On the Queue Number of Planar Graphs

Giuseppe Di Battista, Fabrizio Frati
Dipartimento di Informatica e Automazione
Roma Tre University
Rome, Italy
{*gdb,frati*}*@dia.uniroma3.it*

János Pach
Chair of Combinatorial Geometry
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
janos.pach@epfl.ch

***Abstract*—We prove that planar graphs have polylogarithmic queue number, thus improving upon the previous polynomial upper bound. Consequently, planar graphs admit 3D straight-line crossing-free grid drawings in small volume.**

***Keywords*-planar graph; queue layout; straight-line drawing; volume;**

I. Introduction and Overview

A *linear layout* of a graph is a total ordering of its vertices and a partition of its edges such that all the elements of the partition enforce some specific property.

Linear layouts play an important role in Graph Theory and their study goes back to 1973, when Ollmann [28] introduced the concept of *book embedding* (later also called *stack layout*) and *book thickness* (later also called *stack number*, *fixed outer-thickness*, and, most commonly, *page number*) of a graph. A book embedding on k pages of a graph $G(V, E)$ is a linear layout of G in which the partition of E consists of k sets $E_1, E_2, \dots, E_k$, called *pages*, such that no two edges in the same page *cross* (edges (u, v) and (w, z) cross if $u < w < v < z$ or $w < u < z < v$), and the page number is the minimum k such that G has a book embedding on k pages. The literature is rich of combinatorial and algorithmic contributions on the page number of various classes of graphs (see, e.g., [4], [13], [14], [16], [17], [18], [19], [20], [26], [27]). A famous result of Yannakakis [34] states that a planar graph has page number at most four.

Queue layout and *queue number* are the dual concepts of book embedding and page number, respectively. A queue layout on k queues of a graph $G(V, E)$ is a linear layout of G in which the partition of E consists of k sets $E_1, E_2, \dots, E_k$, called *queues*, such that no two edges in the same queue *nest* (edges (u, v) and (w, z) nest if $u < w < z < v$ or $w < u < v < z$), and the queue number is the minimum k such that G has a queue layout on k queues. Queue layouts were introduced by Heath, Leighton, and Rosenberg [21], [25], motivated by applications, e.g., in parallel process scheduling [1], matrix-computations [29], and sorting permutations and networks [30], [33].

Computing the queue number of a graph is $\mathcal{NP}$-complete. Namely, it is known that deciding if a graph has queue number 1 is $\mathcal{NP}$-complete [25]. However, from a combinatorial point of view, a large number of bounds are known on the queue number of several graph classes. For example, graphs with m edges have queue number at most $e\sqrt{m}$ [10], graphs with tree-width w have queue-number at most $3^w \cdot 6^{(4^w-3w-1)/9} - 1$ [9], graphs with tree-width w and degree Δ have queue-number at most $36\Delta w$ [9], graphs with path-width p have queue-number at most p [9], graphs with band-width b have queue-number at most $\lceil b/2 \rceil$ [25], and graphs with track number t have queue-number at most $t - 1$ [9]. Queue layouts of directed graphs [23], [24] and posets [22] have also been studied.

As in many graph problems, a special attention has been devoted to planar graphs and their subclasses. For example, trees have queue number 1 [25], outerplanar graphs have queue number 2 [21], and series-parallel graphs have queue number 3 [31]. However, for general planar graphs the best known upper bound for the queue number is $O(\sqrt{n})$ (a consequence of the results on graphs with $O(n)$ edges [10], [25], [32]), while no super-constant lower bound is known. Heath et al. [21], [25] conjectured that planar graphs have $O(1)$ queue number. Contrastingly, Pemmaraju [29] conjectured that a certain class of planar graphs, namely *planar 3-trees*, have $\Omega(\log n)$ queue number. However, Dujmović et al. [9] disproved such a conjecture by proving that graphs of constant tree-width, and hence also planar 3-trees, have constant queue number. Observe that the problem of determining the queue number of planar graphs is cited into several papers and collections of open problems (see, e.g., [3], [6], [8], [9], [12], [21], [25]).

In this paper, we prove that the queue number of planar graphs is $O(\log^4 n)$. The proof is constructive and is based on a polynomial-time algorithm that computes a queue layout with such a queue number. The result is based on several new combinatorial and algorithmic tools.

First (Sect. II), we introduce *level-2-connected graphs*, that are plane graphs in which each outerplanar level induces a set of disjoint 2-connected graphs. We show that every planar graph has a subdivision with one vertex per edge that is a level-2-connected graph. Such a result, together with a result of Dujmović and Wood [12] stating that the queue number of a graph G is at most the square of the queue number of a subdivision of G with one vertex per edge, allows us to study the queue number of level-2-connected graphs in order to determine bounds on the queue number of general planar graphs.

Second (Sect. III), we introduce *floored graphs*, that are plane graphs whose vertices are partitioned into sets $V_1, V_2, \dots, V_k$ such that some strong topological properties on the subgraph induced by each V_i and on the connectivity among such subgraphs are satisfied. We prove that every level-2-connected graph admits a partition of its vertex set resulting into a floored graph. Floored graphs are then related to the outerplanar levels of a level-2-connected graph. Such levels form a tree hierarchy that can be thought as having one node for each connected component of an outerplanar level and an edge (u, v) if the graph corresponding to v lies inside the graph corresponding to u. Floored graphs are used to explore such hierarchy one path at a time. Moreover, we prove the existence in any floored graph G of a simple subgraph (a path plus few edges) that decomposes G into two smaller floored graphs G' and G''.

Third (Sect. IV), we show an algorithm that constructs a queue layout of a floored graph G in which the different sets of the partition are in consecutive sub-sequences of the total vertex ordering of G. The algorithm is recursive and at each step uses the mentioned decomposition of a floored graph G into two floored graphs G' and G'' several times, each time splitting the floored graph with the greatest number of vertices between the two floored graphs obtained at the previous splitting, until no obtained floored graph has more than half of the vertices of the initial floored graph. The resulting floored graphs have different vertex partitions. However, it is shown how to merge such partitions in such a way that $O(\log^2 n)$ queues are sufficient to accommodate all the edges of the initial n-vertex graph.

Then, we conclude that floored graphs have $O(\log^2 n)$ queue number, hence level-2-connected graphs have $O(\log^2 n)$ queue number, thus planar graphs have $O(\log^4 n)$ queue number.

Our result sheds new light on one of the most studied Graph Drawing problems (see, e.g., [3], [5], [6], [9], [11], [15]): Given an n-vertex planar graph which is the volume required to draw it in 3D, representing edges with straight-line segments that cross only at common endpoints? The previously best known upper bound [11] was $O(n^{1.5})$ volume. We prove that planar graphs have 3D straight-line crossing-free drawings in $O(n \log^c n)$ volume, for some constant c. Such a result comes from our new bound on the queue number of planar graphs and from results by Dujmović, Morin, and Wood [9], [11] relating the queue number of a graph to its track number and to the volume requirements of its 3D straight-line crossing-free drawings.

Because of space limitations, several proofs are omitted and can be found in the full version of the paper [7].

II. Preliminaries

A *planar drawing* of a graph is a mapping of each vertex to a distinct point of the plane and of each edge to a Jordan curve between its endpoints such that no two edges intersect except, possibly, at common endpoints. A planar drawing of a graph determines a circular ordering of the edges incident to each vertex. Two drawings of the same graph are *equivalent* if they determine the same circular ordering around each vertex. A *planar embedding* is an equivalence class of planar drawings. A planar drawing partitions the plane into topologically connected regions, called *faces*. The unbounded face is the *outer face*. A graph together with a planar embedding and a choice for its outer face is called *plane graph*. A plane graph is *maximal* when all its faces are triangles. A plane graph is *internally-triangulated* when all its internal faces are triangles. An *outerplane* graph is a plane graph such that all its vertices are on the outer face.

