Hexagonal Grid Drawings

Goos Kant

RUU-CS-92-06
February 1992

Utrecht University

Department of Computer Science

Padualaan 14, P.O. Box 80.089,
3508 TB Utrecht, The Netherlands,
Tel. : ... + 31 - 30 - 531454

Hexagonal Grid Drawings

Goos Kant

Technical Report RUU-CS-92-06
February 1992

Department of Computer Science
Utrecht University
P.O.Box 80.089
3508 TB Utrecht
The Netherlands

ISSN: 0924-3275

Hexagonal Grid Drawings*

Goos Kant
Dept. of Computer Science, Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, the Netherlands

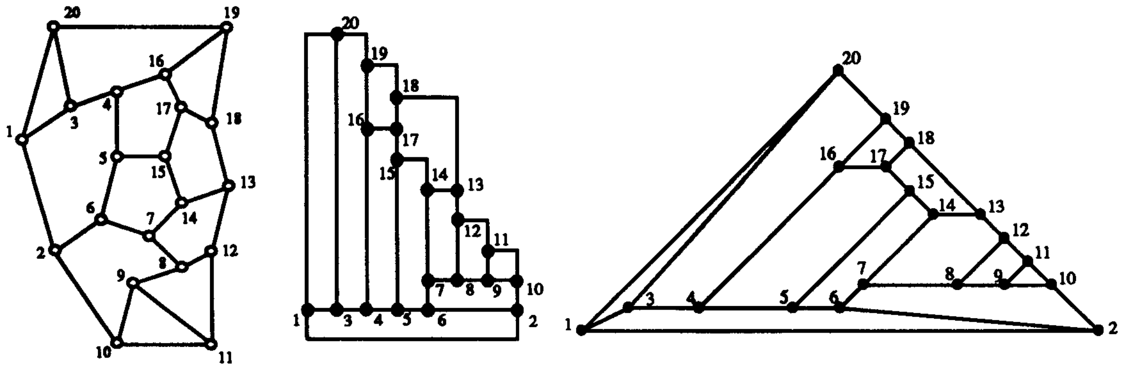
Abstract

In this paper we present a linear algorithm to draw triconnected planar graphs of degree 3 planar on a linear-sized hexagonal grid such that in at most one edge are bends. This algorithm can be used to draw this class of graphs planar with straight lines on a $n/2 \times n/2$ grid, improving the best known grid bounds by a factor 4. We also show how to draw planar graphs of degree at most 3 planar with straight lines such that the minimum angle is $\geq \pi/6$, thereby answering a question of Formann et al.

1 Introduction

Embedding a graph in the rectilinear grid has several applications, like VLSI circuit design, architectural floor plan layout and aesthetic layout of diagrams used in information system design. This problem has been extensively studied for planar graphs of degree at most four such that the vertices are grid points and the edges are alternating sequences of horizontal and vertical segments (which is called orthogonal grid drawings). Storer [10] presented three heuristic algorithms to minimize the number of bends, and requiring an $O(n^2)$ grid. (n denote the number of vertices.) Tamassia & Tollis [13] gave linear implementations of these. These algorithms obtain solutions, whose number of bends is only a constant from the lower bounds. Some tighter existential lower bounds on the minimum number of bends required by layouts are presented by Tamassia, Tollis & Vitter [14]. Tamassia showed that the problem of minimizing bends is polynomially solvable, by presenting an $O(n^2 \log n)$ algorithm to draw a planar graph of degree ≤ 4 orthogonal on an $O(n^2)$ grid with a minimum number of bends [11]. The resulted drawings are of reasonable aesthetic quality, due to the large size of the angles (here rectangles) and a small number of bends.

*This work was supported by the ESPRIT Basic Research Actions of the EC under contract No. 3075 (project ALCOM).



(a) input graph (b) orthogonal drawing (c) straight–line convex drawing

Figure 1: An orthogonal and a straight-line drawing of a planar graph.

However, even for graphs with maximum degree 3 this can imply $\frac{n}{2}$ bends on a rectilinear grid ([10], see also figure 1(b)). On the other side, if we want to draw the graph with straight lines, then the minimum angle can be very small (see figure 1(b)). In this paper we try to find a trade-off between the number of bends and the size of the angles. Hereto we inspect the triangled (or so-called hexagonal grid in VLSI-design), as shown in figure 2. All angles of an hexagonal grid have size $\pi/3$ and we want to draw connected planar graphs of degree at most 3 (here also called 3-planar graphs) on this grid, such that the number of bends is minimized. We first inspect triconnected planar graphs of degree 3. Triconnectivity means that deleting any pair of vertices u, v does not destroy the planarity. Using the interesting characteristics of this class of planar graphs, in which every vertex has degree exactly three, we obtain a linear time algorithm to draw a triconnected 3-planar graph on a linear-sized hexagonal grid, such that there is at most one bent edge. This algorithm can easily be modified such that we can draw any triconnected 3-planar graph with straight lines on a rectilinear grid of size at most $n/2 \times n/2$. The best known grid bound for these graphs was $n - 2 \times n - 2$ (see Schnyder [9]), hence we improve the total size by a factor 4. A small area is important for the finite resolution of display and printing devices (and of the human eye), in which the dimensions cannot be arbitrarily scaled down. Other polynomial algorithms to draw a planar graph on a grid of size $O(n^2)$ are described in [2, 3, 4, 6, 10, 11].

We can generalize the algorithm such that all connected 3-planar graphs can be drawn on a hexagonal grid with at most one bent edge. Finally we show how we can use this algorithm to draw a connected 3-planar graph with straight lines in the plane, such that the minimum angle is at least $\pi/6$ for at most 4 angles, and at least $\pi/3$ for all remaining angles ($n > 4$). This solves an open problem of Formann et al. [5].

The paper is organized as follows: In section 2 we show how we can draw a

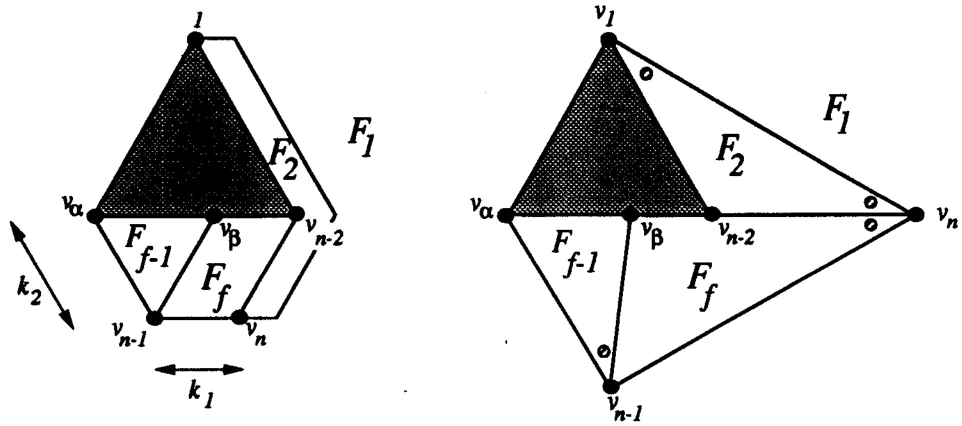


Figure 6: Drawing the 3-planar graph with wide angles.

