Minimum-Width Grid Drawings of Plane Graphs

Marek Chrobak* Shin-ichi Nakano†

Abstract

Given a plane graph G, we wish to draw it in the plane in such a way that the vertices of G are represented as grid points, and the edges are represented as straight-line segments between their endpoints. An additional objective is to minimize the size of the resulting grid. It is known that each plane graph can be drawn in such a way in a $(n-2) \times (n-2)$ grid (for $n \geq 3$), and that no grid smaller than $(2n/3-1) \times (2n/3-1)$ can be used for this purpose, if n is a multiple of 3. In fact, for all $n \geq 3$, each dimension of the resulting grid needs to be at least $\lfloor 2(n-1)/3 \rfloor$, even if the other one is allowed to be unbounded. In this paper we show that this bound is tight by presenting a grid drawing algorithm that produces drawings of width $\lfloor 2(n-1)/3 \rfloor$. The height of the produced drawings is bounded by $4\lfloor 2(n-1)/3 \rfloor - 1$. Our algorithm runs in linear time and is easy to implement.

1 Introduction

The problem of automatic graph drawing has recently attracted a lot of attention, due to its numerous practical applications and the challenging mathematical and algorithmic questions that arise in this area. Generally, given a graph G, the task is to produce an aesthetic drawing of G, one that accurately reflects the topological structure of G in a graphical form. Many versions of this problem have been considered, and there is a variety of techniques, algorithms, and software packages currently available. (See the survey in [DETT94] for more information.)

For planar graphs, this problem is especially interesting. In this case, we typically require that vertices are represented by points in the plane, and edges are drawn as non-intersecting straight-line segments between their endpoints. Additionally, we are often given a *plane graph*, that is a planar graph with a given planar embedding, represented combinatorially by cyclic orderings of edges incident to all vertices, and by the choice of the external face. (For 3-connected graphs the choice of the external face uniquely determines the embedding.) Then the drawing needs to be

*Department of Computer Science, University of California, Riverside, CA 92521. Research partially supported by NSF grants CCR-9112067 and CCR-9503498.

†Department of System Information Sciences, Tohoku University, Sendai 980-77, Japan.

consistent with that given planar embedding, in the sense that for each vertex v, the given cyclic ordering of edges incident to v needs to be the same as their clockwise ordering in the drawing.

In this paper we deal with the following problem: given a plane graph G, we want to map its vertices into integer grid points in such a way that the edges between them can be drawn as straight, non-intersecting line segments. The resulting drawing has to be consistent with the planar embedding of G. We call such mappings *grid drawings*.

Restricting vertex coordinates to integer values has been motivated by the fact that using arbitrary real values leads to problems with rounding errors. Also, integer vertex coordinates facilitate the display of the drawing on raster graphics devices.

It has been proven that each plane graph has a straight-line drawing [Fa48, Wa36, St51]. Some algorithms for computing such drawings were proposed in [CYN84, CON85]. All these methods, however, use real-valued coordinates. Of course, we can approximate real vertex coordinates by rational numbers, and then use appropriate scaling, but the grids obtained by following this method are, unfortunately, of exponential size.

The question whether smaller, polynomial-size, grids can be used for this purpose was open until 1988, when de Fraisseix, Pach and Pollack [FPP88, FPP90] proved that each plane graph with n vertices can be embedded into a $(2n-4) \times (n-2)$ grid. (Throughout the paper we assume that $n \geq 3$.) We will refer to their method as the *shift method*. Their paper initiated intensive research in this area, and led to new results and implementations. Chrobak and Payne [CP95] gave a simple, linear-time implementation of the shift method. Schnyder [Sc90] presented a different technique, based on barycentric representations, that led to smaller grid drawings of size $(n-2) \times (n-2)$. He also pointed out later (personal communication) that there is a close relationship between the shift method and barycentric representations, and that the grid size in [CP95] can be reduced to $(n-2) \times (n-2)$ without affecting time complexity. As it was shown in [Jon93], the shift method produces high quality drawings, and compares favorably with other techniques.

In the work mentioned above, it is usually assumed that the given plane graph is triangulated. Otherwise, one can always triangulate a given graph and remove the added edges after constructing the drawing. This approach, however, leads to poor quality drawings, so the question was raised whether aesthetic drawings can be constructed efficiently without a prior triangulation. For non-triangulated plane graph, one criterion of aestheticity is that the faces should be represented by convex polygons. This can always be achieved when the graph is 3-connected [Tu63]. G. Kant [Ka93], working in this direction, proved that each 3-connected plane graph has a convex drawing in a $(2n-4) \times (n-2)$ grid, and the grid size was recently improved to $(n-2) \times (n-2)$ by Schnyder and Trotter [ST92] and Chrobak and Kant [CK93], independently. All these algorithms can be implemented in linear time.

The obvious question is: what is the minimum size of grid drawings? In their paper, [FPP88] proved that, in the worst case, no grid smaller than $(2n/3-1)\times(2n/3-1)$ is possible for n-vertex plane graphs, if n is a multiple of 3. The simple argument they presented can be easily modified to show that, for all $n \geq 3$, each dimension of the grid needs to be at least $\lfloor 2(n-1)/3 \rfloor$, even if the other one is unbounded.

It is important to note that the distinction between planar and plane graphs is essential. If we are given a *planar* graph G on input, and we are allowed to choose its planar embedding, then the known $\lfloor 2(n-1)/3 \rfloor$ lower bound proof does not apply. We discuss this issue in Section 6.

In this paper we show that this bound is tight, by presenting an algorithm that embeds each n-vertex plane graph into a grid of width at most $\lfloor 2(n-1)/3 \rfloor$. The height of the resulting drawings is at most $4\lfloor 2(n-1)/3 \rfloor - 1 \leq 8n/3 - 3$. Our algorithm runs in linear time, and is easy to implement. However, the correctness proof and the grid size estimate are quite difficult.

To some degree, our method is a continuation of the work from [FPP90, CP95, Sc90, CK93]. The reduction of the grid width, however, is based on a novel technique based on a lemma describing a combinatorial structure of *domino chains* in planar graphs. The number of the domino chains, as we show, is closely related to the width of the graph. This technique is of its own interest, and we believe it will be useful in other algorithms for drawing planar graphs. In fact, recently, He [He95] investigated the class of 4-connected planar graphs whose external face is not a triangle, and used our domino-chain method as a base for his algorithm for drawing such graphs in small grids.

Besides grid drawings of plane graphs, area requirements have been studied for other classes of graphs, under various aestheticity criteria. Rectilinear drawings are often considered, when edges are restricted to be horizontal or vertical line segments, with or without bends (see, for example, [DLT84]). A lot of attention has been given to tree drawing. For example, Crescenzi, Di Battista and Piperno [CDP92], and Garg, Goodrich and Tamassia [GGT93] investigate the problem for upward drawings of trees. Planar upward drawings are studied in [DTT92]. See the survey in [DETT94] for more references.

In Section 2 we introduce our notation and terminology. In Section 3 we present a generic shift method for grid drawings, of which the algorithms from [FPP88, CP95] and the restriction of [CK93] to triangulated plane graphs are special instances. Our algorithm is based on the shift method as well. In Section 4, for the sake of presentation, we introduce a simplified version of our algorithm, called Algorithm $\mathcal{A}$, that illustrates the main idea for reducing grid width. Algorithm $\mathcal{A}$ produces drawings of width $\lfloor 2(n-1)/3 \rfloor$, but the height can be quadratic (it is bounded by $n^2/4$). Later, in Section 5, we present Algorithm $\mathcal{B}$ that uses the same width but reduces the height to $4\lfloor 2(n-1)/3 \rfloor - 1$.

2 Preliminaries

Canonical orderings. Let $G = (V, E)$ be an arbitrary triangulated plane graph with n vertices, where $n \geq 3$, and $\pi = v_1, \ldots, v_n$ an ordering of V such that (v_1, v_2, v_n) is the external face of G in counter-clockwise order. Define G_k to be the subgraph of G induced by $v_1, \ldots, v_k$ and C_k to be its external face. We say that π is a *canonical ordering of* G if the following conditions are satisfied for each $k = 3, \ldots, n$:

(co1) Each G_k is 2-connected and internally triangulated (that is, all internal faces of G_k are triangles).

(co2) C_k contains (v_1, v_2).

(co3) If $k < n$, then v_{k+1} is in the exterior face of G_k.

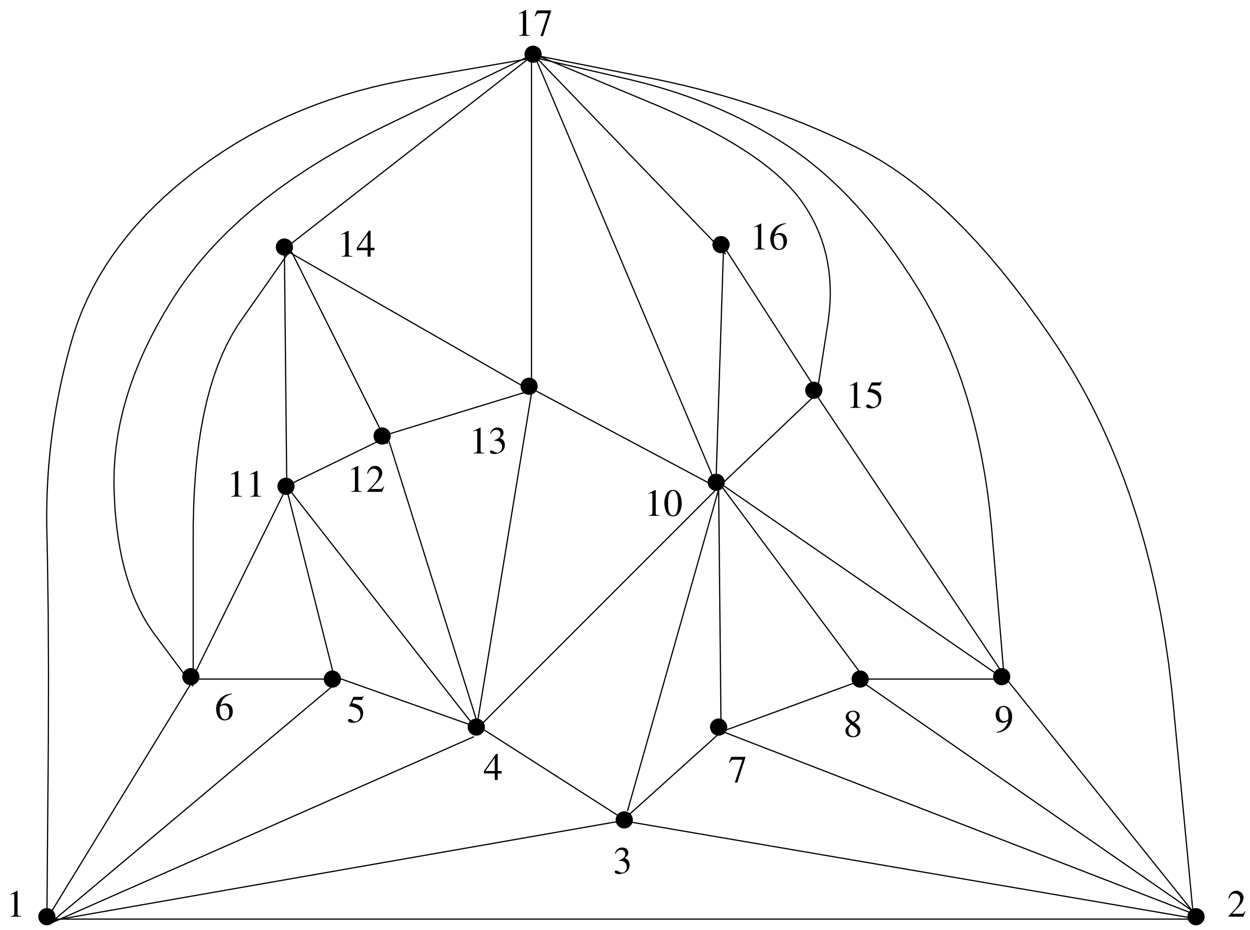

Figure 1: Canonical ordering of a graph G.

It is easy to see that Conditions (co1) and (co3) imply that the neighbors of v_{k+1} in G_k must belong to C_k and must, in fact, be consecutive on C_k. The existence of canonical orderings was proven by de Fraisseix, Pach and Pollack in [FPP88] (See also [Ka93]). Canonical orderings (and their extensions to 3-connected graphs) were used in [FPP88, CP95, Ka92, CK93] for graph drawing algorithms. An example of a canonical ordering is given in Figure 1.

Lemma 1 *[FPP88] Let G be a triangulated plane graph, and (v_1, v_2, v_n) the external face of G in counter-clockwise order. Then there exists a canonical ordering $\pi = v_1, v_2, \ldots, v_n$ of G, and π can be constructed in linear time.*

By an *ordered* triangulated plane graph (G, π) we will mean a triangulated plane graph G with a given canonical ordering $\pi = v_1, \ldots, v_n$, where (v_1, v_2, v_n) is the external face of G in counter-clockwise order. We will use symbol $\prec$ to denote the linear order given by the canonical ordering, that is, if $i < j$, then we will write $v_i \prec v_j$.

By the *contour of* G_k we mean its external face written as

$$C_k = (v_1 = w_1, w_2, \ldots, w_m = v_2).$$

Note that $w_1, \ldots, w_m$ are ordered clockwise on C_k.

Let $3 \leq k < n$, and let the neighbors of $v = v_{k+1}$ in G_k be $w_p, w_{p+1}, \ldots, w_q$. The *in-degree of* v, denoted $deg^-(v)$, is the number of neighbors of v in G_k, that is $deg^-(v) = q - p + 1$. For $i = p, \ldots, q$, we denote $ind_v(w_i) = i - p + 1$ and call it the *index of* w_i *with respect to* v. For $k = 1, 2, 3$, $deg^-(v_k)$ and ind_{v_k} are undefined. Obviously, $deg^-(v_k) \geq 2$ for each $k = 4, \ldots, n-1$ and $deg^-(v_n) \geq 3$.

