

## The Unit Bar Visibility Number of a Graph

Emily Gaub<sup>1</sup> Michelle Rose<sup>2</sup> Paul S. Wenger<sup>3</sup>

<sup>1</sup>Pacific University, Forest Grove, OR 97116

<sup>2</sup>Mount St. Mary's University, Emmitsburg, MD 21727

<sup>3</sup>School of Mathematical Sciences, Rochester Institute of Technology,  
Rochester, NY 14623

### Abstract

A  $t$ -unit-bar representation of a graph  $G$  is an assignment of sets of at most  $t$  horizontal unit-length segments in the plane to the vertices of  $G$  so that (1) all of the segments are pairwise nonintersecting, and (2) two vertices  $x$  and  $y$  are adjacent if and only if there is a vertical channel of positive width connecting a segment assigned to  $x$  and a segment assigned to  $y$  that intersects no other segment. The *unit bar visibility number* of a graph  $G$ , denoted  $ub(G)$ , is the minimum  $t$  such that  $G$  has a  $t$ -unit-bar visibility representation. Our results include a linear time algorithm that determines  $ub(T)$  when  $T$  is a tree, bounds on  $ub(K_{m,n})$  that determine  $ub(K_{m,n})$  asymptotically when  $n$  and  $m$  are asymptotically equal, and bounds on  $ub(K_n)$  that determine  $ub(K_n)$  exactly when  $n \equiv 1, 2 \pmod{6}$ .

Submitted: September 2015	Reviewed: December 2015	Revised: February 2016	Accepted: March 2016	Final: March 2016
Published: April 2016				
Article type: Regular paper		Communicated by: W. S. Evans		

Research supported by REU in Extremal Graph Theory and Dynamical Systems – NSF award #1358583.

*E-mail addresses:* gaub2680@pacificu.edu (Emily Gaub) MMRose1@email.msmary.edu (Michelle Rose) pswsma@rit.edu (Paul S. Wenger)

## 1 Introduction

Motivated by the challenges of very-large-scale integration (VLSI), significant study has focused on graphs with representations having nice geometric descriptions. A *visibility representation* of a graph  $G$  consists of an assignment of pairwise disjoint geometric objects (typical examples include segments, Cartesian products of segments, or spheres) to the vertices of  $G$  so that two vertices are adjacent if and only if there is an uninterrupted “line of sight” (generally a line segment or channel of positive width) joining the objects assigned to those vertices.

A *bar visibility representation* of a graph  $G$  is an assignment of distinct horizontal line segments (henceforth called *bars*) in the plane to the vertices of  $G$  so that two vertices are adjacent if and only if there is an uninterrupted channel of positive width that joins the bars corresponding to those vertices. A graph with a bar visibility representation is a *bar visibility graph*. Tamassia and Tollis [11] and Wismath [14] characterized bar visibility graphs: a graph  $G$  is a bar visibility graph if and only if it is planar and there is a planar embedding of  $G$  such that all cut-vertices appear on the same face.

In a bar visibility representation of a graph, vertices are allowed to have horizontal bars of arbitrary length. From the viewpoint of VLSI design, it is reasonable to assume that the size of components in a circuit design have roughly equal size. A graph is a *unit bar visibility graph* if it has a bar visibility representation in which every bar has the same length. Such a representation is called a *unit bar visibility representation*. For simplicity, throughout the paper we assume that all unit bars have length 1.

Dean and Veytsel [6] introduced unit bar visibility graphs and characterized the trees, complete bipartite graphs, and complete graphs that are unit bar visibility graphs. Wiglesworth [13] continued the study of unit bar visibility graphs, characterizing unit-bar visibility graphs that have representations with width at most 2 (that is, the projection of all bars onto the  $x$ -axis has length at most 2). Wiglesworth also proved other structural relations between a unit bar visibility graph and its unit bar visibility representations. To date, there is no characterization of unit bar visibility graphs.

The family of bar visibility graphs is quite restrictive. Chang et al. [4] introduced a generalization of bar visibility representations that captures all graphs. A  $t$ -bar is the union of (at most)  $t$  horizontal bars in the plane. A graph  $G$  has a  $t$ -bar visibility representation if there is an assignment of  $t$ -bars to the vertices of  $G$  such that two vertices  $u$  and  $v$  are adjacent in  $G$  if and only if there is an uninterrupted vertical channel of positive width joining a bar assigned to  $u$  to a bar assigned to  $v$  (see Figure 1). It is possible that two bars assigned to the same vertex might see each other, and we suppress such loops; this situation does not arise in the proofs in this paper. The *bar visibility number* of  $G$ , denoted  $b(G)$ , is the minimum  $t$  such that  $G$  has a  $t$ -bar visibility representation. Chang et al. [4] proved that  $b(G) \leq 2$  if  $G$  is planar, determined the bar visibility number of complete bipartite graphs within 1, proved that  $b(K_n) = \lceil n/6 \rceil$ , and proved that  $b(G) \leq \lceil |V(G)|/6 \rceil + 2$  for all graphs. The

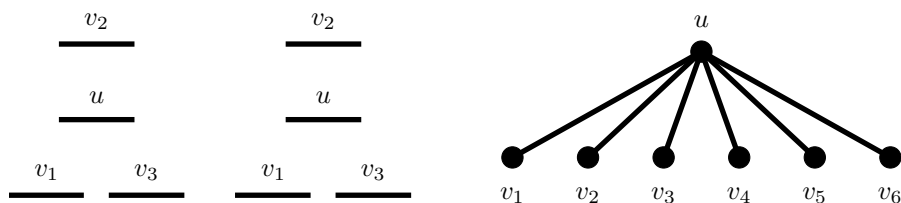


Figure 1: A 2-unit-bar representation of  $K_{1,6}$ . Since  $K_{1,6}$  is not a unit rectangle visibility graph, we conclude that  $ub(K_{1,6}) = 2$ .

notion of bar visibility numbers has since been extended to directed graphs [2].