A graph $G'(V', E')$ is a *subgraph* of a graph $G(V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. A subgraph is *induced* by V' if, for every edge $(u, v) \in E$ such that $u, v \in V'$, $(u, v) \in E'$. The subgraph induced by $V' \subseteq V$ is denoted by $G[V']$.

A graph is *connected* if every pair of vertices is connected by a path. A *k-connected* graph G is such that removing any $k - 1$ vertices leaves G connected. A vertex whose removal disconnects the graph is a *cut-vertex*.

A *k-subdivision* of a graph G is a graph obtained by replacing each edge of G with a path having at most $2 + k$ vertices.

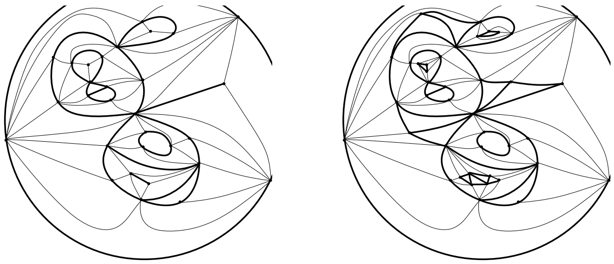
A *chord of a cycle C* is an edge connecting two non-consecutive vertices of C. A *chord of a plane graph G* is a chord of the cycle delimiting the outer face of G.

Given a plane graph G and an edge (u, v) on the outer face of G, we say that G *is to the left (resp. to the right) of* (u, v) *when traversing it from u to v* if an internal face of G is to the left (resp. to the right) of (u, v) when traversing such an edge from u to v.

Given two plane graphs G_1 and G_2 embedded in the plane and possibly sharing some vertices of their outer faces, we say that G_2 *is in the outer face of* G_1 if every vertex of G_2 not in G_1 is to the left of the cycle delimiting the outer face of G_1 when clockwise traversing such a cycle.

The *outerplanar levels* (or simply *levels*) of a plane graph G are defined as follows. Let $G_1 = G$ and let G_{i+1} be the plane graph obtained by removing from G_i ($i \geq 1$) the set V_i of vertices of the outer face of G_i and their incident edges. Set V_i is the *i-th level* of G. Observe that the first level of G is the set of vertices of its outer face. Let k be the maximum index such that $V_k \neq \emptyset$. We say that G has k levels.

A 2-connected internally-triangulated plane graph G is *level-2-connected* if G_i is composed of a set of 2-connected graphs that are pairwise vertex-disjoint and that have each at least three vertices, for each $1 \leq i \leq k$. That is, G_i has no cut-vertex and it has no connected component that is a single



(a) (b)

Figure 1. Two maximal plane graphs G and G^*. The subgraphs induced by the levels of G and G^* are shown by thick lines. (a) G is not level-2-connected. (b) G^* is level-2-connected and contains G as a 1-subdivision.

vertex or a single edge. Fig. 1.a shows a maximal plane graph G that is not level-2-connected and Fig. 1.b shows a maximal plane graph G^* that is level-2-connected and that contains G as a 1-subdivision. We have the following:

Lemma 1: Let G be an n-vertex plane graph. There exists a maximal plane graph G^* such that:

(i) G^* is level-2-connected,
(ii) G^* contains a subgraph G' such that G' is a 1-subdivision of G, and
(iii) G^* has $O(n)$ vertices.

We further observe two properties of a level-2-connected graph G. Let C be a cycle delimiting the outer face of a 2-connected component of the i-th level of G and let G_C be the subgraph of G inside or on the border of C. Let (u, v) be a chord of G and let V_1 and V_2 be the vertex sets of the two connected components of G which are obtained by removing u, v, and their incident edges.

Lemma 2: G_C is a level-2-connected graph.

Lemma 3: $G[V_1 \cup \{u, v\}]$ and $G[V_2 \cup \{u, v\}]$ are level-2-connected graphs.

III. Floored Graphs

In this section we define floored graphs and show how to decompose a floored graph into two smaller floored graphs. A *floored graph* (see Fig. 2) is a graph G with one distinguished vertex or edge on the outer face and whose vertex set is partitioned into sets $F_1, F_2, \dots, F_k$ that induce subgraphs of G, called *floors*, satisfying the topological properties described below. More formally, a floored graph is a triple (G, f, g), where $G(V, E)$ is a 2-connected internally-triangulated plane graph, f is a function $f : V \rightarrow \mathbb{N}$, where $F_i = f^{-1}(i)$, for $i = 1, \dots, k$ (denote by k the largest integer such that $f^{-1}(k) \neq \emptyset$), and g is an edge in E or a vertex in V, such that the following conditions are satisfied:

C1: Graph $G[F_1]$ is *either* a vertex on the outer face of G (then g is such a vertex and let $u_1^1(G) = v_1^1(G) = g$), *or* an edge on the outer face of G (then g is such an edge and let $g = (u_1^1(G), v_1^1(G))$, where G is to the left of g when traversing it from $u_1^1(G)$ to $v_1^1(G)$), *or* a level-2-connected graph (then $g = (u_1^1(G), v_1^1(G))$ is an edge of $G[F_1]$ on the outer face of G, where G is to the left of g when traversing it from $u_1^1(G)$ to $v_1^1(G)$). $G[F_1]$ is the *first floor* of (G, f, g).

C2: For each $2 \leq i \leq k-1$, graph $G[F_i]$ is composed of a sequence $G_1^i, G_2^i, \dots, G_{x(i)}^i$ of graphs which are either single edges or level-2-connected graphs, with $x(i) \geq 1$, such that: (i) G_1^i has a vertex $u_1^i(G)$ on the outer face of G; (ii) $G_{x(i)}^i$ has a vertex $v_{x(i)}^i(G)$ on the outer face of G; (iii) G_j^i has a vertex $v_j^i(G)$ coincident with a vertex $u_{j+1}^i(G)$ of G_{j+1}^i, for $1 \leq j \leq x(i)-1$; such a vertex is on the outer faces of both graphs; (iv) G_j^i and G_{j+1}^i lie each one in the outer face of the other one, for $1 \leq j \leq x(i)-1$; (v) G_j^i and G_l^i do not share any vertex, for $l \neq j-1, j+1$; (vi) edge $(u_j^i(G), v_j^i(G))$ exists and is on the outer face of G_j^i, for $1 \leq j \leq x(i)$; (vii) G_j^i (if such a graph is not just edge $(u_j^i(G), v_j^i(G))$) is to the left of $(u_j^i(G), v_j^i(G))$ when traversing it from $u_j^i(G)$ to $v_j^i(G)$. $G[F_i]$ is the *i-th floor* of (G, f, g).

C3: Graph $G[F_k]$ is either a single vertex $u_1^k(G) = v_{x(k)}^k(G)$ on the outer face of G, or a sequence $G_1^k, G_2^k, \dots, G_{x(k)}^k$ of graphs which are either single edges or level-2-connected graphs, with $x(k) \geq 1$, such that properties (i)–(vii) of Condition C2 hold (in such a case $u_1^k(G)$ and $v_{x(k)}^k(G)$ are defined as in such properties). $G[V_k]$ is the *last floor* of (G, f, g).

C4: G contains no edge connecting the i_1-th floor and the i_2-th floor of (G, f, g), with $i_2 \neq i_1 - 1, i_1 + 1$.

C5: Any floor is in the outer face of each other floor.

