Algorithmica (2010) 57: 381–397
DOI 10.1007/s00453-008-9215-x

Planar Polyline Drawings via Graph Transformations

Huaming Zhang

Received: 2 September 2007 / Accepted: 18 July 2008 / Published online: 15 August 2008

Abstract We study the problem of transforming plane triangulations into irreducible triangulations, which are plane graphs with a quadrangular exterior face, triangular interior faces and no separating triangles. Our linear time transformation reveals important relations between the minimum Schnyder's realizers of plane triangulations (Bonichon et al., Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, vol. 2607, pp. 499–510, Springer, Berlin, 2003; Research Report RR-1279-02, LaBRI, University of Bordeaux, France; Brehm, Diploma thesis, FB Mathematik und Informatik, Freie Universität Berlin, 2000) and the transversal structures of irreducible triangulations (Fusy, Proceedings of 13th International Symposium on Graph Drawing, Lecture Notes in Computer Science, vol. 3843, pp. 177–188, Springer, Berlin, 2005; He, SIAM J. Comput. 22:1218–1226, 1993). The transformation morphs a 3-connected plane graph into an internally 4-connected plane graph. Therefore some of the graph algorithms designed specifically for 4-connected plane graphs can be applied to 3-connected plane graphs indirectly. As an example of such applications, we present a linear time algorithm that produces a planar polyline drawing for a plane graph with n vertices in a grid of size bounded by $W \times H$, where $W \leq \lfloor \frac{2n-2}{3} \rfloor$, and $W + H \leq \lfloor \frac{4n-4}{3} \rfloor$. It uses at most $\lfloor \frac{2n-5}{3} \rfloor$ bends, and each edge uses at most one bend. Our algorithm is area optimal. Compared with the existing area optimal polyline drawing algorithm proposed in Bonichon et al. (Proceedings of the 28th International Workshop on Graph-Theoretic Concepts in Computer Science, Lecture Notes in Computer Science, vol. 2573, pp. 35–46, Springer, Berlin, 2002), our algorithm uses a smaller number of bends. Their bend bound is $(n-2)$.

This research is supported in part by NSF grant CCF-0728830.

H. Zhang (✉)
Computer Science Department, University of Alabama in Huntsville, Huntsville, AL 35899, USA
e-mail: hzhang@cs.uah.edu

Keywords Plane graph · Polyline drawing

1 Introduction

A *plane triangulation* G is a plane graph where every face of G is a triangle (including the exterior face). A *realizer* $\mathcal{R}$ of a plane triangulation G is a partition of the set of its interior edges into three particular trees [12, 13]. All the realizers of G form a distributive lattice [6]. The minimum element of the distributive lattice is called the minimum realizer of G, which has some special properties and they have been widely used in graph encoding, graph enumeration, graph generation [3, 5], and graph drawing [14]. An *irreducible triangulation* G' is an internally triangulated plane graph with four exterior vertices and no separating triangles. G' is internally 4-connected, since adding a vertex in the exterior face and connecting this vertex to all the exterior vertices of G' makes G' 4-connected. Combinatorially, G' is associated with *transversal structures* [7, 8]. A transversal structure $\mathcal{T}$ of an irreducible triangulation G' is a partition of the set of its interior edges into two special subgraphs. The transversal structures have served as a rudimentary step for many graph algorithms, including rectangular dual for irreducible triangulations [8], straight line grid embedding for irreducible triangulations [7, 9, 11], and floor-planning for plane triangulations [10]. In this paper, we introduce a transformation which converts a plane triangulation G into an irreducible triangulation G' by a certain number of operations. These operations include insertion of vertices at appropriate places, insertion and deletion of diagonals. These operations are determined by the minimum realizer of G and can be done in linear time. Furthermore, the structure of the minimum realizer of G is quite well kept in the transversal structure of G'. Compared with the transformations introduced in [15], the transformation introduced in this paper only needs to add $\leq \lfloor \frac{n-1}{3} \rfloor$ vertices into a plane triangulation G with $n \geq 4$ vertices to make G an irreducible triangulation, while the transformation in [15] needs to add up to $\lfloor \frac{2n-5}{3} \rfloor$ vertices to make G an irreducible triangulation. It is conceivable that the transformation introduced in this paper is more preferable in some applications, as will be demonstrated.

This transformation allows us to apply certain graph algorithms specifically designed for 4-connected plane graphs to 3-connected plane graphs. We focus on polyline drawings [1] for plane graphs. A *polyline drawing* is a drawing of a plane graph in which each edge is represented by a polygonal chain and every vertex is placed at a grid point. Bonichon et al. [2] presented a linear time algorithm that produces polyline drawings for a plane graph with n vertices within a grid of area $(n - \lfloor \frac{p}{2} \rfloor - 1) \times (p + 1)$, where $p \leq \frac{2n-5}{3}$. It is area optimal and each edge has at most one bend. However the total number of bends used by this algorithm could be $(n - 2)$.

As an application of our transformation, we present a linear time algorithm that produces a polyline drawing for a plane graph with n vertices in a grid of size bounded by $W \times H$, where $W \leq \lfloor \frac{2n-2}{3} \rfloor$, and $W + H \leq \lfloor \frac{4n-4}{3} \rfloor$. In addition, it uses at most $\lfloor \frac{2n-5}{3} \rfloor$ bends, and each edge uses at most one bend. Our algorithm is area optimal. Compared with the existing area optimal polyline drawing algorithm proposed in [2], our algorithm only uses at most $\lfloor \frac{2n-5}{3} \rfloor$ bends, while their bend bound is $(n-2)$. On the other hand, when compared with the drawing size of the polyline drawing algorithm in [15], our algorithm achieves the same bound on the total number of bends as it but our algorithm keeps the drawing in an area optimal grid, which is smaller than the one used in [15].

The present paper is organized as follows. In Sect. 2, we recall a few definitions. In Sect. 3, we present the transformation. Then we present our drawing algorithm.

2 Preliminaries

All graphs are simple plane graphs in this paper. We abbreviate clockwise and counter clockwise as "cw" and "ccw" respectively.

A *plane triangulation* is a plane graph where every face is a triangle (including the exterior face). Let G be a plane triangulation of n vertices with three exterior vertices v_1, v_2, v_n in ccw order. A *realizer* $\mathcal{R} = \{T_1, T_2, T_n\}$ of G is a partition of its interior edges into three trees T_1, T_2, T_n of directed edges, colored blue, green, red and rooted at v_1, v_2, v_n respectively, such that all of the following hold:

- For each $i \in \{1, 2, n\}$, the interior edges incident to v_i are in T_i and directed toward v_i.
- For each interior vertex v of G, v has exactly one edge leaving v in each of T_1, T_2, T_n. The ccw order of the edges incident to v is: leaving in T_1, entering in T_n, leaving in T_2, entering in T_1, leaving in T_n, and entering in T_2 (see Fig. 1). Each entering block could be empty.

Figure 2(1) shows a realizer of a plane triangulation G. The blue lines (green lines and red lines, respectively) are the edges in T_1 (T_2 and T_n, respectively).

In [12], Schnyder showed that every plane triangulation G has a realizer which can be constructed in linear time. For each tree T_i, we denote by $\overline{T}_i$ the tree composed of T_i augmented with the two exterior edges incident to the root of T_i. Obviously, $\overline{T}_i$ is a spanning tree of G. For example, in Fig. 2(1), $\overline{T}_n$ is T_n (the tree in red lines) augmented with the edge (v_1, v_n) and (v_2, v_n), which is a spanning tree of G.