In this paper we study  $t$ -bar visibility representations of graphs in which every bar (that is, every individual bar in every  $t$ -bar) has the same length, which we can assume to be 1. We call such a representation a  $t$ -unit-bar visibility representation. The unit bar visibility number of  $G$ , denoted  $ub(G)$ , is the minimum  $t$  such that  $G$  has a  $t$ -unit-bar visibility representation. A  $t$ -unit-bar visibility representation of  $G$  with  $t = ub(G)$  is called *optimal*.

By assigning the edges of a graph disjoint intervals of length 1 on the  $x$ -axis and giving vertices  $u$  and  $v$  bars that project onto the interval for the edge  $uv$ , it is clear that  $ub(G) \leq \Delta(G)$  for all  $G$ , where  $\Delta(G)$  denotes the maximum degree of  $G$ . Thus  $ub(G)$  is well-defined. Since a  $t$ -unit-bar visibility representation of  $G$  is also a  $t$ -bar visibility representation of  $G$ , it follows that  $b(T) \leq ub(T)$ .

We study the unit bar visibility number of graphs in various families. In Section 3, we present a linear-time algorithm that determines the unit bar visibility number of trees and generates an optimal representation. In Section 4 we give bounds for the unit bar visibility number of complete bipartite graphs that are asymptotically tight when the partite sets are of asymptotically equal sizes. In Section 5 we study the unit bar visibility number of complete graphs, proving that  $\lceil \frac{n}{6} \rceil \leq ub(K_n) \leq \lceil \frac{n+4}{6} \rceil$ . Section 6 contains open questions and conjectures.

Throughout the paper, all graphs are finite. We let  $d_G(v)$  denote the degree of a vertex  $v$  in a graph  $G$ ; when the graph is clear, we simply write  $d(v)$ . Given a positive integer  $n$ , we let  $[n]$  denote the set  $\{1, \dots, n\}$ . We let  $K_n$  denote the complete graph on  $n$  vertices and let  $K_{m,n}$  denote the complete bipartite graph with partite sets of order  $m$  and  $n$ .

## 2 Preliminaries

When an arrangement of unit bars in the plane is given without it being identified as a representation of a specific graph, we refer to it as a unit bar visibility layout. When we refer to a bar in a  $t$ -unit-bar visibility representation or layout we are referring to one of the unit bars in one of the  $t$ -unit-bars. Since all bars are assumed to have length 1 we may describe each bar by the coordinates of its left endpoint; we denote the left endpoint of a bar  $b$  by  $(x_b, y_b)$ . If  $y_b < y_{b'}$

then we say that  $b$  is *below*  $b'$  and  $b'$  is *above*  $b$ . If  $x_b < x_{b'}$  then we say that  $b$  is *to the left of*  $b'$  and  $b'$  is *to the right of*  $b$ .

We say that two bars *see* each other if there is an uninterrupted vertical channel of positive width between the bars. When such a channel exists, we also say that the corresponding  $t$ -unit-bars and the corresponding vertices see each other. If two bars  $b$  and  $b'$  see each other and  $b$  is below  $b'$ , then we say that  $b$  *sees*  $b'$  *above* and that  $b'$  *sees*  $b$  *below*.

Let  $R$  be a  $t$ -unit-bar visibility layout. We define  $G(R)$  to be the unit bar visibility graph that is represented by  $R$  when all bars represent distinct vertices. Given a bar  $b$  in  $R$ , define  $\text{proj}(b)$  to be the projection of  $b$  onto the  $x$ -axis. Similarly, for a set of bars  $S$ , define  $\text{proj}(S)$  to be  $\bigcup_{b \in S} \text{proj}(b)$ . Two bars  $b$  and  $b'$  in  $R$  are said to *overlap* if  $\text{proj}(b)$  and  $\text{proj}(b')$  have an intersection of positive measure. The *components* of  $R$  are the sets of bars in  $R$  that correspond to the components of  $G(R)$ . Thus distinct components of  $R$  have disjoint projections onto the  $x$ -axis.

Let  $R$  and  $R'$  be two unit bar visibility layouts. A *disjoint union* of  $R$  and  $R'$  is the unit bar visibility layout obtained by arranging  $R$  and  $R'$  in the plane so that  $\text{proj}(R) \cap \text{proj}(R') = \emptyset$  and (without loss of generality)  $x_b < x_{b'}$  for all  $b \in R$  and  $b' \in R'$ . With this terminology, every unit bar visibility layout is a disjoint union of its components. For convenience we label the components of  $R$  as  $R_1, \dots, R_\ell$  so that for  $i < j$  we have that  $x_b < x_{b'}$  whenever  $b \in R_i$  and  $b' \in R_j$ .

**Lemma 1** *Every graph  $G$  has an optimal  $t$ -unit-bar visibility representation in which all bars have distinct  $y$ -coordinates.*

**Proof:** Let  $R$  be an optimal representation of  $G$  in which there are the fewest pairs of bars that share their  $y$ -coordinate. Suppose  $b$  and  $b'$  are two bars that share their  $y$ -coordinate. Since  $G$  is finite, there exists  $\epsilon > 0$  such that no bar has its  $y$ -coordinate in  $(y_b - \epsilon, y_b) \cup (y_b, y_b + \epsilon)$ . Increase the  $y$ -coordinate of  $b$  by  $\epsilon/2$  to obtain the  $t$ -unit-bar visibility layout  $R'$ . It is clear that  $R'$  is also a  $t$ -unit bar visibility representation of  $G$ , contradicting the minimality of  $R$ .  $\square$

### 3 Trees

A tree that is a unit bar visibility graph is a *unit bar visibility tree*; we abbreviate unit bar visibility tree as UBVT. In [6], Dean and Veytsel characterized unit bar visibility trees. A *caterpillar* is a tree in which all vertices with degree at least 2 lie on a single path.

