Bounding Queue-Number in Planar Graphs

Jeffrey Kam

Spring 2020

Contents

1 Overview

Stacks and queues are ubiquitous in algorithm design, so it is natural to define a similar data structure for graphs, namely the stack and queue layout in Section 2. The stack and queue layout corresponds to DFS and BFS respectively, and the stack and queue-number provides a way of quantifying the power of stack and queue in graphs. (See Example 2.6 for details)

However, even for a simple graph class like planar graph, it was unknown to us whether the queue-number and stack-number are bounded. Heath, Leighton, and Rosenberg first conjectured that the queue-number is bounded in 1992 [6]. This remained unproven for 27 years, until recently in 2019. *Dujmovic et al.* finally had a breakthrough and showed that the queue-number of all planar graphs have a constant upper bound of 49 [2]. The tools developed to prove this result have led to the resolution of a few other open problems of related nature, such as bounded non-repetitive chromatic number in planar graphs [4]. However, the problem of whether we can bound the queue-number bt the stack-number, or vice versa, is still unsolved. In other words, we still don't know whether a queue or stack is more powerful in graphs.

In this lecture note, we will cover the following:

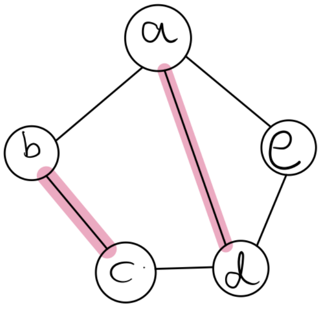
1. Provide the necessary definitions for the proof, accompanied by illustrative examples.
2. Introduce a new technique called layered partition, and prove some useful results on the relation between layered partition and queue-number.
3. Prove that planar graphs have bounded queue-number of 766.
4. Briefly explore the proof for reducing the upper bound to 49, by exploiting some structures called Tripod (see Definition 5.1) in a planar triangulation.

The result of bounded queue-number can be generalized to genus g graphs and proper minor-closed class of graphs, but the details will not be provided here; interested readers should go read the paper by Dujmovic et al [3]. [1]

2 Definitions

Definition 2.1 (Nested Edge)
Given a graph G and a vertex ordering $(\prec)$ of G. Two edges uv, xy of G, and WLOG $u \prec v$ and $x \prec y$, are said to be *nested* if $u \prec x \prec y \prec v$.

Example 2.2 Example of nested edges in a graph.



vertex ordering:
a b c d e

Then, ad, bc are nested since $a < b < c < d$ in the vertex ordering.
Note that ae, bc are also nested.

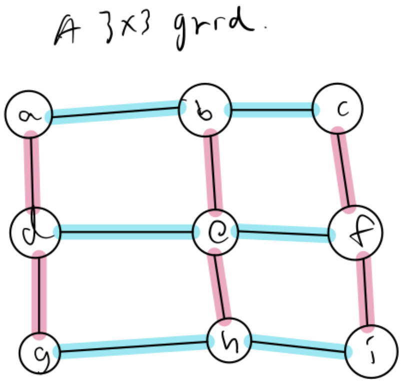
Definition 2.3 (Queue Layout)
A *queue layout* of G with vertex ordering $(\prec)$ is a partition of the edges, say $E_1, \cdots, E_k \subseteq E(G)$, such that for every pair of edges $e, f \in E_i$, no two edges nest. Each of the edge partition in a queue layout is called a *queue*.

[1] I will mainly be covering page 1 to 18 of reference [3] in this lecture note.

Definition 2.4 (*k*-queue Layout and Queue Number)
G is said to have a *k-queue layout* if there is a valid queue layout using only k edge partitions. The *queue-number* of G, denoted $qn(G)$, is the smallest k such that G has a k-queue layout.

Example 2.5 For trees, we have a 1-queue layout simply following BFS traversal ordering.

For a less trivial example, we consider the queue layout for a 3×3 grid graph, where the edge partitions are E_1 and E_2. This uses 2 partitions, so G with this vertex ordering admits a 2-queue layout.



vertex ordering:
a b c d e f g h i

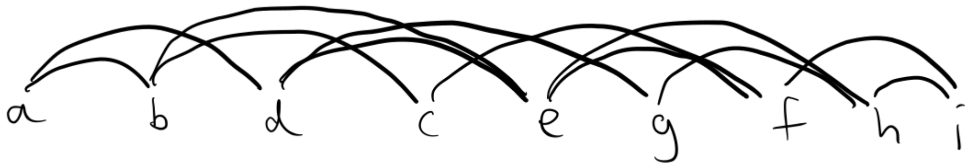
E_1 = all horizontal edges

E_2 = all vertical edges.

So, it has a 2-queue layout.

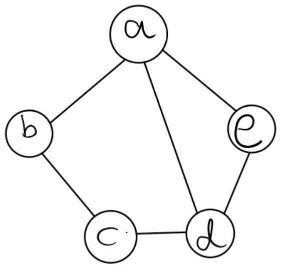
However, when given a different vertex ordering, G can admit a 1-queue layout! So, the queue-number of G is $qn(G) = 1$.

If the vertex ordering is a b d c e g f h i,
we have a 1-queue layout, so $\underline{qn(G) = 1}$, not 2.



Note There is a similar definition with *stack*, where the restriction on edges is that every pair within the same edge partition cannot "cross" (cannot satisfy $u < x < v < y$), and this structure corresponds to DFS in a similar fashion of how queue layout corresponds to BFS in Example 2.6.

Example 2.6 The following example shows how a queue layout corresponds to BFS, and in particular, why a queue layout could not have nested edges. Assume the following graph admits a 1-queue layout using the vertex ordering *abcde* shown below.



vertex ordering:
a b c d e.

$v_i = a \begin{cases} \text{push: } ab \\ \text{push: } ad \\ \text{push: } ae \end{cases}$ $\cdots$ $\begin{bmatrix} ae \\ ad \\ ab \end{bmatrix} \downarrow$

$v_i = b \begin{cases} \text{pop: } ab \\ \text{push: } bc \end{cases}$ $\begin{bmatrix} bc \\ ae \\ ad \end{bmatrix} \downarrow$

$v_i = c \begin{cases} \text{pop: } bc \\ \text{push: } cd \end{cases}$ ← we have a conflict,
bc is not at the bottom of queue.

In the BFS, we want to traverse all the edges by the vertex ordering. In particular, the root of the BFS traversal is the first vertex in the ordering. Let v be some vertex in the ordering, and let $a_1, a_2, \cdots, a_k$ be the neighbors of v to the left of v in the vertex ordering, with $a_1 < a_2 < \cdots < a_k$. Similarly, let $b_1, \cdots, b_j$ be neighbors to the right of v and $b_1 < b_2 < \cdots < b_j$. Then, when we reach vertex v, we first remove $a_1v, \cdots, a_kv$ from the queue and push $vb_1, \cdots, vb_m$ to the queue. However, the example above shows that if there is a pair of nested edges, then the queue structure will be invalid.

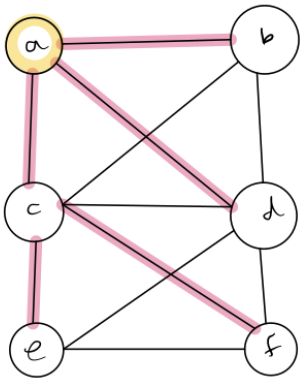
Recall $P = (P_1, \cdots, P_n)$ is a *partition* of G if each $v \in V(G)$ belongs to exactly one of P_i. Each of these P_i is called a *part*.

Definition 2.7 (Layered Partition)
A *layered partition* of G is an ordered vertex partition $(V_0, V_1, \cdots, V_n)$ such that if $xy \in E(G)$, then one of them holds:

- $x, y \in V_i$ for some i. This is called an *intra-layer edge.*
- $x \in V_i$ and $y \in V_{i+1}$ for some i. This is called an *inter-layer edge.*

Definition 2.8 (BFS Layering)
Let r be a root in a connected graph G. A *BFS layering* is a layering where each partition $V_i = \{v : dist(r, v) = i, v \in V(G)\}$. More intuitively, we can consider a BFS layering of G in terms of a BFS spanning tree T, where each layer i of the tree T serves as the partition V_i.

Example 2.9 An example of a (BFS) layering for the following graph.