Normally, realizers of a plane triangulation G are not unique. In [6], Brehm introduced an order structure to the set of all the realizers of G and proved that they form a distributive lattice. Among all the realizers, the unique minimum element in

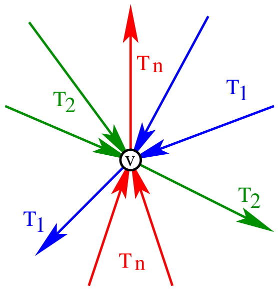


Fig. 1 (Color online) Edge directions around an interior vertex v

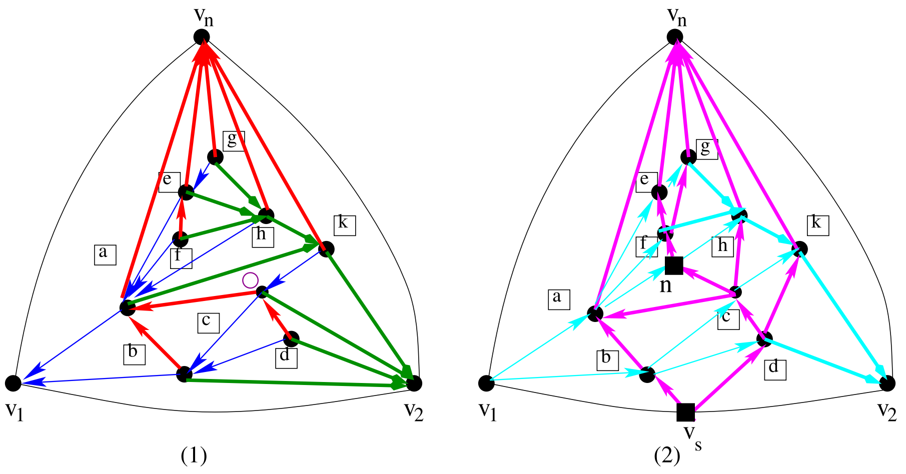


Fig. 2 (Color online) (**1**) A plane triangulation G and the minimum realizer $\mathcal{R}_0$ of G. (**2**) An irreducible triangulation G'_1 and a transversal structure $\mathcal{T}(G'_1)$ for G'_1

the distributive lattice is the realizer $\mathcal{R}_0$ of G, where according to the edge directions in $\mathcal{R}_0$, there are no ccw cyclic triangles. (Here, ccw cyclic triangles include both ccw cyclic faces and complex ccw cyclic triangles.) This realizer $\mathcal{R}_0$ of G will be called the *minimum realizer* of G. For example, in Fig. 2(1), the realizer has exactly one cw cyclic triangle (marked by an empty circle) but no ccw cyclic triangles. It is the minimum realizer of G. We will focus on the minimum realizer from now on in this paper.

A plane graph G' is an *irreducible triangulation* if G' has an quadrangular exterior face, triangular interior faces, and no separating triangles, i.e., each 3-cycle is the boundary of an interior face of G'. Irreducible triangulations form an important class of triangulations, as they are internally 4-connected. Namely, if we add one vertex into the exterior face of G', and connect this vertex to all the exterior vertices of G', G' becomes 4-connected. Next, we introduce the concept of *transversal structures* [7, 8] for irreducible triangulations. (In [8], it is called a *regular edge labeling*.)

Definition 1 Let G' be an irreducible triangulation, v_1, v_s, v_2, v_n be its four exterior vertices in ccw order. A *transversal structure* $\mathcal{T}(G')$ of G' is a partition of its interior edges into two sets, say in magenta and cyan edges, such that the following conditions are satisfied:

1. In cw order around each interior vertex v, its incident edges form: a non empty interval of magenta edges entering v, a non empty interval of cyan edges entering v, a non empty interval of magenta edges leaving v, and a non empty interval of cyan edges leaving v.
2. All interior edges incident to v_n are magenta edges entering v_n, all interior edges incident to v_s are magenta edges leaving v_s, all interior edges incident to v_1 are cyan edges leaving v_1, and all interior edges incident to v_2 are cyan edges entering v_2. Each such block is non empty.

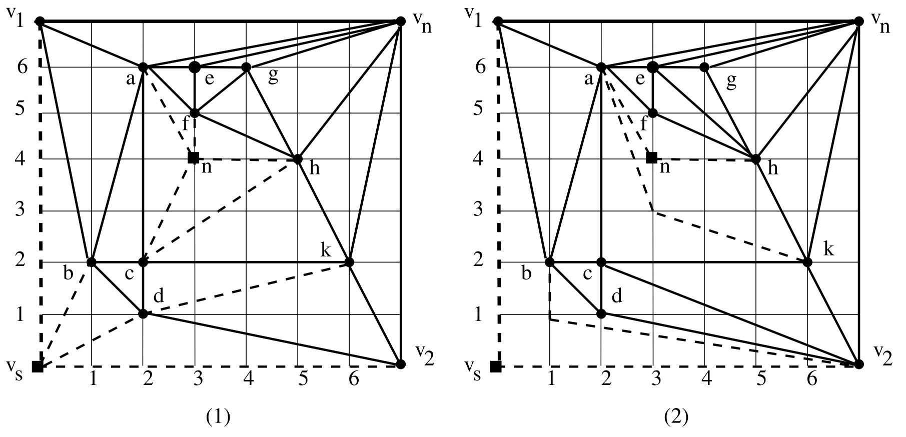


Fig. 3 (**1**) A straight-line grid drawing of the graph G'_1 in Fig. 2(2). (**2**) A polyline drawing of G in Fig. 2(1)

Consider an irreducible triangulation G' with a transversal structure $\mathcal{T}(G')$ as defined above. The subgraph of G' with all its magenta colored edges (cyan colored edges respectively) and all its four exterior edges is called the *magenta map* (*cyan map* respectively) of G', it is denoted by G'_m (G'_c respectively). For an interior vertex v in G'_m (G'_c respectively), the edges entering v and the edges leaving v form two non empty contiguous blocks. When walking from u to v in G'_m (G'_c respectively), if we always pick the first feasible outgoing edge (i.e., the outgoing edge could still lead to the vertex v) in ccw direction at all its intermediate vertices in the path, then the path is unique and it is called the *rightmost* path from u to v. Similarly, the *leftmost* path always chooses the first feasible cw outgoing edge at all its intermediate vertices in the path. For any interior vertex v of G', let $P_c(v)$ be the unique path from v_1 to v_2 in G'_c such that, the subpath of $P_c(v)$ from v_1 to v is the rightmost one before arriving at v, and the subpath of $P_c(v)$ from v to v_2 is the leftmost one after leaving v. Let $y(v)$ be the number of faces in G'_c enclosed by the path (v_1, v_s, v_2) and $P_c(v)$. Similarly, for any interior vertex v of G', let $P_m(v)$ be the unique path from v_s to v_n in G'_m such that, the subpath of $P_m(v)$ from v_s to v is the rightmost one before arriving at v, and the subpath of $P_m(v)$ from v to v_n is the leftmost one after leaving v. Let $x(v)$ be the number of faces in G'_m enclosed by the path (v_s, v_1, v_n) and $P_m(v)$. Let $x(\mathcal{T}(G'))$ be the number of interior faces of G'_m, $y(\mathcal{T}(G'))$ be the number of interior faces of G'_c. For the vertices v_1, v_s, v_2, v_n, we define $x(v_1) = 0$, $y(v_1) = y(\mathcal{T}(G'))$, $x(v_s) = 0$, $y(v_s) = 0$, $x(v_n) = x(\mathcal{T}(G'))$, $y(v_n) = y(\mathcal{T}(G'))$, and $x(v_2) = x(\mathcal{T}(G'))$, $y(v_2) = 0$. We have the following lemma from [7]:

