisting of k edges is called a $k-$path (or a $k-$cycle, respectively). A *triangle* is a 3-cycle. A *quadrangle* is a 4-cycle. A cycle C of G divides the plane into its interior and exterior region. If C contains at least one vertex in its interior, C is called a *separating cycle*.

A *plane triangular graph* is a plane graph all of whose interior faces are triangles. For the rectangular dual problem, as we will see later, we only need to consider plane triangular graphs. Let G be such a graph. Consider an interior vertex v of G. We use $N(v)$ to denote the set of neighbors of v. If $N(v) = \{u_1, \dots, u_k\}$ are in counterclockwise order around v in the embedding, then $u_1, \dots, u_k$ form a cycle, denoted by $Cycle(v)$. The *star* at v, denoted by $Star(v)$, is the set of the edges $\{(v, u_i) \mid 1 \leq i \leq k\}$.

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) (one in $Adj(u)$ and one in $Adj(v)$) are cross-linked to each other. This representation can be constructed as a by-product by using a planarity testing algorithm in linear time.

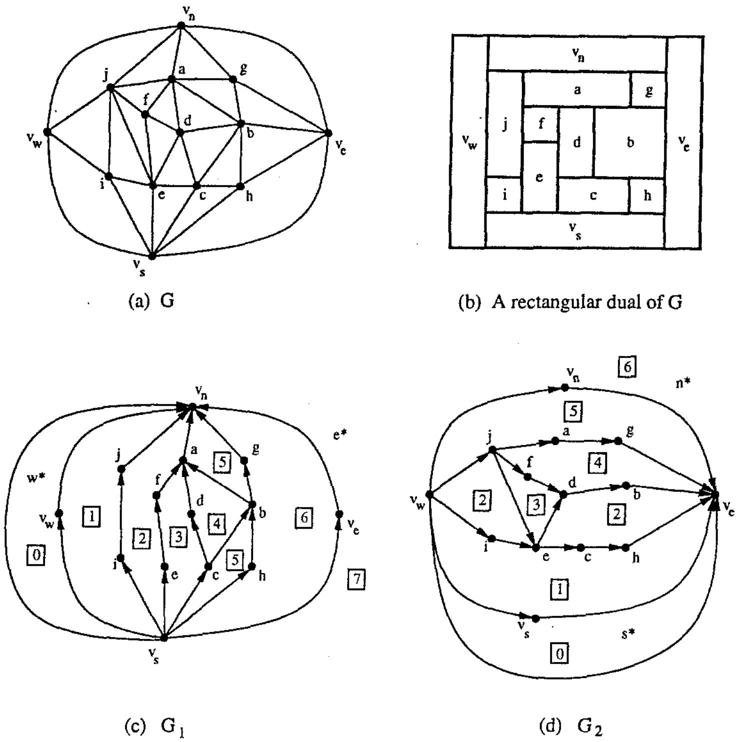

Fig. 1. A PTP graph, its rectangular dual, and the *st*-graphs G_1 and G_2

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's 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 Figure 1 (1) and (2)). The following theorem was proved in [1, 8]:

Theorem 1. *A planar graph G has a rectangular dual R with four rectangles on the boundary of R if and only if (1) every interior face is a triangle and the exterior face is a quadrangle; (2) G has no separating triangles.*

A graph satisfying the conditions in Theorem 1 is called a *proper triangular planar* (PTP) graph. ¿From now on, we will discuss only such graphs. Note the condition (2) of Theorem 1 implies that G is 4-connected. Since G has no separating triangles, the degree of any interior vertex v of G is at least 4. (If $deg(v) = 3$, $Cycle(v)$ would be a separating triangle.)

The rectangular dual algorithm in [5] heavily depends on the concept of *regular edge labeling* (REL) defined as follows [2, 5]:

Definition 2. A regular edge labeling of a PTP graph G is a partition 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, v_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.