Example: Consider the ordered graph (G, π) in Figure 1. We identify vertices by their ranks in the canonical ordering. Then, $deg^-(4) = 2$, $deg^-(9) = 2$, $deg^-(10) = 5$, $deg^-(11) = 3$, $ind_4(3) = 2$, $ind_{10}(8) = 4$, $ind_{14}(12) = 3$. ♠

Grid Drawings. Let N denote the set of non-negative integers, and G be a given plane graph with vertex set V. (Recall that a plane graph is a planar graph which is already embedded in the plane.) Let $P = (x, y) : V \to N \times N$ be a one-to-one function that maps V into an integer grid, where $x(v)$ and $y(v)$ represent the x and y coordinates of a vertex v. Then we can also think of P as a straight-line drawing of G by mapping each edge (u, v) of G into the straight-line segment $P(u, v) = [P(u), P(v)]$. If this mapping P is a correct planar embedding of G which is topologically equivalent to G, then we say that P is a *grid drawing* of G.

When P is understood from context, we will often identify a vertex v with its embedding $P(v)$ and, similarly, an edge (u, v) with the line segment $P(u, v) = [P(u), P(v)]$. For example, we will frequently refer to a slope of an edge, while meaning the slope of the line segment corresponding to this edge.

The *width* of a given drawing P is defined as the difference between the x-coordinates of the leftmost and rightmost vertices, that is $\max_{u,v} |x(u) - x(v)|$. The *height* is defined similarly: $\max_{u,v} |y(u) - y(v)|$.

By a minor modification of the construction in [FPP88], we obtain the following theorem.

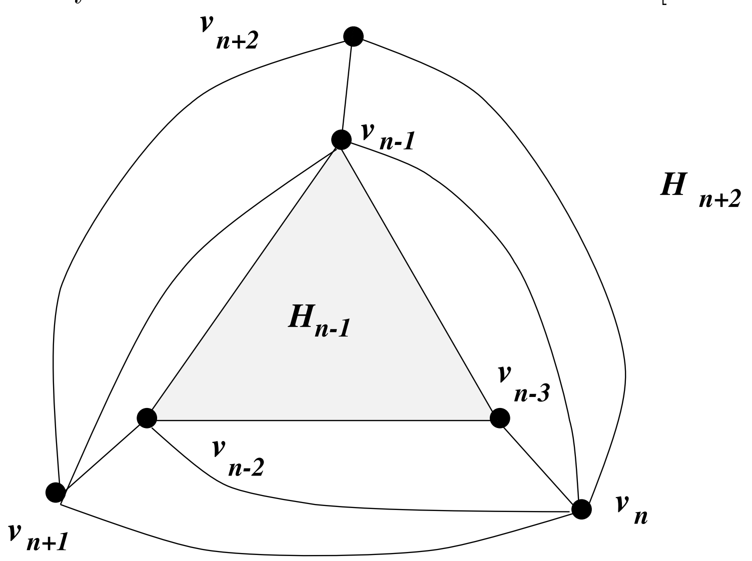

Figure 2: The construction of graph H_n from Theorem 1.

Theorem 1 *For each $n \geq 3$ there is an n-vertex plane graph H_n such that the width and height of each grid drawing of H_n is least $\lfloor 2(n-1)/3 \rfloor$.*

Proof: It is sufficient to consider the width only. We construct H_n recursively. H_3 is the triangle (v_1, v_2, v_3) and, for $n \geq 4$, H_n is obtained by adding vertex v_n to the outer face of H_{n-1} and connecting it to $v_{n-3}, v_{n-2}, v_{n-1}$ in such a way that the outer face of H_n is (v_n, v_{n-1}, v_{n-2}).

First, notice that for $n = 3, 4, 5$, H_n requires width $1, 2, 2$, respectively, which equals $\lfloor 2(n-1)/3 \rfloor$. The theorem follows by induction, since adding v_{n+1}, v_{n+2} and v_{n+3} to H_n forces us to use at least two more x-coordinates. $\square$

3 The Shift Method for Grid Drawings

Throughout the paper we will assume that the given input plane graph is triangulated. Each planar graph can be easily triangulated in linear time (see, for example, [Ka93]), and after the grid drawing is found, the added edges can be removed.

Let (G, π) be a given ordered triangulated plane graph, where $\pi = v_1, \ldots, v_n$ and $n \geq 3$. Our general strategy is similar to the methods from [FPP90, CP95]: we add vertices one at a time, in canonical order. At every time step, the contour C_k satisfies a certain invariant that involves restrictions on the slopes of contour edges. When adding a vertex v_{k+1} we determine its location in the grid and, if necessary, shift some parts of G_k to the right in order to preserve the invariant. The difficult part is to determine which internal vertices of G_k can be shifted to the right without violating planarity. We will describe such a method in this section.

The U-sets. We will maintain a set $U(v)$ for each vertex v. This set will contain vertices located "under" v that need to be shifted whenever v is shifted. Initially, $U(v_i) = \{v_i\}$ for $i = 1, 2, 3$. Suppose that $3 \le k \le n-1$ and that we are about to add v_{k+1} to G_k. Let the contour of G_k be

$$C_k \quad = \quad (v_1 = w_1, w_2, \ldots, w_m = v_2)$$

and let $w_p, \ldots, w_q$ be the neighbors of v_{k+1} in G_k. Then we set

$$U(v_{k+1}) \quad \leftarrow \quad \{v_{k+1}\} \cup \bigcup_{i=p+1}^{q-1} U(w_i).$$

Note that the sets $U(w_1), U(w_2), \ldots, U(w_m)$ form a partition of the vertices in G_k.

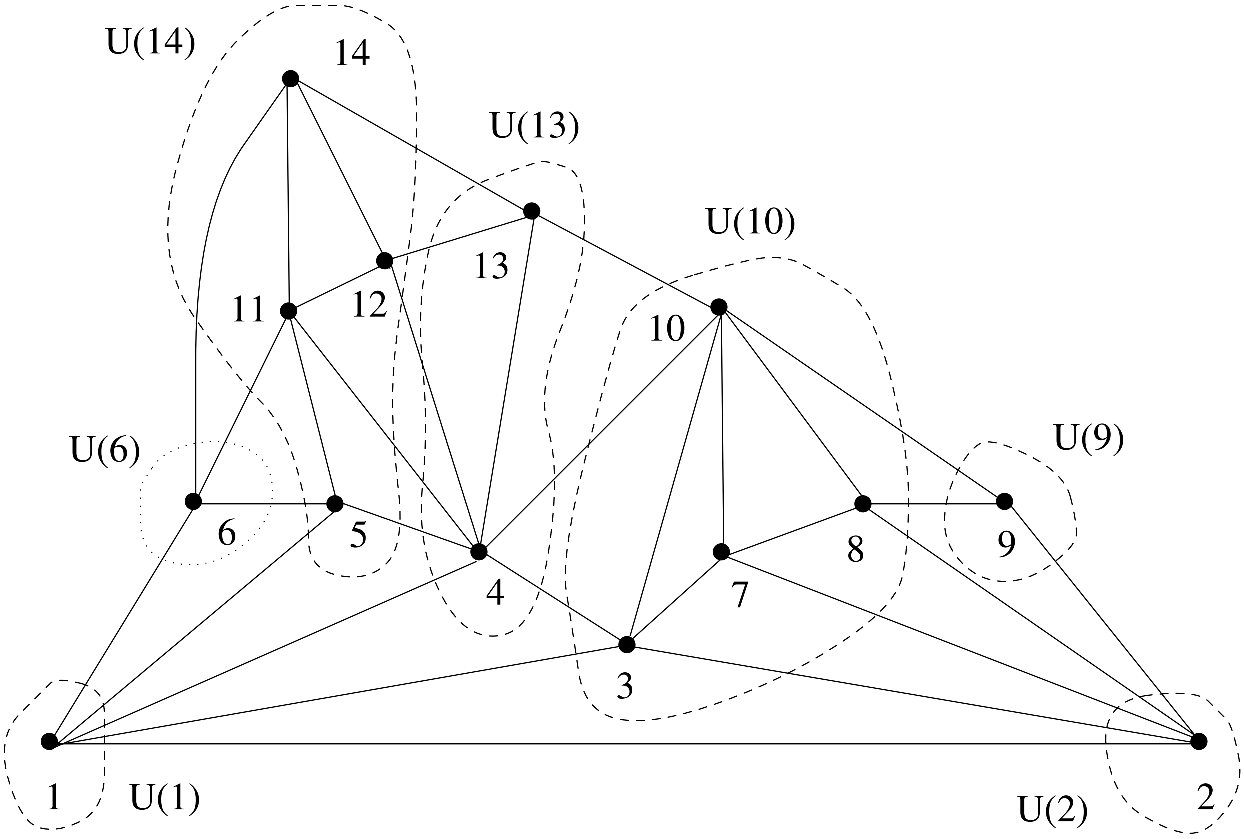

Figure 3: The U-sets for subgraph G_{14} of graph G from Figure 1.

The shift operation. Shifting a contour vertex w_j is achieved by operation *shift*(w_j), that increases the x-coordinate of each $u \in \bigcup_{i=j}^{m} U(w_i)$ by 1.

In the original shift method [FPP88], all slopes in the contour are either 1 or -1. Preserving this invariant requires shifting w_{p+1} and w_q at each step, resulting in the drawing of width $2 + 2(n-3) = 2n-4$ and height $n-2$. This can be improved by using slopes which are either -1 or arbitrary nonnegative numbers (see [CK93]). In this method, each step involves one shift only. This leads to drawings of width $2 + (n-3) = n-1$ and height $n-1$, which can be improved to $(n-2) \times (n-2)$ by handling the last vertex in a special manner. Our method will in fact avoid making any shifts in approximately $n/3$ steps.

Generic Shift Algorithm. Initially, v_1, v_2, v_3 are mapped into different grid points so that $x(v_2) > x(v_1) \geq 0$, and v_3 is located at a point satisfying $x(v_1) \leq x(v_3) \leq x(v_2)$, $y(v_3) \geq \max\{y(v_1), y(v_2)\}$, and the last inequality is strict when $y(v_1) = y(v_2)$.

Inductively, suppose that $3 \leq k \leq n-1$, that G_k has already been embedded, and that we are about to add $v = v_{k+1}$. Let $C_k = (w_1, \ldots, w_m)$ be the contour of G_k, and $w_p, \ldots, w_q$ be the neighbors of v in G_k. Apply *shift*(w_i) to some of $w_1, \ldots, w_m$ (possibly none), so that afterwards there is at least one point (x', y') inside the external face of G_k satisfying the following conditions:

(gsm1) $x(w_p) \leq x' \leq x(w_q)$,

(gsm2) $y' \geq \max\{y(w_{p+1}), y(w_{q-1})\}$, and

(gsm3) all vertices $w_p, \ldots, w_q$ are visible from (x', y').

Pick an arbitrary such point (x', y') and set $(x, y)(v) = (x', y')$.

In (gsm3), the term "visible" means that the edges from v to all $w_p, \ldots, w_q$ do not intersect the edges in C_k.

Lemma 2 *For all choices of shift operations and vertex coordinates, as long as (gsm1), (gsm2) and (gsm3) are satisfied, the Generic Shift Algorithm produces a correct grid drawing.*

Proof: All vertices are mapped into grid points, so we only need to show that the edges will not cross. The lemma follows from the following claim, which we prove by induction: Let $3 \leq k \leq n$, and $C_k = (w_1, \ldots, w_m)$. Then

(i) G_k is correctly embedded,

(ii) $x(w_1) \leq x(w_2) \leq \ldots \leq x(w_m)$, and

(iii) executing an arbitrary number of operations *shift*(w_j) does not introduce edge crossings.

First, note that G_3 satisfies (i)–(iii). For the inductive step, we need to show that adding $v = v_{k+1}$ to G_k preserves conditions (i)–(iii). When we install v, we may execute a number of operations *shift*(w_j). By the inductive assumption (iii), the correctness of the drawing of G_k is preserved, and now (i) follows from (gsm3). The new contour is $C_{k+1} = (w_1, \ldots, w_p, v, w_q, \ldots, w_m)$. Property (ii) is invariant under shifting, and therefore the inductive assumption (ii) and the choice of $x(v)$ in (gsm1) imply that (ii) is preserved as well.

It remains to show that (iii) holds after adding v. By the definition of the U-sets, $\mathit{shift}(w_j)$ in G_{k+1}, for $j > q$, is equivalent to $\mathit{shift}(w_j)$ in G_k, since only sets $U(w_i)$, for $i \geq j$ are shifted; vertex v and the edges incident to v are not affected.

Executing $\mathit{shift}(w_q)$ in G_{k+1} is also equivalent to $\mathit{shift}(w_q)$ in G_k, but in G_{k+1} it also stretches the edge (v, w_q). No edge crossings are introduced in G_k, by induction. In the triangle (w_{q-1}, v, w_q) we have $x(v), x(w_{q-1}) \leq x(w_q)$ and $y(w_{q-1}) \leq y(v)$, so moving w_q to the right does not introduce edge intersections.

Executing $\mathit{shift}(v)$ is equivalent to $\mathit{shift}(w_{p+1})$ in G_k and increasing $x(v)$ by 1. Note that v moves rigidly with all its neighbors except w_p, and therefore this shift is equivalent (up to a rigid shift of the whole graph) to shifting all vertices in $u \in \bigcup_{i=1}^{p} U(w_i)$ by 1 to the *left*, so this case is symmetric to the previous case. Similarly, the case $j \leq p$ is symmetric to the one with $j > q$. This completes the proof. □

4 Minimum-Width Grid Drawings

Let (G, π) be a given ordered, triangulated plane graph, where $\pi = v_1, \ldots, v_n$. For a given $3 \leq k \leq n-1$, let $w_p, \ldots, w_q$ be the neighbors of $v = v_{k+1}$ in C_k. When we add v to G_k, its leftmost and rightmost edges (w_p, v), (v, w_q) become contour edges. We call (w_p, v) a *forward edge* and (v, w_q) a *backward edge*. All vertices and edges that disappear from the contour when we add v are said to be *covered* by v. A vertex $v \neq v_1, v_2, v_3$ of in-degree 2 is called *forward-oriented (backward-oriented)* if it covers a forward (backward) edge.