Lemma 1 *Let G' be an irreducible triangulation, v_1, v_s, v_2, v_n be its four exterior vertices in ccw order. Then*:

1. *G' admits a transversal structure $\mathcal{T}(G')$, which is computable in linear time.*
2. *Applying $\mathcal{T}(G')$, for each vertex v, embed it on the grid point $(x(v), y(v))$. For each edge of G', simply connect its end vertices by a straight line. The drawing*

is a straight-line grid drawing for G'. Its drawing size is $x(\mathcal{T}(G')) \times y(\mathcal{T}(G'))$. This drawing is computable in linear time.

It is fairly straightforward to generalize Lemma 1 to a weighted version. A *legal weight function* $\mathcal{W}$ for $\mathcal{T}(G')$ assigns a positive integer to each interior face of G'_m and each interior face of G'_c; the assigned integer is called the *weight* of the face. The *unit weight function* $\mathcal{U}$ for $\mathcal{T}(G')$ is the legal weight function where $\mathcal{U}$ assigns 1 as the weight for every interior face of G'_m and G'_c. Given a legal weight function $\mathcal{W}$ for $\mathcal{T}(G')$, for any interior vertex v, let $P_c(v)$ and $P_m(v)$ be the same paths as defined before. Let $y_w(v)$ be the total weight of all the interior faces in G'_c enclosed by the path (v_1, v_s, v_2) and $P_c(v)$. Let $x_w(v)$ be the total weight of all the interior faces in G'_m enclosed by the path (v_s, v_1, v_n) and $P_m(v)$. Let $x_w(\mathcal{T}(G'))$ be the total weight of all the interior faces of G'_m. $y_w(\mathcal{T}(G'))$ be the total weight of all the interior faces of G'_c. For the vertices v_1, v_s, v_2, v_n, we define $x_w(v_1) = 0$, $y_w(v_1) = y_w(\mathcal{T}(G'))$, $x_w(v_s) = 0$, $y_w(v_s) = 0$, $x_w(v_n) = x_w(\mathcal{T}(G'))$, $y_w(v_n) = y_w(\mathcal{T}(G'))$, and $x_w(v_2) = x_w(\mathcal{T}(G'))$, $y_w(v_2) = 0$. We then have the following weighted version of Lemma 1:

Lemma 2 *Let G' be an irreducible triangulation, v_1, v_s, v_2, v_n be its four exterior vertices in ccw order. Let $\mathcal{T}(G')$ be a transversal structure for G'. Let G'_m and G'_c be its magenta map and cyan map respectively. Let $\mathcal{W}$ be a legal weight function of $\mathcal{T}(G')$. Then for each vertex v, embed it on the grid point $(x_w(v), y_w(v))$. For each edge of G', simply connect its end vertices by a straight line. The drawing is a straight-line grid drawing for G'. Its drawing size is $x_w(\mathcal{T}(G')) \times y_w(\mathcal{T}(G'))$. This drawing is computable in linear time.*

For example, Fig. 2(2) shows an example of a transversal structure $\mathcal{T}(G'_1)$ for the irreducible triangulation G'_1. For $\mathcal{T}(G'_1)$ in Fig. 2(2), let $\mathcal{W}$ be the legal weight function where for each interior face of its magenta map, its weight is 1; for each interior face of its cyan map, its weight is 1 except for the interior face enclosed by $\{v_1, b, c, k, h, n, a, v_1\}$, whose weight is 2. For the vertex f in G'_1 in Fig. 2(2), its $P_m(f)$ is $(v_s, d, c, n, f, e, v_n)$, and its $x_w(f) = 3$. Its $P_c(f)$ is the path (v_1, a, f, h, k, v_2), and its $y_w(f) = 5$. $x_w(\mathcal{T}(G'_1)) = 7$ and $y_w(\mathcal{T}(G'_1)) = 7$. Applying Lemma 2 to this legal weight function, Fig. 3(1) presents a straight-line grid drawing of the graph G'_1 in Fig. 2(2). For some edges, we use dashed lines here. The reason for this arrangement will be clear later.

From now on in this paper, if the legal weight function $\mathcal{W}$ is clear from the context, we will simply abbreviate $x_w(v)$ into $x(v)$, $y_w(v)$ into $y(v)$ respectively. We have the following simple observations regarding the coordinate monotonicity of the above grid drawing algorithms in Lemmas 1 and 2.

Observation 1

1. If (u, v) is a magenta edge directed from u to v in the magenta map G'_m of $\mathcal{T}(G')$, then $x(u) \le x(v)$, and $y(u) < y(v)$.
2. If (u, v) is a cyan edge directed from u to v in the cyan map of G'_c of $\mathcal{T}(G')$, then $y(u) \ge y(v)$, and $x(u) < x(v)$.

3 Polyline Drawing Algorithm

In this section, we present the transformation from plane triangulations to irreducible triangulations and its application in polyline drawing of plane graphs.

Let T be a rooted tree drawn in the plane, $v_1, v_2, \dots, v_n$ be the ccw preordering of the nodes of T. The subsequence $v_i, \dots, v_j$ is a *branch* of T if it is a chain (i.e., v_t is the parent of v_{t+1} for every $i \le t < j$), and if $(j-i)$ is maximal. The first node of each branch will be called a *glue node* of T. Each branch begins with a glue node and ends with a leaf node (they could be the same node). Therefore the number of glue nodes of T is the same as the number of leaves of T. The nodes of T in this ccw preordering $v_1, v_2, \dots, v_n$ can be partitioned into branches. We will number all the branches of T from the 1-st branch to the $\gamma(T)$-th branch according to the ccw preordering of the glue nodes they contain, where $\gamma(T)$ is the number of leaves of T. We define the *k-th node set* to be the set of all the nodes in the k-th branch, the *k-th edge set* to be set of all the edges in the branch augmented with the edge between its glue node (if it is not the root of T) and its parent in T. For example, consider $\overline{T}_1$ in Fig. 2(1). Its nodes can be partitioned into the following 7 branches: $\{v_1, v_2\}$, $\{b, d\}$, $\{c, k\}$, $\{a, h\}$, $\{f\}$, $\{e, g\}$, and $\{v_n\}$. The glue nodes are $\{v_1, b, c, a, f, e, v_n\}$. The 1-st node set is $\{v_1, v_2\}$, the 1-st edge set is $\{(v_1, v_2)\}$. The 2-nd node set is $\{b, d\}$, the 2-nd edge set is $\{(b, v_1), (d, b)\}$.

It is easy to see that a non-root node v is a glue node if and only if v has at least one sibling node appearing before v (not necessarily right before v) in the ccw preordering of T. Otherwise, the branch containing the parent of v would have included v.