**Theorem 1** (Dean-Veytsel [2003]) *A tree  $T$  is a UBVT if and only if  $\Delta(T) \leq 3$  and  $T$  is a subdivision of a caterpillar.*

We present a linear time algorithm that determines the unit bar visibility number of trees. With a slight modification, this algorithm will generate an optimal  $t$ -unit-bar visibility representation of a tree.

A *unit bar visibility forest* is a graph in which every component is a UBVT. Given a graph  $G$ , let the *unit bar visibility arboricity* of  $G$ , denoted  $\Upsilon_{ub}(G)$ , be the minimum number of unit bar visibility forests needed to decompose  $G$ . Note that a decomposition of a graph into  $k$  forests gives a decomposition of  $G$  into trees such that no vertex is in more than  $k$  of those trees. If  $T$  is a tree, then a decomposition of  $T$  into trees such that no vertex is in more than  $k$  of those trees also yields a decomposition of  $T$  into at most  $k$  forests. A decomposition of a tree  $T$  into UBVTs such that no vertex is in more than  $\Upsilon_{ub}(T)$  of the UBVTs will be called an *optimal* decomposition. As a first step towards giving an algorithm that determines  $ub(T)$  for all trees  $T$ , we prove that for all trees the unit bar visibility number and unit bar visibility arboricity are equal.

**Theorem 2** *If  $T$  is a tree, then  $ub(T) = \Upsilon_{ub}(T)$ .*

**Proof:** First suppose that  $\Upsilon_{ub}(T) = k$ , and let  $\{T_1, \dots, T_\ell\}$  be a decomposition of  $T$  into UBVTs such that each vertex in  $T$  is in at most  $k$  trees in the decomposition. A disjoint union of unit bar visibility representations of  $T_1, \dots, T_\ell$  is a  $k$ -unit-bar visibility representation of  $T$ , and therefore  $ub(T) \leq \Upsilon_{ub}(T)$ .

We now show that  $\Upsilon_{ub}(T) \leq ub(T)$ . By Lemma 1, we know that there is an optimal  $t$ -unit-bar visibility representation  $R$  of  $T$  such that every bar in  $R$  has a distinct  $y$ -coordinate. Among all such representations of  $T$ , let  $R$  be chosen so that it has the minimum number of pairs of bars  $\{b, b'\}$  such that  $b$  and  $b'$  correspond to the same vertex and lie in the same component of  $R$ .

For each  $v \in V(T)$ , let  $b_1(v), \dots, b_t(v)$  be the  $t$  bars assigned to  $v$ . First assume that  $b_i(v)$  and  $b_j(v)$  are in distinct components of  $R$  for all  $v \in V(T)$  and all  $i, j \in [t]$  such that  $i \neq j$ . Let  $R_1$  be the first component of  $R$ . Since the labels on the bars in  $R_1$  are distinct, the labels induce an isomorphism from  $G(R_1)$  to the subgraph of  $T$  induced by the vertices assigned to the bars in  $R_1$ . Therefore  $G(R_1)$  is a tree. Removing  $R_1$  from  $R$  and applying induction then shows that  $R$  is a disjoint union of unit bar visibility representations of trees. Therefore  $\Upsilon_{ub}(T) \leq t$ .

Now let  $\{b(v), b'(v)\}$  be a pair of bars that are assigned to the same vertex and lie in the same component of  $R$ . Without loss of generality, assume that  $y_{b(v)} < y_{b'(v)}$ . Fix a value  $z$  such that  $y_{b(v)} < z < y_{b'(v)}$  and  $z$  is not the  $y$ -coordinate of any bar in  $R$ . Place the horizontal line  $y = z$  through  $R$ . This immediately partitions  $R$  into two sets:  $A$ , the set of bars above the line  $y = z$ ; and  $B$ , the set of bars below the line  $y = z$ . Further partition  $A$  into  $A_1, \dots, A_\ell$ , the components of the unit bar visibility layout consisting of just the bars in  $A$ . Similarly, partition  $B$  into  $B_1, \dots, B_{\ell'}$ , the components of the layout consisting of just the bars in  $B$ . Observe that  $\mathcal{P} = \{A_1, \dots, A_\ell, B_1, \dots, B_{\ell'}\}$  is a partition of  $R$  (see the top left picture in Figure 2).

Form an auxiliary graph  $H$  with  $V(H) = \mathcal{P}$  where two elements of  $\mathcal{P}$  are adjacent if bars from those elements can see each other in  $R$  (see the top right picture in Figure 2). It follows that  $H$  is a bipartite interval graph. Hence  $H$  is a forest with a bipartition corresponding to the partition of  $R$  into  $A$  and  $B$ . Let  $B_i$  be the set that contains  $b(v)$  and let  $A_{i'}$  be the set that contains  $b'(v)$ .