Assume now that $n \geq 4$. Each vertex $v \neq v_1, v_2$ will be classified as *stable* or *unstable*. Also, with each such v we will associate a sequence of vertices called its *domino chain*, denoted $DC(v)$. We will also define the *dominator* of v, denoted $\mathit{dom}(v)$, which is either a vertex or the "undefined" symbol $\perp$.

These concepts are defined as follows: First, for $v = v_n$, we define $DC(v_n) = (v_n)$, $\mathit{dom}(v_n) = \perp$, and v_n is stable. Suppose now $2 \leq k \leq n-2$, $v = v_{k+1}$, and let w_p denote, as usual, the leftmost neighbor of v in G_k, that is $\mathit{ind}_v(w_p) = 1$ (for $k = 2$ we assume $w_p = v_1$). Also, let z be the vertex that covers edge (w_p, v). Such z must exist because $v \neq v_n$. Then:

(dc1) If $\mathit{ind}_z(v) = 2$, then $DC(v) \leftarrow (v)$, $\mathit{dom}(v) \leftarrow z$ and v is unstable.

(dc2) If $\mathit{ind}_z(v) \geq 4$, then $DC(v) \leftarrow (v)$, $\mathit{dom}(v) \leftarrow z$ and v is stable.

(dc3) If $\mathit{ind}_z(v) = 3$ and $DC(z) = (z_1, z_2, \ldots, z_i = z)$, then $DC(v) \leftarrow (z_1, z_2, \ldots, z_i, v)$ and $\mathit{dom}(v) \leftarrow \mathit{dom}(z)$. Also, v is stable iff z is stable.

An unstable vertex $v \neq v_1, v_2, v_3$ of in-degree 2 is called a *room-shift vertex*.

Note that (for $n \geq 4$) $DC(v_3)$ contains only vertex v_3, and the dominator of v_3 is the vertex z that covers edge (v_1, v_3); thus $ind_z(v_3) = 2$. The dominator of v_3 will only play a role in the width estimate.

The intuition is as follows: Our algorithms will try to make as few shifts as possible while preserving the invariant that the consecutive contour vertices satisfy $x(w_i) \leq x(w_{i+1})$, and the equality may hold only when $y(w_i) < y(w_{i+1})$. A stable vertex v can be placed above its leftmost neighbor w_p, saving one x-coordinate, while an unstable one may need to be placed one x-coordinate to the right. In particular, if v is a room-shift vertex, then this can result in putting v directly above its right neighbor and violating the above mentioned invariant. Thus, in that case, we need to shift v's right neighbor to the right in order to "make room" for v.

Domino chains and dominators are used to determine which vertices are stable. Suppose that z is the vertex that covers edge (w_p, v). Consider first the case when $ind_z(v) = 2$. We cannot place v above w_p because w_p is also the leftmost neighbor of z, and the embedding must satisfy $x(z) \geq x(w_p)$. Thus in this case we call v unstable. If $ind_z(v) \geq 4$, however, independently of whether vertex z is stable or not, z will be placed to the left of w_p, and then v can be put above w_p. Thus v is stable. The final case is when $ind_z(v) = 3$. We know that if z is stable, we can put z above its leftmost neighbor, which is located to the left of w_p. Then v can be put above w_p. If z is not stable, however, z might be located above w_p, and we have to put v to the right of w_p. Thus whether v is stable or not depends on whether z is stable or not. This, naturally, leads to the definition of the domino chains. Note that the vertices in $DC(v)$ do indeed behave like dominoes: If the first one is stable, all of them are stable. If the first one is unstable, all are unstable.

Example: Consider the ordered graph in Fig 1. We have $DC(4) = (4)$, $dom(4) = 5$, $DC(7) = (13, 10, 7)$, $dom(7) = 14$, $DC(12) = (17, 14, 12)$, $dom(12) = \perp$, $DC(15) = (15)$ and $dom(15) = 16$. Since $ind_{17}(6) = 2$, vertex 6 is unstable. Since $ind_{14}(13) = 4$, vertices 7,10,13 are stable. Vertices 4, 5, 6, 15 are room-shift vertices. Vertices 7, 8, 9, 12, 16 have in-degree 2, but are stable. Note that domino chains in this example are either disjoint or one is a prefix of another. Also, no two room-shift vertices share a dominator. We will prove in the lemma below that these properties hold in general. ♠

Lemma 3 *Assume $n \geq 4$, and let $u, v \neq v_1, v_2$. Then*

(a) *If $u \in DC(v)$ then $DC(u)$ is a prefix of $DC(v)$.*

(b) *If $u \notin DC(v)$ and $v \notin DC(u)$ then:*

  (b1) *$DC(u) \cap DC(v) = \emptyset$.*

  (b2) *If u, v are unstable, then $dom(u), dom(v) \neq \perp$ and $dom(u) \neq dom(v)$.*

Proof: Part (a) follows directly from the definition of domino chains, since the predecessor of each vertex in a domino-chain is uniquely defined.

We now prove Part (b). The only way to violate (b1) is when $DC(u)$ and $DC(v)$ have a common prefix. Let z be the last vertex in this prefix, and let u' and v' be its successors in $DC(u)$ and $DC(v)$, respectively. By the definition of domino chains, we then have that $u', v' \prec z$, $ind_z(u') = 3$ and $ind_z(v') = 3$, reaching a contradiction. Thus (b1) holds.

Now we prove (b2). Suppose that u, v are unstable. That $dom(u), dom(v) \neq \perp$ is obvious, since otherwise u, v would be stable. Now let u' and v' be the first vertices in $DC(u)$ and $DC(v)$, respectively. By (b1), $u' \neq v'$. Suppose $z = dom(u) = dom(v)$. Then also $z = dom(u') = dom(v')$. We have $u', v' \prec z$. Since u is unstable, all vertices in $DC(u)$, including u', are unstable. Similarly, v' is unstable. We conclude that $ind_z(u') = ind_z(v') = 2$, reaching a contradiction. □

Algorithm $\mathcal{A}$: Let (G, π) be an ordered triangulated plane graph, where $\pi = v_1, \ldots, v_n$. Let a_{f} and a_{b} denote the numbers of in-degree 2 vertices that are, respectively, forward oriented and backward oriented in (G, π). Suppose first that $a_{\mathrm{f}} \leq a_{\mathrm{b}}$. Then we proceed as follows:

If $n = 3$, we define $(x, y)(v_1) = (0, 0)$, $(x, y)(v_2) = (1, 0)$ and $(x, y)(v_3) = (0, 1)$, and the algorithm terminates.

Assume now that $n \geq 4$. We first embed v_1, v_2, v_3, as follows: $(x, y)(v_1) = (0, 0)$, $(x, y)(v_2) = (2, 0)$ and $(x, y)(v_3) = (1, 1)$. After this initialization, we add vertices in order $v_4, \ldots, v_n$. Suppose $3 \leq k \leq n - 1$, and that we are about to add $v = v_{k+1}$. As usual, let $C_k = (w_1, \ldots, w_m)$ and $w_p, \ldots, w_q$ be the neighbors of v in G_k. If v is stable then $x(v) \leftarrow x(w_p)$. Otherwise, $x(v) \leftarrow x(w_p) + 1$ and, additionally, if $deg^-(v) = 2$ then we do $\mathit{shift}(w_q)$.

In both cases the $y(v)$ is chosen to be the smallest integer such that $(x', y') = (x(v), y(v))$ satisfies requirements (gsm2) and (gsm3).

It remains to deal with the case when $a_{\mathrm{f}} > a_{\mathrm{b}}$. Consider (G', π'), where G' is a "mirror" copy of G (that is, the ordering of edges at each vertex is reversed), and π' is the same as π except that the ordering of v_1 and v_2 is reversed. A vertex is forward-oriented (backward-oriented) in π iff it is backward-oriented (forward-oriented) in π'. Thus we can apply the previous case of Algorithm $\mathcal{A}$ to (G', π'). After computing the embedding, we can modify it using the left-right reflection: set $x_0 \leftarrow x(v_2)$ and then $x(v_k) \leftarrow x_0 - x(v_k)$ for all k. This way the resulting embedding will be topologically equivalent to G.

This completes the description of Algorithm $\mathcal{A}$. In Algorithm $\mathcal{A}$ each contour edge (w_i, w_{i+1}) belongs to one of the following four types:

vertical: when $x(w_i) = x(w_{i+1})$ and $y(w_i) < y(w_{i+1})$;

horizontal: when $y(w_i) = y(w_{i+1})$;

upward: when $y(w_i) < y(w_{i+1})$ and $x(w_i) < x(w_{i+1})$;

downward: when $y(w_i) > y(w_{i+1})$ and $x(w_i) < x(w_{i+1})$.

The reader should keep in mind that our definition of the above terms differs slightly from their common English use.

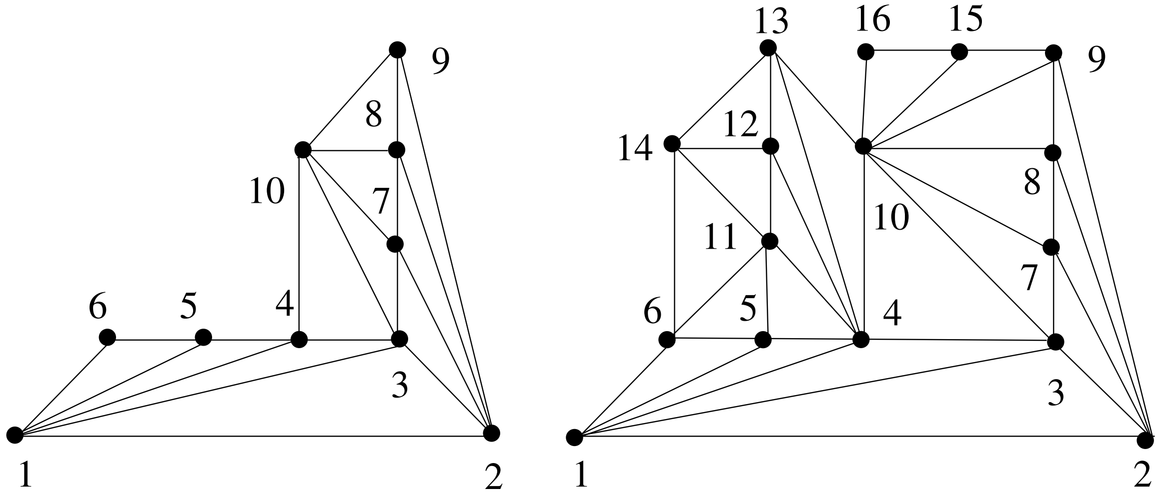

Figure 4: An example of the execution of Algorithm $\mathcal{A}$ on graph G from Figure 1. Drawings of G_{10} and G_{16} are shown.

In Algorithm $\mathcal{A}$ vertical edges are always forward, but horizontal, upward and downward edges could be either forward or backward. Note also that for $deg^-(v) = 2$, Algorithm $\mathcal{A}$ will determine $y(v)$ as follows: If (w_p, w_q) is upward then $y(v) = y(w_q)$. If (w_p, w_q) is horizontal, then $y(v) = y(w_p) + 1 = y(w_q) + 1$. If (w_p, w_q) is downward, then $y(v)$ is either $y(w_p) + 1$ or $y(w_p)$, depending on whether v is stable or not.

Given two vertices u, v, the *slope* of the segment from $(x, y)(u)$ to $(x, y)(v)$ is defined in a standard fashion:

$$slope(u, v) \quad = \quad \frac{y(v) - y(u)}{x(v) - x(u)},$$

where for $x(v) = x(u)$ we assume that $slope(u, v)$ is $\pm\infty$, depending on the sign of $y(v) - y(u)$. If (u, v) is an edge, we will sometimes say that the slope of edge (u, v) is $slope(u, v)$, if the grid drawing functions (x, y) are understood from context. Our algorithm changes the slopes of certain edges during the computation. For edges (u, v) that never belonged to a contour the slopes remain constant, so then $slope(u, v)$ will be defined unambiguously. If (u, v) is a contour edge, then its slope does not change as long as it is in the contour, and then we can write $slope(u, v)$, meaning the slope of (u, v) at the time when it was in the contour. If any ambiguity arises, we will use notation $slope_k(u, v)$ to denote the slope of (u, v) in the embedding of G_k.

Theorem 2 *If G is a given triangulated plane graph with $n \geq 3$ vertices, then Algorithm $\mathcal{A}$ produces a grid drawing of G of width $\lfloor 2(n-1)/3 \rfloor$ and height $n^2/4$.*

Proof: The theorem holds obviously for $n = 3$, so we assume that $n \geq 4$. We prove the correctness, the width estimate, and the height estimate separately.

Correctness: We will show that the following invariant holds at each step $k = 3, \ldots, n$:

(I): Let $C_k = (w_1, \ldots, w_m)$ be the current contour. Then

(I1): For each $j = 1, \ldots, m-1$, we have

(a) $x(w_j) \le x(w_{j+1})$,

(b) $x(w_j) = x(w_{j+1})$ iff $w_j \prec w_{j+1}$ and w_{j+1} is stable. Also, $x(w_j) = x(w_{j+1})$ implies $y(w_j) < y(w_{j+1})$.

(I2) If $k < n$, and $w_p, \ldots, w_q$ are the neighbors of $v = v_{k+1}$ in G_k, then after adding v in the external face of G_k, we have $x(w_p) \le x(v) < x(w_q)$, $y(v) \ge \max\{y(w_{p+1}), y(w_{q-1})\}$, and all $w_p, \ldots, w_q$ are visible from v.

Since (I2) implies that the choices made by Algorithm $\mathcal{A}$ satisfy conditions (gsm1)–(gsm3), the correctness follows directly from Lemma 2.

Thus it is sufficient to show that (I2) holds at each step. Invariant (I1) is true, trivially, for $k = 3$. Assume (I1) holds for some $3 \le k < n$. In the inductive step we will show that (I1) implies (I2), and that (I1) is preserved after adding $v = v_{k+1}$.

Claim 2.1: $x(w_{p+1}) > x(w_p)$.