Let G be a plane triangulation endowed with its minimum realizer $\mathcal{R}_0 = \{T_1, T_2, T_n\}$. As stated by Bonichon et al. in [3] and proved in [4], the tree $\overline{T}_i$, $i = 1, 2, n$ has the following:

Branch Property The interior nodes of G within the same branch of $\overline{T}_1$ ($\overline{T}_2$, $\overline{T}_n$ respectively) have the same parent in $\overline{T}_2$ ($\overline{T}_n$, $\overline{T}_1$ respectively).

For example, in Fig. 2(1), nodes a, h are in the same branch in $\overline{T}_1$, they have the same parent k in $\overline{T}_2$.

Let G be a plane triangulation with $n \ge 4$ vertices, v_1, v_2, v_n be its three exterior vertices in ccw order, $\mathcal{R}_0 = \{T_1, T_2, T_n\}$ be its minimum realizer. Next we will show three ways to transform G into an irreducible triangulation. Each transformation uses a different tree from $\overline{T}_i, i = 1, 2, n$. In Sect. 3.1, we will present the transformation from G to G'_1, which uses $\overline{T}_1$. (The other two transformations to $G'_i, i = 2, n$ use $\overline{T}_i, i = 2, n$ respectively and they can be done similarly.) The main idea is, when constructing incrementally the triangulation G endowed with its minimum realizer, a closely related irreducible triangulation G'_1 endowed with a transversal structure can be constructed in parallel. The principle here is to construct G by adding branches of $\overline{T}_1$ one by one, each branch addition leads to the addition of a face in the cyan map of the associated transversal structure for G'_1. In Sect. 3.2, we define an appropriate legal weight function $\mathcal{W}$ for G'_1 and apply Lemma 2 on $\mathcal{W}$ to obtain a straight-line grid drawing for G'_1. We then manipulate the drawing of G'_1 to make it a polyline drawing for the original graph G. In Sect. 3.3, we analyze the polyline drawing size.

3.1 Graph Transformation

In order to make the following presentation easier for the boundary conditions when using $\overline{T}_1$ to transform G, we color the exterior edges (v_1, v_2) and (v_1, v_n) blue. Both of them are directed towards v_1. We color (v_n, v_2) green, it is directed towards v_2. An edge (q, p) of $\overline{T}_1$ (oriented from q to p) is called a *split edge* for $\overline{T}_1$ if the first sibling g of q appearing after q (not necessarily right after q) in the ccw preordering traversal of $\overline{T}_1$ is a child of q in $\overline{T}_2$. For example, in Fig. 2(1), both (v_2, v_1) and (h, a) are split edges for $\overline{T}_1$. ((v_2, v_1) is a split edge is due to our above boundary coloring scheme.) Given a split edge (q, p), we add a *split node* n in the middle of (q, p) to split it into two corresponding edges (q, n) and (n, p). For $\overline{T}_1$, we define the *k-th expanded node set* to be the k-th node set of $\overline{T}_1$ augmented with all the split nodes for all the split edges in the k-th edge set of $\overline{T}_1$, the *k-th expanded edge set* to be the k-th edge set, except that each split edge in it is replaced by its two corresponding edges.

The construction of G starts from an empty graph. Then it adds one branch of $\overline{T}_1$ at each iteration. We will denote the intermediate graph after adding the k-th branch of $\overline{T}_1$ by G_k for G. G_k is simply the subgraph of G induced by all the vertices of the first k branches of $\overline{T}_1$. Note that, according to the properties of realizers, all G_k are 2-connected. In comparison, when transforming G into G'_1, we will instead add all the vertices and edges in the k-th expanded node set and the k-th expanded edge set at each iteration. In addition, we will manipulate the edge connections and colorings according to different cases as well. For convenience, we will denote the intermediate graph constructed for G'_1 by G'_{1k} after the k-th iteration.

After adding the 1-st branch of $\overline{T}_1$ into the empty graph, G_1 is simply (v_1, v_2). For G'_{11}, after we add the 1-st expanded edge set, G'_{11} becomes a path (v_2, v_s, v_1), where v_s is the split node added to split (v_1, v_2). We then reverse the two edges' directions and recolor them cyan to complete the first iteration.

Consider a k-th branch of $\overline{T}_1$ for $k \geq 2$. Let g be the glue node, l be the leaf node. Then the k-th branch can be denoted by $\{g, \ldots, l\}$, where it is possible that $g = l$ in a degenerated case. Let p be the parent of g in $\overline{T}_1$. Let q be the parent of g in the tree $\overline{T}_2$. According to the Branch Property for $\overline{T}_1$, q is also the parent for all the vertices $\{g, \ldots, l\}$ in $\overline{T}_2$. Obviously, both p and q are on the exterior boundary of G_{k-1} and $p \neq q$. Since g is a glue node, there must be a sibling s of g in $\overline{T}_1$ which is the last sibling before g in the ccw preordering of $\overline{T}_1$. The edge (s, p) is in $\overline{T}_1$. It is directed from s to p. s must also be in the exterior boundary of G_{k-1}. Otherwise, it would not be the last sibling of g in the ccw preordering of $\overline{T}_1$. We have the following two cases of the k-th iteration: (in Figs. 4 and 5, we use a solid line to represent an edge, a dashed line to represent a possibly degenerated path).

Case 1: s is a leaf node of $\overline{T}_1$. Then $s = q$ and (q, p) is a split edge. See Fig. 4(1) for adding the k-th branch to construct G.

To perform the k-th iteration to transform G into G'_1, we first add all the edges in the k-th expanded edge set into $G'_{1(k-1)}$. We then insert the green edge (l, q). However, we do not insert the other green edges (w, q) for $w \in \{g, \ldots, l\} - \{l\}$, even if there are any such edges. Note that $(q, p) = (s, p)$ and it is a split edge, a split node n has been added in (q, p). We add a magenta edge leaving from n to every node in

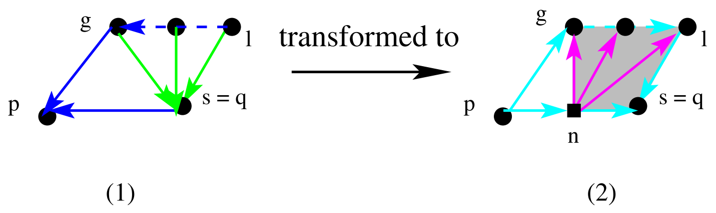


Fig. 4 (Color online) Case 1 of transforming the k-th branch

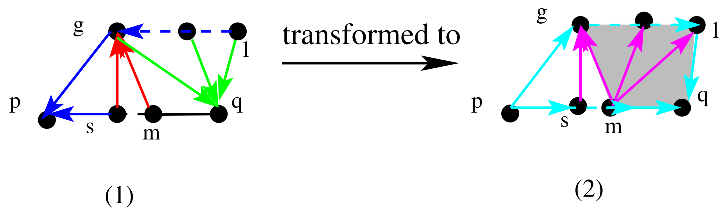


Fig. 5 (Color online) Case 2 of transforming the k-th branch

the k-th expanded node set. We complete this iteration by reversing the directions of all the blue edges, and recoloring blue and green cyan. See Fig. 4(2) for an illustration.

Case 2: s is a non-leaf node of $\overline{T}_1$. Then $s \neq q$. Let m be the vertex right before q on the exterior path of G_{k-1} from p to q in cw direction. This path can then be denoted by $\{p, s, \ldots, m, q\}$, where it is possible that $s = m$. See Fig. 5(1) for adding the k-th branch of G into G_{k-1}. Note that, the edges in the path from s to q on the contour of G_{k-1} could be either blue or green in G, in which case we color them black in the figure.