C6: Paths $B_1 = (u_1^1(G), u_1^2(G), \dots, u_1^k(G))$ and $B_2 = (v_1^1(G), v_{x(2)}^2(G), \dots, v_{x(k)}^k(G))$ exist and are on the outer face of G. Such paths are called the *borders* of (G, f, g). If (G, f, g) has one floor, then B_1 and B_2 are single vertices.

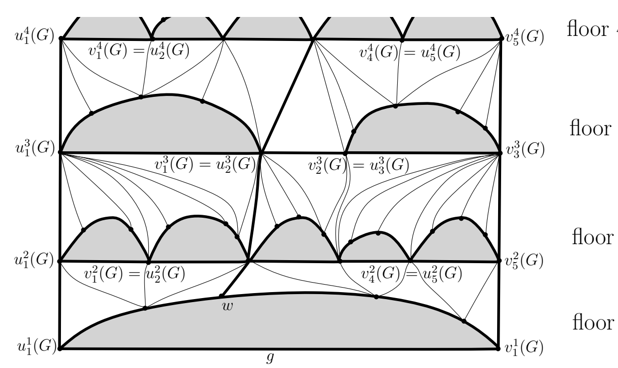


Figure 2. A floored graph (G, f, g) with 4 floors. Level-2-connected graphs are gray. Their outer faces are shown by thick lines. The borders of (G, f, g) are shown by thick lines. In this example $G[F_1]$ is a level-2-connected graph. Hence, $g = (u_1^1(G), v_1^1(G))$. A raising path starting at w is shown by thick lines.

Level-2-connected graphs can be easily turned into floored graphs, as shown in the following.

Lemma 4: Let $G(V, E)$ be a level-2-connected graph. Let f be the function $f : V \to \mathbb{N}$ such that $f(z) = 1$, for every $z \in V$. Let $g = (u_1^1, v_1^1)$ be any edge on the outer face of G, where G is to the left of g when traversing such an edge from u_1^1 to v_1^1. Then, (G, f, g) is a floored graph.

We have the following structural lemma (see Fig. 3).

Lemma 5: Let (G, f, g) be a floored graph. Then, exactly one of the following assertions is true. (1) $G[F_1]$ is vertex g and $(g, u_1^2(G), v_{x(2)}^2(G))$ is an internal face of G. (2) $G[F_1]$ is vertex g and vertices g, $u_1^2(G)$, and $v_{x(2)}^2(G)$ are not on the same internal face of G. (3) $G[F_1]$ is an edge $g = (u_1^1(G), v_1^1(G))$. (4) $G[F_1]$ is a level-2-connected graph and the vertex w of G that forms an internal face with g is on the outer face of $G[F_1]$. (5) $G[F_1]$ is a level-2-connected graph and the vertex w_1 of G that forms an internal face with g is not on the outer face of $G[F_1]$.

We now define and study raising paths, that are paths that will be used in order to split floored graphs into smaller floored graphs. Let (G, f, g) be a floored graph and let $w \neq g$ be a vertex on the outer face of the i-th floor of G, for any $1 \leq i \leq k$. A *raising path starting at* w (see Fig. 2) is a path $R(w) = (w_1 = w, w_2, \dots, w_y)$ such that $f(w_x) = f(w_{x-1}) + 1$, for every $2 \leq x \leq y$, and such that, if a vertex w_x belongs to the border B_1 (resp. to B_2), then all the vertices after w_x in $R(w)$ belong to B_1 (resp. to B_2). We have the following.

Lemma 6: Let (G, f, g) be a floored graph. For every vertex w of the outer face of a floor of G different from the last floor, there exists a vertex z on the outer face of $G[F_{f(w)+1}]$ and adjacent to w.

Corollary 1: Let (G, f, g) be a floored graph. For every vertex $w \neq g$ on the outer face of a floor of G, there exists a raising path starting at w.

Suppose that a raising path $R(w)$ shares vertices with B_1. Then, path $R(w) \setminus B_1$ is the subpath of $R(w)$ starting at w and ending at the first vertex z shared by $R(w)$ and B_1 (z is in $R(w) \setminus B_1$). Further, $B_1 \setminus R(w)$ is the subpath of B_1 starting at $u_1^1(G)$ and ending at the first vertex z shared by $R(w)$ and B_1 (z is in $B_1 \setminus R(w)$). If $R(w)$ shares vertices with B_2, then $R(w) \setminus B_2$ and $B_2 \setminus R(w)$ are defined analogously.

Let (G, f, g) be a floored graph such that G has more than three vertices. Denote by P the subpath of the outer face of G between $u_1^k(G)$ and $v_{x(k)}^k(G)$ and not containing g. Given a vertex w_y in P, let $P_1(w_y)$ (resp. $P_2(w_y)$) be the subpath of P between $u_1^k(G)$ and w_y (resp. between w_y and $v_{x(k)}^k(G)$).

We now discuss how to use a raising path to split a floored graph (G, f, g) into two floored graphs. We distinguish five cases, according to the five mutually-exclusive assertions of Lemma 5.

Case 1. *$G[F_1]$ is vertex g and $(g, u_1^2(G), v_{x(2)}^2(G))$ is an internal face of G.*

See Fig. 3.a. Actually, this case does not use a raising path, but changes G by removing one of its vertices still obtaining a floored graph in which g is now an edge. Let (G', f', g') be the triple defined as follows. $G'(V', E')$ is the graph obtained from G by removing vertex g and its incident edges $(g, u_1^2(G))$ and $(g, v_{x(2)}^2(G))$, $f'(w) = f(w) - 1$, for each vertex $w \in V'$, and $g' = (u_1^2(G), v_{x(2)}^2(G))$.

Case 2. *$G[F_1]$ is vertex g and vertices g, $u_1^2(G)$, and $v_{x(2)}^2(G)$ are not on the same internal face of G.*

See Fig. 3.b. Consider any edge (g, w) internal to G. Observe that such an edge exists, as G is internally-triangulated. Consider any raising path $R(w)$ starting at w.

- If $R(w)$ does not share vertices with B_1 and B_2, then let w_y be the last vertex of $R(w)$. Let $G'(V', E')$ be the subgraph of G inside or on the border of cycle $B_1 \cup P_1(w_y) \cup R(w) \cup (g, w)$ and let $G''(V'', E'')$ be the subgraph of G inside or on the border of cycle $B_2 \cup P_2(w_y) \cup R(w) \cup (g, w)$. Let $f'(z) = f(z)$, for each vertex $z \in V'$, and let $f''(z) = f(z)$, for each vertex $z \in V''$. Finally, let $g' = g$ and $g'' = g$.
- If $R(w)$ shares vertices with B_1 (the case in which it shares vertices with B_2 being analogous), let $G'(V', E')$ be the subgraph of G inside or on the border of cycle $(B_1 \setminus R(w)) \cup (R(w) \setminus B_1) \cup (g, w)$ and let $G''(V'', E'')$ be the subgraph of G inside or on the border of cycle $P \cup R(w) \cup (g, w) \cup B_2$. Let $f'(z) = f(z)$, for each vertex $z \in V'$, and let $f''(z) = f(z)$, for each vertex $z \in V''$. Finally, let $g' = g$ and $g'' = g$.

Case 3. *$G[F_1]$ is an edge $g = (u_1^1(G), v_1^1(G))$.*

See Fig. 3.c. Consider the vertex w of G that forms an internal face with g. Notice that such a vertex exists as G is internally-triangulated and belongs to the second floor of G (by the fact that $G[F_1]$ is an edge and by Condition C4).