Theorem 4.2 *There is a straight-line drawing of a connected 3-planar graph in which every angle has size* $\geq \pi/6$. *If* $n = 4$ *then 6 angles have size* $< \pi/3$*; if* $n > 4$ *then at most 4 angles have size* $< \pi/3$.

By using the grid as shown in figure 5 we can achieve a straight-line drawing of a connected 3-planar graph with smallest angle $> \pi/6$, for $n \geq 6$. We apply the drawing algorithm as described above. Let again $k_2 \geq k1$ and assume $k_2/k_1 = C$. After moving v_{n-1} and v_n to $(k_1, -k_1)$ and $(2k_1 + k_2, k_2)$, respectively, it follows that all angles, except $\gamma_1 = \angle v_\alpha v_{n-1} v_\beta$ and $\gamma_2 = \angle v_{n-1} v_n v_{n-2}$, have size at least $\pi/4$. For γ_1 holds that $\tan(\gamma_1) = k_2/(k1 + k_2) = C/(C+1)$. For γ_2 holds that $\tan(\gamma_2) = (k_1 + k_2)/(2k_1 + k_2) = (C+1)/(C+2)$. For C large enough we achieve that γ_1 and γ_2 come arbitrary close to $\pi/4$. However, this enlarges the size of the grid. We are not able to prove a lower bound of $\geq \pi/4$ for $n \geq 6$ in general and remains as an open problem.

5 Optimizations

In the original algorithm, described in section 2, we always go from the startpoint in Y-direction and from the endpoint in Z-direction to the same height, even when there is no reason to go upwards. For example, assume $y(c_i) > y(c_j)$ and (c_i, c_{i+1}) is in Z-direction downwards, then we can place the new vertices $w_1, \ldots, w_p$ of face F_k on a horizonal line on height $y(c_i)$ instead of $y(c_i) + 1$. To obtain this we change the two lines with (*) both as follows in HEXA-DRAW:

⋮

if $(y(c_i) > y(c_j)$ **and** $y(c_i) > y(c_{i+1}))$ **or** $(y(c_j) > y(c_i)$ **and** $y(c_j) > y(c_{j-1}))$ **then**

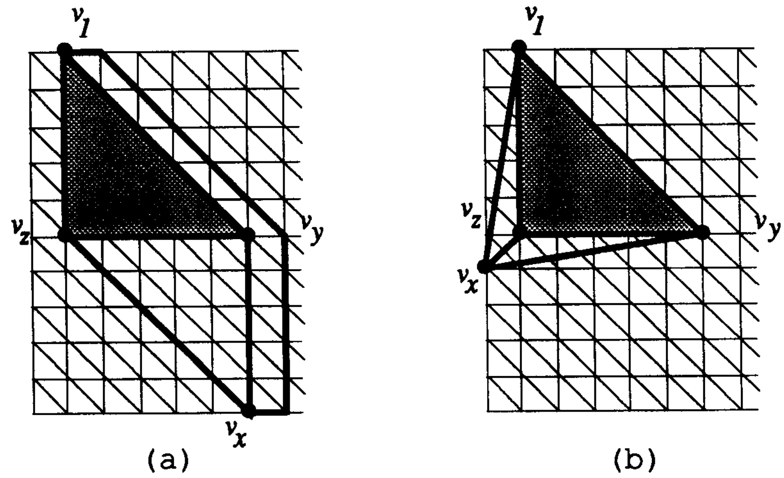


Figure 5: Drawing the triconnected 3-planar graph on gridcoordinates.

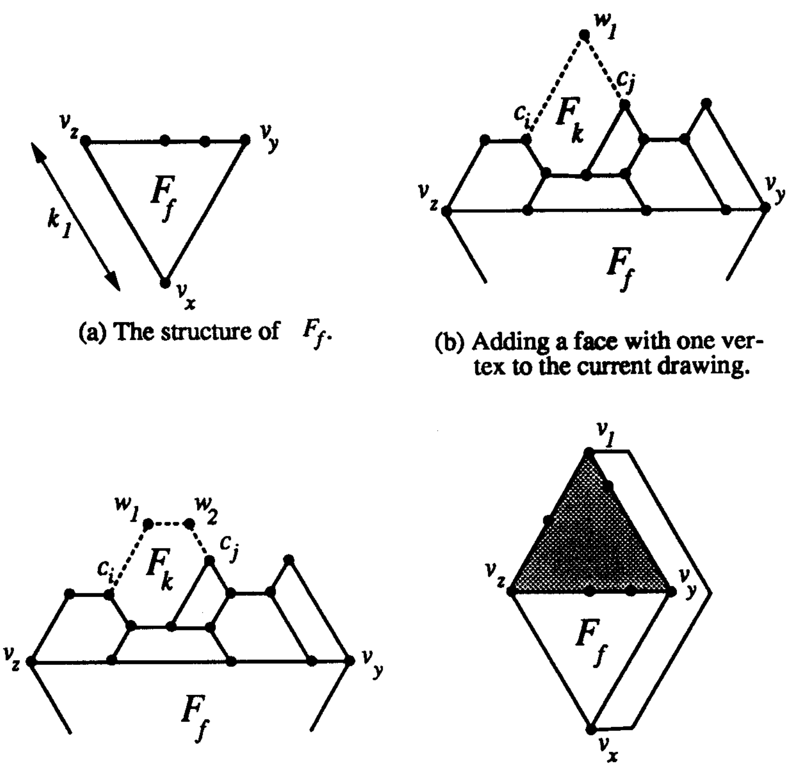
an angle of degree $3\pi/4$ with the X-axis. See figure 5(a). We now again do the algorithm HEXA-DRAW. The coordinates again follow the X- and Y-direction. We finally move vertex v_x to the point $(-k_1 - 1, k_1 - 1)$, as shown in figure 5(b). Now v_x has straight lines to v_y, v_z and v_1. The gridsize in X- and Y-direction is still the same, thereby proving the following theorem.

Theorem 2.7 *There is a simple linear time algorithm to draw a triconnected 3-planar graph planar with straight lines on an* $\frac{n}{2} \times \frac{n}{2}$ *grid.*

3 Drawing Graphs with degree at most 3

In this section we show how we can extend the results of section 2 to planar graphs G of degree at most 3. In this case the dual graph H is not necessarily triangulated and may contain several multiple edges. We assume that G is biconnected. If not, then we start with drawing a biconnected component and draw this component inside or outside another biconnected component later.