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

Let G be a PTP graph and $\{T_1, T_2\}$ be a 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 $v_S \to v_W$, $v_W \to v_N$, $v_S \to v_E$, $v_E \to v_N$), and a new edge (v_S, v_N). 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 the clockwise direction is denoted by $left(v)$. The other face of G_1 that separates the incoming and the outgoing edges of v is denoted by $right(v)$.

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). 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 the clockwise direction is denoted by $above(v)$. The other face of G_2 that separates the incoming and the outgoing edges of v is denoted by $below(v)$.

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. We direct the edges of G_1^* 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 of G, define: $x_{\text{left}}(v) = d_1(left(v))$, and $x_{\text{right}}(v) = d_1(right(v))$. 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 of G, 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 rectangular dual algorithm relies on the following theorem [5].

Theorem 3. *Let G be a PTP graph and $\{T_1, T_2\}$ be a 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 $\{f(v) | v \in V\}$ form a rectangular dual of G.*

Figure 1 shows an example of the theorem. Figure 1 (3) shows the st-graph G_1. The small squares in the figure represent the nodes of G_1^* and the integers in the squares represent their d_1 values. Figure 1 (4) shows the graph G_2. Figure 1 (2) shows the rectangular dual constructed as in Theorem 3. The algorithm for computing a rectangular dual is as follows [5]:

Algorithm 1: Rectangular Dual (Input: a PTP graph $G = (V, E)$).

1. Construct a regular edge labeling $\{T_1, T_2\}$ of G.
2. Construct from $\{T_1, T_2\}$ the planar st-graphs G_1 and G_2.
3. Construct the dual graph G_1^* from G_1 and G_2^* from G_2.
4. Compute $d_1(F)$ for nodes in G_1^* and $d_2(F)$ for nodes in G_2^*.
5. Assign each vertex v of G a rectangle $f(v)$ as in Theorem 3.

The steps 2 through 5 of Algorithm 1 can be easily implemented in linear time [5]. In next two sections we present two algorithms for constructing a REL of PTP graphs.

3 Algorithm based on edge contraction

In this section, we present our first algorithm for computing a REL of a PTP 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. 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. e_1 and e_2 are replaced by a new edge (y, o_e).

e_3 and e_4 are replaced by a new edge (z, o_e). (See Figure 2.) The resulting *contracted graph* is denoted by G/e. The edges e_1, e_2, e_3, e_4 are called the *surrounding edges e*. The edges (y, o_e) and (z, o_e) are called the *residue edges* of e.

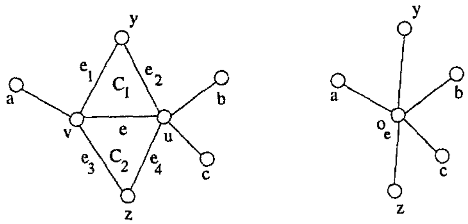

Fig. 2. Edge contraction

The graph $G' = G/e$ has a plane embedding inherited from the embedding of G. Since G is a PTP graph, e is not on any separating triangle. Thus G' has no multiple edges. It's easy to see that G' with the inherited embedding is a plane triangular 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 PTP graph.

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 $N(v)$. The edge (u, v) is contractible if and only if for any neighbor x ($x \neq y, z$) of v, the only common neighbors of u and x are v and possibly y or z. In this case, u is called a *contractible neighbor* of v.

Lemma 4. *Let G be a PTP 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.*

Let e be a contractible edge of a PTP graph G. Suppose a REL $\{T_1', T_2'\}$ of $G' = G/e$ has been found. Then we can *expand* e and obtain a 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 the rotation of the edges around o_e) as shown in Figure 3. 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 (o_e, y) in clockwise order. Depending on the label of (o_e, x) with respect to $\{T_1', T_2'\}$,