To perform the k-th iteration to transform G into G'_1, we first add all the edges in the k-th expanded edge set into $G'_{1(k-1)}$. We then insert the green edge (l, q). However, we do not insert the other green edges (w, q) for $w \in \{g, \ldots, l\} - \{l\}$, even if there are any such edges. Insert all the red edges as in G. If (g, p) is a split edge, then we add a magenta edge from s to the corresponding split node. For every other split node in the k-th expanded node set, we add a magenta edge entering it from m. We complete this iteration by reversing the directions of all the blue edges, recoloring blue and green cyan, and recoloring red magenta. See Fig. 5(2) for an illustration.

After all the iterations, we remove the colors and directions of all the exterior edges for G'_1 to finish the transformation.

Regarding whether the edges of G are in G'_1 or not, we have the following simple observation:

Observation 2

1. All the exterior edges of G are included in G'_1, though the split edge (v_1, v_2) is included as two edges in G'_1.
2. The red edges from G are all included in the magenta map of G'_1. The blue edges of G are all included in the cyan map of G'_1, though each split edge (which is indeed blue) is included as two edges in G'_1.

3. All the green edges in G leaving from leaves of $\overline{T}_1$ are included in the cyan map of G'_1. All the other green edges in G are not included in G'_1. They are called the *missing green edges*.

The first statement of the next technical lemma states that the magenta map and the cyan map constructed above for G'_1 is indeed a transversal structure. It is denoted by $\mathcal{T}(G'_1)$. ($\mathcal{T}(G'_i)$, $i = 2, n$ can be similarly defined.)

Lemma 3 *Let G be a plane triangulation with $n \geq 4$ vertices, v_1, v_2, v_n be its exterior vertices in ccw order, $\mathcal{R}_0 = \{T_1, T_2, T_n\}$ be its minimum realizer. T_i is rooted at v_i. Let γ_i be the number of leaves of T_i, for $i \in \{1, 2, n\}$. Then the following statements hold*:

1. *For every $i \in \{1, 2, n\}$, $\mathcal{T}(G'_i)$ is a transversal structure of G'_i.*
2. *Using the unit weight function $\mathcal{U}$ for $\mathcal{T}(G'_i)$, $x(\mathcal{T}(G'_i)) = \gamma_{i-1} + 1$, $y(\mathcal{T}(G'_i)) = \gamma_i + 1$ for all $i \in \{1, 2, n\}$. Here, $i - 1$ is understood as modular* 3 *subtraction, and n is treated as if it were* 3.
3. *G can be transformed into an irreducible triangulation by adding $\leq \lfloor \frac{n-1}{3} \rfloor$ vertices.*

Proof We only prove the case $i = 1$. The other two cases can be similarly proved.

1. $\mathcal{T}(G'_1)$ is a transversal structure is based on the following facts:
   - All the interior edges adjacent to v_1 are outgoing cyan edges. All the interior edges adjacent to v_2 are incoming cyan edges. All the interior edges adjacent to v_s are outgoing magenta edges. All the interior edges adjacent to v_n are incoming magenta edges.
   - Every interior vertex of G'_1, including the split nodes, has cyan incoming edges (which are the blue outgoing edges in G, though they might be split). For non-leaf nodes of $\overline{T}_1$, their incoming blue edges in G become outgoing cyan edges in G'_1. For leaves of $\overline{T}_1$, their outgoing green edges (which are included in G'_1) become outgoing cyan edges in G'_1.
   - Starting from the 2-nd iteration, each iteration adds an interior face $\mathcal{F}$ into the cyan map of G'_1. The facial cycle of $\mathcal{F}$ is decomposed into two directed paths from p to q. One path is on the bottom of $\mathcal{F}$, the other path is on the top of $\mathcal{F}$. Both start from p and end on q. See Figs. 4(2) and 5(2). For each vertex (including the split nodes) $v \notin \{p, q\}$ on the top of $\mathcal{F}$, its has a non empty contiguous block of magenta edges entering it inside $\mathcal{F}$. For each vertex (including the split nodes) $v \notin \{p, q\}$ on the bottom of $\mathcal{F}$, it has a non empty contiguous block of magenta edges leaving it inside $\mathcal{F}$. Observe that, for any particular interior vertex (including the split nodes) of G'_1, the block of magenta edges entering it and the block of magenta edges leaving it are in different interior faces of G'_1 and hence they are separated by the edges in the cyan map, which includes the cyan edges entering v and the cyan edges leaving v.
   - For each interior vertex of G'_1, its four non empty blocks of incoming magenta edges, incoming cyan edges, outgoing magenta edges, and outgoing cyan edges are in cw order.

Therefore, $\mathcal{T}(G'_1)$ is a transversal structure.

2. For the magenta map of $\mathcal{T}(G'_1)$, we start with n vertices $(n-1)$ edges from the tree $\overline{T}_n$, plus the edge (v_1, v_2). Suppose that we totally add x split nodes. Observe that, for each split node, we add exactly one magenta edge entering it except for v_s during the transformation. v_s splits the edge (v_1, v_2) into two edges, that also increases the edge number by one for the magenta map. In addition, for every leaf v of the red tree T_n, we add exactly one magenta edge entering v. Therefore, the total number of edges added is $x + \gamma_n$. Therefore, according to Euler's formula, we have the total number of faces of the magenta map $= (n - 1 + 1 + x + \gamma_n) - (n + x) + 2 = \gamma_n + 2$. Therefore, the total number of interior faces for the magenta map is $\gamma_n + 1$.

   Consider the cyan map of $\mathcal{T}(G'_1)$. The 1-st iteration adds no interior face. For each following index $k \geq 2$, the k-th iteration adds exactly one interior face to the cyan map. The total number of iterations equals the total number of branches of $\overline{T}_1$, which is $\gamma_1 + 2$. Therefore, the cyan map has $\gamma_1 + 1$ interior faces.
3. G'_i is associated with $\mathcal{T}(G'_i)$. Since the total number of vertices in a transversal structure G'_i equals the total number of the interior faces of its magenta map and cyan map plus one [7], so the total number of vertices in G'_i is $(\gamma_{i-1} + 1) + (\gamma_i + 1) + 1 = \gamma_{i-1} + \gamma_i + 3$. Therefore, the number of nodes added for G'_i is $\gamma_{i-1} + \gamma_i + 3 - n$. The total number of nodes added for all three G'_i, $i = 1, 2, n$ is $2(\gamma_1 + \gamma_2 + \gamma_n) + 9 - 3n$. In [3, 4, 6], it has been shown that $\gamma_1 + \gamma_2 + \gamma_n = (2n - 5 - \delta_0)$, where δ_0 is the number of the cw cyclic faces of G in the minimum realizer $\mathcal{R}_0$. Hence, the total number of nodes added for all three G'_i, $i = 1, 2, n$ is $2(2n - 5 - \delta_0) + 9 - 3n = (n - 2\delta_0 - 1)$. Therefore, one of the transformation only needs to add $\leq \lfloor \frac{n-2\delta_0-1}{3} \rfloor \leq \lfloor \frac{n-1}{3} \rfloor$ nodes. □

As stated in the proof of the above lemma, for the cyan map of $\mathcal{T}(G'_1)$, when $k \geq 2$, exactly one interior face of $\mathcal{T}(G'_1)$ is added to the cyan map when we perform the k-th iteration on G'_1. This interior face will be denoted by $\mathcal{F}_k$.