Assume first that all vertices in G have degree three. Inspect the dual graph H. H is triangulated, but has possibly several multiple edges. If deleting two vertices v_1, v_2 disconnects G into one component V' and one or more components $G' - V'$, then we call V' a *2-subgraph* with *cutting pair* v_1, v_2. Notice that since every vertex has degree 3 in G, every pair of multiple edges in H uniquely corresponds with a 2-subgraph V' of G. Using this observation we can construct in linear time the 2-subgraph-tree T (introduced in [8]): every node v' in T corresponds with a 2-subgraph V' in G. v' is an ancestor of w' in T, if $W' \subseteq V'$. The leaves of T



(a) The structure of F_f.

(b) Adding a face with one vertex to the current drawing.

(c) Adding a face with two vertices to the current drawing.

(d) The completed drawing.

Figure 3: Outline of the outerface.

Let v_x be the unique vertex $\in F_f \cap F_2 \cap F_1$. Let v_y and v_z be the neighbors of v_x in F_f. We start with drawing v_x on $(0,0)$. From v_x we place v_y k_1 steps in Y-direction and v_z k_1 steps in Z-direction. All other vertices of F_f are placed on the horizontal line (of length k_1) between v_z and v_y (see figure 3(a)), such that these horizontal edges e of F_f have length $th(e)$. The horizontal line between v_y and v_z forms the basis for adding the faces $F_{f-1}, F_{f-2}, \ldots, F_3$. When adding a face F_k by adding vertices and edges of $E(F_k)$ to the current drawing of $F_{k+1}, \ldots, F_f$, we call the added vertices and edges new. Let C_{k+1} be the outerface of the current drawing of $F_{k+1}, \ldots, F_f$, consisting of the vertices $v_x, v_z = c_1, c_2, c_3, \ldots, c_l = v_y$. Let c_i and c_j ($j > i$) be two vertices of C_{k+1}, incident to the new edges of F_k, then we call c_i the *startpoint*, c_j the *endpoint* and the vertices $v_{i+1}, \ldots, v_{j-1}$ *internal*. We also call the edges $(c_i, c_{i+1}), \ldots, (c_{j-1}, c_j)$ *internal*.

Adding a face goes as follows: if we add one vertex then we walk from c_i upwards in Y-direction and from c_j upwards in Z-direction. The crossing point is the place for the new vertex (see figure 3(b)). If we add the vertices $w_1, \ldots, w_p$ ($p \geq 2$), then we go from c_i one unit in Y-direction and from c_j in Z-direction to the same

H. Every vertex v in G has degree 3, thus every face in H has three edges. Since G is triconnected, H is triconnected as well, thus H is a triangulated planar graph with f vertices. In [6] there is a canonical numbering defined for triangulated planar graphs, which can be described as follows:

> Assume H is a triangulated planar graph embedded in the plane with exterior face u, v, w. Then there exists a numbering of the vertices $v_1 = u, m_2 = v, v_3, \ldots, v_f = w$ meeting the following requirements for every k, $4 \leq k \leq f$:
>
> - The subgraph $H_{k-1} \subseteq H$ induced by v_1, v_2, v_{k-1} is biconnected, and the boundary of its exterior face is a cycle C_{k-1} containing the edge (u, v);
> - v_k is in the exterior face of H_{k-1} and its neighbors in H_{k-1} form an (at least 2-element) subinterval of the path $C_{k-1} - (u, v)$.

Let the f vertices of H be numbered as in the canonical numbering, then every vertex v_i in H, $i \neq 1, 2, f$, has at least two neighbors v_j, v_k, with $j < i$ and $k < i$ and has at least one neighbor v_l, with $l > i$. Such a numbering can be obtained in linear time [6].

Let face F_i of G correspond with the canonical numbered vertex v_i of H. The idea is to start with drawing F_f, F_{f-1}, ..., and finally F_3, F_2, F_1. F_2 will be the face of G obtained by adding the edge with bends and F_1 will be the outerface. But the faces $F_{f-1}, F_{f-2}, \ldots$, influence the structure of the drawing of F_f and hereto we start with some definitions:

Definition 2.1 *$E(F_k)$ is the set of edges e of F_k, belonging to a face $F_j, j < k$.*

Definition 2.2 *The* basis-edge *of F_k, $be(F_k)$, is the edge $e \in F_k$ that among all edges in F_k, belongs to the highest numbered face F_j that is adjacent to F_k.*

Let $be(F_f)$ be the unique edge $e \in F_f \cap F_1$. By the canonical numbering of H it follows that $|E(F_k)| \geq 2$ for all faces $F_k, k \geq 3$. This means that when we add any face F_k to the current drawing of the faces $F_{k+1}, \ldots, F_f$ on the hexagonal grid, we add at least one vertex v. The basis-edge plays an important role in the drawing algorithm. We first assign a length to each basis-edge, which is calculated as follows:

Set $lth(e) = 1$ for all edges $e \in G$;
for $k := 3$ **to** $f - 1$ **do** $lth(be(F_k)) := \sum_{e \in E(F_k)} lth(e) - 1$;
$lth(be(F_f)) := \sum_{e \in E(F_f)} lth(e)$;

For each edge e, we will show that the length of e in the resulting drawing, denoted by *length*(e), is at least $lth(e)$. Let $k_1 = lth(be(F_f))$. We construct the drawing as follows:

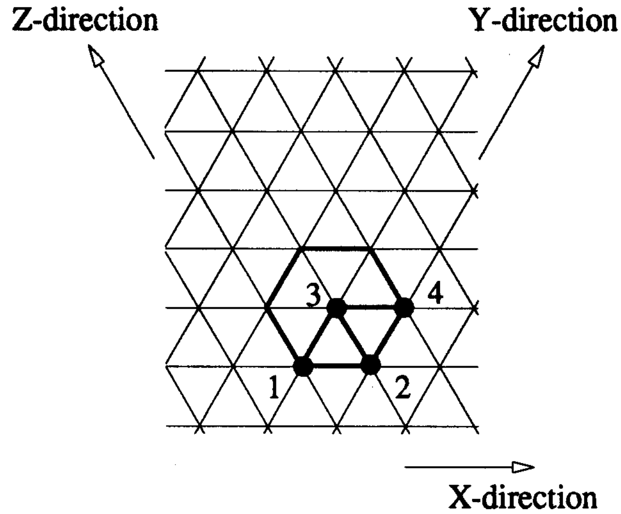


Figure 2: Drawing of a K_4 on a hexagonal grid.

triconnected 3-planar graph on a hexagonal grid. In section 3 we extend these results to general 3-planar graphs. In section 4 we show how we can draw 3-planar graphs in the plane with minimum angle $\geq \pi/6$. In section 5 we give some techniques for decreasing the grid size in most cases. Section 6 contains some final remarks and some open problems.

2 Triconnected 3-Planar Graphs

Let G be a planar graph, i.e., G can be drawn in the plane such that there is no pair of crossing edges. If G is triconnected (biconnected) then deleting any two (one) vertices with incident edges preserves the connectivity. If G is triconnected then the embedding is unique, which means that in every drawing of G, the order of the neighbors around each vertex is equal. We assume that every vertex v in G has degree 3. In a hexagonal grid, there are three directions of the lines (see figure 2): 0 degree lines (here called X-direction), $\pi/3$ degree lines (here called Y-direction) and the $2\pi/3$ degree lines (here called Z-direction). We show in this section that in linear time we can draw any triconnected 3-planar graph on a $\frac{n}{2} \times \frac{n}{2}$ hexagonal grid, such that in at most one edge are bends. This last bent edge cannot always be avoided. For instance, consider the graph K_4 in figure 2.