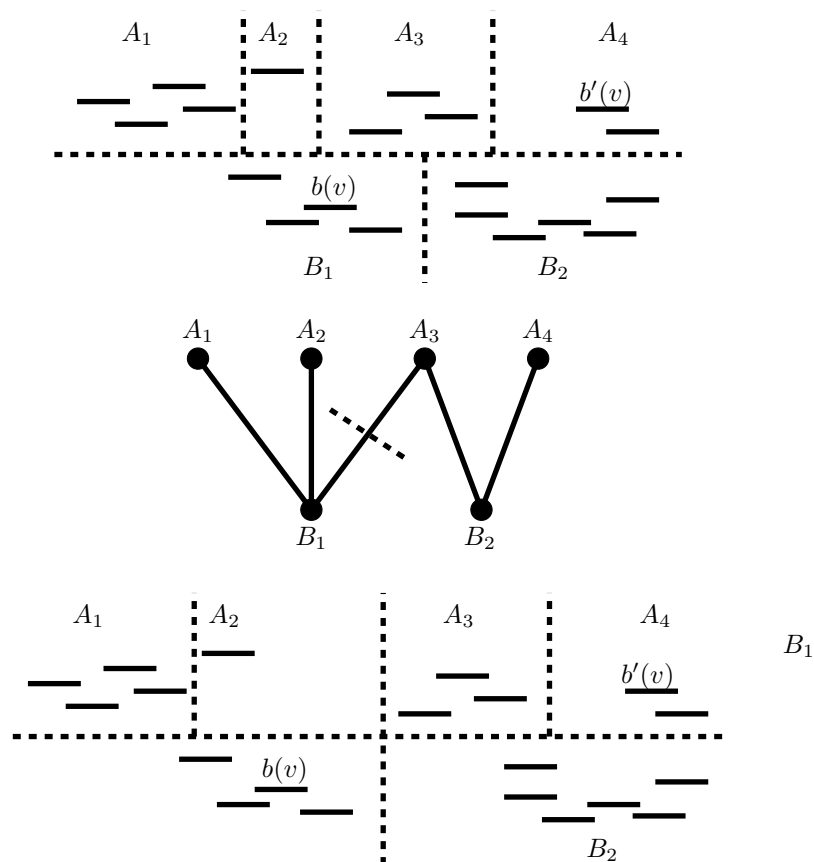


Figure 2: Top: A representation of a tree  $T$  with two bars assigned to  $v$  in the same component, along with a partition of the representation. Middle: The auxiliary bipartite graph. Bottom: A new representation of  $T$  with  $b(v)$  and  $b'(v)$  in different components.

Let  $e = B_i A_j$  be the first edge on the path in  $H$  from  $B_i$  to  $A_j$ . Let  $R'$  be the union of the elements of  $\mathcal{P}$  in the component of  $H - e$  that contains  $A_j$ . Finally, let  $S$  be a  $t$ -unit-bar visibility layout obtained from a disjoint union of  $R \setminus R'$  and  $R'$ . You can picture  $S$  as being obtained by “sliding” the bars in  $R'$  to the right of all other bars in  $R$  (see the bottom picture in Figure 2).

We claim that  $S$  is a  $t$ -unit-bar visibility representation of  $T$ , contradicting the minimality of  $R$  since  $b(v)$  and  $b'(v)$  are no longer in the same component. It is clear that there are no bars that see each other in  $S$  that did not see each other in  $R$ . Also there are no pairs of bars assigned to the same vertex lying in the same component of  $S$  that do not also lie in the same component of  $R$ . Therefore it remains to show that every edge in  $T$  is a visibility in  $S$ . If two bars  $b(w)$  and  $b(w')$  can see each other in  $R$  but cannot see each other in  $S$ , then without loss of generality we can assume that  $b(w) \in B_i$  and  $b(w') \in A_j$ . Both  $b(w)$  and  $b(w')$  lie in components of  $S$  that contain bars corresponding to  $v$ . Since each component is connected, in the component of  $b(w)$  there is a sequence of bars from  $b(w)$  to a bar of  $v$  such that consecutive bars see each other, and in the component of  $b(w')$  there is a similar sequence of bars from  $b(w')$  to a bar of  $v$ . Therefore there is a component of  $S$  that contains a  $w, v$ -walk in  $T$  and a component that contains a  $w', v$ -walk in  $T$ . One of these walks must contain the edge  $ww'$ , so the corresponding component of  $S$  must contain the edge  $ww'$ . Therefore  $S$  contains every edge of  $T$ , contradicting the minimality of  $R$ .  $\square$

An immediate corollary of Theorems 1 and 2 is the following lower bound on the unit bar visibility number of a tree.

**Corollary 3** *If  $T$  is a tree, then  $ub(T) \geq \lceil \Delta(T)/3 \rceil$ .*

**Proof:** Each UBVT has maximum degree at most 3, and therefore a vertex in  $T$  of degree  $d$  must be in at least  $\lceil d/3 \rceil$  elements of a decomposition of  $T$  into UBVTs.  $\square$

We now prove our first upper bound on the unit bar visibility number of trees.

**Theorem 4** *If  $T$  is a tree, then there is a decomposition of  $T$  into UBVTs such that each vertex  $v \in V(G)$  is in at most  $\lceil \frac{d(v)+1}{3} \rceil$  elements of the decomposition. Therefore  $ub(T) \leq \lceil \frac{\Delta(T)+1}{3} \rceil$ .*

**Proof:** We proceed by induction on the number of vertices in  $T$ . For the base case, we decompose  $K_{1,n}$  into  $\lfloor n/3 \rfloor$  copies of  $K_{1,3}$  and, if  $n$  is not divisible by 3, one copy of  $K_{1,r}$  where  $1 \leq r \leq 2$  and  $r \equiv n \pmod{3}$ .