If $w_{p+1} \prec w_p$ then $x(w_{p+1}) > x(w_p)$ by the inductive assumption (I1). Suppose that $w_p \prec w_{p+1}$. Since $ind_v(w_{p+1}) = 2$, we have $v = dom(w_{p+1})$ and w_{p+1} is unstable. Claim 2.1 follows now from the inductive assumption (I1).

Claim 2.2: If v is unstable and $deg^-(v) \ge 3$, then $x(w_{p+2}) > x(w_{p+1})$.

If $w_{p+2} \prec w_{p+1}$, then Claim 2.2 follows from the inductive assumption (I1). Suppose that $w_{p+1} \prec w_{p+2}$. Since v covers edge (w_{p+1}, w_{p+2}), and $ind_v(w_{p+2}) = 3$, the fact that v is unstable implies that w_{p+2} is unstable, and now Claim 2.2 follows from the inductive assumption (I1).

Suppose now that v is stable. By Claim 2.1 and (I1), if we set $x(v) = x(w_p)$ and choose $y(v)$ to be large enough, then v will satisfy (gsm2) and (gsm3). We also have $x(w_p) = x(v) < x(w_{p+1}) \le x(w_q)$, completing the proof of (I2). Since the new contour is $C_{k+1} = (w_1, \ldots, w_p, v, w_q, \ldots, w_m)$, these inequalities, together with the inductive assumption imply that (I1) is preserved.

The second case is when v is unstable and $deg^-(v) \ge 3$. Claims 2.1 and 2.2 imply that $x(w_p) < x(w_{p+1}) < x(w_{p+2})$. Therefore, using (I1), after setting $x(v) = x(w_p) + 1$ and choosing $y(v)$ large enough, v will satisfy (gsm2) and (gsm3). Since we also have $x(w_p) < x(v) \le x(w_{p+1}) < x(w_{p+2}) \le x(w_q)$, (I2) follows. These inequalities, together with the inductive assumption, also imply that (I1) is preserved.

Finally, consider the case when v is unstable and $deg^-(v) = 2$, that is v is a room-shift vertex. By Claim 2.1, after executing the shift operation, we have $x(w_q) \ge x(w_p) + 2$. Therefore, by setting $x(v) = x(w_p)+1$, and choosing $y(v)$ as in the algorithm, we make (gsm2) and (gsm3) true. Since we also have $x(w_p) < x(v) < x(w_q)$, (I2) follows. These last inequalities, together with the inductive assumption imply that (I1) is preserved.

Width estimate: Denote by a and b the number of vertices of in-degree 2 and in-degree ≥ 3, respectively, not counting v_1, v_2, v_3. Thus we have $n = a + b + 3$. As in Algorithm $\mathcal{A}$, by a_{f} and a_{b} we denote the numbers of in-degree 2 vertices that are, respectively, forward oriented and backward oriented. Without loss of generality, we can assume that $a_{\mathrm{f}} \leq a_{\mathrm{b}}$.

Let ω be the width of the drawing constructed by algorithm $\mathcal{A}$, and denote by a^{rs} the number of room-shift vertices (other than v_1, v_2, v_3). Then $\omega = a^{\mathrm{rs}} + 2$. Observe that a dominator cannot be a backward-oriented vertex of in-degree 2. By Lemma 3, dominators of room-shift vertices are distinct, and they are distinct from the dominator of v_3. Thus $a^{\mathrm{rs}} + 1 \leq a_{\mathrm{f}} + b$, and we get

$$\omega \;=\; a^{\mathrm{rs}} + 2 \;\leq\; (a_{\mathrm{f}} + b - 1) + 2 \;\leq\; a/2 + b + 1 \;=\; n - a/2 - 2.$$

On the other hand, $\omega \leq a + 2$. Therefore

$$\omega \;\leq\; \min(a+2, n - a/2 - 2) \;\leq\; 2(n-1)/3,$$

as required.

Height estimate: Let γ_k be the smallest slope among the edges in C_k. By the invariant (I1) we have $-\infty < \gamma_k < 0$.

Claim 2.3: If a^{rs}_k is the number of room-shift vertices among the $v_4, \ldots, v_k$, then $\gamma_k \geq a^{\mathrm{rs}}_k + 2 - k$.

First we show that Claim 2.3 implies the height estimate. For $k = n$, we have $\gamma_n \geq a^{\mathrm{rs}} + 2 - n = \omega - n$, and thus the height of the drawing is at most $\omega(n - \omega) \leq n^2/4$.

The proof of Claim 2.3 is by induction on k. For $k = 3$, $a^{\mathrm{rs}}_3 = 0$ and $\gamma_3 = -1$, so the claim holds. Let us now assume that the lemma holds for some k, $3 \leq k < n$, and we are about to add $v = v_{k+1}$. We only need to consider new contour edges (w_p, v) and (v, w_q), and only when they are downward.

The first case is when v is a room-shift vertex. Then $a^{\mathrm{rs}}_{k+1} = a^{\mathrm{rs}}_k + 1$, so it is sufficient to show that the slopes of the new contour edges are at least γ_k. In this case edge (w_p, v) is always upward or horizontal. Suppose that (v, w_q) is downward. If (w_p, w_q) is horizontal, then $\mathit{slope}(v, w_q) \geq -1$. If (w_p, w_q) is downward, then $\mathit{slope}_{k+1}(v, w_q) = \mathit{slope}_k(w_p, w_q)$, because of the shift.

The second case is when v is not a room-shift vertex, so $a^{\mathrm{rs}}_{k+1} = a^{\mathrm{rs}}_k$. We want to show that the slopes of the new contour edges are at least $\gamma_k - 1$. Consider first edge (w_p, v) and suppose it is downward. Then $\mathit{slope}_{k+1}(w_p, v) > \mathit{slope}_k(w_p, w_{p+1})$ so (w_p, v) cannot cause a problem. Consider now the case when (v, w_q) is downward. If $\mathit{deg}^-(v) = 2$, then $x(v) = x(w_p)$, $y(v) = y(w_p) + 1$, and thus $\mathit{slope}_{k+1}(v, w_q) \geq \mathit{slope}_k(w_p, w_q) - 1 \geq \gamma_k - 1$. If $\mathit{deg}^-(v) \geq 3$, by the minimality of $y(v)$, there exists a downward edge (w_i, w_{i+1}) for some $p \leq i \leq q - 1$, for which

$slope(w_i, w_{i+1}) > slope(v, w_{i+1}) \geq slope(w_i, w_{i+1}) - 1$. Since $slope(v, w_q) \geq slope(v, w_{i+1})$, we get $slope(v, w_q) \geq slope(w_i, w_{i+1}) - 1 \geq \gamma_k - 1$. □

5 Reducing Height

In this section we will show how we can reduce the grid height to $4\lfloor 2(n-1)/3 \rfloor - 1$. The new algorithm, called Algorithm $\mathcal{B}$, follows the same general approach as Algorithm $\mathcal{A}$, but it reduces the height by ensuring that downward edges in the contours are not too steep.

Let (G, π) be a given ordered, triangulated plane graph, where $\pi = v_1, \ldots, v_n$. Given a grid embedding $P = (x, y)$ of G, we define the *slack* between u and v by

$$\begin{aligned} slack(u, v) &= y(v) + 4[x(v) - x(u)] - y(u) \\ &= 4\Delta x(u, v) + \Delta y(u, v), \end{aligned}$$

where $\Delta x(u, v) = x(v) - x(u)$ and $\Delta y(u, v) = y(v) - y(u)$. Note that if a contour edge (u, v) is horizontal or upward then $slack(u, v) \geq 4$. Thus $slack(u, v)$ can only become nonpositive if (u, v) is downward. We also have the following relationship between slack and slope:

$$slope(u, v) = -4 + \frac{slack(u, v)}{\Delta x(u, v)}.$$

For simplicity, we do not specify P in the notation for slack and slope, since P will always be understood from context.

Fact 1 *Let $t_1, \ldots, t_\ell$ be any path in G. Then*

(a) $slack(t_1, t_\ell) = \sum_{i=1}^{\ell-1} slack(t_i, t_{i+1})$,

(b) Suppose that $slack(t_i, t_{i+1}) \geq 1$ for all $i = 1, \ldots, \ell - 1$ and $1 \leq a < b \leq \ell$. Then $slack(t_1, t_\ell) \geq slack(t_a, t_b)$. If, additionally, $1 < a$ or $b < \ell$ then $slack(t_1, t_\ell) \geq slack(t_a, t_b) + 1$.

Proof: Part (a) can be proven by simple summation. Part (b) follows directly (a). □

In Algorithm $\mathcal{A}$ vertical edges were always forward, but horizontal, upward and downward edges could be either forward or backward. Algorithm $\mathcal{B}$ will also satisfy this property. In order to reduce the height of the drawing, we use the simple but fruitful idea from [FPP90], which is to control the grid height by ensuring that the slopes of all contour edges are bounded (from below) by a constant. If all such slopes are $\geq -c$ then, quite obviously, the height of the drawing will be bounded by c times its width. In [FPP90, CP95, CK93] all edges in the contour had slope at least -1. Our algorithm will use slopes greater than -4. Preserving this invariant will require, in some cases, making more shifts than in Algorithm $\mathcal{A}$. Nevertheless, as we will show, the width of the grid drawing produced by Algorithm $\mathcal{B}$ will still be bounded by $\lfloor 2(n-1)/3 \rfloor$.

We will distinguish two types of shifts. Let v be a vertex to be installed. As in Algorithm $\mathcal{A}$, a *room-shift* occurs when v is a room-shift vertex. A *slope-shift* occurs when we shift the rightmost neighbor w_q of v in order to reduce the absolute value of the slope of edge (v, w_q). We will call such v *slope-shift vertices*. No two shifts will occur simultaneously. A *shift vertex* is either a room-shift or a slope-shift vertex.

The main intuition behind our method can be explained as follows: Suppose that we are about to install a vertex $v = v_{k+1}$ with $deg^-(v) \geq 4$, and let w_q be its rightmost neighbor in C_k. Assume that the edge (w_{q-1}, w_q) is downward. In order for w_q to be visible from v, the slope of (v, w_q) must be smaller than that of (w_{q-1}, w_q). (Recall that both values are negative.) Thus it might seem, at first, that by repeatedly adding such vertices the slope can decrease to below -4. This is not so, however. To understand why, it is better to think in terms of slacks instead of slopes. Note that a slope of an edge is greater than -4 iff its slack is positive. Since $deg^-(v) \geq 4$, we can install v so that $x(v) < x(w_{q-1})$ (similarly as in Algorithm $\mathcal{A}$). But then, by setting $y(v) = y(w_q) + 4\Delta x(v, w_q) - slack(w_{q-1}, w_q)$, we have $slack(v, w_q) = slack(w_{q-1}, w_q)$. So we have $slope(v, w_q) > -4$ even though $slope(v, w_q) < slope(w_{q-1}, w_q)$. Because of this property, such vertices v will be called *slack-preserving*. (The formal definition is given later.)

There are other vertices though for which this approach does not work. These will be called *slack-reducing*. Suppose, for instance, that $deg^-(v) = 2$, v is stable and (w_p, w_q) is downward. Then v will be installed at $x(v) = x(w_p)$ and $y(v) = y(w_p)+1$, in which case we have $slack(v, w_q) = slack(w_{q-1}, w_q) - 1$. In particular, if $slack(w_{q-1}, w_q) = 1$, the slack of (v, w_q) will be zero, in which case we will have to shift w_q to maintain the slope invariant.

Let $v = v_{k+1}$ be a vertex whose neighbors in G_k are $w_p, \ldots, w_q$, and let $p + 1 \leq r \leq q$. Then the vertex w_r is called *pivotal for* v if r is the smallest index such that for each $i = r + 1, \ldots, q$ vertex w_i is stable and $w_{i-1} \prec w_i$. From the definition of stable vertices we obtain the following fact.

Fact 2 *The pivotal vertex w_r is well defined for each v. If $deg^-(v) = 2$ then $r = p + 1$. If v is unstable and $deg^-(v) \geq 3$ then $r \geq p + 2$.*

Algorithm $\mathcal{B}$: The choice of the canonical ordering $\pi = v_1, \ldots, v_n$ satisfying $a_{\mathrm{f}} \leq a_{\mathrm{b}}$, and the initialization are exactly the same as in Algorithm $\mathcal{A}$.

Assume now that $n \geq 4$, and that we are now about to add $v = v_{k+1}$, for $3 \leq k \leq n - 1$. As usual, we denote $C_k = (w_1, \ldots, w_m)$, $w_p, \ldots, w_q$ are the neighbors of v in G_k, and w_r is the pivotal neighbor of v.

```
if v is stable then x(v) ← x(w_p)
   else x(v) ← x(w_p) + 1;
if deg⁻(v) = 2 then begin
   if v is stable then y(v) ← max{y(w_p) + 1, y(w_q)}
   else begin { v is unstable }
      shift(w_q);
      if (w_p, w_q) is upward then y(v) ← y(w_q)
         else y(v) ← max{y(w_p), y(w_q) + 1}
   end
end else begin { deg⁻(v) ≥ 3 }
   ỹ ← y(w_r) + 4Δx(v, w_r) − slack(w_{r−1}, w_r);
   if r = p + 1 or (v is unstable and r = p + 2) then ỹ = ỹ + 1;
   y(v) ← max{ỹ, y(w_{q−1})};
end;
if slack(v, w_q) = 0 then shift(w_q);
```

This completes the description of Algorithm $\mathcal{B}$. Figure 5 shows an example of an execution of Algorithm $\mathcal{B}$.