Given a layering (Recall this is the BFS layering)

$V_1 = \{a\}$, $V_2 = \{b, c, d\}$, $V_3 = \{e, f\}$.

Given a partition $P = \{a, b\}, \{c, d\}, \{e\}, \{f\}$.

P has layered width 2, because

$|\{c, d\} \cap V_2| = 2$.

Note that $\{ab, ac, ce\}$ and $\{bc, cd, ef\}$ are examples of inter-layer and intra-layer edges, respectively.

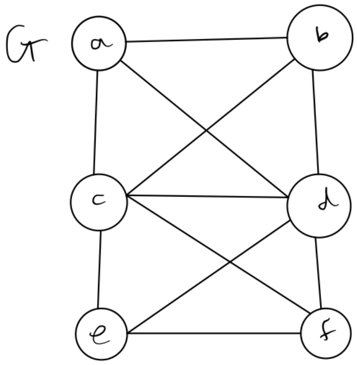
Definition 2.10 (*H*-partition)
A *H-partition* of G is a partition parameterized by the graph H, namely $(A_x : x \in V(H))$, where each A_x is a partition. For every edge $uv \in E(G)$, if u and v belong to partitions A_x and A_y respectively, then one of them holds:

- $x = y$ (u, v are in the same partition). In this case, uv is called an *intra-bag edge*.
- $xy \in E(H)$. In this case, uv is called an *inter-bag edge*.

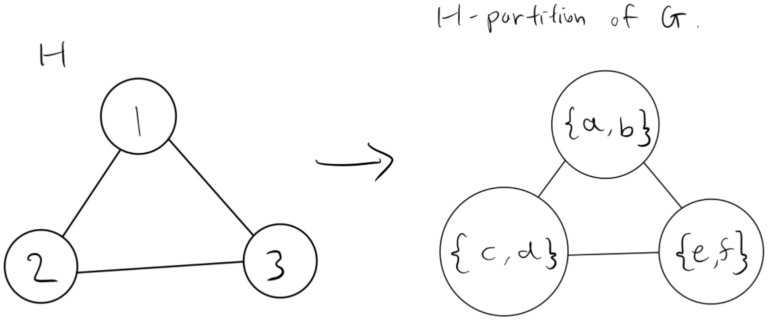
The *width* of the H-partition is $\max |A_x| : x \in V(H)$.

Note A special case of H-partition is where H is a tree T, known as a tree-partition. This is unlike tree-decomposition in the sense that our bags are partition, and thus, the intersection of our bags is always empty here.

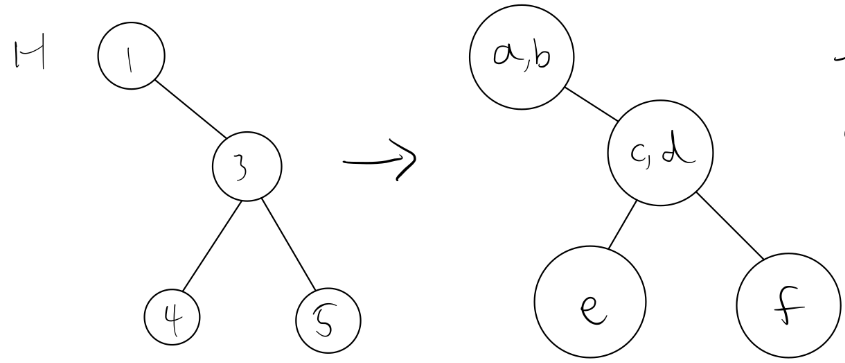
Example 2.11 Given the following graph G, we have two examples of H decompositions, where in the second one, H is a tree, so it is a tree partition.



For the C_3-decomposition below (since $H = C_3$), $\{ac, ad, bc, bd\}$ are examples of some inter-bag edges, and $\{ab, cd, ef\}$ are examples of intra-bag edges.



This H-partition has width 2.
Each bag has size 2

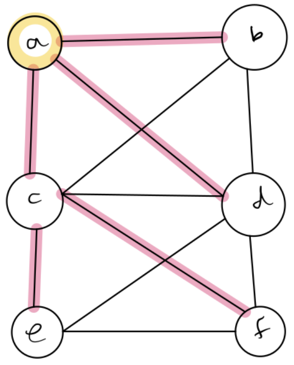


This is a tree decomposition, since H is a tree.
It has width 2.

Errata: "This is a tree partition, since H is a tree."

Definition 2.12 (Layered Width of a Partition)
Given a layering $(V_0, \cdots, V_k)$ and a partition $P = (P_1, \cdots, P_m)$ of G, the *layered width* of P is l if each partition in P has at most l vertices in each layer V_i. In other words, P has layered width l if and only if $P_i \cap V_j \leq l$ for all $1 \leq i \leq m$ and $1 \leq k \leq j$.

Example 2.13 Given the same graph in Example 2.9, and the same layering, we show that it has layered width 2.

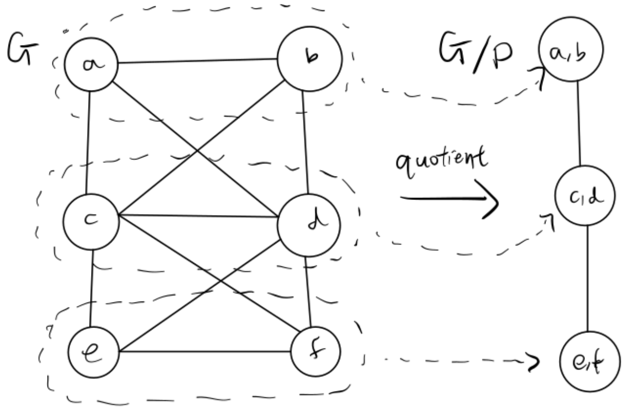


Given a layering (Recall this is the BFS layering)
$V_1 = \{a\}$, $V_2 = \{b, c, d\}$, $V_3 = \{e, f\}$.
Given a partition $P = \{a,b\}, \{c,d\}, \{e\}, \{f\}$.
P has layered width 2, because
$|\{c,d\} \cap V_2| = 2$.

Definition 2.14 (Quotient)
Given a partition P of G, the *quotient* of P, denoted G/P, is a graph where each part in P_i is represented by a vertex p_i and $p_i p_j \in E(G/P)$ if and only if there are edges in $E(G)$ connecting some vertices between partition P_i and P_j.

Example 2.15 Given a partition P of G (same graph as Example 2.9), we show an example of G/P.

Let $P = \{a,b\}, \{c,d\}, \{e,f\}$ be a partition.



Note that since $\{a,b\}, \{c,d\}, \{ef\}$ each of these partitions are connected. G/P is a minor of G by contracting each partition into a single vertex.

It is clear from here that the structure of G/P satisfy as an H-partition.

Remark 2.16 (Some notes on the properties of quotient)

- The structure we retain in the quotient is the adjacency structure between different partitions.
- If each of the partition is connected, the quotient of P is equivalent to the resulting graph by contracting each partition P_i into p_i, which means G/P is a minor of G. (See Example 2.15)
- Another way to define an H-partition P is that G has an H-partition if G/P is isomorphic to a spanning subgraph of H.
- Given a partition $P = (P_1, \cdots, P_n)$, G has a G/P-partition, where each $x \in V(G/P)$ corresponds a distinct part P_i of the partition.

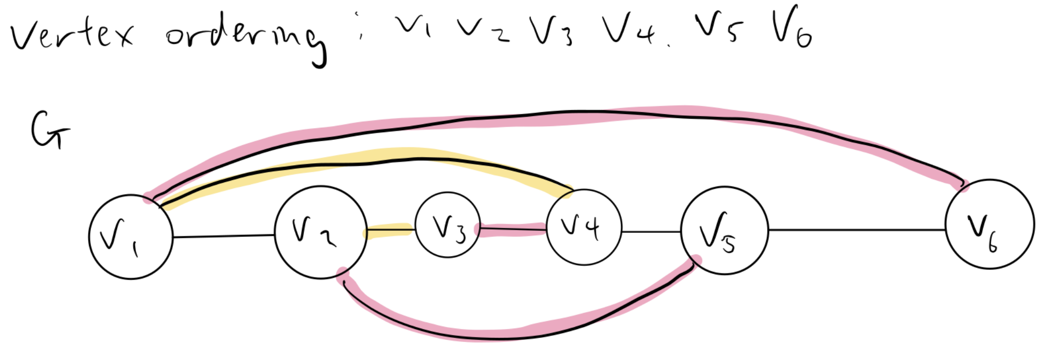
3 Relating Queues and Layered Partition

In this section, we will introduce two more definitions, then show a main result relating the queue-number of H and G, where the H-partition of G has layered width l with respect to some layering.