Now assume that  $T$  is not a star and let  $v$  be a vertex in  $T$  with exactly one neighbor that is not a leaf. If  $d(v) \geq 4$ , then let  $T'$  be obtained by deleting three neighbors of  $v$  that are leaves. The addition of a copy of  $K_{1,3}$  to the decomposition of  $T'$  from the inductive hypothesis yields the desired decomposition of

$T$ . If  $d(v) = 3$ , then let  $T'$  be obtained by deleting the two neighbors of  $v$  that are leaves. The addition of a copy of  $K_{1,2}$  to the decomposition of  $T'$  from the inductive hypothesis yields the desired decomposition of  $T$ . If  $d(v) = 2$ , then let  $v'$  be the leaf neighbor of  $v$  and let  $T' = T - v'$ . By induction,  $T'$  has a decomposition into UBVTs where  $v$  lies in exactly one UBVT. By adding  $v'$  and the edge  $vv'$  to the element of the decomposition that contains  $v$ , we obtain the desired decomposition of  $T$ .  $\square$

Corollary 3 and Theorem 4 actually determine the unit bar visibility number of any tree whose maximum degree is not a multiple of 3. However, both Corollary 3 and Theorem 4 can be sharp when  $\Delta(T)$  is a multiple of 3. As a simple example, given a tree  $T$  with maximum degree 3,  $ub(T) = 1$  if  $T$  is a subdivision of a caterpillar and  $ub(T) = 2$  otherwise. While it is not difficult to show that both bounds are also sharp for larger maximum degrees that are multiples of 3, we choose to omit such constructions for brevity.

We now present a linear-time algorithm called `UNIT_BAR_TREE` that determines if  $ub(T) = \lceil \Delta(T)/3 \rceil$ . The algorithm decomposes trees into UBVTs, and in Theorem 6 we prove that each vertex is in at most  $\Upsilon_{ub}(T)$  elements in the decomposition. Therefore, by Theorem 2, the algorithm determines  $ub(T)$ .

Given a tree  $T$ , `UNIT_BAR_TREE`( $T$ ) selects a root of  $T$  and performs a post-order traversal of  $T$ , calling two subroutines, called `PRUNE` and `COLOR`, at each vertex. When `PRUNE` is called at a vertex  $v$  that is not the root, a maximum of  $\lceil \frac{\Delta(T)}{3} \rceil - 1$  UBVTs containing  $v$  and its descendants are added to the decomposition of  $T$ . When `PRUNE` is called at the root, up to  $\lceil \frac{\Delta(T)}{3} \rceil$  UBVTs containing the root may be added to the decomposition. After `PRUNE` is called at  $v$ , the algorithm calls `COLOR` at  $v$ . If  $v$  is not the root, `COLOR` either assigns a color to the edge joining  $v$  and its parent, or halts and declares  $ub(T) = \lceil \frac{\Delta(T)+1}{3} \rceil$ . If  $v$  is the root, `COLOR` declares  $ub(T) = \lceil \frac{\Delta(T)}{3} \rceil$  or  $ub(T) = \lceil \frac{\Delta(T)+1}{3} \rceil$ . The decision to assign a particular color or halt depends on the complexity of the portion of the tree that remains below  $v$ .

We now describe the `PRUNE` routine in detail. When `PRUNE` is called at  $v$ , `COLOR` has already been called at all children of  $v$  without halting. Consequently each edge joining  $v$  to a child has been colored green, yellow, or red (as explained in the `COLOR` routine below). Let  $T'$  be the tree that remains when `PRUNE` is called at  $v$ . The `PRUNE` routine selects subtrees of  $T'$  rooted at  $v$  and removes their edges and vertices (excluding  $v$ ) from  $T'$ ; we say that these subtrees are *pruned* from  $T'$ . If  $v$  is not the root of  $T$ , then `PRUNE` stops when  $v$  has no remaining children or  $\lceil \frac{\Delta(T)}{3} \rceil - 1$  trees have been pruned at  $v$ . If  $v$  is the root, then `PRUNE` stops when  $v$  has no remaining children or  $\lceil \frac{\Delta(T)}{3} \rceil$  trees have been pruned at  $v$ .

For each child  $x$  of  $v$  in  $T'$ , let  $T'_{vx}$  denote the subtree of  $T'$  consisting of  $v$ ,  $x$ , and all descendants of  $x$  in  $T'$ . The choices of the subtrees rooted at  $v$  to prune are made greedily according to the following priority ordering of edge



colors joining  $v$  to its children (this ordering is summarized in Table 1). In the following list, let  $x$ ,  $x'$ , and  $x''$  be children of  $v$ , though it is possible that  $v$  does not have three children.

1. If  $vx$  is red and  $vx'$  is green, then prune  $T'_{vx} \cup T'_{vx'}$ .
2. If  $vx$  is red, then prune  $T'_{vx}$ .
3. If  $vx$  and  $vx'$  are yellow and  $vx''$  is green, then prune  $T'_{vx} \cup T'_{vx'} \cup T'_{vx''}$ .
4. If  $vx$  and  $vx'$  are yellow, then prune  $T'_{vx} \cup T'_{vx'}$ .
5. If  $vx$  is yellow and  $vx'$  and  $vx''$  are green, then prune  $T'_{vx} \cup T'_{vx'} \cup T'_{vx''}$ .
6. If  $vx$  is yellow and  $vx'$  is green, then prune  $T'_{vx} \cup T'_{vx'}$ .
7. If  $vx$  is yellow, then prune  $T'_{vx}$ .
8. If  $vx$ ,  $vx'$  and  $vx''$  are green, then prune  $T'_{vx} \cup T'_{vx'} \cup T'_{vx''}$ .
9. If  $vx$  and  $vx'$  are green, then prune  $T'_{vx} \cup T'_{vx'}$ .
10. If  $vx$  is green, then prune  $T'_{vx}$ .

Most preferred	Least preferred
RG > R > YYG > YY > YGG > YG > Y > GGG > GG > G	

Table 1: Preferred pruning order in PRUNE, and ordering of color types in the proof of Theorem 6.