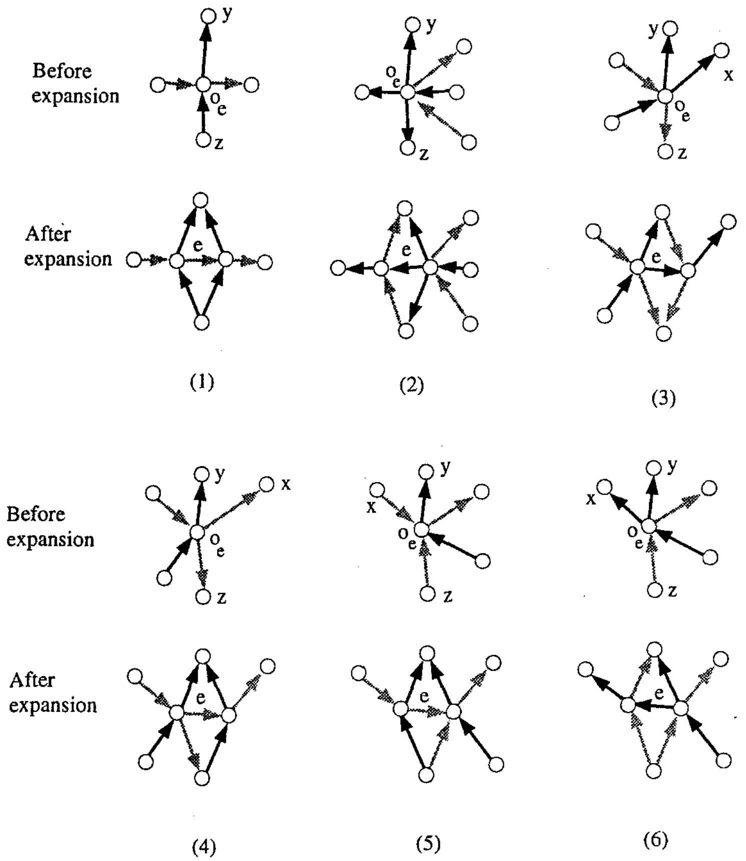

Fig. 3. Edge expansion

there are two cases as shown in Fig 4.3 and 4.4. 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 Fig 4.5 and 4.6. 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. Since the minimum degree of G is at most 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 a REL for the graph $G' = G/e$. Finally expand e to obtain a 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 or not. Since the degree of u and w can be large, this checking can 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 $V_i = \{v \in V | deg(v) = i\}$ and $V_{[i,j]} = \{v \in V | i \leq deg(v) \leq j\}$. 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 5. *Any PTP 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 $4n_4 + 5n_5 + 6n_{[6,19]} + 20n_{[20,\infty)} \leq \sum_{4 \leq i} in_i = \sum_{v \in V} deg(v) = 2|E| = 6n - 14 = 6(n_4 + n_5 + n_{[6,19]} + n_{[20,\infty)}) - 14$. This gives: $14n_{[20,\infty)} + 2n_6 \leq 2n_4 + n_5 + 2n_6 - 14 \leq 2n - 14$. Hence:

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

Let p_4 (p_5, respectively) be the number of good degree-4 (degree-5, respectively) vertices. So 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. So the contribution of v to S is at least 3. Thus $3(n_4 - p_4) + 2(n_5 - p_5) \leq S$, which gives $3n_4 + 2n_5 - (3p_4 + 2p_5) \leq \sum_{v \in V_{[20,\infty)}} deg(v)$. This in turn implies: $3n_4 + 2n_5 + \sum_{v \in V_{[4,5]}} deg(v) + \sum_{v \in V_{[6,19]}} deg(v) - (3p_4 + 2p_5) \leq \sum_{v \in V} deg(v) = 2|E| = 6n - 14 = 6(n_4 + n_5 + n_6 + n_{[7,19]} + n_{[20,\infty)}) - 14$. Simplifying this inequality, we get: $n_4 + n_5 + n_{[7,19]} - (3p_4 + 2p_5) \leq 6n_{[20,\infty)} - 14$. Hence:

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

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

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

Algorithm 2: REL (Input: A PTP 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:

Put w_i back into G. Expand the corresponding contracted edge.