Let G be a triconnected 3-planar graph with n vertices. Then n must be even, the number of edges $m = \frac{3}{2}n$ and the number of faces $f = \frac{n}{2} + 2$, since by Euler's formulae, $m - n - f + 2 = 0$. Let H be the dual graph of G, i.e., every face F_k in G is represented by a vertex v_{F_k} in H and there is an edge (v_{F_k}, v_{F_j}) in H if F_k and F_j share a common edge in G. Every vertex v in G corresponds with a face in height (assume $y(c_i) \geq y(c_j)$) and add the new vertices on the horizontal line in between (see figure 3(c)). Adding face F_2 is obtained by going from v_x one step in X-direction, k_1 steps in Y-direction, k_1 steps in Z-direction and one step in negative X-direction to the last added vertex, say v_1 (see figure 3(d)). Adding the other faces can formally be described as follows:

```
HEXA-DRAW
    x(v_x) := y(v_x) := 0;
    let w_1, ..., w_p be the other vertices of F_f, with w_1 = v_z and w_p = v_y;
    y(w_1) := y(w_2) := ... := y(w_p) := k_1;
    x(w_1) := -k_1;
    for l := 2 to p do x(w_l) := x(w_{l-1}) + lth((w_l, w_{l-1}));
    for k := f - 1 downto 3 do
        let w_1, ..., w_p be the new vertices from startpoint c_i to endpoint c_j of F_k;
        x(w_1) := x(c_i);
        if p = 1 then
(*)         y(w_1) := y(c_j) + x(c_j) - x(c_i)
        else
(*)         y(w_1) := y(w_2) := ... := y(w_p) := max{y(c_i), y(c_j)} + 1;
            for l := 2 to p - 1 do x(w_l) := x(w_{l-1}) + lth((w_l, w_{l-1}));
            x(w_p) := x(c_j) + y(c_j) - y(w_1)
    rof;
```

It is easy to see that the algorithm can be implemented to run in linear time and space. To prove the correctness of the algorithm, we need the following lemma's:

Lemma 2.1 *At least one of the internal edges of a face F_k is horizontal.*

Proof: Suppose not. Let c_i and c_j be the start- and endpoint, resp., of F_k. c_i and c_j had degree 2 before adding F_k and thus (c_i, c_{i+1}) must have Z-direction downwards and (c_{j-1}, c_j) must have Y-direction upwards, if they are not horizontal. But there cannot be a vertex c_α, $i < \alpha < j$ such that $(c_{\alpha-1}, c_\alpha)$ has Z-direction and $(c_\alpha, c_{\alpha+1})$ has Y-direction, because then by HEXA-DRAW, c_α would have degree 4. Thus there must be at least one horizontal internal edge when adding F_k. □

Lemma 2.2 *The internal edges of a new face F_k are: first ≥ 0 edges in Z-direction downwards, then one horizontal edge and then ≥ 0 edges in Y-direction upwards, in this order from left to right.*

Proof: If for an internal vertex c_α holds that $(c_{\alpha-1}, c_\alpha)$ is of Y-direction and $(c_\alpha, c_{\alpha+1})$ is of Z-direction then by definition c_α has degree 2 in G_k and hence cannot be internal. Similar when one or two of these edges are horizontal. Hence there is correspond with the triconnected components of G and the root r of T corresponds with G.

Notice that adding an edge between the cutting pair of a triconnected component V' of G makes V' triconnected and 3-planar, hence we can apply HEXA-DRAW to all triconnected components V' of G, where the added edge between the cutting pair is the edge with bends in HEXA-DRAW. The idea is now as follows. We start with calculating the places of the triconnected components V' with respect to the startvertex v_x of V', by the algorithm HEXA-DRAW. We know from lemma 2.5 that we can do it such that its size in X- and Y-direction is at most $\frac{|V'|}{2} - 1$ (because we do not count the increase of the size by the virtual bent edge (v_x, v_1)). We now replace V' by the edge (v_x, v_1) between the cutting pair of V', with labels $lth_X((v_x, v_1))$ and $lth_Y((v_x, v_1))$, denoting the length in X- and Y-direction, respectively, of V'.

Assume all triconnected components V' of a 2-subgraph W' are replaced by a vertex between their cutting pair. We first eliminate vertices $v \in W'$ of degree two while connecting their neighbours v', v'', with $lth_X((v', v'')) = lth_X((v, v')) + lth_X((v, v''))$ and similar for lth_Y. Similar we replace multiple edges by one edge e with $lth_X(e)$ and $lth_Y(e)$ the sum of lth_X and lth_Y of the multiple edges. Notice that after this W' is a triconnected component and 3-planar as well, hence after adding a virtual edge between the cutting pair (v_x, v_1) of W', we can apply HEXA-DRAW.

However, for every edge e we also add $lth_X(e)$ and $lth_Y(e)$ to $lth(e')$, with e' the basis-edge of F_k, if e is a new vertex of F_k. Also in the algorithm we must have that $lth(e) \geq lth_X(e) + lth_Y(e)$, which can be obtained as follows:

Set $lth(e) = \max\{1, lth_X(e) + lth_Y(e)\}$, for all edges $e \in W'$;
for $k := 3$ **to** $f - 1$ **do**
$\quad lth(be(F_k)) := \max\{\sum_{e \in E(F_k)} lth(e) - 1, lth_X(be(F_k)) + lth_Y(be(F_k))\}$;
$lth(be(F_f)) := \max\{\sum_{e \in E(F_f)} lth(e), lth_X(be(F_f)) + lth_Y(be(F_f))\}$;

We now modify HEXA-DRAW as follows such that for all edges (a, b) holds that $x(v_b) \geq x(v_a) + lth_X((v_a, v_b))$ and $y(v_b) \geq y(v_a) + lth_Y((v_a, v_b))$. Suppose we add the new vertices $w_1, \ldots, w_p$ ($p \geq 2$) of face F_k from c_i to c_j, then we place w_1 on $(x(c_i) + lth_X((c_i, w_1)), y(c_i) + \max\{1, lth_Y((c_i, w_1))\})$. Given the coordinates of w_i, we can place w_{i+1} on $(x(w_i) + \max\{1, lth_X((w_i, w_{i+1}))\}, y(w_i) + lth_Y((w_i, w_{i+1})))$. Assume $y(c_i) \geq y(c_j)$. Since we added $lth_X((w_i, w_{i+1}))$ and $lth_Y((w_i, w_{i+1}))$ to $lth(e')$, with $e' = be(F_k)$, we obtain that there is enough space to place $w_1, w_2, \ldots, w_p$ between c_i and c_j. Similar to lemma 2.4 we can prove that for all edges $e \in W'$, $length_X(e) \geq lth_X(e)$ and $length_Y(e) \geq lth_Y(e)$, where $length_X(e)$ and $length_Y(e)$ denote the length of e in X- and Y-direction of the drawing, respectively.