We now describe COLOR in detail. First, assume that  $v$  is not the root of  $T$ . Let  $T''$  be the subtree of  $T$  that remains after PRUNE is called at  $v$ , and let  $u$  be the parent of  $v$ . The COLOR routine assigns a color to  $uv$  or halts and declares  $ub(T) = \left\lceil \frac{\Delta(G)+1}{3} \right\rceil$  as follows (summarized in Table 2).

Edges to remaining children at $v$ after pruning	Action
None or G	color $uv$ G
GG, Y, or YG	color $uv$ Y
YY or R	color $uv$ R
At least three, RR, RY, or RG	declare $ub(T) = \lceil \frac{\Delta(G)+1}{3} \rceil$

 Table 2: Action of  $\text{COLOR}(v)$ , where  $u$  denotes the parent of  $v$ .

- 1) Color  $uv$  green (G) if
  - a)  $v$  has no children in  $T''$ , or
  - b)  $v$  has exactly one child  $x$  in  $T''$  and  $vx$  is green.
- 2) Color  $uv$  yellow (Y) if
  - a)  $v$  has exactly two children  $x$  and  $x'$  in  $T''$ , and  $vx$  and  $vx'$  are both green, or
  - b)  $v$  has exactly one child  $x$  in  $T''$  and  $vx$  is yellow, or
  - c)  $v$  has exactly two children  $x$  and  $x'$  in  $T''$ , and  $vx$  is yellow and  $vx'$  is green.
- 3) Color  $uv$  red (R) if
  - a)  $v$  has exactly two children  $x$  and  $x'$  in  $T''$ , and  $vx$  and  $vx'$  are both yellow, or
  - b)  $v$  has exactly one child  $x$  in  $T''$  and  $vx$  is red.
- 4) Declare  $ub(T) = \lceil \frac{\Delta(G)+1}{3} \rceil$  if
  - a)  $v$  has at least three children in  $T''$ , or
  - b)  $v$  has two children  $x$  and  $x'$  in  $T''$  and  $vx$  is red.

If  $v$  is the root of  $T$ , then  $\text{COLOR}$  declares  $ub(T) = \lceil \frac{\Delta(T)}{3} \rceil$  if  $v$  has no children after  $\text{PRUNE}$  runs at  $v$ , and declares  $ub(T) = \lceil \frac{\Delta(T)+1}{3} \rceil$  otherwise.

We now present  $\text{UNIT\_BAR\_TREE}(T)$  in pseudocode.

**Algorithm 5** ( $\text{UNIT\_BAR\_TREE}(T)$ )

*Input:* A tree  $T$ .

*Output:*  $ub(T)$ . If  $ub(T) = \lceil \frac{\Delta(T)}{3} \rceil$ , then also a decomposition  $\mathcal{T}$  of  $T$  into UBVTs so that every vertex is in at most  $\lceil \Delta(T)/3 \rceil$  trees in  $\mathcal{T}$ .

1. Initialize: Choose a vertex  $r$  in  $T$  and let  $T_r$  be  $T$  rooted at  $r$ . Set  $\mathcal{T} = \emptyset$ .
2. Do a postorder traversal of  $T_r$ . Let  $v$  be the current vertex.
  - a. Run  $\text{PRUNE}(v)$  and add each UBVT pruned at  $v$  to  $\mathcal{T}$ .
  - b. Run  $\text{COLOR}(v)$ .
3. Return  $ub(T)$ . If  $ub(T) = \lceil \Delta(T)/3 \rceil$ , also return  $\mathcal{T}$ .

An example of Algorithm 5 is provided in Figure 3.

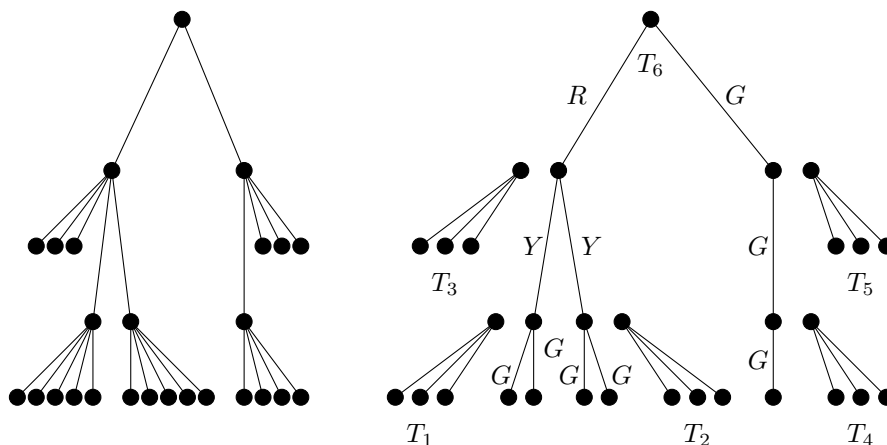


Figure 3: A tree with unit bar visibility number 2 and the decomposition produced by Algorithm 5. The edges in  $T_6$  are labeled with the colors of their respective branches.

Before proving that Algorithm 5 produces an optimal decomposition of  $T$ , we present a lemma that allows us to modify UBVTs rooted at the same vertex. Given a tree  $T$  with root  $v$ , we call each maximal subtree of  $T$  that contains  $v$  as a leaf a *branch* of  $T$ . We assign a color to each branch  $B$  in  $T$  depending on its structure as described below. The color that is assigned to  $B$  is the same as the color that the COLOR routine would assign to the edge in  $B$  that is incident to  $v$ .

1.  $B$  is red (R) if  $B$  contains vertices of degree 3, all of which lie in a single path, and no such path also contains  $v$  (i.e.  $B$  is a subdivided caterpillar with maximum degree 3 and  $v$  is not on the spine);
2.  $B$  is yellow (Y) if  $B$  contains vertices of degree 3, all of which lie in a single path that also contains  $v$  (i.e.  $B$  is a subdivided caterpillar with maximum degree 3 and  $v$  is on the spine);
3.  $B$  is green (G) if  $B$  contains no vertices of degree 3 (i.e.  $B$  is a path).