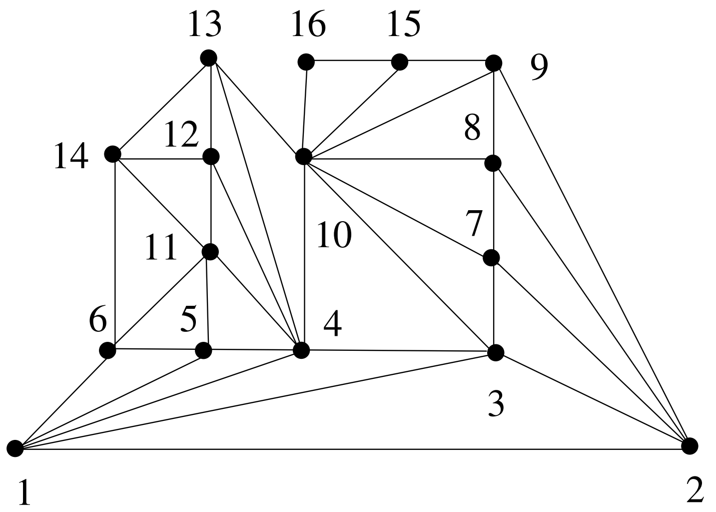

Figure 5: An example of an execution of Algorithm $\mathcal{B}$ on graph G from Figure 1. The drawing of G_{16} is shown.

Throughout the rest of this section we will adopt the following convention: if (u, v) is an edge, then $slack(u, v)$ denotes the slack of (u, v) at the time when (u, v) was added to the graph. If (u, v) is not a contour edge then its slack does not change. If (u, v) is a contour edge, its slack does not change as long as it belongs to the contour, but it may change after it is covered by another vertex because of shifts.

Lemma 4 *Algorithm $\mathcal{B}$ produces a correct grid drawing, and the height of this drawing is less than 4 times its width, that is $y(v_n) < 4x(v_2)$.*

Proof: The lemma is trivial for $n = 3$. So assume $n \geq 4$. We prove the following invariant:

(J) Let $3 \leq k \leq n$, and $C_k = (w_1, \ldots, w_m)$. The drawing produced by Algorithm $\mathcal{B}$ satisfies the following conditions:

- (J1) For all $i = 1, \ldots, m-1$, we have
  - (a) $x(w_i) \leq x(w_{i+1})$, and $x(w_i) = x(w_{i+1})$ iff $w_i \prec w_{i+1}$ and w_{i+1} is stable.
  - (b) $\mathit{slack}(w_i, w_{i+1}) \geq 1$. Additionally, if vertex w_{i+1} is unstable and $w_i \prec w_{i+1}$ then $\mathit{slack}(w_i, w_{i+1}) \geq 2$.
- (J2) If $k < n$ then let $v = v_{k+1}$, and let $w_p, \ldots, w_q$ denote the neighbors of v in G_k. Denote by $\mathit{slack}'(v, w_q)$ the slack of (v, w_q) before Line 15 is executed. Then
  - (a) $x(w_p) \leq x(v) < x(w_q)$.
  - (b) $y(v) \geq \max\{y(w_{p+1}), y(w_{q-1})\}$, v is in the outer face of G_k, and $w_p, \ldots, w_q$ are visible from v.
  - (c) Let $\mathit{deg}^-(v) = 2$. If (w_p, w_q) is upward or horizontal then $\mathit{slack}'(v, w_q) \geq 3$. If (w_p, w_q) is downward, then either v is stable and $\mathit{slack}'(v, w_q) = \mathit{slack}(w_p, w_q) - 1$, or v is unstable and $\mathit{slack}'(v, w_q) = \mathit{slack}(w_p, w_q)$.
  - (d) Let $\mathit{deg}^-(v) \geq 3$. If $\tilde{y} < y(w_{q-1})$ then (v, w_q) is upward. If $\tilde{y} \geq y(w_{q-1})$ then v has the following property: If $r = p+1$, or if v is unstable and $r = p + 2$, then $\mathit{slack}'(v, w_q) \geq \mathit{slack}(w_{r-1}, w_r) + q - r - 1$. Otherwise, $\mathit{slack}'(v, w_q) \geq \mathit{slack}(w_{r-1}, w_r) + q - r$.

Observe first that Lemma 4 follows from Invariant (J). The correctness of the drawing follows directly from (J2.a), (J2.b), and Lemma 2. Invariant (J1.b), implies that $\mathit{slope}(v_n, v_2) > -4$, and therefore $y(v_n) < 4x(v_2)$, as required.

We prove Invariant (J) by induction on k. (J1) holds for $k = 3$, by inspection. Suppose G_k satisfies (J1) for some $k \geq 3$. In the inductive step we will show that (J1) implies (J2), and that (J1) is preserved after adding v.

Proof that (J1) implies (J2). As in the proof of Theorem 2, we have the following claim.

Claim 4.1: $x(w_{p+1}) > x(w_p)$. If v is unstable and $\mathit{deg}^-(v) \geq 3$, then $x(w_{p+2}) > x(w_{p+1})$.

(a) If v is stable then $x(v) = x(w_p) < x(w_q)$ by Claim 4.1. If v is unstable and $\mathit{deg}^-(v) \geq 3$ then $x(v) = x(w_p) + 1 \leq x(w_{p+1}) < x(w_q)$ by Claim 4.1. If v is a room shift vertex, then $x(v) = x(w_p) + 1 < x(w_q)$ because of the shift. Therefore $x(w_p) \leq x(v) < x(w_q)$ in all cases.

(b) If $\mathit{deg}^-(v) = 2$, then (J2.b) follows directly from the algorithm, lines 4-8. Thus we can assume now that $\mathit{deg}^-(v) \geq 3$. Define $\tilde{v}$ to be the point $\tilde{v} = (x(v), \tilde{y})$. From the algorithm, lines 11-12, $\tilde{v}$

has the following important property: If v is stable and $r > p+1$, or if v is unstable and $r > p+2$, then $slack(\tilde{v}, w_{r-1}) = slack(\tilde{v}, w_r) - slack(w_{r-1}, w_r) = 0$. Otherwise, $slack(\tilde{v}, w_{r-1}) = -1$.

Denote by H_i the open half-plane which is on the left-hand side of edge (w_i, w_{i+1}) if it's traversed from w_i to w_{i+1}. Thus H_i will be above (w_i, w_{i+1}) if (w_i, w_{i+1}) is not vertical. Note that for $x(v) \le x(w_{i+1})$, $slack(\tilde{v}, w_{i+1}) < 0$ implies $\tilde{v} \in H_i$.

Claim 4.2: $\tilde{y} \ge \max_{p<i<r} y(w_i)$, and $\tilde{v} \in \bigcap_{i=p}^{r-1} H_i$.

We show first that Claim 4.2 implies (J2.b). Since $y(w_r) < \ldots < y(w_q)$ and $y(v) = \max(\tilde{y}, y(w_{q-1}))$, we get $y(v) \ge \max_{p<i<q} y(w_i)$. We also have $\tilde{v} \in H_i$ for $i = r, \ldots, q-1$, because $w_r, \ldots, w_q$ are embedded vertically and $x(v) < x(w_r)$. Thus $\tilde{v} \in \bigcap_{p \le i < q} H_i$. This implies that $\tilde{v}$ is in the outer region of G_k and that the vertices $w_p, \ldots, w_q$ are visible from $\tilde{v}$. Since v is installed directly above $\tilde{v}$, v also satisfies this condition.

We prove now Claim 4.2. Assume first that v is stable. If $r = p+1$ then $x(v) = x(w_p)$ and $\tilde{y} = y(w_p) + 1$, so the claim is obvious. If $r > p+1$ then, from induction, Claim 4.1, and Fact 1, we obtain that $x(w_i) > x(v)$ and $slack(w_i, w_r) \ge slack(w_{r-1}, w_r)$ for $i = p+1, \ldots, r-1$, and that (for $i = p$) $x(w_p) = x(v)$ and $slack(w_p, w_r) > slack(w_{r-1}, w_r)$. Therefore, for all $i = p, \ldots, r-1$,

$$\begin{aligned} \tilde{y} &= y(w_r) + 4\Delta x(v, w_r) - slack(w_{r-1}, w_r) \\ &> y(w_r) + 4\Delta x(w_i, w_r) - slack(w_i, w_r) \ = \ y(w_i). \end{aligned}$$

Now, for each $i = p, \ldots, r-1$, we have

$$\begin{aligned} slack(\tilde{v}, w_i) &= slack(\tilde{v}, w_{r-1}) - slack(w_i, w_{r-1}) \\ &\le slack(\tilde{v}, w_{r-1}) \ = \ 0, \end{aligned}$$

which implies that $\tilde{v} \in H_i$.

The second case is when v is unstable. If $r = p+2$ then $x(w_p) < x(v) \le x(w_{p+1})$, and

$$\begin{aligned} \tilde{y} &= y(w_r) + 4\Delta x(v, w_r) - slack(w_{p+1}, w_r) + 1 \\ &> y(w_r) + 4\Delta x(w_{p+1}, w_r) - slack(w_{p+1}, w_r) \ = \ y(w_{p+1}). \end{aligned}$$

We have $slack(\tilde{v}, w_r) = slack(w_{p+1}, w_r) - 1$, which implies that $\tilde{v} \in H_{p+1}$. Since $slack(\tilde{v}, w_{p+1}) = slack(\tilde{v}, w_r) - slack(w_{p+1}, w_r) = -1 < slack(w_p, w_{p+1})$, we obtain $\tilde{v} \in H_p$.

If $r > p+2$, the argument is very similar to the case when v is stable and $r > p+1$, so we only sketch it here. From induction, Claim 4.1, and Fact 1, we obtain that $x(w_i) > x(v)$ and $slack(w_i, w_r) \ge slack(w_{r-1}, w_r)$ for $i = p+2, \ldots, r-1$, and that $x(w_{p+1}) \ge x(v)$ and $slack(w_{p+1}, w_r) > slack(w_{r-1}, w_r)$. Therefore, for all $i = p+1, \ldots, r-1$, $\tilde{y} > y(w_i)$. For each $i = p, \ldots, r-1$, we have $slack(\tilde{v}, w_i) \le 0$, which implies that $\tilde{v} \in H_i$.

(c) If (w_p, w_q) is upward then (v, w_q) is horizontal, so $slack(v, w_q) \ge 4$. If (w_p, w_q) is horizontal, then $x(v) < x(w_q)$ and $y(\tilde{v}) = y(w_q) + 1$, implying that $slack(v, w_q) \ge 3$.

Suppose now that (w_p, w_q) is downward. If v is stable then

$$\begin{aligned} \mathit{slack}'(v, w_q) &= \Delta y(v, w_q) + 4\Delta x(v, w_q) \\ &= \Delta y(w_p, w_q) - 1 + 4\Delta x(w_p, w_q) \;=\; \mathit{slack}(w_p, w_q) - 1. \end{aligned}$$

If v is unstable, then denoting by $x'(w_q)$ the x-coordinate of w_q after the room shift we have

$$\begin{aligned} \mathit{slack}'(v, w_q) &= \Delta y(v, w_q) + 4[x'(w_q) - x(v)] \\ &= \Delta y(w_p, w_q) + 4\Delta x(w_p, w_q) \;=\; \mathit{slack}(w_p, w_q). \end{aligned}$$

(d) Let $\mathit{deg}^-(v) \geq 3$. Suppose first that $\tilde{y} < y(w_{q-1})$, that is $y(v) = y(w_{q-1})$. If $r < q$ then $y(w_q) > y(w_{q-1})$, and we are done. If $r = q$, then by the proof of (J2.b), w_{q-1} and w_q are visible from $\tilde{v}$. Therefore, since $x(\tilde{v}) < x(w_{q-1})$, we must have $y(w_q) > y(w_{q-1})$, so (v, w_q) is upward.

Suppose now that $\tilde{y} \geq y(w_{q-1})$, that is $y(v) = \tilde{y}$. For $i = r, \ldots, q-1$ edges (w_i, w_{i+1}) are vertical, so we have $\mathit{slack}'(v, w_{i+1}) > \mathit{slack}(v, w_i)$. Thus $\mathit{slack}'(v, w_q) \geq \mathit{slack}(v, w_r) + q - r$. Now (J2.d) follows, since $\mathit{slack}(v, w_r) = \mathit{slack}(w_{r-1}, w_r) - 1$ if v is stable and $r = p+1$ or if v is unstable and $r = p+2$, and $\mathit{slack}(v, w_r) = \mathit{slack}(w_{r-1}, w_r)$ otherwise.

Proof that (J2) implies that (J1) is preserved. By the inductive assumption, we only need to consider the two new edges (w_p, v) and (v, w_q).

(a) If v is stable, then $x(v) = x(w_p) < x(w_{p+1}) \leq x(w_q)$, by Claim 4.1. If v is a room-shift vertex, then $x(v) = x(w_p) + 1 < x'(w_q)$, because of the shift. ($x'(w_q)$ is the new x-coordinate of w_q.) If $\mathit{deg}^-(v) \geq 3$ and v is unstable, then $x(v) = x(w_p) + 1 \leq x(w_{p+1}) < x(w_q)$, by Claim 4.1. This completes the proof of (J1.a).

(b) We show (J1.b) for edge (w_p, v). The case when v is stable is obvious, since then $x(v) = x(w_p)$ and $y(v) > y(w_p)$. It is also obvious when v is a room-shift vertex, by inspection. So let us assume that $\mathit{deg}^-(v) \geq 3$ and v is unstable. In this case $r \geq p+2$. We have

$$\mathit{slack}(w_p, v) \;=\; \mathit{slack}(w_p, w_{r-1}) + \mathit{slack}(w_{r-1}, w_r) - \mathit{slack}(v, w_r).$$

If $r \geq p+3$ then $\mathit{slack}(w_p, w_{r-1}) \geq 2$ and $\mathit{slack}(v, w_r) = \mathit{slack}(w_{r-1}, w_r)$. If $r = p+2$ then $\mathit{slack}(w_p, w_{r-1}) \geq 1$ and $\mathit{slack}(v, w_r) = \mathit{slack}(w_{r-1}, w_r) - 1$. In both cases, we get $\mathit{slack}(w_p, v) \geq 2$.

For edge (v, w_q), we just need to show that $\mathit{slack}(v, w_q) \geq 1$. Suppose first that $\mathit{deg}^-(v) = 2$. By (J2.c), if (w_p, w_q) is upward, horizontal, or if $\mathit{slack}'(v, w_q) = \mathit{slack}(w_p, w_q)$, we are done. The only missing case is when (w_p, w_q) is downward and v is stable. Then $\mathit{slack}'(v, w_q) = \mathit{slack}(w_p, w_q) - 1 \geq 0$. If $\mathit{slack}'(v, w_q) > 0$, we are done. If $\mathit{slack}'(v, w_q) = 0$, a shift will occur in line 15, resulting in $\mathit{slack}(v, w_q) = 4$.