Definition 3.1 (Rainbow)
Given a vertex ordering of G, a *rainbow* of G with respect to the ordering is a set of pairwise nested edges.

Note Recall that nested edges requires that endpoints of edges are all distinct, thus, all of such edges in a rainbow must form a matching.

Example 3.2 Examples of 2 different rainbows in the graph G.



$\{v_1v_4, v_2v_3\}$ and $\{v_1v_6, v_2v_5, v_3v_4\}$ are both rainbows of G.

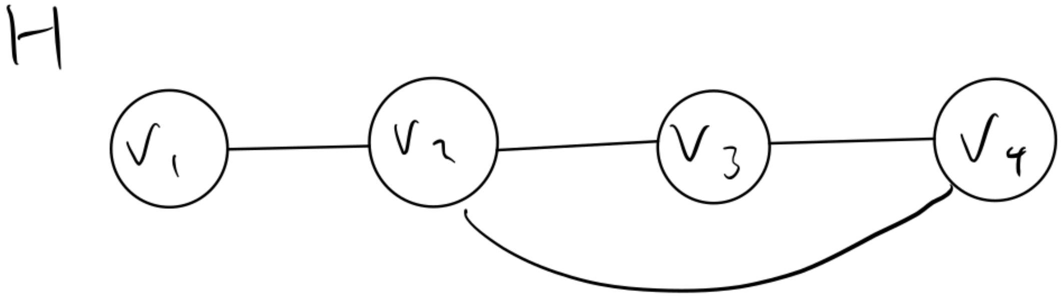
Note that they are rainbows because edges within those sets are pairwise nested. On a related note, the second rainbow $\{v_1v_6, v_2v_5, v_3v_4\}$ does look like a rainbow, with all the arcs nested within each other.

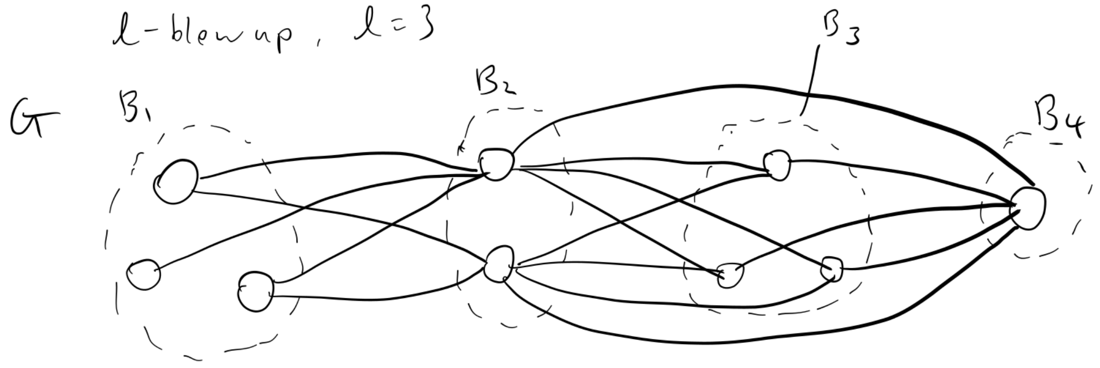
Definition 3.3 (l-blowup)
Let H be a graph with vertices $v_1, \cdots, v_n$. Define G as follows:

- $V(G)$: The vertex set are $B_1, B_2, \cdots, B_n$ where each B_i contains at most l vertices, and all vertices between the B_i's are pairwise distinct.
- $E(G)$: For each edge v_iv_j in H, we have an edge between every vertex in B_i and B_j. We can think of it as a complete bipartite graph with vertex class B_i and B_j.

G is called the *l-blowup* of H, and each B_i is called a *block* of G.

Example 3.4 Example of an l-blowup of H, with $l = 3$.





This is a 3-blow up because each B_i, $|B_i| \leq 3$.

Each of the groups circled by a dashed line is a block. Each of these blocks B_i corresponds to $v_i \in V(H)$ and has size ≤ 3, since G is a 3-blowup of H.

Before we prove our main lemma (Lemma 3.9) and its immediate result (Corollary 3.10), we need to first introduce the following three lemmas, without proof.

Lemma 3.5 (Heath and Rosenberg)
A vertex ordering in G admits a k-queue layout if and only if every rainbow with respect to the ordering has size at most k. [5]

Lemma 3.6 (Heath and Rosenberg)
A complete graph of n vertices, K_n, has queue-number $\lfloor \frac{n}{2} \rfloor$. [5]

Note The proofs of the lemmas above utilize extra lemmas and theorems that are an integral part of the paper's contribution. It will be too long if included in this lecture note. For the second lemma, the main line of attack is to show $qn(G) \leq \lfloor \frac{n}{2} \rfloor$ (the easy direction), and then invoke a central lemma to that paper, and show $qn(G) \geq \lfloor \frac{n}{2} \rfloor$ (the hard direction).

Lemma 3.7 (Wiechert)
Let G be a graph with treewidth k. Then, $qn(G) \leq 2^k - 1$. [8]

Note The crux of the proof involves a colouring argument.

This lemma by Wiechert (Lemma 3.7) is important because if we are able to bound the treewidth of some graphs related to the planar graph by some constant, we can then hope to apply the bound and see if it can provide some constant upper bound for the planar graph. The following lemma is necessary for our proof of Lemma 3.9.

Lemma 3.8 Let H be a graph with a 1-queue layout and G be an l-blowup of H. Then, G has an l-queue layout.

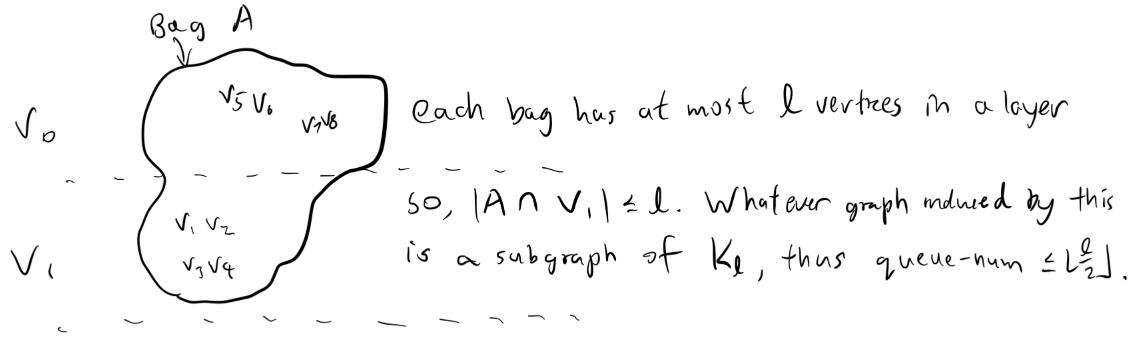

Bag A
V_0
$v_5\, v_6$
$v_7 v_8$
V_1
$v_1\, v_2$
$v_3 v_4$
each bag has at most l vertices in a layer
so, $|A \cap V_1| \leq l$. Whatever graph induced by this
is a subgraph of K_l, thus queue-num $\leq \lfloor \frac{l}{2} \rfloor$.