- If w belongs to B_2, that is $w = v_{x(2)}^2(G)$, then let $G'(V', E')$ be the subgraph of G inside or on the border of cycle $B_1 \cup P \cup (B_2 \setminus \{(v_1^1(G), w)\}) \cup (u_1^1(G), w)$, let $f'(z) = f(z)$, for each vertex $z \in V'$, and let $g' = u_1^1(G)$.
- If w belongs to B_1, that is $w = u_1^2(G)$, then let $G''(V'', E'')$ be the subgraph of G inside or on the border of cycle $B_2 \cup P \cup (B_1 \setminus \{(u_1^1(G), w)\}) \cup (v_1^1(G), w)$, let $f''(z) = f(z)$, for each vertex $z \in V''$, and let $g'' = v_1^1(G)$.
- If w belongs neither to B_1 nor to B_2, then consider any raising path $R(w)$ starting at w.
  - If $R(w)$ does not share vertices with B_1 and B_2, then let w_y be the last vertex of $R(w)$. Let $G'(V', E')$ be the subgraph of G inside or on the border of cycle $B_1 \cup P_1(w_y) \cup R(w) \cup (u_1^1(G), w)$ and let $G''(V'', E'')$ be the subgraph of G inside

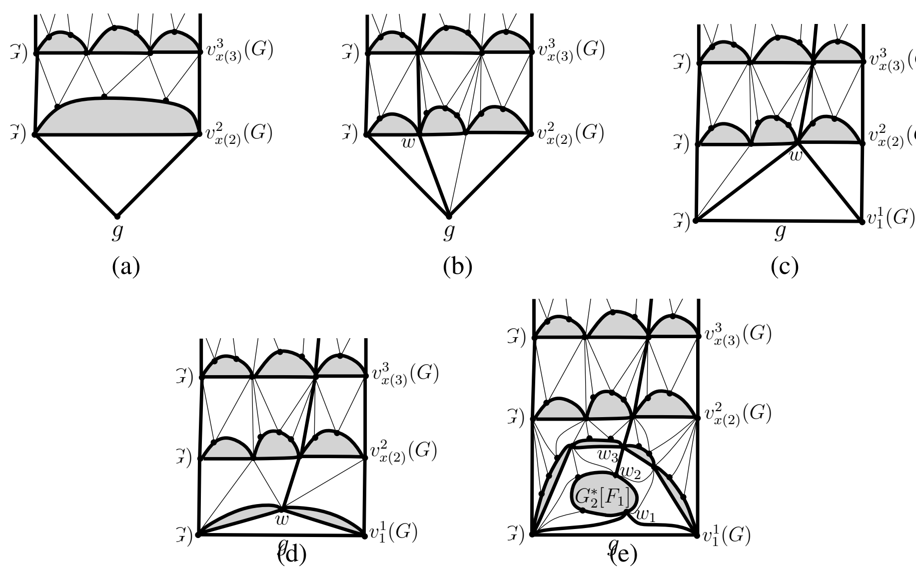


Figure 3. Illustration for Cases 1–5. The borders of (G, f, g), of (G', f', g'), and of (G'', f'', g'') and the cycles delimiting the outer faces of the floors of (G, f, g), of (G', f', g'), and of (G'', f'', g'') are shown by thick lines. (a) Case 1. (b) Case 2. (c) Case 3. (d) Case 4. (e) Case 5.

or on the border of cycle $B_2 \cup P_2(w_y) \cup R(w) \cup (v^1_1(G), w)$. Let $f'(z) = f(z)$, for each vertex $z \in V'$, and let $f''(z) = f(z)$, for each vertex $z \in V''$. Finally, let $g' = u^1_1(G)$ and $g'' = v^1_1(G)$.

– If $R(w)$ shares vertices with B_1 (the case in which it shares vertices with B_2 being analogous), let $G'(V', E')$ be the subgraph of G inside or on the border of cycle $(B_1 \setminus R(w)) \cup (R(w) \setminus B_1) \cup (u^1_1(G), w)$ and let $G''(V'', E'')$ be the subgraph of G inside or on the border of cycle $P \cup R(w) \cup (v^1_1(G), w) \cup B_2$. Let $f'(z) = f(z)$, for each vertex $z \in V'$, and let $f''(z) = f(z)$, for each vertex $z \in V''$. Finally, let $g' = u^1_1(G)$ and $g'' = v^1_1(G)$.

Case 4. *$G[F_1]$ is a level-2-connected graph and the vertex w of G that forms an internal face with g is on the outer face of $G[F_1]$.*

See Fig. 3.d. Consider any raising path $R(w)$ starting at w.

- If $R(w)$ does not share vertices with B_1 and B_2, then let w_y be the last vertex of $R(w)$. Let $G'(V', E')$ be the subgraph of G inside or on the border of cycle $B_1 \cup P_1(w_y) \cup R(w) \cup (u^1_1(G), w)$ and let $G''(V'', E'')$ be the subgraph of G inside or on the border of cycle $B_2 \cup P_2(w_y) \cup R(w) \cup (v^1_1(G), w)$. Let $f'(z) = f(z)$, for each vertex $z \in V'$, and let $f''(z) = f(z)$, for each vertex $z \in V''$. Finally, let $g' = (u^1_1(G), w)$ and $g'' = (v^1_1(G), w)$.
- If $R(w)$ shares vertices with B_1 (the case in which it shares vertices with B_2 being analogous), let $G'(V', E')$ be the subgraph of G inside or on the border of cycle $(B_1 \setminus R(w)) \cup (R(w) \setminus B_1) \cup (u^1_1(G), w)$ and let $G''(V'', E'')$ be the subgraph of G inside or on the border of cycle $P \cup R(w) \cup (v^1_1(G), w) \cup B_2$. Let $f'(z) = f(z)$, for each vertex $z \in V'$, and let $f''(z) = f(z)$, for each vertex $z \in V''$. Finally, let $g' = (u^1_1(G), w)$ and $g'' = (v^1_1(G), w)$.

Case 5. *$G[F_1]$ is a level-2-connected graph and the vertex w_1 of G that forms an internal face with g is not on the outer face of $G[F_1]$.*

See Fig. 3.e. By planarity and since (G, f, g) satisfies Condition C5, w_1 is in $G[F_1]$. Since w_1 is not on the outer face of $G[F_1]$, it is an internal vertex of $G[F_1]$. By planarity, w_1 is in the second level of $G[F_1]$. Since $G[F_1]$ is level-2-connected, the subgraph of $G[F_1]$ induced by the vertices in its second level consists of a set of vertex-disjoint 2-connected graphs. Let $G^*_2[F_1]$ be the one of such graphs w_1 belongs to. Since $G^*_2[F_1]$ is 2-connected, its outer face is delimited by a cycle C. Orient C so that $G^*_2[F_1]$ is to the right of every edge of C when traversing such an edge according to its orientation; let w_2 be the vertex preceding w_1 in C. Since G is internally-triangulated and since w_2 is on the outer face of $G^*_2[F_1]$, w_2 has at least one incident edge e whose end-vertex $w_3 \neq w_2$ is not in $G^*_2[F_1]$. We prove that w_3 is on the outer face of $G[F_1]$. By planarity, w_3 is not in the i-th level of $G[F_1]$, for any $i \geq 3$. If w_3 is in the second level of $G[F_1]$, then it belongs to a 2-connected component, say $G^+_2[F_1]$, of the graph induced by the second level of $G[F_1]$. However, since no edge of $G[F_1]$ connects vertices of two distinct connected components induced by the second level of $G[F_1]$, it follows that $G^+_2[F_1] = G^*_2[F_1]$, hence w_3 belongs to $G^*_2[F_1]$, contradicting the assumptions on e. Then w_3 is in the first level of $G[F_1]$, that is, it is on the outer face of $G[F_1]$. Consider any raising path $R(w_3)$ starting at w_3.