Theorem 6. *Algorithm 2 computes a REL of a PTP 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 5, 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 non-trivial part is Step 4.2. We need to find a contractible neighbor of v in $O(1)$ time. Suppose $deg(v) = 5$ and u_i $(0 \leq i \leq 4)$ 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 and u_j $(1 \leq i, j \leq 4)$, 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 u_i $(1 \leq i \leq 4)$ is contractible, then u_0 is contractible by Lemma 4. 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 $deg(v) \leq 5$, 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 operation 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 Algorithm based on canonical ordering

In this section we consider 4-connected planar triangular graphs (all of whose face, including the exterior face, are triangles). We introduce the *canonical ordering* for such graphs, which is the basis for our second algorithm for finding a REL of a PTP graph G. Note that adding an edge connecting two non-adjacent exterior vertices of a PTP-graph G leads to a 4-connected planar triangular graph. The applications of the canonical ordering to other classes of planar graphs have been studied in [4, 7].

4.1 The canonical ordering of 4-connected planar triangular graphs

Let G be a 4-connected planar triangular graph with three exterior vertices u, v, w.

Theorem 7. *There exists a labeling of the vertices $v_1 = u, v_2 = v, v_3, \ldots, v_n = w$ of G meeting the following requirements for every $4 \leq k \leq n$:*

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 neighbors in G_{k-1} form a (at least 2-element) subinterval of the path $C_{k-1} - \{(u, v)\}$. If $k \leq n-2$, v_k has at least 2 neighbors in $G - G_{k-1}$.*

Proof. The vertices $v_n, v_{n-1}, \ldots, v_3$ are defined by reverse induction. Number the three exterior vertices u, v, w by v_1, v_2 and v_n. Let G_{n-1} be the subgraph of G after deleting v_n. By 4-connectivity of G, G_{n-1} is triconnected, and its exterior face C_{n-1} is a cycle and, hence, admits the constraints of the theorem. Let $v_{n-1} \neq v_1$ be the vertex of C_{n-1} adjacent to both v_2 and v_n in G. By the 4-connectivity, $G - \{v_n, v_{n-1}\}$ is biconnected and its exterior face C_{n-1} is a cycle and, hence, admits the constraints.

Let $k < n-1$ be fixed and assume that v_i has been determined for every $i > k$ such that the subgraph G_i induced by $V - \{v_{i+1}, \ldots, v_n\}$ satisfies the constraints of the theorem. Let C_k denote the boundary of the exterior face of G_k. Assume first that C_k has no interior chords. Suppose $v_1, c_{k_1}, \ldots, c_{k_p}, v_2$ 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 $p \geq 2$. If all vertices $c_{k_1}, \ldots, c_{k_p}$ have only one edge to the vertices in $G - G_k$, then since G is a planar triangular graph, they are adjacent to the same vertex v_j for some $k < j < n$. 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_{k_α}, has at least 2 neighbors in $G - G_k$. c_{k_α} is 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. Let also (c_d, c_e) be a chord with $e > d \geq b$ such that $e - d$ is minimal. (If there is no such a chord, let $(c_a, c_b) = (c_d, c_e)$ and number the vertices in clockwise order around C_k such that $a = 1 < b = d$ and $e = 1$.) Assume, without loss of generality, 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 triangular graph, they are adjacent to the same vertex v_j, and we also have $(v_a, v_j), (v_b, v_j) \in G$. But then $\{(v_a, v_j), (v_j, v_b), (v_b, v_a)\}$ 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 having no incident chords. c_α is the next vertex v_k in our ordering.

Theorem 8. *The canonical ordering can be computed in linear time.*

Proof. We label each vertex v by $Interval(v)$, which can have the following values: (a): not yet visited, (b): visited once, or (p): visited more than once and the visited edges form p intervals in $Adj(v)$. We also maintain a variable $Chords(v)$ for each vertex v on the exterior face, denoting the number of incident chords of v.