Figure 1: Explaining the intra-level intra-bag case

Proof. Let $v_1, \cdots, v_n$ be the ordering that admits a 1-queue layout in H, and $B_1, \cdots, B_n$ (the ordering within B_i can be arbitrary) be the vertex ordering of G. Let R be a rainbow of G. By Lemma 3.5, it is sufficient to show that $|R| \leq l$. The main idea is to show that all the left (or right) endpoints of the rainbow must lie in the same block in G. Suppose, for a contradiction, that all left and right endpoints lie in at least 2 different blocks respectively. For simplicity, say they are in a total of 4 different blocks, where B_1, B_2 are for left endpoints, and B_3, B_4 are for right endpoints. Then there must exist some edges connecting B_1, B_4, and another connecting B_2, B_3. This implies that there is an edge v_1v_4 and v_2v_3 in H, which is a contradiction, since they are nested. Thus, all left (or right) endpoints must lie in a single block. Recall that edges in a rainbow must nest and nesting edges must have distinct endpoints. Since each block has at most l vertices, the rainbow size $|R| \leq l$ as required. □

Lemma 3.9 If G has an H-partition with layered width l with respect to some layering $(V_0, V_1, \cdots, V_n)$, and H has a k-queue layout, then G has a $(3lk + \lfloor \frac{3}{2} l \rfloor)$-queue layout. Thus,

$$qn(G) \leq 3l\ qn(H) + \lfloor \frac{3}{2} l \rfloor$$

Proof. Let $x_1, \cdots, x_h$ be the vertex ordering of H that admits a k-queue layout, with queues $E_1, \cdots, E_k$. For each layer V_i, we define V_i' as $A_{x_1} \cap V_i, A_{x_2} \cap V_i, \cdots, A_{x_h} \cap V_i$, where the ordering within each $A_x \cap V_i$ is arbitrary. Then, let $V_1', \cdots, V_n'$ be the vertex ordering of G. We will show that G admits the desired queue layout with this vertex ordering. In particular, we have 4 cases to consider, but we will only consider the following 2 cases because the remaining 2 cases follow a similar argument.

- (Intra-level intra-bag edges) Consider the subgraph G' induced by $A_x \cap V_i$ for some $x \in V(H)$ and $0 \leq i \leq n$. Note that $|A_x \cap V_i| \leq l$ since it has layered width l, so G' is a subgraph of an l-complete graph. (See Figure 1) Thus, by Lemma 3.6, $qn(G') \leq qn(K_l) = \lfloor \frac{l}{2} \rfloor$. These subgraphs are located in different layers or in different bags, and since each of the bags and layers are located sequentially in the ordering, namely $V'_1, \cdots, V'_n$, the edges between the subgraphs cannot nest. Thus, $\lfloor \frac{l}{2} \rfloor$ queues suffice.

- (Inter-level inter-bag edges) We want to consider all edges going from A_x at V_i to A_y at V_{i+1}. Let the subgraph consisting of all these edges be G'. Fix a queue, E_α, consider 2 subgraphs G_1 and G_2 with inter-level inter-bag edges restricted within E_α. Let G_1 be defined as all inter-level inter-bag edges $xy \in E_\alpha$ where $x \in V_i, y \in V_{i+1}$ and $x \prec y$; and similarly, let G_2 be defined by $y \prec x$.

  Consider the following auxiliary graph Z_1 (See Figure 2). The main idea here is that each $z_{i,x}$ here represents vertices in A_x in layer V_i. Let Z_1 have the following vertex ordering from top left to bottom right, and the inferred vertex set

  $$z_{0,x_1}, \cdots z_{0,x_h}$$
  $$z_{1,x_1}, \cdots z_{1,x_h}$$
  $$\vdots$$
  $$z_{n,x_1}, \cdots z_{n,x_h}$$

  For the edges of Z_1, there is an edge from z_{i,x_j} to z_{i+1,x_k} if and only if $x_j x_k \in E_\alpha$ with $x_j \prec x_k$. Note that no two edges here nest. The only case it could have nested is if we have edges $z_{0,x_i} z_{1,x_j}$ and $z_{0,x_k} z_{1,x_l}$, where $i \prec k \prec l \prec j$. However, this would mean the edges $x_i x_j$ and $x_k x_l$ by construction of Z_1. So Z_1 admits a 1-queue layout.

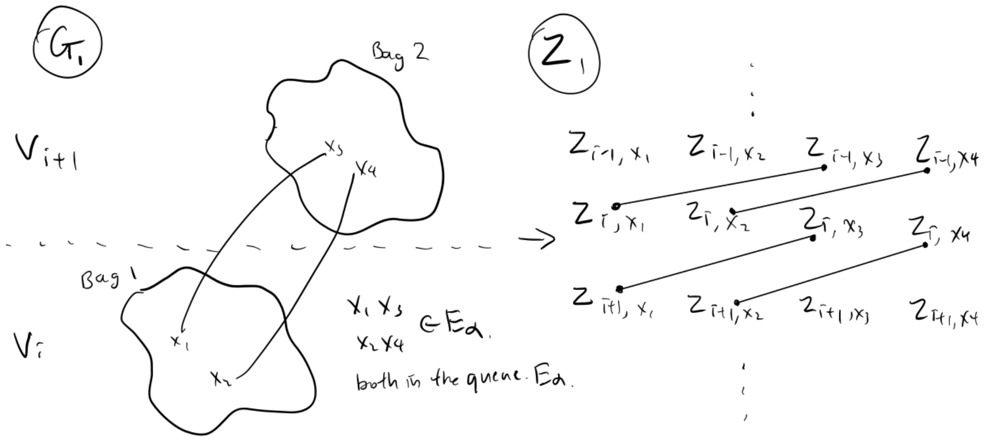
  Consider a maximum l-blowup of Z_1 (by replacing each vertex z_{0,x_1} by blocks of l vertices), say Z'_1. It is clear that G_1 is isomorphic to the subgraph of Z'_1, where each blocks of Z'_1 corresponds to partition bags A_x of G_1 in some layer V_i. Then, by Lemma 3.8, Z'_1 admits an l-queue layout, and since G_1 is a subgraph of it, G_1 also admits an l-queue layout. We can construct Z_2, Z'_2 similarly for G_2 for it to have an l-queue layout.

  Note that how G_1 and G_2 are parameterized by the edges in E_α. So, for different queues, say E_β, the G_1 and G_2 will be edge disjoint from the ones for E_α. Thus, the union of all these G_1 G_2 for each queue gives us all the inter-level inter-bag edges, and that each of these G_1 and G_2 serve as a queue. So, G' has a $2kl$-queue layout, since G_1, G_2 have an l-queue layout and each of them are parameterized by the k queues of H.

For the remaining two cases, intra-level inter-bag edges admit a kl-queue layout, and inter-level intra-bag edges admit a l-queue layout. Combining them all together,

$$\lfloor \frac{l}{2} \rfloor + kl + l + 2kl = 3kl + \lfloor \frac{3}{2} l \rfloor = 3l\ qn(H) + \lfloor \frac{3}{2} l \rfloor$$

we have the desired $(3l\ qn(H) + \lfloor \frac{3}{2} l \rfloor)$-queue layout for G. □


G_1
Bag 2
V_{i+1}
x_3
x_4
Bag 1
V_i
x_1
x_2
$x_1x_3 \in E_\alpha$.
x_2x_4
both in the queue E_α.
Z_1
Z_{i-1,x_1} Z_{i-1,x_2} Z_{i-1,x_3} Z_{i-1,x_4}
Z_{i,x_1} Z_{i,x_2} Z_{i,x_3} Z_{i,x_4}
Z_{i+1,x_1} Z_{i+1,x_2} Z_{i+1,x_3} Z_{i+1,x_4}

These are edges x_ix_j going from V_i to V_{i+1} that satisfy $x_i < x_j$, thus in Z_1.

We can think of Z_1 as all the possible edges xy in E_α and make them inter-level for all levels. Then, when we construct the l-blowup of Z_1, it must have contained G_1.

Figure 2: Explaining the inter-level inter-bag case

The following immediate result is central to both the main proofs for bounded queue-number in planar graphs and also for reducing the upper bound.

Corollary 3.10 Let P be a partition of G with layered width l such that G/P has treewidth at most k, then $qn(G) \leq 3l(2^k - 1) + \lfloor \frac{3}{2}l \rfloor$.