The multiset of the colors of the branches of  $T$  at  $v$  is the *color-type* of  $T$ , denoted  $c(T)$ . For convenience we suppress set notation and record each color-type as a string of Rs followed by Ys followed by Gs. It is clear that a tree is a UBVT if and only if it is trivial or its color-type is RG, R, YYG, YY, YGG, YG, Y, GGG, GG, or G. We rank these strings according to the lexicographic ordering arising from the ordering  $R \succ Y \succ G$ . This ranking is shown in Table 1.

Let  $T_1$  and  $T_2$  be two trees with a common root  $v$ , and let  $B_1$  and  $B_2$  be branches of  $T_1$  and  $T_2$  respectively. A *branch-swap* of  $B_1$  and  $B_2$  exchanges  $B_1$  and  $B_2$  yielding the trees  $(T_1 - B_1) \cup B_2$  and  $(T_2 - B_2) \cup B_1$ . We say that  $T_1$

absorbs  $B_2$  and  $T_2$  gives  $B_2$  if we add  $B_2$  to  $T_1$  yielding the trees  $T_1 \cup B_2$  and  $T_2 - B_2$ .

**Lemma 2** *Let  $T_1$  and  $T_2$  be two UBVTs that are rooted at  $v$ , let  $B_1$  be a branch of  $T_1$ , and let  $B_2$  be a branch of  $T_2$ .*

1. *If  $B_1$  and  $B_2$  have the same color, then the branch-swap of  $B_1$  and  $B_2$  yields two UBVTs.*
2. *If  $c(T_2) \in \{YGG, YG, GGG, GG, G\}$ ,  $B_1$  is yellow, and  $B_2$  is green, then the branch-swap of  $B_1$  and  $B_2$  yields two UBVTs.*
3. *If  $c(T_2) \in \{GG, G\}$ ,  $B_1$  is red, and  $B_2$  is green, then the branch-swap of  $B_1$  and  $B_2$  yields two UBVTs.*
4. *If  $d_{T_1}(v) \leq 2$ , there is no red branch in  $T_1$ , and  $B_2$  is green, then the absorption of  $B_2$  by  $T_1$  yields two UBVTs.*
5. *If  $d_{T_1}(v) = 1$ , the only branch in  $T_1$  is yellow, and  $B_2$  is green or yellow, then the absorption of  $B_2$  by  $T_1$  yields two UBVTs.*
6. *Suppose that  $B_1$  is red and  $c(T_2) \in \{GGG, GG\}$ . If  $B_2$  and  $B_3$  are both branches of  $T_2$ , then  $(T_1 - B_1) \cup B_2 \cup B_3$  and  $(T_2 - (B_2 \cup B_3)) \cup B_1$  are both UBVTs.*
7. *If  $B_1$  is red and  $c(T_2) = G$ , then  $(T_1 - B_1) \cup B_2$  and  $(T_2 - B_2) \cup B_1$  are both UBVTs.*

**Proof:** In all cases, it is trivial to check that the resulting trees have color-types of UBVTs.  $\square$

**Theorem 6** *Given a tree  $T$ ,  $\text{UNIT\_BAR\_TREE}(T)$  determines  $ub(T)$  in time linear in the order of the tree. Furthermore, if  $ub(T) = \lceil \Delta(T)/3 \rceil$ , then  $\text{UNIT\_BAR\_TREE}(T)$  also generates a decomposition of  $T$  into UBVTs so that each vertex is in at most  $\lceil \Delta(T)/3 \rceil$  of the UBVTs.*

**Proof:** First assume that  $\text{UNIT\_BAR\_TREE}(T)$  returns  $ub(T) = \lceil \Delta(T)/3 \rceil$ . In this case,  $\text{UNIT\_BAR\_TREE}(T)$  also returns  $\mathcal{T}$ , a decomposition of  $T$  into UBVTs. If  $v$  is a vertex that is not the root of  $T$ , then  $\mathcal{T}$  contains at most  $\lceil \Delta(T)/3 \rceil - 1$  UBVTs that are rooted at  $v$  and exactly one UBVT containing  $v$  that is rooted at an ancestor of  $v$ . If  $v$  is the root, then  $\mathcal{T}$  contains at most  $\lceil \Delta(T)/3 \rceil$  UBVTs that contain  $v$ . Therefore, if  $\text{UNIT\_BAR\_TREE}(T)$  declares  $ub(T) = \lceil \Delta(T)/3 \rceil$ , then by Theorem 2  $\mathcal{T}$  serves as a certificate.

For the rest of the proof we assume to the contrary that there is a tree  $T$  such that  $\text{UNIT\_BAR\_TREE}(T)$  returns  $ub(T) = \left\lceil \frac{\Delta(T)+1}{3} \right\rceil$  while  $ub(T) = \lceil \Delta(T)/3 \rceil$ . Note that there is only a discrepancy in these two values if  $\Delta(T)$  is a multiple of 3. Thus we assume that  $\Delta(T)$  is a multiple of 3. We will write  $\frac{\Delta(T)}{3} + 1$  for  $\left\lceil \frac{\Delta(T)+1}{3} \right\rceil$  and we will suppress the ceiling notation in  $\left\lceil \frac{\Delta(T)}{3} \right\rceil$ . By

Theorem 2 we may assume that  $T$  has a decomposition into UBVTs so that each vertex is in at most  $\Delta(T)/3$  of the UBVTs. Let  $\mathcal{T} = \{T_1, \dots, T_\ell\}$  be the partial decomposition of  $T$  into UBVTs when Algorithm 5 halts and declares  $ub(T) = \frac{\Delta(T)}{3} + 1$ . Order  $\mathcal{T}$  so that  $T_i$  is pruned before  $T_j$  when  $i < j$ . Let  $\widehat{\mathcal{T}}$  be a decomposition of  $T$  into UBVTs so that each vertex is in at most  $\Delta(T)/3$  elements of  $\widehat{\mathcal{T}}$ .