For example, the irreducible triangulation G'_1 and its transversal structure $\mathcal{T}(G'_1)$ in Fig. 2(2) is transformed from G in Fig. 2(1) by using $\overline{T}_1$. The transformation added two split nodes, v_s and n. It also deleted some green edges such as (b, v_2) and added some new edges such as (v_s, b). $\mathcal{F}_4$ is the face enclosed by $\{v_1, b, c, k, h, n, a, v_1\}$ in the cyan map in Fig. 2(2).

Note that, for the transformation introduced in [15], the number of vertices added could be $\lfloor \frac{2n-5}{3} \rfloor$, while the number of vertices added for the transformation in this paper is $\leq \lfloor \frac{n-1}{3} \rfloor$. As we will see in the remaining of the paper, this is crucial in controlling the size of the polyline drawing of G.

3.2 Recovering the Missing Green Edges

Given any legal weight function $\mathcal{W}$ for $\mathcal{T}(G'_1)$, applying Lemma 2, we obtain a straight-line grid drawing of G'_1. By removing the added new magenta edges and the added split nodes (as they are not from G), but keeping the split blue edges in the drawing, it becomes a partial polyline drawing of G. Note that, for those blue edges which have been split, each of them is drawn as a two-segment polyline in the partial drawing. According to Observation 2, the edges which are not included in the partial drawing are those missing green edges. Next, we will show how we can "recover" those missing green edges, by selecting an appropriate legal weight function $\mathcal{W}$ for $\mathcal{T}(G'_1)$.

First, we note that, the above transformation from the minimum realizer $\mathcal{R}_0$ of G to the transversal structure $\mathcal{T}(G'_1)$ of G'_1 keep all the red edges and their directions in the magenta map of $\mathcal{T}(G'_1)$. The transformation keeps all the blue edges, possibly split, of the minimum realizer in the cyan map of $\mathcal{T}(G'_1)$, but it reverses all its directions. For a green edge from the realizer, if it is not a missing green edge, it is in the cyan map and its original direction is kept. Therefore, according to Observation 1, we obtain:

Observation 3 Let u, v be two vertices of G'_1, then using any legal weight function $\mathcal{W}$, we have:

1. If (u, v) is a red edge directed from u to v in the minimum realizer, then $x(u) \leq x(v)$, and $y(u) < y(v)$.
2. If (u, v) is a blue edge directed from u to v in the minimum realizer, then $y(u) \leq y(v)$, and $x(u) > x(v)$.
3. If (u, v) is a green edge directed from u to v in the minimum realizer and it is not a missing green edge in the transformation, then $y(u) > y(v)$, and $x(u) < x(v)$.

Statements 1, 2, and the second conclusion of Statement 3 in Observation 3 are valid from Observation 1. Consider the first conclusion of Statement 3. Observe that if an interior vertex v of G is in the k-th branch of $\overline{T}_1$ for $k \geq 2$, then the region enclosed by $P_c(v)$ and (v_1, v_s, v_2) in the cyan map contains all the interior faces $\{\mathcal{F}_2, \dots, \mathcal{F}_k\}$. If (u, v) is a green edge directed from u to v and it is not a missing green edge in the transformation, then u is in a larger-indexed branch of $\overline{T}_1$ than v. Therefore, the set of the interior faces in the cyan map enclosed by $P_c(v)$ and (v_1, v_s, v_2) is a proper subset of the set of the interior faces in the cyan map enclosed by $P_c(u)$ and (v_1, v_s, v_2). Since a legal weight function always assigns positive integer weight to any interior face of the cyan map. Therefore, $y(u) > y(v)$.

Next, we will show how to select a legal weight function $\mathcal{W}$ such that: (1) all the missing green edges can be recovered, and (2) each such green edge has at most one bend after its recovery. $\mathcal{U}$ will refer to the unit weight function. $x_w(v)$ and $y_w(v)$ will be denoted by $x(v)$ and $y(v)$, while $x_u(v)$ and $y_u(v)$ will refer to the coordinates calculated from $\mathcal{U}$. For each interior face of the magenta map of $\mathcal{T}(G'_1)$, $\mathcal{W}$ always assigns 1 as its weight. We have the following two subcases for re-inserting the missing green edges corresponding to Case 1 of the construction:

Case 1a: g is a leaf node of $\overline{T}_1$ and the k-th branch of $\overline{T}_1$ is degenerated. $\mathcal{W}$ assigns 1 as the weight for $\mathcal{F}_k$. In this case, no green edge is missing when adding the k-th branch, hence nothing has to be done.

Case 1b: g is a leaf node of $\overline{T}_1$ and the k-th branch is not degenerated. $\mathcal{W}$ still assigns 1 as the weight for $\mathcal{F}_k$. In the drawing of G'_1, according to Observations 1 and 3, we have $x(n) \leq x(g)$, and $y(g) \geq \dots \geq y(l) = y(n) + 1 = y(s) + 1$. For any vertex v in the branch $\{g, \dots, l\}$, if $y(v) = y(l)$, we connect the vertex s to it directly. Otherwise, $y(v) > y(l)$, we then connect s to v through an intermediate grid point,

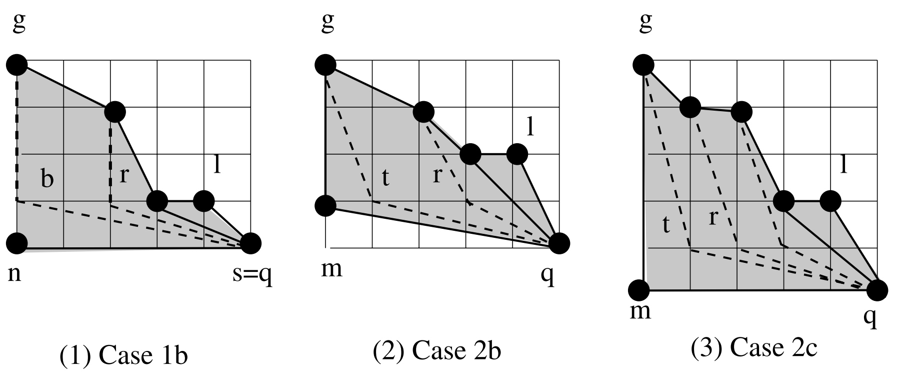


Fig. 6 Recover missing green edges

whose coordinate is $(x(v), y(n)+1)$. See Fig. 6(1) for an illustration for the scenario $x(n) = x(g)$. The other scenario for $x(n) < x(g)$ can be done similarly. Note that, in Fig. 6(1), g connects to s through the intermediate grid point b. The segment from g to b overlaps with the line between n and g. However, this will not create a problem since the edge (n, g) is not in G.

Corresponding to Case 2 of the construction, we have three subcases:

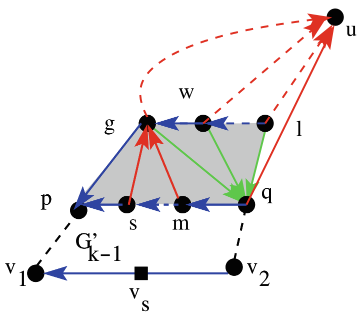
Case 2a: g is a non-leaf node of $\overline{T}_1$ and the k-th branch of $\overline{T}_1$ is degenerated. $\mathcal{W}$ assigns 1 as the weight for $\mathcal{F}_k$. In this case, no green edge is missing when adding the k-th branch, hence nothing has to be done.