Proof. Recall that G has a G/P-partition (Remark 2.16). By assumption, G/P has treewidth at most k, so by Lemma 3.7, $qn(G/P) \leq 2^k - 1$. Combining this with Lemma 3.9, where $H = G/P$, we have

$$qn(G) \leq 3l\ qn(G/P) + \lfloor \frac{3}{2}l \rfloor \leq 3l(2^k - 1) + \lfloor \frac{3}{2}l \rfloor$$

as desired. □

4 Proof of Bounded Queue-number

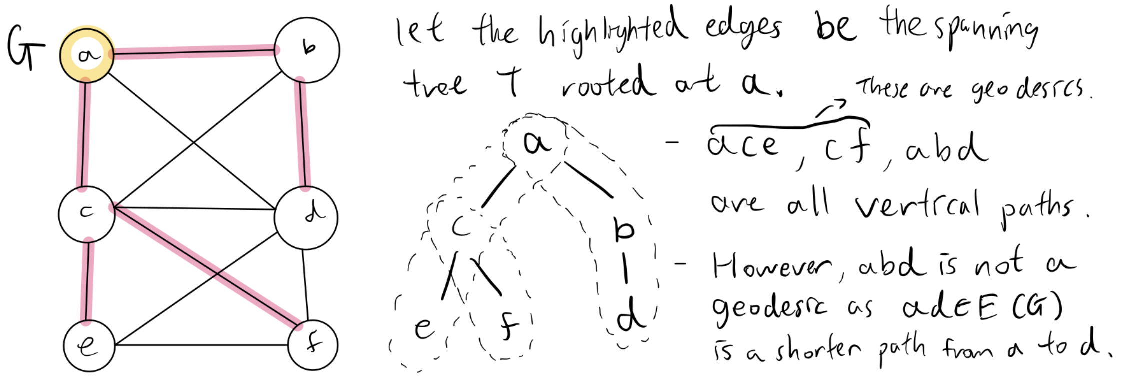
Of important note here is that if H is a spanning subgraph of G, it must satisfy $qn(H) \leq qn(G)$, because for H, we can achieve $qn(G)$ by simply copying the queues used in G and removing the edges not in H. This idea works for general subgraphs as well, but we will also have to remove vertices. Thus, a very natural way to approach this conjecture, showing planar graphs have bounded queue-number, is to think of what structures in a triangulated graph of n vertices we can exploit, such that we can get a constant bound for any planar graphs of n vertices. This leads to the core of our main lemma, where we show that any triangulated disc has a partition P such that the quotient has a constant treewidth and layered width. Then, we can apply Corollary 3.10 above, to get the desired queue-number bound for planar graphs.

Before we can prove the main lemma described above (shown later as Lemma 4.5), we need to introduce the idea of vertical paths and a famous result related to vertex coloring.

Definition 4.1 (Vertical Path)
Let T be a spanning tree of G rooted at r. A *vertical path* P of T is a path $(v_1, \cdots, v_k)$, where $dist_T(v_i, r) = d + i$ for some $d \geq 0$. The vertex v_1 is called an *upper endpoint* and v_k is called a *lower endpoint*.

Note A vertical path is a less restrictive definition than that of a geodesic, a shortest path between 2 endpoints, because it does not enforce the shortest condition within the graph; the vertical path just has to be a "downward" path in T from a high to low layer, with respect to the root r. Philipczuk and Sibertz showed a weaker version of our main lemma, where they showed G has a partition P of geodesics instead of vertical paths. [7]

Example 4.2 Examples of a geodesic and a vertical path.



let the highlighted edges be the spanning tree T rooted at a.

These are geodesics.

- ace, cf, abd are all vertical paths.

- However, abd is not a geodesic as ad∈E(G) is a shorter path from a to d.

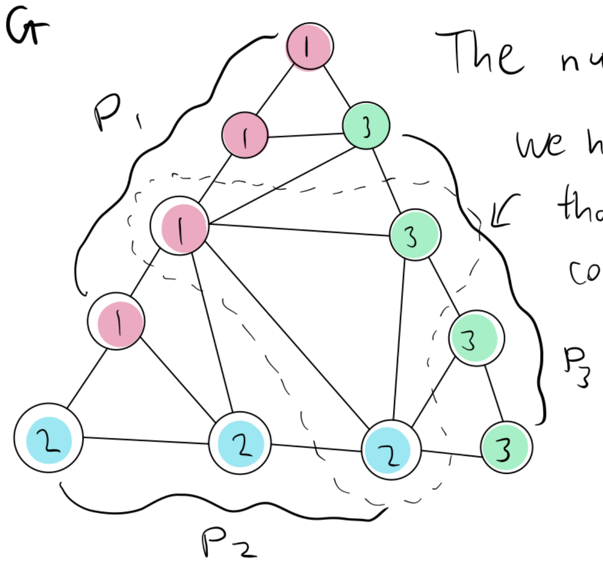
Note that in a BFS tree, all vertical paths are geodesics because paths like ad cannot exist, as it will first be traversed from a to create the BFS tree.

A vertical path is vertical in the sense that from the upper endpoint to the lower endpoint, it keeps getting farther away from the root with respect to T.

Lemma 4.3 (Sperner's Lemma - A 2D variant)
Given a set of colors $\{1, 2, 3\}$. Let G be a triangulated disc, where the outer-face is bounded by the cycle induced by P_1, P_2, P_3, where each P_i admits color i. Then, there must exist an inner face of G where it uses all 3 colors in its vertices.

Note It is not too tricky to prove this, but the proof I know of, for a close variant of Sperner's lemma, utilizes results in game theory, which is not in the spirit of this lecture note.

Example 4.4 Example of Sperner's Lemma with the Sperner's coloring and the inner face that utilizes all 3 colors.



The number denotes the colouring.

We have an inner triangle that utilizes all three colours on its vertices.

The next lemma is central to the proof of bounded queue-number for planar graphs.

Lemma 4.5 Let G^+ be a triangulation and T be a spanning tree of G^+ rooted at r, where r is on the outerface of G^+. Let $P_1, \cdots, P_k$ be vertical paths of T, with $k \in \{1, \cdots, 6\}$, such that $P_1 \cdots P_k$ forms a cycle C in G^+.

Let G be the triangulated disc induced by vertices of C and in the interior of C. Then, G has a partition $\mathcal{P}$ of vertical paths with $\{P_1 \cdots P_k\} \subseteq \mathcal{P}$, and that $G/\mathcal{P}$ has treewidth at most 8 where there exists some bag in $G/\mathcal{P}$ containing all the vertices corresponding to $P_1 \cdots P_k$.

Notation 4.6 Let P be a path. We use p^i for the i^{th} vertex on the path. Specifically, we denote p^s as the first vertex and p^l as the last vertex.

Proof. The main idea is to do induction on the number of vertices in a triangulated disc, exploit some structures in its subgraphs (also triangulated disc), and partition G into vertical paths with respect to some spanning tree T. The inductive proof consists of 3 main parts:

1. Reduce from a triangulated disc of n vertices to a few smaller triangulated disc subgraphs, on which we apply the inductive hypothesis, through Sperner's Lemma. From this, we create the desired partition of G.

2. Construct a tree decomposition from the subtree decompositions by induction.

3. Verify that this decomposition is indeed a valid tree decomposition.

We will only explore steps 1 and 2, as the verification step is relatively algorithmic and does not provide much new insight.

We will prove by induction on $|V(G)|$. Our base case is $|V(G)| = 3$, which is simple. Note that a single vertex of G is a vertical path (a singleton path) in T. Let $\mathcal{P}$ be distinct vertices of G, then $G/\mathcal{P}$ has a tree decomposition of just a bag with all 3 vertices, so $tw(G/\mathcal{P}) \leq 8$. Now, let's assume $|V(G)| > 3$.

Step 1: Assume a clockwise orientation on the cycle, such that traveling in the clockwise direction on $P_1 \cdots P_k$ gives us the cycle C. We want to define 3 paths R_1, R_2, R_3 that forms the cycle boundary. In particular,

- If $k = 1$, and assume $P_1 = p_1^s P_1' p_1^l$, we define $R_1 = p_1^s$, $R_2 = p_1'$, $R_3 = p_1^l$.

- If $k = 2$, and assume $_1 = p_1^s P_1'$, we define $R_1 = p_1^s$, $R_2 = P_1'$, $R_3 = P_2$.

- If $k \in \{3, 4, 5, 6\}$, we define $R_i = P_i \cdots P_{\lfloor ik/3 \rfloor}$.

Essentially, we combine P_i if we have too many (≥ 3) and split P_i if there is not enough. Let G admit a Sperner's coloring by coloring vertices on R_i with color i. Then, we will color the other vertices in G as follows. (See Figure 3)

- Observe that r lies on or in the exterior of C. For any vertex v in the interior of C, namely $v \in D$ that is not on C, we have a path, say P, from r to v that is in T since T is spanning.

- Since P connects r to v, it must use a vertex of C somewhere, say c, because r is on or outside of C and v is in the interior of C. Then, we color v with color i where some vertex of R_i is the first vertex P intersects.