- If $R(w_3)$ does not share vertices with B_1 and B_2, then let w_y be the last vertex of $R(w_3)$. Let $G'(V', E')$ be the subgraph of G inside or on the border of cycle $B_1 \cup P_1(w_y) \cup R(w_3) \cup (w_2, w_3) \cup (w_1, w_2) \cup (u_1^1(G), w_1)$ and let $G''(V'', E'')$ be the subgraph of G inside or on the border of cycle $B_2 \cup P_2(w_y) \cup R(w_3) \cup (w_2, w_3) \cup (w_1, w_2) \cup (v_1^1(G), w_1)$. Let $f'(z) = 1$, for each vertex z that is inside or on the border of C, let $f'(z) = 2$, for each vertex z that is in G', that is in $G[F_1]$, and that is neither inside nor on the border of C, and let $f'(z) = f(z) + 1$, for each vertex z that is in G' and that is in $G[F_i]$, for every $i \geq 2$. Let $f''(w_1) = 1$, let $f''(w_2) = 1$, let $f''(z) = 2$, for each vertex z that is in G'', that is in $G[F_1]$, and that is different from w_1 and w_2, and let $f''(z) = f(z) + 1$, for each vertex z that is G'' and that is in $G[F_i]$, for every $i \geq 2$. Finally, let $g' = (w_1, w_2)$ and $g'' = (w_1, w_2)$.
- If $R(w_3)$ shares vertices with B_1 (the case in which it shares vertices with B_2 being analogous), let $G'(V', E')$ be the subgraph of G inside or on the border of cycle $(B_1 \setminus R(w_3)) \cup (R(w_3) \setminus B_1) \cup (w_2, w_3) \cup (w_1, w_2) \cup (u_1^1(G), w_1)$ and let $G''(V'', E'')$ be the subgraph of G inside or on the border of cycle $B_2 \cup P \cup R(w_3) \cup (w_2, w_3) \cup (w_1, w_2) \cup (v_1^1(G), w_1)$. Let $f'(z) = 1$, for each vertex z that is inside or on the border of C, let $f'(z) = 2$, for each vertex z that is in G', that is in $G[F_1]$, and that is neither inside nor on the border of C, and let $f'(z) = f(z) + 1$, for each vertex z that is in G' and that is in $G[F_i]$, for every $i \geq 2$. Let $f''(w_1) = 1$, let $f''(w_2) = 1$, let $f''(z) = 2$, for each vertex z that is in G'', that is in $G[F_1]$, and that is different from w_1 and w_2, and let $f''(z) = f(z)+1$, for each vertex z that is G'' and that is in $G[F_i]$, for every $i \geq 2$. Finally, let $g' = (w_1, w_2)$ and $g'' = (w_1, w_2)$.

We have the following.

Lemma 7: In any of Cases 1–5, (G', f', g') and (G'', f'', g'') are floored graphs.

IV. Queue Layouts

In this section, we show an algorithm for constructing a queue layout of a floored graph (G, f, g). The algorithm splits (G, f, g) into smaller floored graphs using raising paths, recursively constructs queue layouts of such smaller floored graphs, and then combines such layouts to get a queue layout of (G, f, g).

The algorithm receives as an input a floored graph (G, f, g) such that G has n internal vertices and has k floors, and it performs a *balanced raising-path decomposition*, that is, it repeatedly uses raising paths, according to Cases 1–5 of Section III, to split (G, f, g) into several floored graphs, each with at most $n/2$ internal vertices. More precisely, a balanced raising-path decomposition works as follows. Graph $(G_0^*, f_0^*, g_0^*) = (G, f, g)$ is split into two floored graphs (G_1, f_1, g_1) and (G_1^*, f_1^*, g_1^*), where the number of internal vertices of G_1^* is not less than the number of internal vertices of G_1, then (G_1^*, f_1^*, g_1^*) is split into (G_2, f_2, g_2) and (G_2^*, f_2^*, g_2^*), where the number of internal vertices of G_2^* is not less than the number of internal vertices of G_2, etc., until floored graph $(G_{l-1}^*, f_{l-1}^*, g_{l-1}^*)$ is split into floored graphs (G_l, f_l, g_l) and $(G_l^*, f_l^*, g_l^*) = (G_{l+1}, f_{l+1}, g_{l+1})$ such that both G_l and G_{l+1} have at most $n/2$ internal vertices. The split of a graph (G_j^*, f_j^*, g_j^*) into two floored graphs $(G_{j+1}, f_{j+1}, g_{j+1})$ and $(G_{j+1}^*, f_{j+1}^*, g_{j+1}^*)$ is actually done by applying one of Cases 2–5. When Case 1 is applied to (G_j^*, f_j^*, g_j^*), then just one floored graph is obtained, with the same number of internal vertices of (G_j^*, f_j^*, g_j^*). In such a case, denote again by (G_j^*, f_j^*, g_j^*) the obtained graph and proceed. Denote by $k(G_j)$ the number of floors of graph (G_j, f_j, g_j), for $j = 1, 2, \ldots, l+1$, and by $k(G_j^*)$ the number of floors of graph (G_j^*, f_j^*, g_j^*), for $j = 0, 1, \ldots, l$.

Before giving more details on the algorithm for constructing a queue layout of (G, f, g), we state the following lemma relating the floors of graphs (G_j, f_j, g_j) and (G_j^*, f_j^*, g_j^*) constructed during the balanced raising-path decomposition to the floors and levels of graph (G, f, g).

Lemma 8: There exist a floor i_j of (G, f, g) and a floor p_j of (G_j, f_j, g_j) (resp. of (G_j^*, f_j^*, g_j^*)) such that (see Fig. 4):

(i) for $q = 1, 2, \ldots, p_j$, the q-th floor of (G_j, f_j, g_j) (resp. of (G_j^*, f_j^*, g_j^*)) is a graph whose outer face consists of vertices all belonging to the $(p_j - q + 1)$-th level of the i_j-th floor of (G, f, g);

(ii) for $q = p_j + 1, p_j + 2, \ldots, k(G_j)$ (resp. for $q = p_j + 1, p_j+2, \ldots, k(G_j^*)$), the q-th floor of (G_j, f_j, g_j) (resp. of (G_j^*, f_j^*, g_j^*)) is a graph whose outer face consists of vertices all belonging to the first level of the $(i_j + q - p_j)$-th floor of (G, f, g).

The algorithm to construct a queue layout of (G, f, g) builds an ordered list L of vertices and, at the end of its execution, the order given by L will be the total order of the vertices of G. The algorithm maintains two invariants.

- *Invariant A:* All the vertices of the i-th floor of (G, f, g) come in L before all the vertices of the $(i+1)$-th floor of (G, f, g), for each $1 \leq i \leq k - 1$.
- *Invariant B:* The border vertices of the i-th floor of (G, f, g) come in L before all the non-border vertices of the i-th floor of (G, f, g), for each $1 \leq i \leq k$.

Invariant A corresponds to partition L into sublists, where sublist L_i contains all and only the vertices of the i-th floor of (G, f, g). Hence, the vertices of each floor of (G, f, g) appear consecutively in L. Recall that the splits of Cases 1–5 may originate floored graphs whose floors are different from the ones of (G, f, g). However, when a vertex is inserted into L, it is inserted in the sublist L_i of the floor it belongs to in (G, f, g).