Case 2b: g is a non-leaf node of $\overline{T}_1$ and the k-th branch of $\overline{T}_1$ is not degenerated, but p is not an ancestor of q in $\overline{T}_1$. The edge (s, p) is in $\overline{T}_1$, so there is a green edge within the path from s to q in cw direction along the contour of G_{k-1}. For this case, $\mathcal{W}$ assigns weight 1 to $\mathcal{F}_k$. According to Observation 3, we have $y(g) \geq \cdots \geq y(l) = y(s) + 1 > y(q) + 1$. So $y(l) - y(q) \geq 2$. In addition, we always have $y(l) > y(m) \geq y(q)$. For a vertex v in the branch $\{g, \ldots, l\}$, if $y(v) = y(l)$, we connect the vertex q to it directly. Otherwise, $y(v) > y(l)$, we then connect q to v through an intermediate grid point, whose coordinate is $(x(v)+1, y(m))$. See Fig. 6(2) for an illustration for the scenario $x(g) = x(m)$. Note that, when $m = s$ and (p, g) is split, we have the other scenario $x(m) < x(g)$, which can be recovered similarly.

Case 2c: g is a non-leaf node of $\overline{T}_1$, the k-th branch of $\overline{T}_1$ is not degenerated, and p is an ancestor of q in $\overline{T}_1$. Therefore, every edge in the path $\{q, m, \ldots, s, p\}$ in ccw direction on the contour of G_{k-1} is blue. For this case, $\mathcal{W}$ assigns weight 2 to $\mathcal{F}_k$. Therefore $y(m) = y(q)$, $y(l) - y(m) = 2$ and $x(m) \leq x(g)$. For a vertex v in the branch $\{g, \ldots, l\}$, if $y(v) = y(l)$, we connect the vertex q to it directly. Otherwise, $y(v) > y(l)$, we then connect q to v through an intermediate grid point, whose coordinate is $(x(v)+1, y(m)+1)$. See Fig. 6 (3) for an illustration for the scenario $x(m) = x(g)$. Note that it is possible that $x(m) < x(g)$ when $m = s$ and (p, g) is a split edge. The scenario of $x(m) < x(g)$ can be recovered similarly.

The edge (q, m) is a blue edge when Case 2c happens. Hence the face $\{q, m, g\}$ is a cw cyclic face in the minimum realizer, and it is the only cw cyclic face contained in $\mathcal{F}_k$. In addition, in Case 2c, the path from g to l in the k-branch of $\overline{T}_1$ is not degenerated. Hence, we can denote it by $\{g, w, \ldots, l\}$. See Fig. 7. We will call

Fig. 7 (Color online) Bounding the total number of expanders



the unique cw cyclic face $\{q,m,g\}$ an *expander* for G'_1. (Similarly, we can define expanders for G'_2 and G'_n.) For example, in the cyan map in Fig. 2(2), the face $\mathcal{F}_4$ enclosed by $\{v_1,b,c,k,h,n,a,v_1\}$ belongs to Case 2c. Its weight is assigned 2. The unique cw cyclic face $\{c,a,k\}$ contained in $\mathcal{F}_4$ is an expander for G'_1.

3.3 Analysis of the Grid Size

Let ϵ_i, $i=1,2,n$ be the number of expanders for G'_i, $i=1,2,n$ respectively. Next lemma provides an upper bound for $\epsilon_1+\epsilon_2+\epsilon_n$.

Lemma 4 *Let G be a plane triangulation with $n\geq 4$ vertices, v_1,v_2,v_n be its three exterior vertices in ccw order, $\mathcal{R}_0=\{T_1,T_2,T_n\}$ be its minimum realizer. Let G'_i be the irreducible triangulations as constructed above for $i=1,2,n$. Then $\epsilon_1+\epsilon_2+\epsilon_n\leq\delta_0$, where δ_0 is the number of cw cyclic faces in the minimum realizer $\mathcal{R}_0$.*

Proof We utilize the symmetrical roles of the three indexes $\{1,2,n\}$ tremendously in the proof. First, we recall that an expander can only exist in Case 2c of the recovery process. It implies: (1) All the edges in the path $\{q,m,\ldots,s,p\}$ are blue in the minimum realizer. (Note, it is possible that $m=s$.) (2) The k-th branch $\{g,w,\ldots,l\}$ is not degenerated. (Note, it is possible that $w=l$.) Denote the unique cw cyclic face (g,m,q) by f. Consider the unique green edge (g,q) in f. The resulting non-degenerated rectangle $\{g,w,q,m\}$, after deleting (g,q) from the boundary of the cyclic face f has two non-adjacent blue edges on its boundary. Symmetrically, if f were an expander for G'_2, then the resulting rectangle after deleting the unique red edge (m,g) from the boundary of f should have two non-adjacent green edges on its boundary. However the resulting rectangle after deleting the unique red edge (m,g) from the boundary of f has either two consecutive blue edges if $s\neq m$; or three blue edges if $s=m$. Hence, f cannot to be an expander for G'_2. See Fig. 7.

Observe that, in $\mathcal{R}_0$, the region enclosed by the blue path starting from g, the blue path starting from m, and the red edge (m,g), has no vertex in its interior. In contrast, the region enclosed by the red path starting from g, the red path starting from q, and the green edge (g,q), clearly contains the vertex w in its interior. Hence, f cannot be an expander for G'_n. See Fig. 7.

Hence, any cw cyclic face of $\mathcal{R}_0$ can only be an expander for at most one of G'_i, $i=1,2,n$. Therefore, $\epsilon_1+\epsilon_2+\epsilon_n\leq\delta_0$. □

For example, in Fig. 2(1), the cw cyclic face $\{a, c, k\}$ is an expander for G'_1 as shown in Fig. 2(2). According to our above lemma, $\{a, c, k\}$ cannot be an expander for G'_2 or G'_n. The total number of expanders for all three G'_i, $i = 1, 2, n$ is therefore 1.

Applying Lemma 2 to G'_1, using the legal weight function $\mathcal{W}$, combined with the above recovery process, we obtain a polyline drawing for G. It will be denoted by $\mathcal{D}_1$. It is easy to observe that for any edge of G, it uses at most one bend in $\mathcal{D}_1$. We have the following lemma regarding the total number of bends in $\mathcal{D}_i$.

Lemma 5 *Let $\mathcal{D}_i$, $i = 1, 2, n$ be the polyline drawing of G obtained as above from G'_i. Then the total number of bends is $\leq \gamma_{i-1}$. Here, $i - 1$ is understood as modular 3 operation. n is treated as if it were 3.*

Proof We only prove the case $i = 1$. The other two cases can be similarly proved.