After computing the coordinates of all 2-subgraphs $V' \subseteq G$, for which v' is a child of w' in T, we replace them by an edge between its cutting pair in W'. After eliminating multiple edges and vertices of degree two, we compute the coordinates of the vertices in W'. We repeat this until we are at the root r of T. Computing the final coordinates of all vertices in the 2-subgraph V' is now straightforward:

exactly one horizontal edge e and all left internal edges of e are of Z-direction and all edges right from e are of Y-direction. □

Lemma 2.3 *e is drawn horizontal $\Longleftrightarrow$ e is a basis-edge.*

Proof: $\Longrightarrow$ Let $(c_\alpha, c_{\alpha+1})$ be the horizontal internal edge when adding F_k to G_{k+1}. By lemma 2.1, such an edge exists. All internal edges of F_k left (right) from c_α have Z-direction (Y-direction) upwards by lemma 2.2. But these edges are added after c_α, because c_α is the rightmost vertex of the face when adding $(c_{\alpha-1}, c_\alpha)$ by the algorithm HEXA-DRAW. Similar for $c_{\alpha+1}$. But then $(c_\alpha, c_{\alpha+1})$ belongs to the highest numbered adjacent face of F_k, hence $(c_\alpha, c_{\alpha+1})$ is the basisedge.

$\Longleftarrow$ Suppose e is a horizontal edge, belonging to F_i and $F_j, j > i$. Suppose $be(F_i) = e'$, with $e' \neq e$. We know already that e' is horizontal, but then there are two horizontal internal edges when adding F_i. But this contradicts with lemma 2.2.□

Lemma 2.4 *For each edge e, $length(e) \geq lth(e)$.*

Proof: By reverse induction on the faces F_k. The basis-edge of F_f is drawn with length k_1. The distance between v_z and v_y is equal to k_1, which is equal to the sum of $lth(e)$ for all edges e between v_z and v_y, hence the lemma is correct for F_f.

Suppose the lemma is correct for $i = k+1, \ldots, f$. We show that we add F_k by HEXA-DRAW such that $length(e) \geq lth(e)$ for all edges in G_k. Let $e' = (c_\alpha, c_{\alpha+1}) = be(F_k)$. From c_α we have ≥ 0 edges in Z-direction upwards to c_i (startpoint) and from $c_{\alpha+1}$ we have ≥ 0 edges in Y-direction upwards to c_j (endpoint). Assume w.l.o.g. that $y(c_i) \geq y(c_j)$ and that we add at least two vertices $w_1, \ldots, w_p$. From c_i we go one step in Y-direction to place w_1. From c_j we go in Z-direction to the same height to place w_p. (c_i, w_1) and (w_p, c_j) are not basis-edges, thus $lth(e) = 1$ and thus $length(e) \geq lth(e)$ for (c_i, w_1) and (w_p, c_j). Moreover, $x(w_p) - x(w_1) = x(c_j) + y(c_j) - y(w_1) - x(w_1) = x(c_j) - x(c_i) - (y(c_i) + 1 - y(c_j))$. Notice that from c_α to c_i we go in Z-direction upwards, thus $x(c_i) + y(c_i) = x(c_\alpha) + y(c_\alpha)$, and from $c_{\alpha+1}$ to c_j we go in Y-direction upwards, thus $x(c_j) + y(c_j) \geq x(c_{\alpha+1}) + y(c_{\alpha+1})$. Note also that $y(c_{\alpha+1}) = y(c_\alpha)$, as $(c_\alpha, c_{\alpha+1}) = be(F_k)$, hence horizontal. Thus $x(w_p) - x(w_1) = x(c_j) - x(c_i) - (y(c_i) + 1 - y(c_j)) \geq x(c_{\alpha+1}) - x(c_\alpha) - 1 \geq lth(e') - 1$ by induction. $lth((c_i, w_1)) = lth((w_p, c_j)) = 1$, thus $\sum_{1 \leq i < p} lth((w_i, w_{i+1})) = lth(e') - 1$, hence also in G_k all edges e have length at least $lth(e)$. □

Since $lth(e) \geq 1$, this lemma proves the correctness of the algorithm HEXA-DRAW.

Lemma 2.5 *The size of the hexagonal grid is $\frac{n}{2} \times \frac{n}{2}$.*

Proof: There are $\frac{3}{2}n$ edges and $\frac{n}{2} + 2$ faces. F_2 and F_1 do not have basis-edges, hence there are $\frac{n}{2}$ basis-edges. Each edge, except two incident edges of v_x, are added

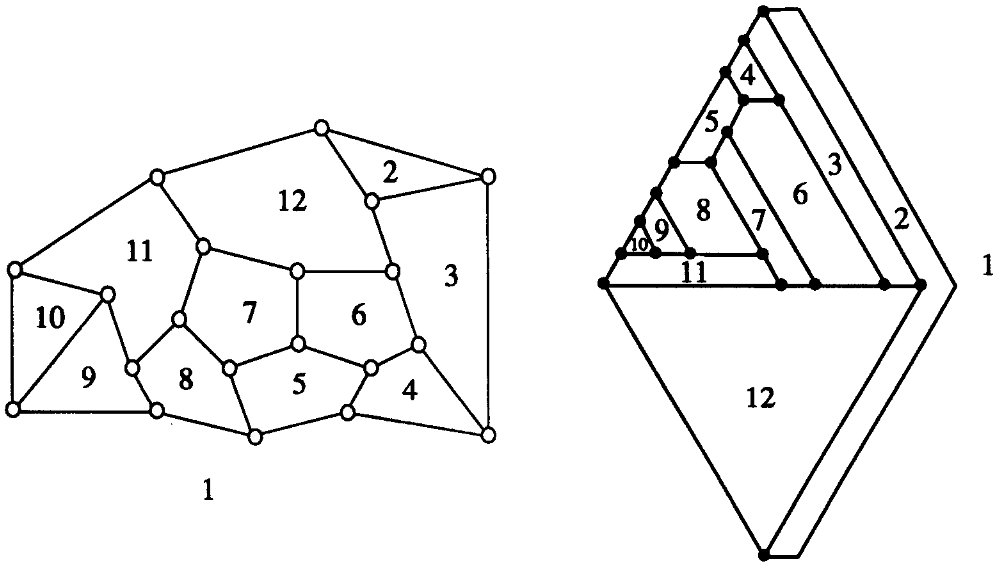


Figure 4: A graph with canonical numbered faces and the corresponding drawing by HEXA-DRAW.

to lth of a basis-edge. The initial length of the $\frac{n}{2}$ basis-edges is ignored when calculating $lth(e)$. In the calculation, $\frac{n}{2} - 1$ times the value 1 is subtracted from $lth(e)$. Thus $k_1 = lth(be(F_f)) = \frac{3}{2}n - 2 - \frac{n}{2} - (\frac{n}{2} - 1) = \frac{n}{2} - 1$. Adding the edge from v_x to v_1 increases the size of both the Y- and Z-direction by one. From v_x we can now visit all other points by walking at most $\frac{n}{2}$ in Y- and Z-direction, thereby proving the lemma. □

In figure 4 an example is given of a drawing of a triconnected 3-planar graph.