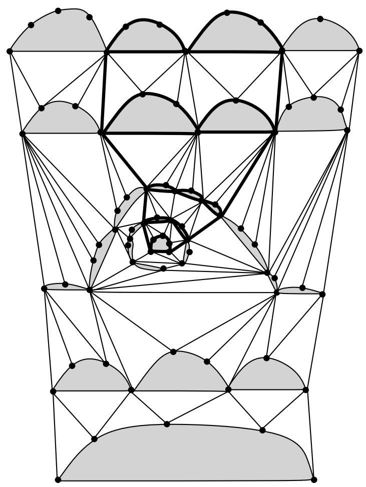


Figure 4. Illustration of the statement of Lemma 8. Graphs (G, f, g) and (G_j, f_j, g_j) are shown. The borders and the outer faces of the floors of (G_j, f_j, g_j) are shown by thick lines. In this example $i_j = 3$ and $p_j = 3$.

Recall that the i-th floor of (G, f, g) is composed of a sequence of edges or level-2-connected graphs G^i_j, where such a sequence can degenerate to be a single vertex for the first and/or the last floor. Denote by $m(G^i_j)$ the number of levels of G^i_j (if G^i_j is an edge, then let $m(G^i_j) = 1$) and denote by $m(i)$ the maximum among the $m(G^i_j)$, for $j = 1, \dots, x(i)$. If the i-th level is a vertex, then let $m(i) = 1$. For each $i = 1, 2, \dots, k$, partition L_i into consecutive sublists $L_{i,1}, L_{i,2}, \dots, L_{i,m(i)}$. Each list $L_{i,j}$ is in turn partitioned into two consecutive sublists $L'_{i,j}$ and $L''_{i,j}$. See Fig. 5.

We now sketch the algorithm for constructing L. It starts by placing, for each $1 \leq p \leq k$, the border vertices on the p-th floor of (G, f, g) at the first positions of $L'_{p,1}$ so that Invariant B is satisfied. Then, the graphs (G_j, f_j, g_j) obtained by the balanced raising-path decomposition are processed one at a time. When (G_j, f_j, g_j) is processed, an ordering J of the vertices of G_j is recursively constructed. The ordering of the vertices of G_j in L is *almost* the same as in J. Namely, each floor of (G_j, f_j, g_j) has in L the same vertex ordering as in J and the vertices of each floor of (G_j, f_j, g_j) appear consecutively in L (except for the border vertices of (G_j, f_j, g_j) on such a floor). However, the order of the floors of (G_j, f_j, g_j) in L may differ from the order of the floors of (G_j, f_j, g_j) in J. Namely, while the floors of (G_j, f_j, g_j) are ordered in J according to the definition of floored graph, such floors are ordered in L according to the level of the floor of (G, f, g) their outer faces belong to. Then, there exists an index p_j for (G_j, f_j, g_j) such that the ordering of the floors of (G_j, f_j, g_j) in L is the p_j-th first, then the $(p_j - 1)$-th, then the $(p_j - 2)$-th, ..., then the first, then the $(p_j + 1)$, then the $(p_j + 2)$-th, ..., then the last. We now formally state the algorithm.

Algorithm 1 VERTEX-ORDERING

Require: A floored graph (G, f, g).
Ensure: A vertex ordering of G in a list L.

for $p = 1, 2, \dots, k$ **do**
  insert vertex $u^p_1(G)$ into $L'_{p,1}$;
end for
for $p = 1, 2, \dots, k$ **do**
  if vertex $v^p_{x(p)}(G)$ does not belong to L **then**
    append $v^p_{x(p)}(G)$ to $L'_{p,1}$;
  end if
end for
let $(G_1, f_1, g_1), (G_2, f_2, g_2), \dots, (G_{l+1}, f_{l+1}, g_{l+1})$ be the graphs obtained by performing a balanced raising-path decomposition of (G, f, g);
for $j = 1, 2, \dots, l + 1$ **do**
  process graph (G_j, f_j, g_j);
  for $y = 1, 2, \dots, k(G_j)$ **do**
    consider the border vertices $u^y_1(G_j)$ and $v^y_{x(y)}(G_j)$ of (G_j, f_j, g_j);
    if $u^y_1(G_j)$ does not belong to L **then**
      let p be the floor of $u^y_1(G_j)$ in (G, f, g);
      let q be the level of $u^y_1(G_j)$ in $G[F_p]$;
      append $u^y_1(G_j)$ to $L'_{p,q}$;
    end if
    if $v^y_{x(y)}(G_j)$ does not belong to L **then**
      let p be the floor of $v^y_{x(y)}(G_j)$ in (G, f, g);
      let q be the level of $v^y_{x(y)}(G_j)$ in $G[F_p]$;
      append $v^y_{x(y)}(G_j)$ to $L'_{p,q}$;
    end if
    recursively construct a vertex ordering J of (G_j, f_j, g_j);
    for $1 \leq r \leq k(G_j)$ **do**
      let p and q be such that all the vertices on the outer face of the r-th floor of (G_j, f_j, g_j) are on the q-th level of the p-th floor of (G_j, f_j, g_j) (by Lemma 8);
      append all the non-border vertices of the r-th floor of (G_j, f_j, g_j) to $L''_{p,q}$ in the same order as they appear in J;
    end for
  end for
end for
return L;

Figure 5. The partition of L into sublists.

Observe that, for $j = 1, 2, \ldots, l + 1$, the algorithm first inserts into L the border vertices of (G_j, f_j, g_j) and then inserts into L the non-border vertices of (G_j, f_j, g_j). Further, when the algorithm processes (G_j, f_j, g_j), all the border vertices of $(G_{j-1}, f_{j-1}, g_{j-1})$ have been already inserted into L.

We now study the edges of (G, f, g) relating the end-vertices of such edges to their position in L and to the floor of (G, f, g) they belong to.

A *visible edge* is an edge of G that has one end-vertex in a list $L'_{p,q}$, for some p and q, and one end-vertex in a list $L'_{r,s}$, for some r and s. A *semi-visible edge* is an edge of G that has one end-vertex in a list $L'_{p,q}$, for some p and q, and one end-vertex in a list $L''_{r,s}$, for some r and s. An *invisible edge* is an edge of G that has one end-vertex in a list $L''_{p,q}$, for some p and q, and one end-vertex in a list $L''_{r,s}$, for some r and s. Intuitively, visible, semi-visible, and invisible edges are such that both end-vertices, one end-vertex, and no end-vertex, respectively, belong to the borders of graphs (G_j, f_j, g_j). The *inter-floor edges* are the edges of G that connect vertices on consecutive floors of (G, f, g). The *intra-floor edges* are the edges of G that connect vertices on the same floor of (G, f, g).

We have the following lemmata.

Lemma 9: Every edge of G is either a visible edge, or a semi-visible edge, or an invisible edge.

Lemma 10: Every edge of G is either an intra-floor edge or a inter-floor edge.

Lemma 11: Every inter-floor edge of (G, f, g) is an inter-floor edge of a graph (G_j, f_j, g_j), for some $1 \leq j \leq l + 1$.

We now introduce some definitions and notation to compute the queue number of G once the vertices of G have the order specified by L. Let $q(G, f, g)$, $q_{intra}(G, f, g)$, and $q_{inter}(G, f, g)$ denote the number of queues needed to embed all the edges, only the intra-floor edges, and only the inter-floor edges, respectively, of a floored graph (G, f, g) once the vertices of G have the order computed by Algorithm Vertex-Ordering. Let $q(n)$, $q_{intra}(n)$, and $q_{inter}(n)$ denote the maximum of $q(G, f, g)$, $q_{intra}(G, f, g)$, and $q_{inter}(G, f, g)$, respectively, over all possible graphs (G, f, g) with n non-border vertices.