Suppose now that $\mathit{deg}^-(v) \geq 3$. We can assume that $\tilde{v} = v$, since otherwise (v, w_q) is upward. Since $\mathit{slack}(w_{r-1}, w_r) \geq 1$ and $q \geq r$, (J2.d) implies that $\mathit{slack}'(v, w_q) \geq 0$, and that $\mathit{slack}'(v, w_q) = 0$ is possible only when $r = q = p+2$ (which means that $\mathit{deg}^-(v) = 3$) and v is unstable. But then, a shift in line 15 will result in $\mathit{slack}(v, w_q) = 4$. □

Mates of shift vertices. Let $v = v_{k+1}$ be a vertex with neighbors $w_p, \ldots, w_q$ in G_k, for $3 \leq k < n$. We say that v is *slack-preserving* if either $deg^-(v) \geq 4$, or v is stable and $deg^-(v) = 3$. We call v *slack-reducing* if either v is unstable and $deg^-(v) = 3$, or v is stable and $deg^-(v) = 2$. We say that v is *slack-critical*, if v is slack-reducing and edge (w_{q-1}, w_q) is downward.

Here are some basic observations: All slope-shift vertices are slack-critical. If v is slack-reducing then its pivotal vertex is w_q, by Fact 2. Note also that the three sets: room-shift vertices, slack-preserving vertices and slack-reducing vertices, form a partition of the set $\{v_4, \ldots, v_n\}$.

If v is a backward-oriented room-shift-vertex, we would like to assign to v a *mate* $\sigma(v)$ that belongs to one of the three categories:

(mr1) a forward-oriented stable, but not slack-critical, vertex of in-degree 2, or

(mr2) an unstable, but not slack-critical, vertex of in-degree 3, or

(mr3) a slack-preserving vertex.

We will use the term σ-mates to describe all vertices $\sigma(v)$. Our goal is to define σ so that all σ-mates are different. The lemma below shows that this is indeed possible. The general idea of the proof is to pick a mate $\sigma(v)$ of v to be a certain vertex that is "responsible" for v being unstable. (However, it is not necessarily the same vertex as $dom(v)$.)

Lemma 5 *There exists an assignment σ of mates to all backward-oriented room-shift vertices, such that*

(a) *each σ-mate satisfies one of the conditions (mr1)-(mr3),*

(b) *all σ-mates are distinct,*

(c) *vertices v_1, v_2, v_3, v_n are not σ-mates.*

Proof: Let v be a backward-oriented room-shift vertex, and t its left neighbor, that is $ind_v(t) = 1$. Note that, by Algorithm $\mathcal{B}$, (t, v) is horizontal or upward. Let z be last vertex in the ordering $\prec$ such that $ind_z(t) = 1$.

If z is stable, then we set $\sigma(v) = z$. Note that in this case $z \neq v$. If $deg^-(z) = 2$, then z is forward-oriented and not slack-critical, satisfying (mr1). If $deg^-(z) \geq 3$, z satisfies (mr3).

Suppose now that z is unstable. In this case it may happen that $z = v$. Then there is z' that covers (t, z). Since z is unstable and (by the choice of z) $ind_{z'}(t) \neq 1$, we have $deg^-(z') \geq 3$ and $ind_{z'}(t) = 2$. Then we set $\sigma(v) = z'$. Note that z' must be unstable, because it precedes z in $DC(z)$. If $deg^-(z') \geq 4$ then z' satisfies (mr3). Otherwise, if $deg^-(z') = 3$, z' satisfies (mr2), since (t, z) cannot be a downward edge, because (t, v) is not downward.

To prove (c), note that $\sigma(v) \neq v_1, v_2, v_3$ is obvious. If z is stable, then $ind_z(t) = 1$, and $z \neq v_n$ because $ind_{v_n}(v_1) = 1$ and $t \neq v_1$. If z is unstable, then also z' is unstable, and thus $z' \neq v_n$, since

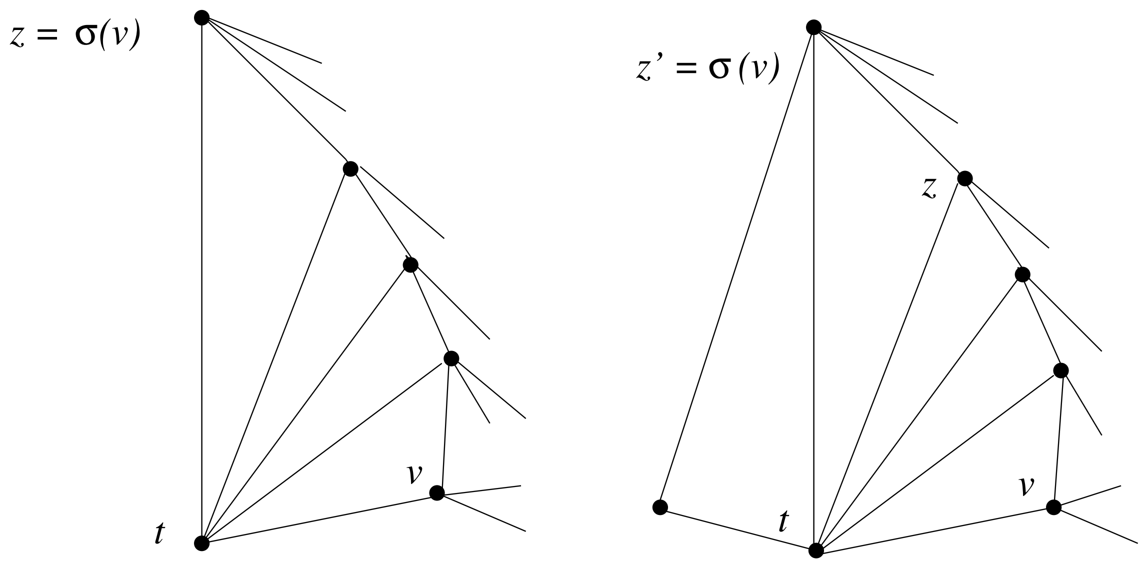

Figure 6: The construction of σ-mates. In the first example z is stable, in the second example z is unstable.

v_n is stable.

It remains to prove (b), that all σ-mates are different. Pick two different backward-oriented room-shift vertices u, v. Let t^u and t^v denote, respectively, the left neighbors of u and v, as in the above construction. Note that $t^u \neq t^v$. For otherwise, if $t^u = t^v$ and, without loss of generality, $u \prec v$, then v couldn't be backward-oriented.

Suppose now that $\sigma(u) = \sigma(v) = z$. By the construction of mates, we have $t^u, t^v \prec z$. Furthermore, either $ind_z(t^u) = 1$ and z is stable, or $ind_z(t^u) = 2$ and z is unstable. But v satisfies the same condition, and since $ind_z(t^u) \neq ind_z(t^v)$, we reach a contradiction. Thus $\sigma(u) \neq \sigma(v)$, completing the proof. □

If v is a slope-shift vertex, we will assign to it two mates $\mu_1(v)$ and $\mu_2(v)$ such that each $\mu_i(v)$ satisfies the following condition:

(ms) $\mu_i(v)$ is slack-critical, but not a slope-shift vertex.

By the definition of slack-critical vertices, $\mu_i(v)$ has in-degree 2 or 3. The vertices $\mu_i(v)$ will be referred to as μ_i-mates, or μ-mates if i is not specified. The μ-mates of v will be the vertices "responsible" for the slope-shift of w_q when installing v. Before we shift w_q, we must have had $slack(v, w_q) = 0$, and thus $slack(w_{q-1}, w_q) = 1$. Let $t = w_q$ and consider all vertices z such that $t \prec z \prec v$ and t is the last neighbor of z in the contour (at the time when we install z). The intuition is that $\mu_1(v)$ and $\mu_2(v)$ will be, respectively, these slack-reducing vertices z that cause the decrease of the slack of (z, t) from 2 to 1, and 3 to 2. In certain situations, however, the construction of the $\mu_i(v)$ will be complicated by the presence of edges (z,t) for which $z \prec t$.

Lemma 6 *There exist two assignments μ_1, μ_2 of mates to all slope-shift vertices such that*

(a) *each μ-mate satisfies condition (ms),*

(b) *all μ-mates are distinct,*

(c) *vertices v_1, v_2, v_3, v_n are not μ-mates.*

Proof: For a vertex $v = v_{k+1}$, by $\gamma_i(v)$ we denote the i-th last neighbor of v in G_k, that is $ind_v(\gamma_i(v)) = deg^-(v) - i + 1$. Call v a *slack-a* vertex, if $slack(v, \gamma_1(v)) = a$. We start with the following useful claims.

Claim 6.1: Suppose that (u,w) is a contour edge with $slack(u,w) = 1$. (i) If $w \prec u$ then u is a slack-1 vertex. (ii) If $u \prec w$ then w is stable and (u,w) is vertical.

Part (i) follows from the definition of slack-1 vertices. Part (ii) follows directly from Invariant (J1.b) and the algorithm.

Claim 6.2: Suppose that (u,w) is a contour edge with $slack(u,w) = 2$. (i) If $w \prec u$ then u is a slack-2 vertex. (ii) If $u \prec w$ and w is unstable then u is a slack-1 vertex. (iii) If $u \prec w$ and w is stable then (u,w) is vertical.

Part (i) follows from the definition of slack-2 vertices. Part (iii) follows from (J1.b) and the algorithm. We now prove Part (ii). Let (u,z) be the first edge covered by w. By the algorithm, $x(u) < x(w) \leq x(z)$, and $y(w) > y(z)$. Then $slack(u,z) < slack(w,z) = 2$, and thus $slack(u,z) = 1$. Since (u,z) is downward, Claim 6.1 implies that u is a slack-1 vertex.

Let now v be a slope-shift vertex. We will associate with v a sequence of vertices

$$S(v) \quad = \quad (\, v = b_0 \, , \, b_1 \, , \, \ldots \, , b_f = c_0 \, , \, c_1 \, , \, \ldots \, , c_g = d_1 \, , \, \ldots \, , d_h \,),$$

where $f, g, h \geq 1$. The μ-mates of v are $\mu_1(v) = b_f$ and $\mu_2(v) = d_h$.

Step I: Construction of the $\{b_i\}$: Let $b_1 = \gamma_2(v)$ and $w = \gamma_1(v)$. Since v is a slope-shift vertex, we have $slack(b_1, w) = 1$ and (b_1, w) is downward. Thus, by Claim 6.1, b_1 is a slack-1 vertex. Suppose that we already have defined a slack-1 vertex b_i. If b_i is slack-reducing then $f = i$, completing the construction of $b_0, \ldots, b_f$. Otherwise, b_i is either a room-shift vertex or it's slack preserving, and we define the next vertex b_{i+1}. Let now $w = \gamma_1(b_i)$ and $w' = \gamma_2(b_i)$.

Case 1: b_i is a room shift vertex. Then, by (J2.c), (w', w) is downward and $slack(w', w) = slack(b_i, w) = 1$. From Claim 6.1, w' is a slack-1 vertex. We set $b_{i+1} = w'$.

Case 2: b_i is slack-preserving.

Case 2.1: w is pivotal for b_i. Then we must have $w \prec w'$. (Otherwise, if $w' \prec w$, by the definition of slack-preserving vertices, w would be stable and thus couldn't be pivotal.)

We also have $slack(w', w) = 1$ and thus, from Claim 6.1, w' is a slack-1 vertex. We set $b_{i+1} = w'$.

Case 2.2: The pivotal vertex of b_i is not w. Let $w'' = \gamma_3(b_i)$. Since $slack(b_i, w) = 1$, (J2.d) implies that w' is the pivotal vertex for b_i and $slack(w'', w') = 1$. (This could happen only when $deg^-(b_i) = 3$ and b_i is stable, or $deg^-(b_i) = 4$ and b_i is unstable.) Similarly as in the previous case, we get that $w'' \prec w'$ is impossible, and thus, using Claim 6.1, w'' is a slack-1 vertex. We set $b_{i+1} = w''$.

Step II: Construction of the $\{c_i\}$*:* Let now $c_0 = b_f$. Let $w = \gamma_1(c_0)$ and $c_1 = \gamma_2(c_0)$. Since c_0 is slack-reducing, we have $slack(c_1, w) = 2$ and (c_1, w) is downward. Thus c_1 is either a slack-1 or a slack-2 vertex. Suppose we already have defined some c_i, which is either a slack-1 or a slack-2 vertex. If c_i is slack-2 and slack-reducing, we set $g = i$ and $h = 1$, and the construction of the sequence $S(v)$ is complete. If c_i is slack-1, we set $g = i$, completing the construction of the $c_1, \ldots, c_g$, and go to Step III. Otherwise, let $w = \gamma_1(c_i)$, $w' = \gamma_2(c_i)$, and we construct c_{i+1}, as follows.

Case 1: c_i is room-shift. Then, by (J2.c), (w', w) is downward and $slack(w', w) = 2$. By Claim 6.2, w' is either a slope-1 or a slope-2 vertex. We set $c_{i+1} = w'$.

Case 2: c_i is slack-preserving. Let $w'' = \gamma_3(c_i)$.

Case 2.1: w is pivotal for c_i. Then (w', w) is downward and $slack(w', w) = 2$. By Claim 6.2, w' is either a slope-1 or a slope-2 vertex. We set $c_{i+1} = w'$.

Case 2.2: w' is pivotal for c_i. By the definition of pivotal vertices, either $w' \prec w''$, or $w'' \prec w'$ and w' is unstable. From (J2.d) we have $slack(w'', w') \in \{1, 2\}$. By Claim 6.1 and Claim 6.2, w'' is either a slope-1 or a slope-2 vertex. We set $c_{i+1} = w''$.

Case 2.3: w and w' are not pivotal for c_i. Let $\tilde{w} = \gamma_4(c_i)$. By (J2.d), this case is possible only when w'' is pivotal and $slack(\tilde{w}, w'') = 1$. By Claim 6.1, $\tilde{w}$ is a slope-1 vertex. We set $c_{i+1} = \tilde{w}$.