We can use the algorithm HEXA-DRAW as follows to draw triconnected planar graphs of degree at most 6 on a hexagonal grid: replace every vertex v by a cycle $C(v)$ of length $deg(v)$ where every vertex of $C(v)$ has an edge to a neighbor of v. Applying HEXA-DRAW to the augmented graph and finally replacing every cycle $C(v)$ by vertex v, one can easily prove the following result:

Theorem 2.6 *There is a linear algorithm to draw a triconnected planar graph of degree ≤ 6 on an $O(n^2)$ hexagonal grid such that there are at most $O(n)$ bends.*

In [13] a linear algorithm is presented to draw a 4-planar graph on an $O(n^2)$ rectilinear grid with at most $O(n)$ bends, hence theorem 2.6 extends this result in a positive way to 6-planar graphs.

We can use a modification of HEXA-DRAW, such that we obtain straight line drawings of a triconnected 3-planar graph G with vertices on an $\frac{n}{2} \times \frac{n}{2}$ rectilinear grid. For this take the Y-axis perpendicular to the X-axis, and let the Z-axis make assume $V' \subseteq W'$ and we already know the final coordinates of vertices $\in W' - V'$, then we replace the virtual edge (v_x, v_1) in W' by the corresponding 2-subgraph V', and adding $x(v_x)$ and $y(v_x)$ to the coordinates of each vertex in V" implies the final coordinates. It follows that we can place v_x and v_1 such that $x(v_1) = x(v_x) + lth_X((v_x, v_1))$ and $y(v_1) = y(v_x) + lth_Y((v_x, v_1))$. It is not hard to find the precise construction, but a little tedious. We omit the details here. Hence we can place V' inside W' with edges of correct length and without crossing edges.

Notice finally that vertices of degree two are no problem in G, because we can eliminate them while connecting their neighbors and updating $lth_X(e)$ and $lth_Y(e)$. This completes the following theorem:

Theorem 3.1 *There is a linear algorithm to draw a connected 3-planar graph on a hexagonal grid with at most one bent edge.*

One may observe that for connected 3-planar graphs, the size of the hexagonal grid is not necessarily polynomially bounded. Hereto we define a class of 3-planar graphs as follows: G_1 consists of four vertices p_1, q_1, s_1 and t_1; an edge (p_1, q_1) and edges from s_1 and t_1 to both p_1 and q_1. G_{n+1} is obtained from G_n by adding four vertices $p_{n+1}, q_{n+1}, s_{n+1}$ and t_{n+1}; the edges $(p_{n+1}, s_n), (q_{n+1}, t_n)$ and edges from s_{n+1} and t_{n+1} to both p_{n+1} and q_{n+1} (see the drawing of figure 8(a)).

Theorem 3.2 *Any drawing of G_n with at most one bent edge on a hexagonal grid requires an $\Omega(n2^n)$ hexagonal grid.*

Proof: Let $X(G_n)$ and $Y(G_n)$ be the required distance in X- and Y-direction of any hexagonal drawing of G_n. By geometric considerations we can prove that $X(G_{n+1}) \geq 2X(G_n) + 2$ and $Y(G_{n+1}) \geq Y(G_n) + 2$. Since $X(G_1) = Y(G_1) = 1$, we have $X(G_n) \geq \Omega(2^n)$ and $Y(G_n) = 2n - 1$. □

4 Drawings with straight lines

In this section we answer the following question, posted by Michael Formann et al. [5]:

> Does every degree 3 planar graph have a planar embedding with smallest angle at least a constant, independent of the number of vertices?

The algorithm HEXA-DRAW does not give a direct answer to this problem, since there is one edge in which bends may occur. But we will show here how the algorithm can be used to draw every 3-planar graph with smallest angle $\geq \pi/6$. This angle is also best possible, since a K_4 can only be drawn with smallest angle equal to $\pi/6$.

Lemma 4.1 *There exists a canonical numbering of a triangulated planar graph H with f vertices such that v_{f-1} is a neighbor of both v_1 and v_f.*

Proof: Assume we have a canonical numbering in which $deg(v_f)$ is as small as possible, thus $deg(v_f) \leq 5$. If $deg(v_f) = 3$, then v_{f-1} is a neighbor of v_1. If $deg(v_f) = 4$ then v_{f-1} is a neighbor of v_1 or v_2. Swapping v_1 and v_2 if necessary gives the result. Suppose further that $deg(v_f) = 5$. The neighbors of v_f are v_1, v_i, v_j, v_k and v_2 respectively in counterclockwise order. If $f - 1 = i$ or k then by swapping v_1 and v_2 if necessary we obtain that v_{f-1} is a neighbor of v_1, so assume that we cannot number the vertices such that $i = f - 1$ or $k = f - 1$, thus $j = f - 1$. Assume $\max\{i, k\} = i$, then vertex v_{i+1} must have only vertex v_i and v_k as lower numbered neighbors. Suppose not, then we could add v_{i+1} before v_i in the canonical numbering. If $j \neq i + 1$, then since v_{i+1} is not a neighbor of v_f, v_{i+1} and v_k have a common higher numbered neighbor v_l, which can be added before v_i. Repeat the argument with $j \neq l$, etc., and it follows that we can add v_j before v_i by the canonical numbering which means that $i = f - 1$. Contradiction, thus v_{i+1} has v_i and v_k as only lower numbered neighbors, thus there is an edge (v_i, v_j) in H. Hence H contains a K_4 on the vertices v_i, v_j, v_k, v_f.

Do now again the canonical numbering $v'_1, v'_2, \ldots, v'_f$ such that $v'_1 = v_i, v'_2 = v_j$ and $v'_f = v_f$. v'_f again has degree 5, viz., the neighbors $v_i, v'_\alpha, v'_\beta, v_k$ and v_j, in this order. If v'_α or $v_k = v'_{f-1}$ then v'_{f-1} is a neighbor of v'_1, so assume $\beta = f - 1$, and $v_k = v'_\gamma$. Then all vertices $v'_3, \ldots, v'_{\gamma-1}$ are inside triangle v'_1, v'_2, v'_γ and these numbers are independent from the numbering of $v'_{\gamma+1}, \ldots, v'_{f-1}$. We now renumber the vertices of triangle v'_1, v'_2, v'_γ such that v'_1 and v'_γ are swapped, i.e., the outerface is now v_k, v_j, v_f. But we did not change the numbers $v'_{\gamma+1}, \ldots, v'_{f-1}$, thus $v'_\beta = v'_{f-1}$ is a neighbor of v'_1. Thus there exists a numbering such that v_{f-1} is a neighbor of both v_1 and v_f. □