Let G be a graph and let $\prec$ be a vertex ordering of G. A set of edges $(a_1, b_1), (a_2, b_2), \ldots, (a_m, b_m)$ is a *rainbow* of *size* m if $a_1 \prec a_2 \prec \ldots \prec a_m \prec b_m \prec \ldots \prec b_2 \prec b_1$. We use the following result.

Lemma 12: [25] The queue number of a graph G is the minimum, taken among all vertex orderings $\prec$ of G, of the maximum size of a rainbow in $\prec$.

We now prove that the edges of (G, f, g) can be embedded into $O(\log^2 n)$ queues, once the vertices of G have the order of L.

Lemma 13: The following statements hold:

(1) The size of a rainbow of visible inter-floor edges of (G, f, g) is at most five.
(2) The size of a rainbow of semi-visible inter-floor edges of (G, f, g) is at most eight.
(3) The size of a rainbow of invisible inter-floor edges of (G, f, g) is at most the maximum size of a rainbow of inter-floor edges in one of graphs $(G_1, f_1, g_1), \ldots, (G_{l+1}, f_{l+1}, g_{l+1})$.

Lemma 14: $q_{inter}(n) = O(\log n)$.

Proof: Every inter-floor edge is either visible, or semi-visible, or invisible, by Lemma 9. By definition, the size of a rainbow of inter-floor edges among graphs $(G_1, f_1, g_1), \ldots, (G_{l+1}, f_{l+1}, g_{l+1})$ is at most $q_{inter}(n/2)$. By Lemmata 13 and 12, all the visible, semi-visible, and invisible inter-floor edges of G can be embedded into at most 5, 8, and $q_{inter}(n/2)$ queues, respectively. Hence, $q_{inter}(n) = 13 + q_{inter}(n/2) = O(\log n)$. ■

Lemma 15: The following statements hold:

(1) The size of a rainbow of visible intra-floor edges of (G, f, g) is at most seven.
(2) The size of a rainbow of semi-visible intra-floor edges of (G, f, g) is at most twelve.
(3) The size of a rainbow of invisible intra-floor edges of (G, f, g) is at most the maximum size of a rainbow of inter-floor edges among graphs $(G_1, f_1, g_1), \ldots, (G_{l+1}, f_{l+1}, g_{l+1})$ plus the maximum size of a rainbow of intra-floor edges among graphs $(G_1, f_1, g_1), \ldots, (G_{l+1}, f_{l+1}, g_{l+1})$.

Lemma 16: $q_{intra}(n) = O(\log^2 n)$.

Proof: Every intra-floor edge is either visible, or semi-visible, or invisible, by Lemma 9. By definition, the size of a rainbow of inter-floor edges among graphs $(G_1, f_1, g_1), \ldots, (G_{l+1}, f_{l+1}, g_{l+1})$ is at most $q_{inter}(n/2)$ and the size of a rainbow of intra-floor edges among graphs $(G_1, f_1, g_1), \ldots, (G_{l+1}, f_{l+1}, g_{l+1})$ is at most $q_{inter}(n/2)$. By Lemmata 15 and 12, all the visible, semi-visible, and invisible inter-floor edges of G can be embedded into at most 7, 12, and $q_{inter}(n/2) + q_{intra}(n/2)$ queues, respectively. By Lemma 14, $q_{inter}(n/2) = O(\log n)$. Hence, $q_{intra}(n) =$

$O(\log n) + q_{intra}(n/2) = O(\log^2 n)$. ■

Theorem 1: Every n-vertex level-2-connected graph G has $O(\log^2 n)$ queue number.

Proof: Consider any level-2-connected graph G with n vertices. Let $f(v) = 1$, for every vertex v in G. Let g be any edge on the outer face of G. By Lemma 4 (G, f, g) is a floored graph. Further, G has $n - 2$ non-border vertices. By Lemma 10, every edge of (G, f, g) is either an inter-floor edge or an intra-floor edge. Hence, $q(G, f, g) \leq q_{intra}(G, f, g) + q_{inter}(G, f, g)$. By definition, $q_{intra}(G, f, g) \leq q_{intra}(n)$ and $q_{inter}(G, f, g) \leq q_{inter}(n)$. Hence, $q(G, f, g) \leq q_{intra}(n) + q_{inter}(n)$. By Lemmata 14 and 16, $q(G, f, g) = O(\log^2 n) + O(\log n) = O(\log^2 n)$, and the theorem follows. ■

Lemma 17: (Dujmović and Wood [12]) Let D be a q-queue subdivision of a graph G with at most one subdivision vertex per edge. Then G has a $2q(q + 1)$-queue layout.

Theorem 2: Every n-vertex planar graph has $O(\log^4 n)$ queue number and a queue layout with such a queue number can be computed in polynomial time.

Proof: By Lemma 1, for every planar graph G, an $O(n)$-vertex level-2-connected graph G^* exists such that G^* contains a 1-subdivision G' of G as a subgraph. By Theorem 1, G^* has $O(\log^2 n)$ queue number, hence G' has $O(\log^2 n)$ queue number. By Lemma 17, G has $O(\log^4 n)$ queue number. Finally, it is easy to see that the algorithm for constructing a vertex ordering of G can be implemented in polynomial time. ■

The bound of Theorem 2, together with the following theorem, immediately implies an $O(n \text{ polylog } n)$ upper bound on the volume requirements of 3D straight-line crossing-free drawings of planar graphs.

Theorem 3: (Dujmović, Morin, and Wood [9]) Let $\mathcal{G}$ be a proper minor-closed family of graphs, and let $\mathcal{F}(n)$ be a family of functions closed under multiplication. The following are equivalent:

(a) Every n-vertex graph in $\mathcal{G}$ has a $\mathcal{F}(n) \times \mathcal{F}(n) \times O(n)$ drawing,
(b) $\mathcal{G}$ has track number $tn(\mathcal{G}) \in \mathcal{F}(n)$, and
(c) $\mathcal{G}$ has queue number $qn(\mathcal{G}) \in \mathcal{F}(n)$.

A result of Dujmović and Wood [11] related to Theorem 3 lead us to precisely determine the value k for the $O(n \log^k n)$ volume upper bound for 3D straight-line crossing-free drawings of planar graphs.

Theorem 4: Every planar graph has a 3D straight-line crossing-free drawing with $O(n \log^{16} n)$ volume.

Proof: Every graph G with *acyclic chromatic number* c and queue number q has track number $tn(G) \leq c(2q)^{c-1}$ [9], where the acyclic chromatic number of a graph G is the minimum number of colors such that G admits a proper coloring in which each pair of colors induces a forest. Since every planar graph has acyclic chromatic number at most five [2] and queue number $O(\log^4 n)$ (by Theorem 2), then every planar graph has track number $O(\log^{16} n)$. Dujmović and Wood [11] proved that every c-colorable graph G with n vertices and track-number $tn(G) \leq t$ has a 3D straight-line crossing-free drawing with $O(c^7 tn)$ volume. Since planar graphs are 4-colorable, the theorem follows. ■

Further, Theorem 2, together with results in [12], implies the following:

Corollary 2: Every graph admitting a drawing in the plane with at most k crossings per edge has queue number $O(\log^{4(k+1)} n)$.

Corollary 3: Every planar graph has a 3D poly-line crossing-free drawing with $O(n \log \log n)$ volume and with $O(\log \log n)$ bends per edge.