Step III: Construction of the $\{d_i\}$*:* We execute this step only when c_g is slack-1. Let $d_1 = c_g$. All $d_1, \ldots, d_h$ are slack-1 vertices, and they are constructed in exactly the same fashion as the sequence $b_1, \ldots, b_f$. Vertex d_h is the first slope-reducing vertex in this sequence.

Claim 6.3: All μ-mates are different and they satisfy the property (ms).

That $\mu_1(v) \neq \mu_2(v)$ follows directly from the construction, since $g \geq 1$ and $\mu_2(v) \prec \mu_1(v)$. In the construction, the successor of each vertex in $S(v) - \{\mu_2(v)\}$ is uniquely defined. Furthermore, each vertex in $S(v) - \{v\}$ is covered by its predecessor. Therefore, for $v \neq v'$, the sequences $S(v)$, $S(v')$ are disjoint. □

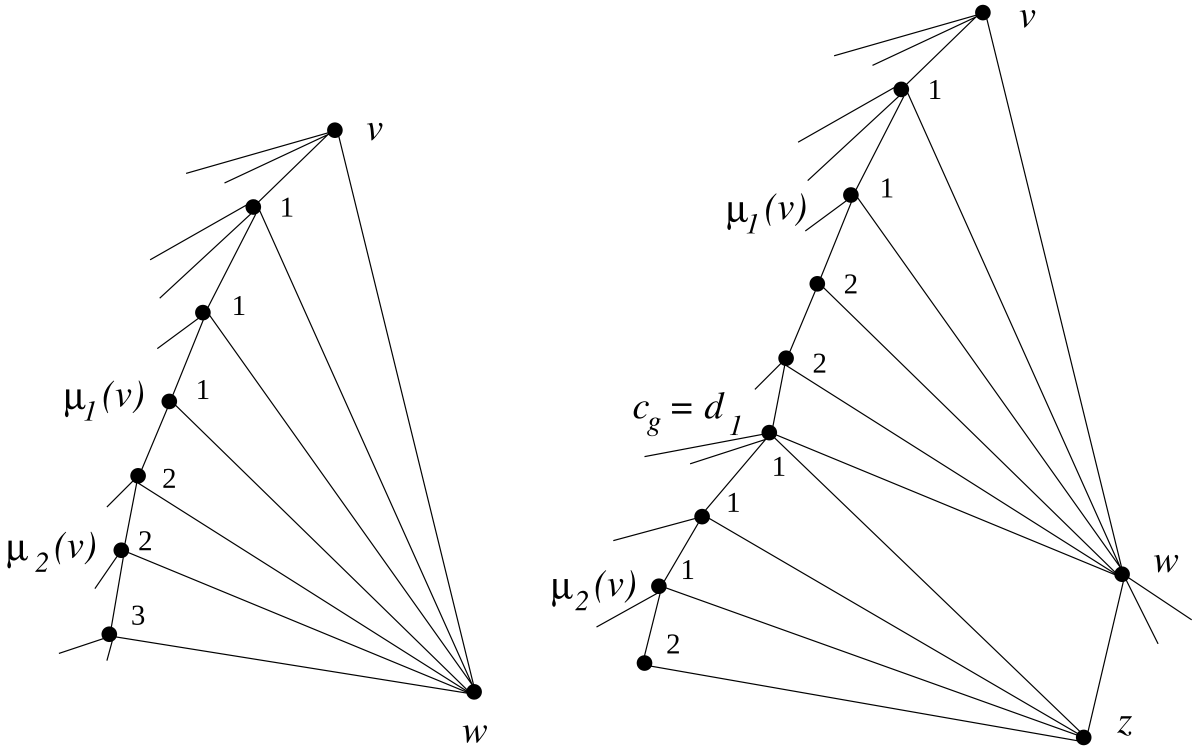

Figure 7: To examples of the construction of μ-mates. In the first example $\mu_2(v)$ is a slack-2 vertex, in the second example $\mu_2(v)$ is a slack-1 vertex.

Width estimate. We use Lemmas 5 and 6 to estimate the grid width. We show first a simple and intuitive proof of the $2(n-1)/3$ upper bound.

We refer the reader to Figure 8 that illustrates our argument. Some of the arrows correspond to mate assignments. The arrows from backward-oriented room-shift vertices go to their σ-mates. The arrows from slope-shift vertices go to their two μ-mates. Because $a_{\mathrm{f}} \leq a_{\mathrm{b}}$, to each in-degree 2 forward-oriented vertex we can assign a different in-degree 2 backward-oriented vertex. This is illustrated by the arrows from forward-oriented room-shift vertices.

Now we reason as follows: Suppose that each shift vertex receives initially a charge of 1, and non-shift vertices get no charge. We show that these charges can be distributed among the vertices in such a way that each vertex ends up with a charge of at most $2/3$. The distribution of charges is indicated by the numbers on the arrows. For example, slope-shift vertices transfer a charge of $1/3$ to each of their μ-mates. The reader should have no difficulty verifying that each vertex indeed receives a charge of at most $2/3$. One useful observation is that, because of Invariant (J1.b) in the proof of Lemma 4, non-vertical forward edges must have slack at least 2, and thus in-degree-2 forward-oriented vertices cannot be slope-shift.

If ω is the width of the graph, then the number of shifts is $\omega - 2$. Since v_1, v_2, v_3, v_n are not

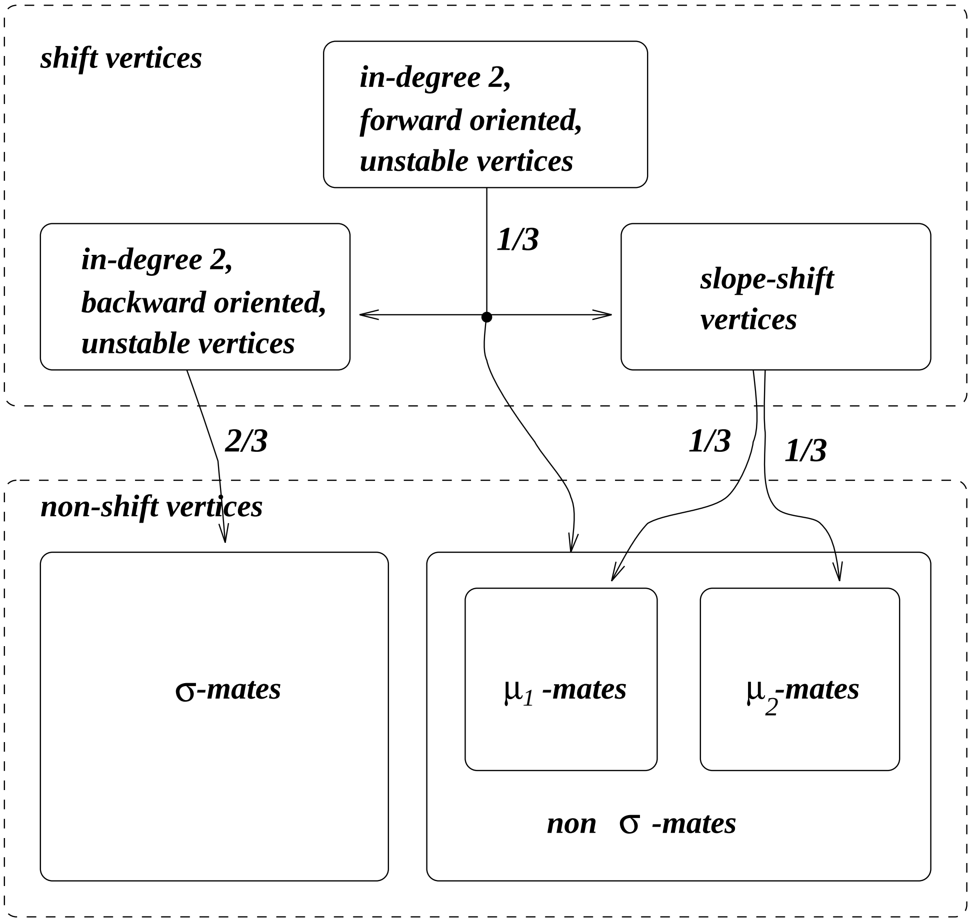

Figure 8: Proving the $2(n-1)/3$ width estimate using charges.

mates, and receive no charges, we have $\omega - 2 \le 2(n-4)/3$, and therefore $\omega \le 2(n-1)/3$.

A more rigorous proof of the width estimate is given below.

Theorem 3 *Given a triangulated n-vertex plane graph G, Algorithm $\mathcal{B}$ constructs a grid drawing of G into a $\omega \times (4\omega - 1)$ grid, for $\omega \le \lfloor 2(n-1)/3 \rfloor$.*

Proof: By Lemma 4, it is sufficient to estimate the width of the drawing. We can assume that we have found mate assignments σ, μ_1 and μ_2 that satisfy Lemmas 5 and 6. Since conditions (mr1)-(mr3) contradict (ms), all mates are distinct. We show that this implies the $\lfloor 2(n-1)/3 \rfloor$ upper bound on the grid width.

As before, let a be the number of vertices of in-degree 2, and let b be the number of vertices of in-degree at least 3. We will refer to them as a-vertices and b-vertices, for short. By a_{f} and a_{b} we denote the numbers of forward and backward-oriented vertices of in-degree 2. For $\xi = a, b$, by ξ^{rs}, ξ^{ss} and ξ^{ns} we denote the number of ξ-vertices which are room-shift, slope-shift and no-shift, respectively. Similarly, by ξ^{mr} and ξ^{ms} we denote the number of ξ-vertices that satisfy one of the conditions (mr1)–(mr3), and condition (ms), respectively. As usual, in all the quantities defined above we take into account only vertices $v_4, \ldots, v_n$ and ignore v_1, v_2, v_3.

We will also combine subscripts and superscripts, with an obvious interpretation. For example, $a_{\mathrm{f}}^{\mathrm{ns}}$ is the number of no-shift forward-oriented vertices of in-degree 2. Some such combinations will be void, for example, by the definition of room-shift vertices, we have $b^{\mathrm{rs}} = 0$. Also, by Invariant (J1.b) in the proof of Lemma 4, downward forward edges must have slack at least 2, implying that $a_{\mathrm{f}}^{\mathrm{ss}} = 0$ and $a^{\mathrm{ss}} = a_{\mathrm{b}}^{\mathrm{ss}}$.

Let ω be the width of the grid drawing of G produced by Algorithm $\mathcal{B}$. First, we have the following equations:

$$
\begin{aligned}
\omega &= a^{\mathrm{rs}} + a^{\mathrm{ss}} + b^{\mathrm{ss}} + 2 && (1)\\
n &= a + b + 3\\
&= a^{\mathrm{rs}} + a^{\mathrm{ss}} + b^{\mathrm{ss}} + a^{\mathrm{ns}} + b^{\mathrm{ns}} + 3\\
&= \omega + (a^{\mathrm{ns}} + b^{\mathrm{ns}}) + 1 && (2)
\end{aligned}
$$

Since $a_{\mathrm{f}} \leq a_{\mathrm{b}}$ and $a_{\mathrm{f}}^{\mathrm{ss}} = 0$, we have

$$
a_{\mathrm{f}}^{\mathrm{rs}} + a_{\mathrm{f}}^{\mathrm{ns}} \leq a_{\mathrm{b}}^{\mathrm{rs}} + a_{\mathrm{b}}^{\mathrm{ss}} + a_{\mathrm{b}}^{\mathrm{ns}} \qquad (3)
$$

By the existence of the σ- and μ-mates, we have

$$
\begin{aligned}
a_{\mathrm{b}}^{\mathrm{rs}} &\leq a_{\mathrm{f}}^{\mathrm{mr}} + b^{\mathrm{mr}} - 1 && (4)\\
2(a_{\mathrm{b}}^{\mathrm{ss}} + b^{\mathrm{ss}}) &\leq a_{\mathrm{f}}^{\mathrm{ms}} + a_{\mathrm{b}}^{\mathrm{ms}} + b^{\mathrm{ms}} && (5)
\end{aligned}
$$

We can subtract 1 in inequality (4) because v_n satisfies condition (mr3) but it cannot be a σ-mate, by Lemma 5. Multiply inequality (4) by 2, and add it to (5). This yields

$$
\begin{aligned}
2(a_{\mathrm{b}}^{\mathrm{rs}} + a_{\mathrm{b}}^{\mathrm{ss}} + b^{\mathrm{ss}}) &\leq 2a_{\mathrm{f}}^{\mathrm{mr}} + a_{\mathrm{f}}^{\mathrm{ms}} + a_{\mathrm{b}}^{\mathrm{ms}} + 2b^{\mathrm{mr}} + b^{\mathrm{ms}} - 2\\
&\leq 2a_{\mathrm{f}}^{\mathrm{ns}} + a_{\mathrm{b}}^{\mathrm{ns}} + 2b^{\mathrm{ns}} - 2, && (6)
\end{aligned}
$$

where the second inequality follows from that fact that no mate is a shift-vertex. Now add (6) and (3), getting

$$
\begin{aligned}
a_{\mathrm{f}}^{\mathrm{rs}} + a_{\mathrm{b}}^{\mathrm{rs}} + a_{\mathrm{b}}^{\mathrm{ss}} + 2b^{\mathrm{ss}} &\leq a_{\mathrm{f}}^{\mathrm{ns}} + 2a_{\mathrm{b}}^{\mathrm{ns}} + 2b^{\mathrm{ns}} - 2\\
&\leq 2(a^{\mathrm{ns}} + b^{\mathrm{ns}}) - 2 && (7)
\end{aligned}
$$

and then, using (7) and (1), we obtain

$$
\begin{aligned}
\omega &= a^{\mathrm{rs}} + a^{\mathrm{ss}} + b^{\mathrm{ss}} + 2\\
&\leq a_{\mathrm{f}}^{\mathrm{rs}} + a_{\mathrm{b}}^{\mathrm{rs}} + a_{\mathrm{b}}^{\mathrm{ss}} + 2b^{\mathrm{ss}} + 2\\
&\leq 2(a^{\mathrm{ns}} + b^{\mathrm{ns}}). && (8)
\end{aligned}
$$

Finally, inequalities (2) and (8) imply that

$$\omega \;\leq\; 2(n-1)/3$$

as required. $\Box$

Theorem 4 *Algorithm $\mathcal{B}$ can be implemented in linear time.*

Proof: The implementation is very similar to the one in [CP95], so we only sketch it briefly here.