We may assume w.l.o.g. that the planar graph is triconnected. If not, then we can apply the generalization of section 3. Let a canonical numbering of the faces of G be given, satisfying lemma 4.1. Such a numbering can be achieved in linear time. Let $k_2 = lth(be(F_{f-1}))$ and let $k_1 = \frac{n}{2} - 1 - k_2$. We assume that $k_2 \geq k_1$, otherwise we simply put $k_2 = k_1$ (this only enlarges the drawing somewhat). Let $v_n \in F_f, F_2$ and F_1, with neighbors v_{n-1} and v_{n-2} in F_f, such that $v_{n-1} \in F_{f-1}$. Let v_α and v_β be the other neighbors of v_{n-1}, with v_β also $\in F_f$. Then we draw v_{n-1} and v_n on positions $(0,0)$ and $(k_1, 0)$, resp., and the vertices v_α, v_β and v_{n-2} on positions $(-k_2, k_2), (0, k_2)$ and (k_1, k_2), resp., as shown in figure 6(a). We draw the remaining vertices of F_{f-1} and F_f on the horizontal line between v_α and v_{n-2} with respect to the length of the basisedges. We now apply HEXA-DRAW to draw the faces $F_{f-2}, \ldots, F_3, F_2$. This gives a hexagonal drawing with only in (v_1, v_n) bends. To remove these bends, we move v_{n-1} and v_n to $(k_1, -k_1)$ and $(2k_1 + k_2, k_2)$, resp., as shown in figure 6(b). As we used the underlying hexagonal grid, it follows that all angles have size at least $\pi/6$, and only the angles $\angle v_{n-2}v_1v_n, \angle v_1v_nv_2, \angle v_{n-2}v_nv_{n-1}$ and $\angle v_\beta v_{n-1}v_\alpha$ can have size $< \pi/3$ (see the marked angles in figure 6(b)). If $n = 4$, then 6 angles have size $\pi/6$. This completes the following result:

$y(w_1) := y(w_2) := \ldots := y(w_p) := \max\{y(c_i), y(c_j)\}$

else

$\vdots$

In case we add one vertex w_1 and ($y(c_i) > y(c_j)$ **and** $y(c_i) > y(c_{i+1})$) holds, then $x(w_1) := x(c_j) + y(c_j) - y(c_i)$. To prove that the drawing algorithm HEXA-DRAW works correct after this modification, we prove the following variant of lemma 2.2. (Notice that lemma 2.1 still holds and that still holds: e is a basis-edge $\Longrightarrow$ e is drawn horizontal.)

Lemma 5.1 *All internal edges of a face F_k left from $be(F_k)$ are horizontal or upwards in Z-direction. All internal edges of F_k right from $be(F_k)$ are horizontal or upwards in Y-direction.*

Proof: Suppose there are edges $(c_\alpha, c_{\alpha+1})$ and $(c_\beta, c_{\beta+1})$ in Y- and Z-direction on one side of a horizontal edge, with $\alpha + 1 \leq \beta$. If $\alpha + 1 = \beta$ then by definition c_β has degree 2 in G_{k+1} and hence must be a start- or endpoint. If $\alpha + 2 = \beta$ then there is only one horizontal edge between the edges of Y- and Z-direction, thus $c_{\alpha+1}$ or c_β must have degree 2 in G_{k+1}. If $\beta > \alpha + 2$ then there are more horizontal consecutive edges. If these edges belong to one face of G_{k+1} then the internal vertices $c_{\alpha+2}, \ldots, c_{\beta-1}$ have degree 2 in G_{k+1}, otherwise again $c_{\alpha+1}$ or c_β has degree 2 in G_{k+1}.

Hence there is a horizontal edge $(c_\alpha, c_{\alpha+1})$ such that left from c_α all internal edges are horizontal or upwards in Z-direction. Right from $c_{\alpha+1}$ all internal edges are horizontal or upwards in Y-direction. Similar to lemma 2.3 we can prove that we can choose α such that $(c_\alpha, c_{\alpha+1})$ is the basis-edge. □

This lemma implies that in some cases we can decrease the total height considerably. Another optimization is the following. Let a canonical numbering of H be given such that v_{f-1} is a neighbor of v_f and v_1 (satisfying lemma 4.1) and the hexagonal drawing as defined in section 4. Let k_1, k_2, v_α and v_{n-2} as defined in section 4. If k_2 is small, then the first horizontal line between v_α and v_{n-2} is on lower height. Hence we have to find a canonical numbering of H such that in G the length k_2 is as small as possible. Though this is not easy in general, it becomes solvable when there is a triangle F_t in G. Thus there is a vertex $v_t \in H$ with degree 3. Number H using lemma 4.1 such that $t = f - 1$. v_t has v_1, v_{f-2} and v_f as neighbors. This implies that no vertex $v \in H$ has v_{f-1} as highest numbered neighbor, because v_{f-2} has both v_{f-1} and v_f as neighbors. Using the modified hexagonal drawing algorithm of section 4 this leads to a hexagonal drawing in which $k_2 = 1$. Hence this decreases the height of the drawing considerably.

The last optimization we notice is when $x(w_p) - x(w_1) > \sum_{1 \leq i < p} lth((w_i, w_{i+1}))$ when adding face F_k. This is the case when $y(c_j) > y(c_{\alpha+1})$, with $(c_\alpha, c_{\alpha+1})$ the basis-edge of F_k. In HEXA-DRAW this leads to a drawing with $length((w_{p-1}, w_p)) >$

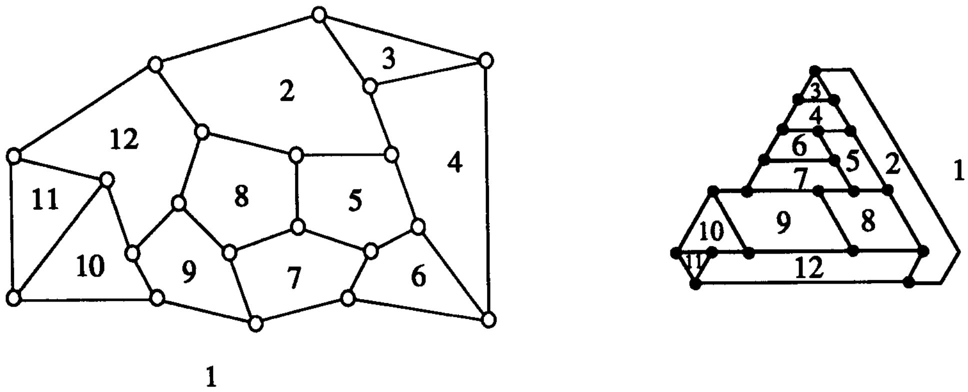


Figure 7: Optimizing the drawing of the graph in figure 4.