Then, by Sperner's lemma (Lemma 4.3), some inner face $\Delta = \{t_1, t_2, t_3\}$ must use all 3 colors in its vertices. We assume WLOG that t_i admits color i.

Let Q_i be the path in the tree T from t_i to R_i, and let p_i be the vertex at which it touches R_i. This is possible because from the coloring, we know that t_i is colored i only if it crosses R_i when traveling from the root. Let Q_i' be the truncated path of Q_i, removing the vertex p_i. WLOG, we assume a clockwise orientation on the cycle. Let R_i^+ be the subpath in R_i that comes before p_i, since p_i touches R_i, and R_i^- be the path that comes after. Then, consider $C_i = R_i^+ Q_i' Q_{i+1}' R_{i+1}^-$ for all $i \in \{1, 2, 3\}$ (with $R_4 = R_1$). The C_i's that are not cycles are called "degenerate", and we won't consider them. In particular, they happen when $Q_{i+1},' Q_i'$ are empty and R_{i+1}^-, R_i^+ are both single vertices.

For each cycle C_i ($i \in \{1, 2, 3\}$ assuming no "degenerate" cycles), define its induced subgraph as D_i. Note that each D_i is a triangulated disc and uses at most $n - 1$ vertices because C_i captures only 2 of the 3 vertices on Δ. (See Figure 4) Recall that

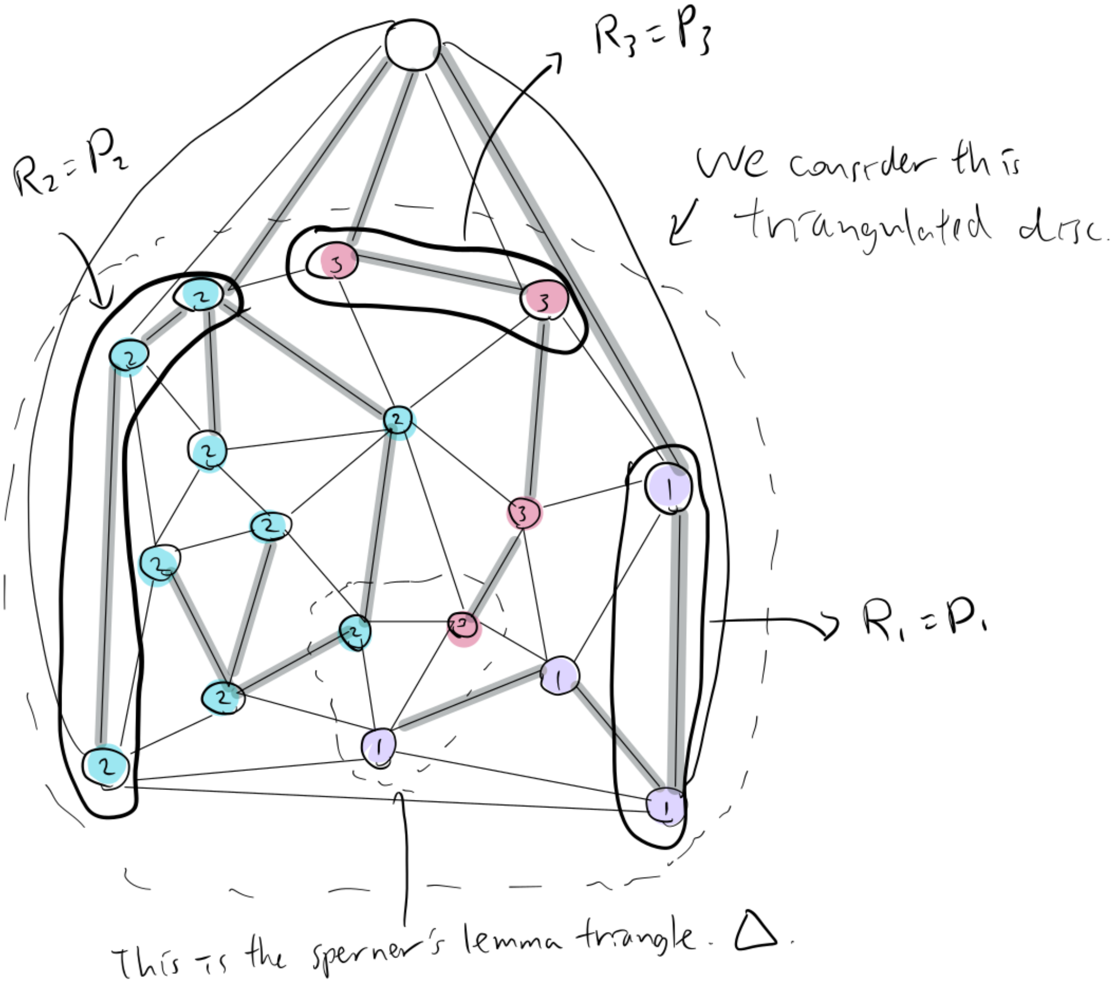


Figure 3: Example of the triangulated disc and its Sperner's coloring

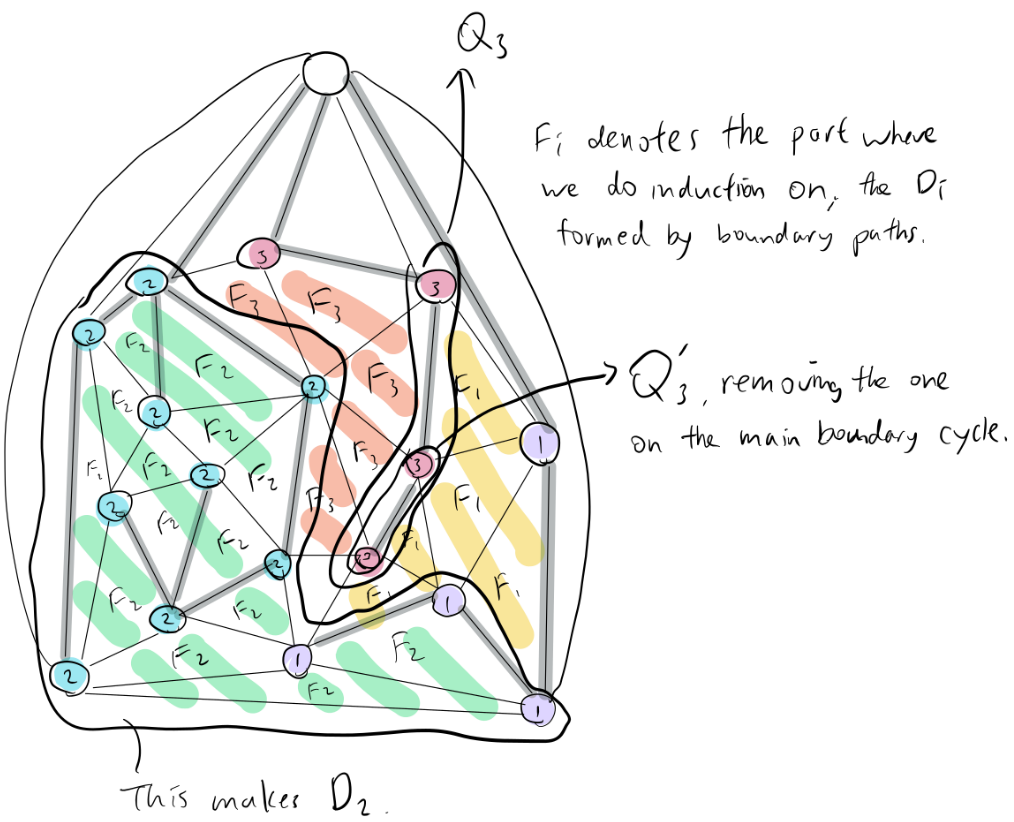

Q_3
F_i denotes the part where we do induction on, the D_i formed by boundary paths.
Q_3', removing the one on the main boundary cycle.
This makes D_2.
F_1
F_2
F_3
1
2
3

Figure 4: Example of forming the smaller triangulated discs for induction

$C_i = R_i^+ Q_i' Q_{i+1}' R_{i+1}^-$, and by construction, each R_i is joined by at most 2 vertical paths (i.e. the case when $k > 3$). Thus, the cycle C_i can be re-represented with at most 6 vertical paths, by replacing R_i and R_{i+1} with the respective P_i's. This means, D_i is a triangulated disc bounded by a cycle generated from 6 vertical paths of T. Thus, we can apply the inductive hypothesis on D_i. D_i has a partition $\mathcal{P}_{D_i}$ of vertical paths such that $D_i/\mathcal{P}_{D_i}$ has a tree decomposition of bag size at most 9, where we denote the underlying tree J_i. Also, $D_i/\mathcal{P}_{D_i}$ has a bag that contains all the vertical paths that forms the boundary of D_i, which we will call boundary paths. Denote these bags as $(B_x^i : x \in V(J_i))$. Let the bag $B_{u_i}^i$ be the one that contains all of those (at most) 6 boundary paths. We perform this for all non degenerate C_i.