To each leaf of $\overline{T}_n$ corresponds either a re-inserted green edge or a split edge (which is the possible bend of a blue edge). These are the only possible cases for a bend. Hence the total number of bends in $\mathcal{D}_1$ is $\leq \gamma_n$. □

Now we have all the ingredients to state our main theorem:

Theorem 1 *Let G be a plane graph with n vertices. Then G admits a polyline drawing in a grid with size $W \times H$, where $W \leq \lfloor \frac{2n-2}{3} \rfloor$, and $(W + H) \leq \lfloor \frac{4n-4}{3} \rfloor$. The total number of bends is $\leq \lfloor \frac{2n-5}{3} \rfloor$, and each edge uses at most one bend. The drawing can be obtained in linear time.*

Proof Without loss of generality, let's assume G is a plane triangulation and $n \geq 4$. Let v_1, v_2, v_n be its exterior vertices in ccw order. Let $\mathcal{R}_0 = \{T_1, T_2, T_n\}$ be its minimum realizer. T_i is rooted at v_i. Let γ_i be the number of leaves of T_i, $i \in \{1, 2, n\}$. Let δ_0 be the number of the cw cyclic faces of $\mathcal{R}_0$. Using above construction, we have G'_i, $i = 1, 2, n$ and three drawings $\mathcal{D}_i$, $i = 1, 2, n$. Let ϵ_i, $i = 1, 2, n$ be the number of expanders for G'_i, $i = 1, 2, n$ respectively.

According to Lemma 4, we have $\epsilon_1 + \epsilon_2 + \epsilon_n \leq \delta_0$. We also have $\gamma_1 + \gamma_2 + \gamma_n = 2n - 5 - \delta_0$ from [3, 4, 6].

According to Lemma 3 and 5, we have the following:

1. The drawing size of $\mathcal{D}_1$ is $(\gamma_n + 1) \times (\gamma_1 + 1 + \epsilon_1)$, which is bounded by $(\gamma_n + 1 + \epsilon_n) \times (\gamma_1 + 1 + \epsilon_1)$. The number of bends is $\leq \gamma_n \leq (\gamma_n + 1 + \epsilon_n) - 1$.
2. The drawing size of $\mathcal{D}_2$ is $(\gamma_1 + 1) \times (\gamma_2 + 1 + \epsilon_2)$, which is bounded by $(\gamma_1 + 1 + \epsilon_1) \times (\gamma_2 + 1 + \epsilon_2)$. The number of bends is $\leq \gamma_1 \leq (\gamma_1 + 1 + \epsilon_1) - 1$.
3. The drawing size of $\mathcal{D}_n$ is $(\gamma_2 + 1) \times (\gamma_n + 1 + \epsilon_n)$, which is bounded by $(\gamma_2 + 1 + \epsilon_2) \times (\gamma_n + 1 + \epsilon_n)$. The number of bends is $\leq \gamma_2 \leq (\gamma_2 + 1 + \epsilon_2) - 1$.

We have $(\gamma_n + 1 + \epsilon_n) + (\gamma_1 + 1 + \epsilon_1) + (\gamma_2 + 1 + \epsilon_2) \leq 2n - 5 - \delta_0 + 3 + \delta_0 = 2n - 2$. Therefore, one of $(\gamma_n + 1 + \epsilon_n)$, $(\gamma_1 + 1 + \epsilon_1)$, and $(\gamma_2 + 1 + \epsilon_2)$ is $\leq \frac{2n-2}{3}$. Without loss of generality, let's assume that $(\gamma_n + 1 + \epsilon_n) \leq \frac{2n-2}{3}$. If $(\gamma_n + 1 + \epsilon_n) + (\gamma_1 + 1 + \epsilon_1) \leq \frac{4n-4}{3}$, then $\mathcal{D}_1$ satisfies the requirement. On the other hand, if $(\gamma_n + 1 + \epsilon_n) + (\gamma_1 + 1 + \epsilon_1) > \frac{4n-4}{3}$, then $(\gamma_2 + 1 + \epsilon_2) < \frac{2n-2}{3}$. Therefore $\mathcal{D}_n$ satisfies the requirement. Note that all the numbers W, $W + H$, and the total number of bends are integers, so we can use floor function to state the results in the end.

The linear running time of our polyline drawing algorithm comes from the following simple facts:

- The minimum realizer $\mathcal{R}_0$ can be constructed in linear time [3, 4, 6].
- The transformation from $\mathcal{R}_0$ to the transversal structures can be done in linear time by using our transformation.
- The legal weight function $\mathcal{W}$ used to draw the transversal structures can be decided in linear time.
- According to Lemma 2, the drawing algorithm applied to the transversal structures runs in linear time.
- It is obvious that the recovery process for the missing edges runs in linear time. $\square$

Observe that, our drawing algorithm is area optimal, and it uses fewer bends in contrast with the results in [2]. Figure 3(2) shows a polyline drawing of the original graph G in Fig. 2(1), where the edges represented as two-segment polylines are drawn in dashed lines.

Acknowledgement The author is grateful to the anonymous referees for their insightful comments, which help the author greatly on simplifying the proofs as well as on improving the general presentation of the paper.

References

1. di Battista, G., Eades, P., Tammassia, R., Tollis, I.: Graph Drawing: Algorithms for the Visualization of Graphs. Princeton Hall, Princeton (1998)
2. Bonichon, N., Le Saëc, B., Mosbah, M.: Optimal area algorithm for planar polyline drawings. In: Proceedings of the 28th International Workshop on Graph-Theoretic Concepts in Computer Science. Lecture Notes in Computer Science, vol. 2573, pp. 35–46. Springer, Berlin (2002)
3. Bonichon, N., Gavoille, C., Hanusse, N.: An information-theoretic upper bound of planar graphs using triangulation. In: Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science. Lecture Notes in Computer Science, vol. 2607, pp. 499–510. Springer, Berlin (2003)
4. Bonichon, N., Gavoille, C., Hanusse, N.: An information-theoretic upper bound of planar graphs using triangulation. Research Report RR-1279-02, LaBRI, University of Bordeaux, France
5. Bonichon, N., Gavoille, C., Hanusse, N.: Canonical decomposition of outerplanar maps and application to enumeration, coding and generation. J. Graph Algorithms Appl. **9**, 185–204 (2005)
6. Brehm, E.: 3-orientations and Schnyder 3-tree-decompositions: construction and order structure. Diploma thesis, FB Mathematik und Informatik, Freie Universität Berlin (2000)
7. Fusy, É.: Transversal structures on triangulations, with application to straight-line drawing. In: Proceedings of 13th International Symposium on Graph Drawing. Lecture Notes in Computer Science, vol. 3843, pp. 177–188. Springer, Berlin (2005)
8. He, X.: On finding the rectangular duals of planar triangular graphs. SIAM J. Comput. **22**, 1218–1226 (1993)
9. He, X.: Grid embedding of 4-connected plane graphs. Discrete Comput. Geom. **17**, 339–358 (1997)
10. He, X.: On floor-plan of plane graphs. SIAM J. Comput. **28**, 2150–2167 (1999)
11. Rahman, M.S., Nakano, S.-I., Nishizeki, T.: Rectangular grid drawings of plane graphs. Comput. Geom. Theory Appl. **10**, 203–220 (1998)
12. Schnyder, W.: Planar graphs and poset dimension. Order **5**, 323–343 (1989)

13. Schnyder, W.: Embedding planar graphs on the grid. In: Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 138–148 (1990)
14. Zhang, H., He, X.: An application of well-orderly trees in graph drawing. Int. J. Found. Comput. Sci. **17**, 1129–1142 (2006)
15. Zhang, H., Sadasivam, S.: On planar polyline drawings. In: Proceedings of the 15th International Symposium on Graph Drawing. Lecture Notes in Computer Science, vol. 4875, pp. 213–218. Springer, Berlin (2007)