Canonical orderings can be computed in linear time, as described in [FPP88, Ka93]. To determine the final canonical ordering, we determine which edges are forward and which are backward, and compute the number of forward-oriented and backward-oriented vertices of in-degree 2. If necessary, we will replace (G,π) by (G',π'). All of this can be done easily in linear time.

The straightforward implementation of the construction of the drawing runs in $\Omega(n^2)$ time, since the shift operations can cost as much as $\Omega(n)$ time each. In order to speed it up, we need to install each v_k in time $O(deg^-(v_k))$, which adds up to $O(n)$. This is achieved by postponing the shift operations, and computing only relative x-distances between vertices whenever necessary.

Represent the structure of U-sets in a directed tree T. At each time step, T contains contour edges $w_i \to w_{i+1}$. Also, if v covers u, then T will contain edge $v \to u$. Vertex v_1 is the root of T.

For each vertex v, store $y(v)$. The y-coordinates do not change during the algorithm. For each edge $u \to v$ in T, store the offset value $\Delta x(u,v)$. However, we do *not* store the x-coordinates. If (w_j, w_{j+1}) is a contour edge, then a shift operation *shift*(w_{j+1}) affects only one offset value, namely $\Delta x(w_j, w_{j+1})$. Also, given all $\Delta x(w_i, w_{i+1})$ for contour edges, we can determine, in $O(deg^-(v_k))$ time, the shape of the path $w_p, \ldots, w_q$ (but not its exact location in the plane), and this information is sufficient to determine $y(v_k)$, $\Delta x(w_p, v_k)$ and $\Delta x(v_k, w_j)$ for all $j = p+1, \ldots, q$. $\Box$

6 Final Comments

We have shown that plane graphs have grid drawings of width $\lfloor 2(n-1)/3 \rfloor$, which is optimal, and height $4\lfloor 2(n-1)/3 \rfloor - 1$. Our height analysis for Algorithm $\mathcal{B}$ is nearly tight, since there are examples of graphs on which it uses grids of height $4\lfloor 2(n-1)/3 \rfloor - O(1)$.

Improving Height. The most intriguing question is whether the grid height can be improved. One possibility for improving the height is to use a more restrictive invariant on the slope, by replacing the bound of -4 by, say, -3. It is not hard to see that one obtains correct grid drawings with this change. Unfortunately, we have an example showing that such a modified algorithm

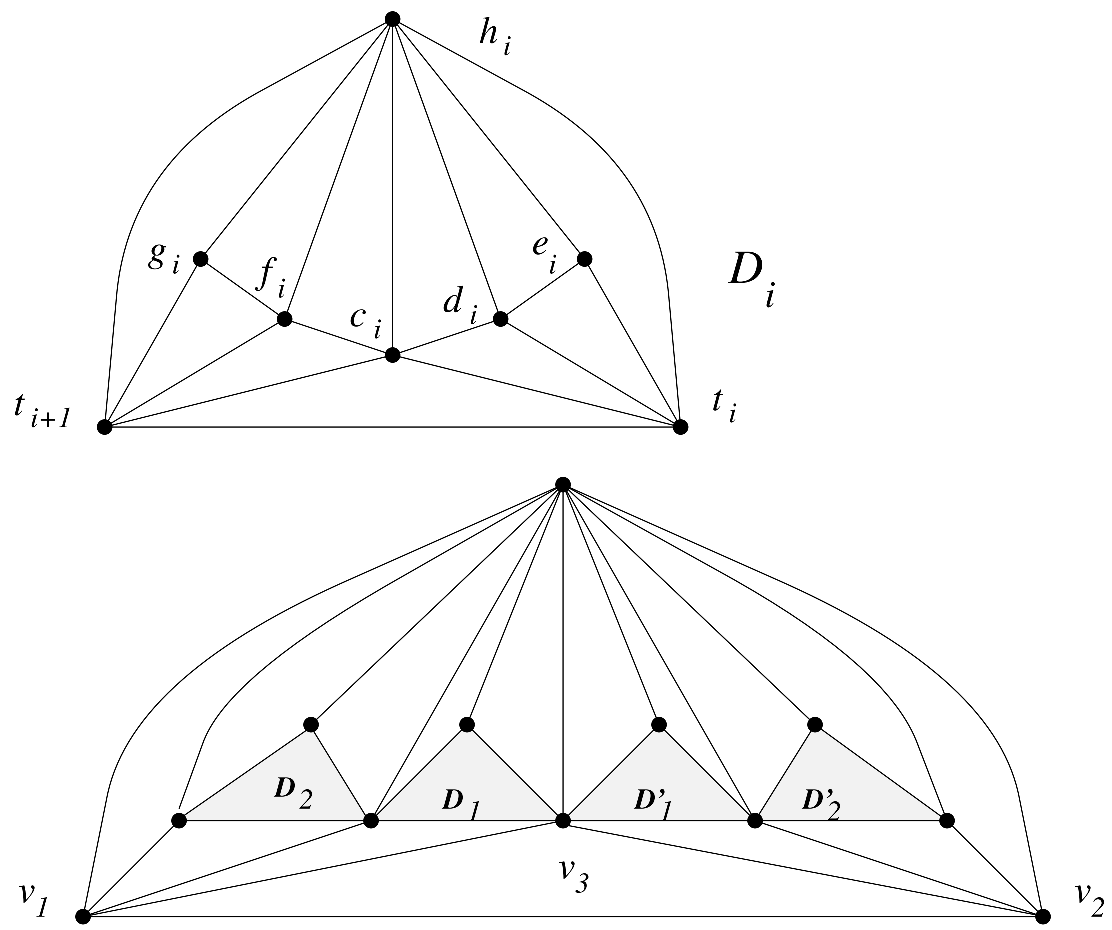

Figure 9: An example showing that replacing slope bound -4 by -3 can increase the width to $5n/7 + O(1)$.

uses width $5n/7 + O(1)$: We start with the triangle (v_1, v_2, v_3). Identify $t_0 = v_3$, and add vertices $t_1, \ldots, t_m$, where each t_i, for $i > 0$, is connected to v_1 and t_{i-1}. Then, for each $i = 0, \ldots, m-1$ we add the component $D_i = \{t_i, c_i, d_i, e_i, f_i, g_i, h_i\}$ with edges: (c_i, t_i), (c_i, t_{i+1}), (d_i, c_i), (d_i, t_i), (e_i, d_i), (e_i, t_i), (f_i, t_{i+1}), (f_i, c_i), (g_i, f_i), (g_i, t_{i+1}), (h_i, t_{i+1}), (h_i, g_i), (h_i, f_i), (h_i, c_i), (h_i, d_i), (h_i, e_i) and (h_i, t_i). We add, symmetrically, vertices t'_i, and components $D'_i = \{t'_i, c'_i, \ldots, h'_i\}$, on the other side of v_3. Finally, we add one more vertex v_n and connect it to all vertices on the outer face. (See Fig. 9.) It is easy to see that, independently of how we choose the canonical ordering of $v_4, \ldots, v_{n-1}$, we have $a_{\mathrm{f}} = a_{\mathrm{b}}$. In each component D_i, we make room-shifts for t_i, c_i, f_i, g_i and one slope-shift for e_i. Thus, for each group of 7 vertices, we make 5 shifts.

Convex Drawings. Chrobak and Kant [CK93] and, independently, Schnyder and Trotter [ST92], proved that 3-connected planar graphs have convex drawings in a $(n-2) \times (n-2)$ grid. Is it possible to modify Algorithm $\mathcal{B}$ to produce convex drawings of 3-connected graphs without increasing the grid size? We conjecture that the answer is positive.

Drawing planar graphs. As we already pointed out in the introduction, for the purpose of grid drawings it is important to distinguish between *planar* graphs and *plane* graphs. If we wish to draw a planar graph, our algorithm is allowed to choose an embedding, and this additional flexibility can be used to reduce the grid size. The proof of the $\lfloor 2(n-1)/3 \rfloor$ lower bound (Theorem 1) does not apply to planar graphs. If the embedding of the graph H_n from this proof is not fixed, one can only prove a $n/3 + \Omega(1)$ lower bound on the width. In fact, H_n can be drawn in a grid of width $n/3 + O(1)$ if a suitable external face is chosen. (We leave the proofs of both facts as an exercise.)

Overall, very little is known about width and area requirements for grid drawings of planar graphs, that is, when we are allowed to choose which face is external. We believe it is an interesting problem that requires further study.

Drawing in a window. In some graph visualization applications one may want to draw a given graph in a prescribed rectangular region, e.g. window on a computer screen. The techniques from [FPP90, CP95, Sc90, CK93] can be used if a window's proportions are 1:1 or 1:2. For windows that are tall and thin, the algorithm from this paper may be applicable. One of the goals of this area of research should be to extend this further, and to determine an optimal width-height trade-off for grid drawings. Then, for any feasible pair (w, h) we could apply a method that gives best drawings in grids of size $w \times h$.

Minimizing the width and area. Although in general one cannot reduce the grid width to below $\lfloor 2(n-1)/3 \rfloor$, many plane graphs can be drawn in grids of smaller width. Is it possible to determine in polynomial time whether a given plane graph G can be drawn in a grid of width w? A number of related problems have been proven NP-complete [ELL93, DLT84, Br88], see also [DETT94]. None of those proofs seems to carry over to the case of grid drawings without restrictions on edge slopes.

One can also ask a similar question for a prescribed window, or area. A related problem for upward tree drawings was studied in [GGT93].

Acknowledgements. We would like to thank John Noga and the anonymous referees for insightful comments, useful suggestions, and for pointing out several mistakes in the earlier version of this paper.

References

[Br88] F. J. Brandenburg, *Nice drawings of graphs and of trees are computationally hard*, Technical Report MIP-8820, Fakultät für Mathematik und Informatik, Universität Passau, 1998.

[CDP92] P. Crescenzi, G. Di Battista, A. Piperno, *A note on optimal area algorithms for upward drawings of binary trees*, Computational Geometry: Theory and Applications 2 (1992) 187-200.

[CK93] M. Chrobak, G. Kant, *Convex grid drawings of 3-connected planar graphs*, Technical Report RUU-CS-93-45, Department of Computer Science, Utrecht University, 1993.

[CON85] N. Chiba, K. Onoguchi, and T. Nishizeki, *Drawing planar graphs nicely*, Acta Informatica 22 (1985) 187-201.

[CP95] M. Chrobak, T. Payne, *A linear-time algorithm for drawing a planar graph on a grid*, Information Processing Letters 54 (1995) 241-246.

[CYN84] N. Chiba, T. Yamanouchi, T. Nishizeki, *Linear algorithms for convex drawings of planar graphs*, in Progress in Graph Theory, J.A. Bondy and U.S.R. Murty (eds.), Academic Press, 1984, pp. 153-173.

[DETT94] G. Di Battista, P. Eades, R. Tamassia, I.G. Tollis, *Automatic graph drawing: an annotated bibliography*, Computational Geometry: Theory and Applications 4 (1994) 235-282.

[DLT84] D. Dolev, T. Leighton, H. Trickey, *Planar embeddings of planar graphs*, Advances in Computing Research 2 (1984) 147-161.

[DTT92] G. Di Battista, R. Tamassia, I. G. Tollis, *Area requirements and symmetry display of planar upward drawings*, Discrete Computational Geometry 7 (1992) 381–401.

[ELL93] P. Eades, T. Lin, X. Lin, *Two tree drawing conventions*, International Journal of Computational Geometry & Applications 3 (1993) 133-153.

[Fa48] I. Fáry, *On straight lines representation of plane graphs*, Acta. Sci. Math. Szeged 11 (1948) 229-233.

[FPP88] H. de Fraysseix, J. Pach, R. Pollack, *Small sets supporting Straight-Line Embeddings of planar graphs*, Proc. 20th Annual Symposium on Theory of Computing, 1988, pp. 426-433.

[FPP90] H. de Fraysseix, J. Pach, R. Pollack, *How to draw a planar graph on a grid*, Combinatorica 10 (1990) 41-51.

[GGT93] A. Garg, M.T. Goodrich, R. Tamassia, *Area-efficient upward tree drawings*, Proc. 9th Annual ACM Symposium on Computational Geometry, 1993, pp. 359–368.

[He95] X. He, *Grid embedding of 4-connected plane graphs*, in Proc. Graph Drawing'95, Sep, 1995, Passau, Germany. Also to appear in Discrete and Computational Geometry.

[Jon93] S. Jones, P. Eades, A. Moran, N. Ward, G. Delott, R. Tamassia, *A note on planar graph drawing algorithms*, Technical Report 216, Dept. of Computer Science, Univ. of Queensland, 1991.

[Ka92] G. Kant, *Drawing planar graphs using the lmc-ordering*, Proc. 33rd Symposium on Foundations of Computer Science, Pittsburgh, 1992, pp. 101-110.

[Ka93] G. Kant, *Algorithms for Drawing Planar Graphs*, Ph.D. Dissertation, Department of Computer Science, University of Utrecht, 1993.

[Sc90] W. Schnyder, *Embedding planar graphs in the grid*, Proc. 1st Annual ACM-SIAM Symp. on Discrete Algorithms, San Francisco, 1990, pp. 138-147.

[ST92] W. Schnyder, W. Trotter, *Convex drawings of planar graphs*, Abstracts of the AMS, 13, 5 (1992), 92T-05-135.

[St51] S. K. Stein, *Convex maps*, Proc. Amer. Math. Soc., 2 (1951) 464-466.

[Tu63] W.T. Tutte, *How to draw a graph*, Proc. London Math. Soc. 13 (1963) 743-768.

[Wa36] K. Wagner, *Bemerkungen zum vierfarbenproblem*, Jahresbericht der Deutschen Mathematiker-Vereinigung 46 (1936) 26-32.