We start with v_n and v_{n-1} and initialize the labels of their neighbors. We compute the ordering in reverse order and update the labels after choosing a vertex v_k as follows: we visit each neighbor v of v_k along the edge connecting them. Let $c_i, \ldots, c_j$ $(j > i)$ be the neighbors (in this order) of v_k in G_{k-1}. If $j = i+1$, then there was a chord (c_i, c_j) in G_{k-1}, hence we decrease $Chords(c_i)$ and $Chords(c_j)$ by one, since (c_i, c_j) becomes part of C_{k-1}. If $j > i+1$, then for each c_l $(i < l < j)$, we compute $Chords(c_l)$. If c_l has a chord to v, then we also increase $Chords(v)$ with one. This is done by marking the vertices that are part of the exterior face. For each c_l $(i \leq l \leq j)$, we update $Interval(c_l)$: if c_l has label $Interval(c_l) = (a)$, label (b) replaces label (a). If $Interval(c_l)$ has label (b) and v_k in $Adj(c_l)$ is adjacent to a previous visited vertex in $Adj(c_l)$, then $Interval(c_l)$ becomes (1), otherwise it becomes (2). Otherwise assume $Interval(c_l) = (p)$, with $p \geq 1$. If the two incident vertices v' and v'' of vertex v_k in $Adj(c_l)$ are already visited, then $Interval(c_l)$ becomes $(p-1)$. If none of v' and v'' is visited, then $Interval(c_l)$ becomes $(p+1)$, else $Interval(c_l)$ is not changed. It is clear that $Interval(c_l) = (p)$ means that the vertices already visited and incident to c_l are composed of p intervals in $Adj(c_l)$.

By Theorem 7 it follows that, if $k \geq 3$, then there is a vertex v with $Interval(v)$ $= 1$ and $Chords(v) = 0$, and this can be chosen as the next vertex v_k in our ordering. We mark v as being visited. Since there are only a linear number of edges, we can find the canonical ordering in linear time.

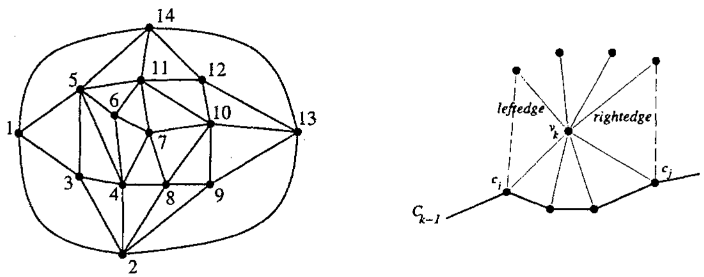

Fig. 4. The canonical ordering from the graph of Figure 1.

4.2 From a canonical ordering to a REL

To compute a REL of a PTP graph G, we first add an edge connecting two non-adjacent exterior vertices of G. This gives a 4-connected planar triangular graph G'. We compute a canonical numbering 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 a REL of G can be easily derived from the canonical ordering.

First, for each edge (v_i, v_j) of G, direct it from v_i to v_j, if $i < j$. Define the *base-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 C_{k-1} (the exterior face 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* of v_k.

Lemma 9. *A base-edge cannot be a leftedge or a rightedge.*

Lemma 10. *An edge is either a leftedge, a rightedge or a base-edge.*

We construct a REL for G as follows: all leftedges belong to T_1, all rightedges belong to T_2. The base-edge (c_α, v_k) of v_k is added to T_1, if $\alpha = j$, to T_2, if $\alpha = i$, and otherwise arbitrary to either T_1 or T_2. (The four exterior edges belong to neither T_1 nor T_2.)

Lemma 11. *$\{T_1, T_2\}$ 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 $(3 \leq k \leq n-2)$. It follows from Theorem 7 that $d \geq 2$. 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 base-edges of $v_{k_2}, \ldots, v_{k_{d-1}}$, respectively. Let the vertex v_{k_β} $(1 \leq \beta \leq d)$ 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} $(1 < l < d,\ l \neq \beta)$, either $k_{l-1} < k_l < k_{l+1}$ or $k_{l-1} > k_l > k_{l+1}$. Thus, by the construction of T_1 and T_2, the edges (v_k, v_{k_l}) are added to T_1, if $1 \leq l < \beta$, and to T_2, if $\beta < l \leq d$. 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 $v_{1_d} = v_2$, and by the argument described above, $(v_1, v_{1_2}), \ldots, (v_1, v_{1_{d-1}})$ 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 8 and Lemma 11 constitute our linear time REL algorithm. See Figure 4 for the construction of a REL from a canonical ordering.