For all D_i's that we have generated, we will use them to define the desired partition $\mathcal{P}$ for G. First, we initialize $\mathcal{P}$ with all the initial vertical paths $P_1, \cdots, P_k$ that defines $\mathcal{P}$. Then, we add all the Q_i' to $\mathcal{P}$. Note that for each of the D_i, by our inductive hypothesis, the outer boundary of D_i is generated by 6 vertical paths. Thus, the partition $\mathcal{P}_{D_i}$ can be considered as 2 separate classes: the boundary paths defining D_i, and the paths that are strictly contained within C_i, which we call internal paths. Then, for all internal paths of D_i, we simply add it to $\mathcal{P}$. This is valid partition for G because the internal paths do not share any vertices with $P_1, \cdots, P_k, Q_1', Q_2', Q_3'$.

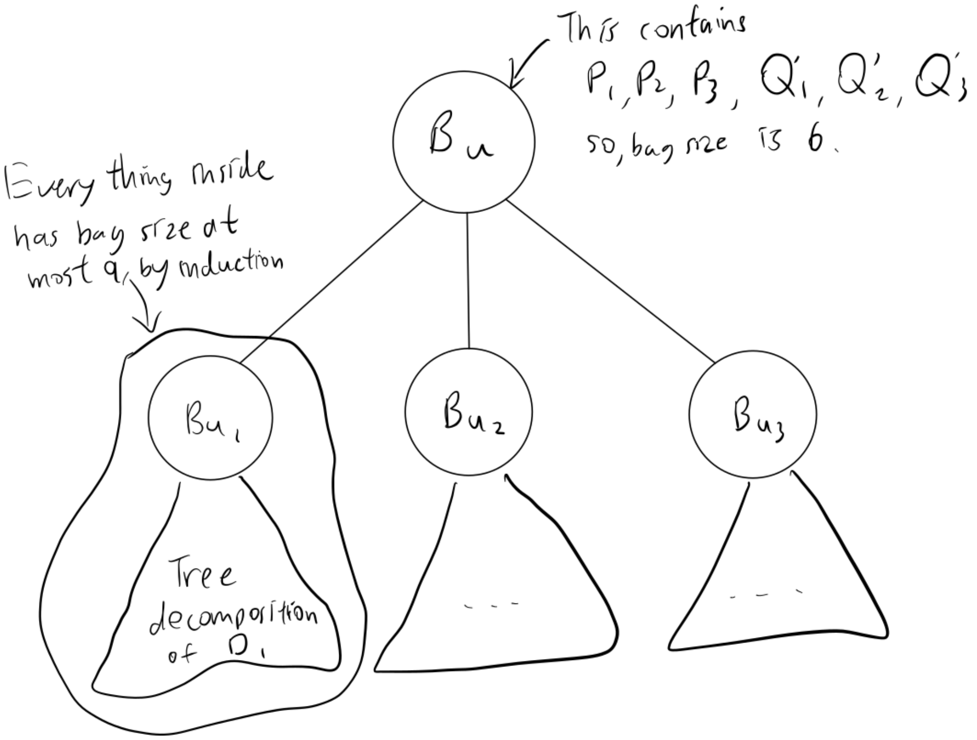

This contains
$P_1, P_2, P_3, Q'_1, Q'_2, Q'_3$
so, bag size is 6.
Everything inside has bag size at most 9, by induction
B_u
B_{u_1}
B_{u_2}
B_{u_3}
Tree decomposition of D_1
...
...

Figure 5: Example of the structure of tree J

So, all paths in $\mathcal{P}$ are indeed vertical paths of T.

Step 2: Now, we construct the tree-decomposition for $G/\mathcal{P}$. Recall that $D_i/\mathcal{P}_{D_i}$ has a J_i-decomposition, and that $B^i_{u_i}$ contains all the vertical paths of D_i. For simplicity sake, say we have J_1, J_2, J_3, implying that all D_i are valid, and suppose, WLOG, that all J_i are disjoint. We construct the tree J as follows, which will serve as the desired J-decomposition for G/P. (See Figure 5)

- $V(J) = \{u\} \cup (\bigcup_{i=\{1,2,3\}} V(J_i))$
- $E(J) = \{uu_1, uu_2, uu_3\} \cup (\bigcup_{i=\{1,2,3\}} E(J_i))$

. For each vertex of J, we initialize the bag B_x as follows

- For $x \neq u$ and $x \in J_i$, $B_x = B^i_x$.
- For $x = u$, B_x consists of all vertices corresponding to $P_1, \cdots, P_k$ and Q'_1, Q'_2, Q'_3.

However, it is not yet a valid tree decomposition because some vertices in B^i_x are not actually vertices of $G/\mathcal{P}$. Recall that B^i_x contains vertices corresponding to vertical path partitions of D_i. From step 1, we reasoned that the partition D_i is separated into 2 classes: boundary paths and internal paths. For internal paths, we know that they are actually in the partition $\mathcal{P}$, by construction, but things are trickier for boundary paths. For each boundary paths of D_i, each of them is a either a sub-path of P_i for some $i \in \{1, \cdots, k\}$ or of Q'_j for some $j \in \{1, 2, 3\}$. This is because, by construction, the boundary paths of D_i are either Q'_i, Q'_{i+1} or two sub-paths of $P_1, \cdots, P_k$ from splitting R_i. Now, for each vertex in B_x that are boundary paths of D_i, we replace it with the corresponding vertex in G/P that denotes the corresponding vertical paths $Q'_1, Q'_2, Q'_3, P_1, \cdots, P_k$. Since we are just replacing 1 vertex with another, not introducing any new vertices into any bag, the maximum bag size of the whole J-decomposition is still 9. Thus, the treewidth is at most 8.

Step 3: We have to check that it is indeed a tree decomposition. Main point to note here is that when it concerns the vertices that correspond to those replaced paths (i.e. the boundary paths of D_i), the bag B_u serves as a bridge between different subtrees (the J_i). Then, by induction, the bags that contain those vertices are connected as a subtree within J_i. The remaining details can be found in reference [3]. □

The lemma (Lemma 4.5) we proved above is a stronger statement than we need for proving bounded queue-number. Below, we will provide the main idea on how to apply a special case of it to prove this theorem.

Theorem 4.7 Let G be planar. Then G has an H-partition of layered width 1, where H has treewidth at most 8.

Proof. Let G^+ be the planar triangulation of G. Let T be a BFS spanning tree rooted at one of the vertices on the outerface. Define the layering to be the BFS layering with respect to T. So, by Lemma 4.5, it has a partition P of paths vertical in T where G^+/P has treewidth at most 8, and since G/P is a subgraph of G^+/P, the treewidth of $G/P \leq 8$. Let P_v be any path in P. Each step in P_v goes down a layer from V_i to V_{i+1}, so $|P_v \cap V_i| \leq 1$ for all layers V_i, and so, P has layered width 1. Let H be G/P where each vertex of H corresponds to a bag of vertical paths in P, then we have the desired H-partition. □

Corollary 4.8 Every planar graph G has $qn(G) \leq 766$.