V. Conclusion

In this paper we have shown that planar graphs have $O(\log^4 n)$ queue number, improving upon the previously best known $O(\sqrt{n})$ bound. Determining the asymptotic behavior of the queue number of planar graphs remains a challenging open problem for which, as far as we know, no super-constant lower bound is known. While we find unlikely that the techniques introduced in this paper can lead to determine a constant upper bound for the queue number of planar graphs, it is possible that, by directly handling cut-vertices in a planar graph decomposition similar to the one we presented, an $O(\log^2 n)$ upper bound can be achieved. We also leave to further research work to design a time-efficient implementation of our algorithm.

As a consequence of our results on the queue number of planar graphs and of a correspondence between queue layouts and 3D straight-line crossing-free drawings introduced by Dujmović et al. [9], planar graphs admit 3D straight-line crossing-free drawings in $O(n \log^c n)$ volume, for some constant c. The question of whether such a volume bound can be reduced to linear remains one of the main unsolved problems in Graph Drawing. In particular, we find fundamental to understand whether small volume can be achieved while obtaining a good aspect ratio for the drawing.

Acknowledgment

Thanks to David Wood and to an anonymous referee for helpful suggestions on how to use the results in [9], [11], [12]. Such suggestions allowed us to improve the bounds in Theorem 4 and to state Corollaries 2 and 3.

References

[1] S. N. Bhatt, F. R. K. Chung, F. T. Leighton, and A. L. Rosenberg. Scheduling tree-dags using FIFO queues: A control-memory trade-off. *Journal of Parallel and Distributed Computing*, 33(1):55–68, 1996.

[2] O. V. Borodin. On acyclic colourings of planar graphs. *Discrete Mathematics*, 25:211–236, 1979.

[3] F. J. Brandenburg, D. Eppstein, M. T. Goodrich, S. G. Kobourov, G. Liotta, and P. Mutzel. Selected open problems in graph drawing. In G. Liotta, editor, *Graph Drawing (GD '03)*, volume 2912 of *LNCS*, pages 515–539, 2004.

[4] J. F. Buss and P. W. Shor. On the pagenumber of planar graphs. In *Symposium on Theory of Computing (STOC '84)*, pages 98–100. ACM, 1984.

[5] R. F. Cohen, P. Eades, T. Lin, and F. Ruskey. Three-dimensional graph drawing. *Algorithmica*, 17(2):199–208, 1997.

[6] E. D. Demaine, J. S. B. Mitchell, and J. O'Rourke. The Open Problems Project. http://maven.smith.edu/∼orourke/TOPP/Welcome.html.

[7] G. Di Battista, F. Frati, and J. Pach. On the queue number of planar graphs. Tech. Report RT-DIA-169-2010, Dept. of Computer Science and Automation, Roma Tre University, 2010. http://web.dia.uniroma3.it/ricerca/rapporti/rt/2010-169.pdf.

[8] V. Dujmović. *Track Layouts of Graphs*. PhD thesis, School of Computer Science, McGill University, Montreal, Canada, 2003.

[9] V. Dujmović, P. Morin, and D. R. Wood. Layout of graphs with bounded tree-width. *SIAM Journal on Computing*, 34(3):553–579, 2005.

[10] V. Dujmović and D. R. Wood. On linear layouts of graphs. *Discrete Mathematics & Theoretical Computer Science*, 6(2):339–358, 2004.

[11] V. Dujmović and D. R. Wood. *Three-dimensional grid drawings with sub-quadratic volume*, pages 55–66. In J. Pach, editor, Towards a Theory of Geometric Graphs. Contemporary Mathematics, American Mathematical Society, 2004.

[12] V. Dujmović and D. R. Wood. Stacks, queues and tracks: Layouts of graph subdivisions. *Discrete Mathematics & Theoretical Computer Science*, 7(1):155–202, 2005.

[13] T. Endo. The pagenumber of toroidal graphs is at most seven. *Discrete Mathematics*, 175(1-3):87–96, 1997.

[14] H. Enomoto, T. Nakamigawa, and K. Ota. On the pagenumber of complete bipartite graphs. *Journal of Combinatorial Theory, Series B*, 71(1):111–120, 1997.

[15] S. Felsner, G. Liotta, and S. K. Wismath. Straight-line drawings on restricted integer grids in two and three dimensions. *Journal of Graph Algorithms & Applications*, 7(4):363–398, 2003.

[16] R. A. Games. Optimal book embeddings of the FFT, Benes, and barrel shifter networks. *Algorithmica*, 1(2):233–250, 1986.

[17] J. L. Ganley and L. S. Heath. The pagenumber of k-trees is O(k). *Discrete Applied Mathematics*, 109(3):215–221, 2001.

[18] T. Hasunuma and Y. Shibata. Embedding De Bruijn, Kautz and shuffle-exchange networks in books. *Discrete Applied Mathematics*, 78(1-3):103–116, 1997.

[19] L. S. Heath. Embedding planar graphs in seven pages. In *Foundations of Computer Science (FOCS '84)*, pages 74–83. IEEE, 1984.

[20] L. S. Heath and S. Istrail. The pagenumber of genus g graphs is $O(g)$. *Journal of the ACM*, 39(3):479–501, 1992.

[21] L. S. Heath, F. T. Leighton, and A. L. Rosenberg. Comparing queues and stacks as mechanisms for laying out graphs. *SIAM J. Discrete Math.*, 5(3):398–412, 1992.

[22] L. S. Heath and S. V. Pemmaraju. Stack and queue layouts of posets. *SIAM Journal on Discrete Mathematics*, 10(4):599–625, 1997.

[23] L. S. Heath and S. V. Pemmaraju. Stack and queue layouts of directed acyclic graphs: Part II. *SIAM Journal on Computing*, 28(5):1588–1626, 1999.

[24] L. S. Heath, S. V. Pemmaraju, and A. N. Trenk. Stack and queue layouts of directed acyclic graphs: Part I. *SIAM Journal on Computing*, 28(4):1510–1539, 1999.

[25] L. S. Heath and A. L. Rosenberg. Laying out graphs using queues. *SIAM Journal on Computing*, 21(5):927–958, 1992.

[26] S. M. Malitz. Genus g graphs have pagenumber $O(\sqrt{g})$. *Journal of Algorithms*, 17(1):85–109, 1994.

[27] S. M. Malitz. Graphs with e edges have pagenumber $O(\sqrt{e})$. *Journal of Algorithms*, 17(1):71–84, 1994.

[28] L. T. Ollmann. On the book thicknesses of various graphs. In F. Hoffman, R. B. Levow, and R. S. D. Thomas, editors, *Southeastern Conference on Combinatorics, Graph Theory and Computing*, volume VIII of *Congressus Numerantium*, page 459, 1973.

[29] S. V. Pemmaraju. *Exploring the Powers of Stacks and Queues via Graph Layouts*. PhD thesis, Virginia Polytechnic Institute and State University, U.S.A., 1992.

[30] V. R. Pratt. Computing permutations with double-ended queues, parallel stacks and parallel queues. In *Symposium on Theory of Computing (STOC '73)*, pages 268–277, 1973.

[31] S. Rengarajan and C. E. Veni Madhavan. Stack and queue number of 2-trees. In D. Z. Du and M. Li, editors, *Conference on Computing and Combinatorics (COCOON '95)*, volume 959 of *LNCS*, pages 203–212, 1995.

[32] F. Shahrokhi and W. Shi. On crossing sets, disjoint sets, and pagenumber. *Journal of Algorithms*, 34(1):40–53, 2000.

[33] R. E. Tarjan. Sorting using networks of queues and stacks. *Journal of the ACM*, 19(2):341–346, 1972.

[34] M. Yannakakis. Embedding planar graphs in four pages. *Journal of Computer and System Sciences*, 38(1):36–67, 1989.