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 [9, 11]. In this section, we show that the canonical ordering can be used to construct a more compact visibility representation for a 4-connected planar triangular graph G.

First let the edges of G be directed 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. The algorithm for constructing the visibility representation of Rosenstiehl & Tarjan [9] and Tamassia & Tollis [11] is almost identical to the rectangular dual algorithm.

Algorithm 3: Visibility Representation
Input: A 4-connected planar triangular 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:
   Draw horizontal line between $(d^*(left(v)), d(v))$ and $(d^*(right(v)) - 1, d(v))$.
5. For each edge (u, v) of G do:
   Draw vertical line between $(d^*(left(u, v)), d(u))$ and $(d^*(left(u, v)), d(v))$.

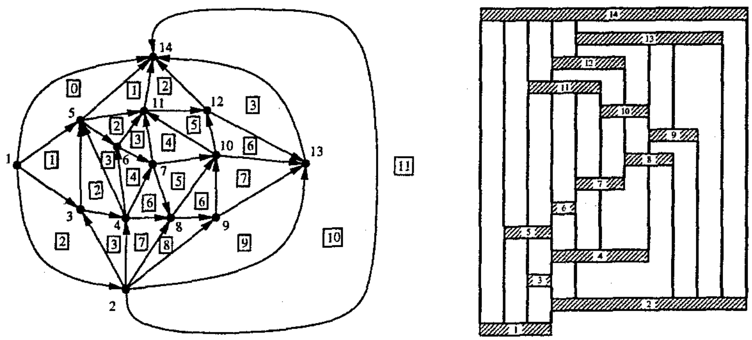

Fig. 5. The canonical ordering leads to a compact visibility representation.

Theorem 12. VISIBILITY(G) *constructs a visibility representation of G on a grid of size at most $(n-1) \times (n-1)$.*

Proof. The correctness of VISIBILITY(G) is shown in [9, 11]. We show that the grid size is at most $(n-1) \times (n-1)$. This follows directly for the height, since the length of the longest path from v_1 to v_n is at most $n-1$.

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 ≥ 2 incoming and ≥ 2 outgoing edges, hence the two directed paths from $low(F_v)$ to $high(F_v)$ both have length ≥ 2. Let $G^{*'}$ be the graph obtained from G^* by removing the sink node t^* and its incident edges. (In Figure 5, 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 every face $F \neq F_{v_{n-1}}$ of $G^{*'}$, the two directed paths of F between $low(F)$ and $high(F)$ in $G^{*'}$ have length ≥ 2.

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

We claim that P_{long} has at most one consecutive sequence of edges in common with any face F of $G^{*'}$. Toward a contradiction assume the claim is not true. Suppose that P_{long} visits some nodes of F, assume that w_1 is the last one, then $l \geq 1$ 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 Figure 6.) Suppose $F = right((w_1, w_2))$. (If $F = left((w_1, w_2))$, the proof is similar.) Let $F_1 = left((w_1, w_2))$. Notice that $w_1 = low(F_1)$. The directed path of F_1, starting with edge (w_1, w_2), has length ≥ 2. Hence w_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.

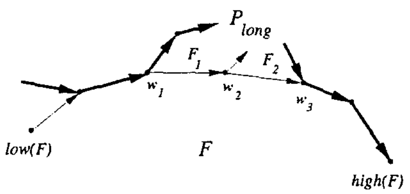

Fig. 6. Example of the proof of Theorem 5.1.

When traversing an edge e of P_{long}, we visit either $left(e)$ or $right(e)$ (or both) for the first time. We *assign* each edge e to the face F, with $e \in F$, which we visit for the first time now. $G^{*'}$ has $n-2$ faces. To every face F of $G^{*'}$, by the claim, at most one edge $e \in P_{long}$ is assigned. Hence the longest path from s^* to t^* in G^* has length $\leq n-1$.