Proof. By applying Corollary 3.10 on Theorem 4.7, we have $qn(G) \leq 3(1)(2^8 - 1) + \lfloor \frac{3}{2}(1) \rfloor = 766$ as desired. □

5 Reducing the upper bound

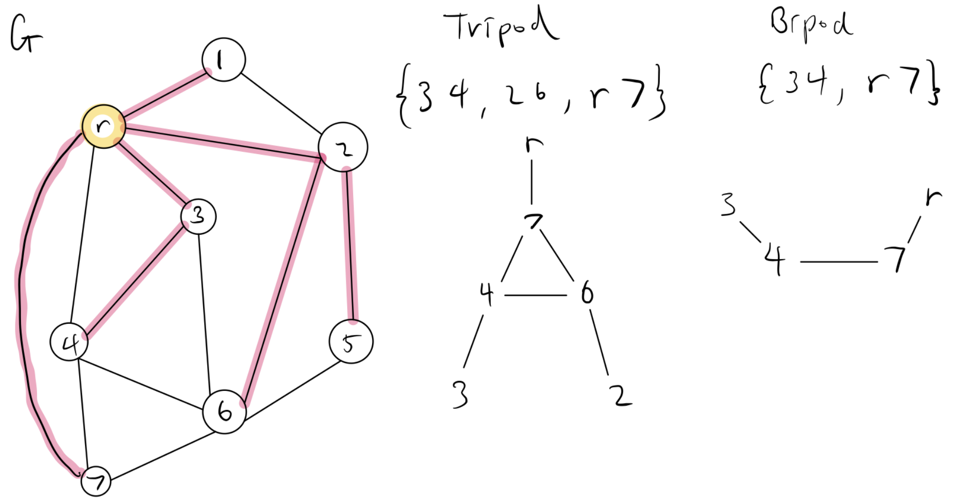
Now that we have successfully shown that planar graphs have bounded queue-number, the natural next step is to see if we can improve the bound. The answer to this is "Yes, we can!". In fact, we can improve the bound from 766 to 49 with only slight modifications to each of the proofs in the above section and a new lemma. In particular, instead of looking for individual vertical paths in G, we look for tripods (see Definition 5.1) in G, and the proofs will be similar to the ones above.

Definition 5.1 (Tripod and Bipod)
Given G and a spanning tree T. A *tripod* is a set of at most 3 pairwise disjoint vertical paths where the lower endpoints form a clique. A *bipod* is a specific case in which we have 2 disjoint vertical paths where the lower endpoints are connected.

Remark 5.2 (Some notes on tripod)

- A vertical path of length ≥ 2 is a bipod simply by splitting into 2 vertical paths; one being the lower endpoint as a singleton path, and the other being the remaining path.
- A bipod is a tripod

Example 5.3 Example of a tripod and a bipod in a graph G with spanning tree T that is rooted at r.

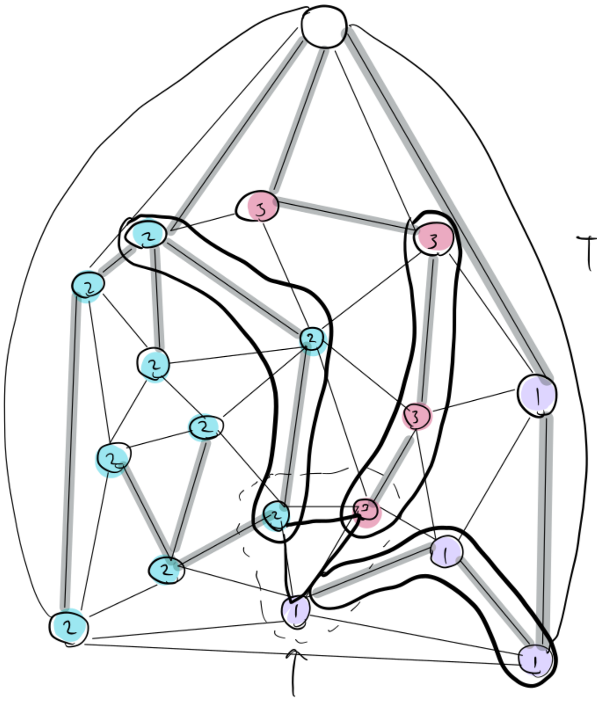


Before we can prove the upper bound of 49, we need to introduce this lemma. This will be used later when our new partition P we get from tripods yields $tw(G/P) \leq 3$. If we simply apply Corollary 3.10, we will have a bigger upper bound of $3(3)(2^3 - 1) + \lfloor \frac{3}{2}(3) \rfloor = 67$ instead of 49.

Lemma 5.4 (Alam et al.)
Let G be planar with treewidth at most 3. Then $qn(G) \leq 5$. [1]

Lemma 5.5 Let G^+ be a triangulation and T be a spanning tree of G^+ rooted at r, where r is on the outerface of G^+. Let $P_1, \cdots, P_k$ be pairwise disjoint bipods of T, with $k \in \{1, \cdots, 3\}$, such that $P_1 \cdots P_k$ form a cycle C in G^+ with r outside of C.

Let G be the triangulated disc induced by vertices of C and in the interior of C. Then, G has a partition P into tripods with $P_1 \cdots P_k$ being inside P, and G/P is planar with treewidth at most 3, where there exists some bag in G/P containing all the vertices corresponding to $P_1 \cdots P_k$.



The three circled
path forms a tripod,
and the lower endpoints
form a clique since it
is a triangle, in particular,
it is Δ.

Δ. sperner's lemma
triangle

Figure 6: Example of the tripod formed by Sperner's triangle

We will not be going through the full proof here, as the main idea is very similar to the one above. However, in the proof above, we partition G into vertical paths. Here, we partition the cycle into bipods (and thus tripods). Then, like above, we use Sperner's Lemma to reduce the problem into smaller triangulated discs, which we can then apply the inductive hypothesis to. Of important note here is that the triangle from Sperner's Lemma and the paths Q_i' that go from P_i to the triangle form a tripod, which is useful for the construction of the partition P of G. (See Figure 6) Then, like we did above, we replace the boundary paths in the bags with the corresponding bipods partition in P. Finally, we show that G/P is indeed a valid tree decomposition.

Theorem 5.6 Let G be planar. Then G has an H-partition of layered width 3, where H has treewidth at most 3.

The proof of this is analogous to that of Theorem 4.7. We show that G/P has treewidth 3 and G has layered width 3, because any tripod T consists of at most 3 vertical paths, so $|T \cap V_i| \leq 3$ for all layers V_i, and then let H be G/P. With Theorem 5.6, Lemma 5.4, and a small variation of Corollary 3.10, we have $qn(G) \leq 3(3)(5) + \lfloor \frac{3}{2}(3) \rfloor = 49$, our current best upper bound for queue-number on planar graphs.

Corollary 5.7 Every planar graph G has $qn(G) \leq 49$.

Acknowledgements

Thanks to Prof. Therese Biedl for suggesting related papers and helpful feedbacks. Also, thanks to Rachel Lau, Steven Wong, and Queenie Chen for proofreading.

References

[1] Jawaherul Md. Alam, Michael A. Bekos, Martin Gronemann, Michael Kaufmann, and Sergey Pupyrev. Queue layouts of planar 3-trees. *Algorithmica*, 2020. doi: 10.1007/s00453-020-00697-4.

[2] Vida Dujmovic, Gwenael Joret, Piotr Micek, Pat Morin, Torsten Ueckerdt, and David Wood. Planar graphs have bounded queue-number. *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, 2019. doi: 10.1109/focs.2019.00056.

[3] Vida Dujmović, Gwenaël Joret, Piotr Micek, Pat Morin, Torsten Ueckerdt, and David R. Wood. Planar graphs have bounded queue-number, 2019. URL `https://arxiv.org/abs/1904.04791`.

[4] Vida Dujmović, Louis Esperet, Gwenaël Joret, Bartosz Walczak, and David Wood. Planar graphs have bounded nonrepetitive chromatic number. *Advances in Combinatorics*, 2020. doi: 10.19086/aic.12100.

[5] Lenwood S. Heath and Arnold L. Rosenberg. Laying out graphs using queues. *SIAM Journal on Computing*, 21(5):927–958, 1992. doi: 10.1137/0221055.

[6] Lenwood S. Heath, Frank Thomson Leighton, and Arnold L. Rosenberg. Comparing queues and stacks as machines for laying out graphs. *SIAM Journal on Discrete Mathematics*, 5(3):398–412, 1992. doi: 10.1137/0405031.

[7] Michał Pilipczuk and Sebastian Siebertz. Polynomial bounds for centered colorings on proper minor-closed graph classes. *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, page 1501–1520, 2019. doi: 10.1137/1.9781611975482.91.

[8] Veit Wiechert. On the queue-number of graphs with bounded tree-width. *The Electronic Journal of Combinatorics*, 24(1), 2017. doi: 10.37236/6429.

Reference [3] is the main reference for this lecture note.