$lth((w_{p-1}, w_p))$. We can now subtract $length((w_{p-1}, w_p)) - lth((w_{p-1}, w_p))$ from $lth((c_\alpha, c_{\alpha+1}))$. We update $lth((a, b))$ for all basis-edges (a, b) by visiting $F_3, F_4, \ldots, F_f$, in this order after HEXA-DRAW. Using the new lth's of the basis-edges we again apply HEXA-DRAW to draw G in linear time on a hexagonal grid of smaller size.

In figure 7, these optimizations have been applied to the example, given in figure 4.

6 Final Remarks

In this paper we considered drawings of planar graphs with degree at most 3 on a hexagonal grid. A simple linear algorithm HEXA-DRAW for this problem is described, leading to a linear-sized grid in the case of triconnected 3-planar graphs. Using HEXA-DRAW we can draw every triconnected 3-planar graph with straight lines on a $\frac{n}{2} \times \frac{n}{2}$ grid, improving the best bound known by a factor 4. Whether there exists triconnected 3-planar graph for which any straight-line drawing requires an $\frac{n}{2} \times \frac{n}{2}$ grid remains as an open problem. In figure 8(a) an example of a planar graph of degree 3 is given, requiring an $(\frac{n}{2} + 1) \times (\frac{n}{4} + 1)$ grid for a straight-line embedding ($n = 8k$, for some integer $k > 0$), but it is not triconnected. In figure 8(b) a planar graph of degree at most 4 is shown, for which every straight-line drawing requires an $\frac{2}{3}(n-1) \times \frac{2}{3}(n-1)$ grid, if this embedding is used, and $\frac{n+1}{2} \times \frac{n-1}{3}$ otherwise ($n = 6k+1$, for some integer $k > 0$). This gives some indication of the tightness of our algorithm. The triconnected 3-planar graph of figure 8(c) requires an $(\frac{n-1}{3} + 2) \times (\frac{n-1}{3} + 2)$ grid, if this embedding is used, and $(\frac{n-1}{4} + 4) \times (\frac{n-1}{6} + 3)$ grid otherwise ($n = 12k + 1$, for some integer $k > 0$).

Recently it has been shown that this canonical numbering can be generalized to triconnected planar graphs. This leads to a general method, working in linear time for drawing planar graphs in several ways, dealing with the grid size, minimum angle

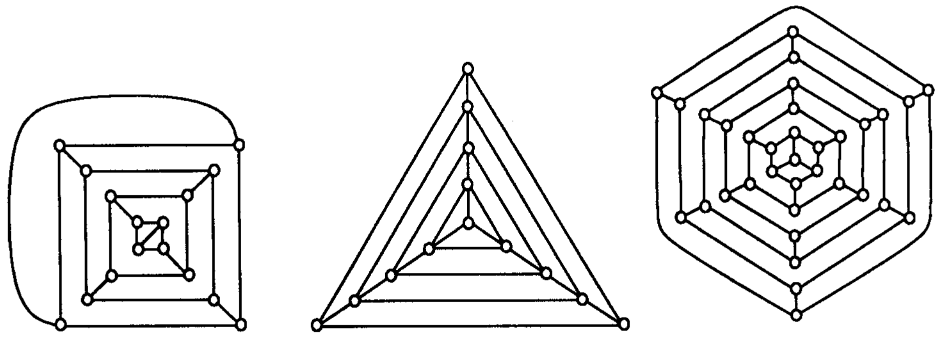


Figure 8: Examples of planar graphs for grid size lower bounds.

and number of bends [7]. This also leads to several improvements for orthogonal grid drawings and convex straight-line drawings.

Open problems include:

- Devise a dynamic algorithm to draw a triconnected 3-planar graph hexagonal. Recently a framework for dynamic graph drawing is presented, requiring $O(\log n)$ amortized time for insertions and output-sensitive time for drawing queries in the case of undirected planar graphs [1].
- Devise a fast algorithm to draw planar graphs planar on the octagonal grid: the rectilinear grid plus the $\pi/4$ and $3\pi/4$ axis. We conjecture that triconnected planar graphs of degree 4 can be drawn on an octagonal grid such that at most K edges have bends, with K constant, independent of n. We are able to prove $K \leq n/2$ on an $n \times n/2$ octagonal grid.
- Develop algorithms that draw planar graphs orthogonal or hexagonal with the minimum number of bends and are simpler or faster than the $O(n^2 \log n)$ algorithm of Tamassia [11].

Acknowledgements

The author wishes to thank Hans Bodlaender for reading earlier drafts. He also thanks Klaus Kriegel for pointing out a small error in an earlier version.

References

[1] Cohen, R.F., G. Di Battista, R. Tamassia, I.G. Tollis and P. Bertolazzi, A Framework for dynamic graph drawing, to appear in: *Proc. ACM Symp. on Computational Geometry*, Berlin, 1992.

[2] Di Battista, G., and R. Tamassia, Algorithms for plane representations of acyclic digraphs, *Theoret. Comp. Science* 61 (1988), pp. 436–441.

[3] Di Battista, G., R. Tamassia and I.G. Tollis, Area requirement and symmetry display in drawing graphs, *Proc. ACM Symp. on Computational Geometry*, Saarbrücken, 1989, pp. 319–321.

[4] Di Battista, G., R. Tamassia and I.G. Tollis, Constrained visibility representations of graphs, *Inf. Proc. Letters* 41 (1992), pp. 1–7.

[5] Formann, M., T. Hagerup, J. Haralambides, M. Kaufmann, F.T. Leighton, A. Simvonis, E. Welzl and G. Woeginger, Drawing graphs in the plane with high resolution, *Proc. 31th Ann. IEEE Symp. on Found. of Comp. Science*, St. Louis, 1990, pp. 86–95.

[6] Fraysseix, H. de, J. Pach and R. Pollack, How to draw a planar graph on a grid, *Combinatorica* 10 (1990), pp. 41–51.

[7] Kant, G., *A New Method for Planar Graph Drawings on a Grid*, Tech. Report, Dept. of Comp. Science, Utrecht University, 1992 (to appear).

[8] Kant, G., and H.L. Bodlaender, Triangulating Planar Graphs While Minimizing the Maximum Degree, to appear in: *Proc. 3rd Scandinavian Workshop on Algorithm Theory (SWAT '92)*, Lecture Notes in Comp. Science, Springer-Verlag, Berlin/Heidelberg, 1992.

[9] Schnyder, W., Embedding planar graphs on the grid, in: *Proc. 1st Annual ACM-SIAM Symp. on Discr. Alg.*, San Fransisco, 1990, pp. 138–147.

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

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

[12] Tamassia, R., G. Di Battista and C. Batini, Automatic graph drawing and readability of diagrams, *IEEE Trans. on Systems, Man and Cybernetics* 18 (1988), pp. 61–79.

[13] Tamassia, R., and I.G. Tollis, Efficient embedding of planar graphs in linear time, *Proc. IEEE Int. Symp. on Circuits and Systems*, Philadelphia, pp. 495–498, 1987.

[14] Tamassia, R., I.G. Tollis and J.S. Vitter, Lower bounds for planar orthogonal drawings of graphs, *Inf. Proc. Letters* 39 (1991), pp. 35–40.