VISIBILITY(G) can be applied to a general 4-connected planar graph by first triangulating it. (The triangulation of a 4-connected planar graph is clearly still 4-connected.) Since the worst-case bounds for visibility representation by applying an arbitrary st-numbering is $(2n-5) \times (n-1)$ [9, 11], our algorithm reduces the width of the visibility representation by a factor 2 in the case of 4-connected planar graphs. Maybe this approach can be used to obtain better grid bounds in general, by splitting the graph into 4-connected components. Consider for this problem a planar triangular graph G. Let C be a separating triangle, such that there are no separating triangles inside C. The subgraph inside C yields a 4-connected component, say B_1. B_1 can be drawn within the required bounds. Drawing B_1 inside $G-B_1$ may increase the drawing of $G-B_1$ by at most $|B_1|-1$ in height and width. If the face F on the vertices u, v, w is not a rectangle in the visibility representation of $G-B_1$, then this is no problem. The difficult case is when F is a rectangle. Solving this remaining problem gives an important improvement in the visibility representations, which plays a major role in a lot of practical commercial environments.

The canonical ordering, presented in this paper, implies an acyclic orientation of the graph, in which every vertex (except v_1, v_2, v_{n-1}, v_n) has ≥ 2 incoming and ≥ 2 outgoing edges. This extends the results for the *st*-ordering for biconnected planar graphs [9] (in which every vertex $v, v \neq v_1, v_n$, has ≥ 1 incoming and ≥ 1 outgoing edge in the acyclic orientation), and the canonical ordering for planar triangular graphs [7] (in which every vertex $v, v \neq v_1, v_2, v_n$, has ≥ 2 incoming and ≥ 1 outgoing edge in the acyclic orientation). Another observation is that the canonical ordering, presented in section 4, gives a simple algorithm to test whether a planar triangular graph is 4-connected.

An interesting research field is to problem of computing a canonical ordering of a 4-connected planar graph such that v_{i+1} is a neighbor of v_i. This would yield a simple algorithm for constructing hamiltonian circuits in 4-connected triangular planar graphs. We leave this question open for the interested reader.

References

1. Bhasker, J., and S. Sahni, A linear algorithm to check for the existence of a rectangular dual of a planar triangulated graph, Networks 7 (1987), pp. 307-317.
2. Bhasker, J., and S. Sahni, A linear algorithm to find a rectangular dual of a planar triangulated graph, *Algorithmica* 3 (1988), pp. 247–178.
3. Eades, P., and R. Tamassia, *Algorithms for Automatic Graph Drawing: An Annotated Bibliography*, Dept. of Comp. Science, Brown Univ., Technical Report CS-89-09, 1989.
4. Fraysseix, H. de, J. Pach and R. Pollack, How to draw a planar graph on a grid, *Combinatorica* 10 (1990), pp. 41–51.
5. He, X., On finding the rectangular duals of planar triangulated graphs, *SIAM J. Comput.*, to appear.
6. He, X., *Efficient Parallel Algorithms for two Graph Layout Problems*, Technical Report 91-05, Dept. of Comp. Science, State Univ. of New York at Buffalo, 1991.
7. Kant, G., Drawing planar graphs using the *lmc*-ordering, *Proc. 33th Ann. IEEE Symp. on Found. of Comp. Science*, Pittsburgh, 1992, pp. 101-110.
8. Koźmiński, K., and E. Kinnen, Rectangular dual of planar graphs, *Network* 5 (1985), pp. 145–157.
9. Rosenstiehl, P., and R. E. Tarjan, Rectilinear planar layouts and bipolar orientations of planar graphs, *Discr. and Comp. Geometry* 1 (1986), pp. 343–353.
10. Schnyder, W., Embedding planar graphs on the grid, in: *Proc. 1st Annual ACM-SIAM Symp. on Discr. Alg.*, San Francisco, 1990, pp. 138–147.
11. Tamassia, R., and I. G. Tollis, A unified approach to visibility representations of planar graphs, *Discr. and Comp. Geometry* 1 (1986), pp. 321–341.