First suppose that  $\mathcal{T} \subseteq \widehat{\mathcal{T}}$ . Let  $y$  be the vertex where Algorithm 5 halts. If  $y$  is the root of  $T$ , then  $y$  is in  $\Delta(T)/3$  trees in  $\widehat{\mathcal{T}}$ , and not all edges at  $y$  are contained in those trees. Hence  $y$  is in at least  $\frac{\Delta(T)}{3} + 1$  elements of  $\widehat{\mathcal{T}}$ , a contradiction. Now assume that  $y$  is not the root of  $T$ . After pruning,  $y$  has degree at least 3 or is still joined to a child by a red edge. In the latter case,  $y$  has descendants of degree 3, and they do not all lie in a single path with  $y$ . It follows that the tree consisting of  $y$ , its remaining descendants, and the edge joining  $y$  to its parent is not a subgraph of a UBVT. Therefore one of  $y$  or its non-leaf descendants is in at least two elements of  $\widehat{\mathcal{T}}$  that are not in  $\mathcal{T}$ . Since  $y$  and all of its remaining non-leaf descendants are in  $\Delta(T)/3 - 1$  elements of  $\mathcal{T}$  (since they each survived the pruning of  $\Delta(T)/3 - 1$  branches), some vertex is in at least  $\Delta(T)/3 + 1$  elements of  $\widehat{\mathcal{T}}$ , a contradiction.

We now assume that  $\mathcal{T} \not\subseteq \widehat{\mathcal{T}}$ . Choose  $\widehat{\mathcal{T}}$  to maximize  $k$  so that  $T_i \in \widehat{\mathcal{T}}$  for all  $i \leq k$ ; note that  $k < \ell$ . Let  $v$  be the vertex where  $T_{k+1}$  is pruned in Algorithm 5. Let  $T'$  be the subtree of  $T$  that remains when Algorithm 5 calls PRUNE at  $v$ . Let  $x_1, \dots, x_m$  be the children of  $v$ , ordered so that when  $i < j$  the index of the element of  $\mathcal{T}$  containing  $x_i$  is less than or equal to the index of the element of  $\mathcal{T}$  containing  $x_j$ .

Let  $T'_{vx_i}$  be the subtree of  $T'$  consisting of  $v, x_i$ , and all descendants of  $x_i$ . Let  $\widehat{T}_{vx_i}$  be the tree in  $\widehat{\mathcal{T}}$  that contains  $vx_i$ . We claim that  $T'_{vx_i} \subseteq \widehat{T}_{vx_i}$ . If  $T'_{vx_i} \not\subseteq \widehat{T}_{vx_i}$ , then there is a descendant  $x'$  of  $v$  in  $\widehat{T}_{vx_i}$  such that some child of  $x'$  from  $T'$  is not in  $\widehat{T}_{vx_i}$ . It follows that  $x'$  lies in at least two elements of  $\widehat{\mathcal{T}} - \{T_1, \dots, T_k\}$ . Since  $x'$  is a descendant of  $v$ , Algorithm 5 has already run PRUNE at  $x'$  when it begins to run PRUNE at  $v$ . Since  $x'$  is not a leaf in  $T'$ , it follows that  $\Delta(T)/3 - 1$  trees are pruned at  $x'$  by Algorithm 5. Furthermore, the trees that are pruned at  $x'$  are in the set  $\{T_1, \dots, T_k\}$  and hence are also in  $\widehat{\mathcal{T}}$ . It follows that  $x'$  lies in  $\Delta(T)/3 + 1$  elements of  $\widehat{\mathcal{T}}$ , a contradiction. Therefore, for all  $i \in [m]$ ,  $T'_{vx_i}$  is a subtree of an element of  $\widehat{\mathcal{T}}$ .

We now impose an additional extremal condition on  $\widehat{\mathcal{T}}$ . Since  $T_{k+1}$  is pruned at  $v$ , it follows that  $T_{k+1} = T'_{vx_i} \cup \dots \cup T'_{vx_{i+j}}$  for some  $i \in [m]$  and some  $j \in \{0, 1, 2\}$ . Order  $vx_i, \dots, vx_{i+j}$  so that  $vx_i$  has the highest priority color with respect to the order  $R \succ Y \succ G$ . Among all optimal decompositions that contain  $\{T_1, \dots, T_k\}$ , choose  $\widehat{\mathcal{T}}$  so that  $\widehat{T}_{vx_i}$ , the element of  $\widehat{\mathcal{T}}$  that contains  $vx_i$ , has as many branches in common with  $T_{k+1}$  as possible. We will modify  $\widehat{T}_{vx_i}$  to obtain a new optimal decomposition of  $T$  that contradicts the extremality of  $\widehat{\mathcal{T}}$ . We proceed by cases depending on the relative color-types of  $T_{k+1}$  and  $\widehat{T}_{vx_i}$  (refer to Table 1). The specific modifications to  $\widehat{T}_{vx_i}$  are summarized in Table 3.

		$c(\widehat{T}_{vx_i})$	
$c(T_{k+1})$		RG	R
RG		Swap G-G	Abs. G
R		Give G	Same

		$c(\widehat{T}_{vx_i})$				
$c(T_{k+1})$		YYG	YY	YGG	YG	Y
YYG		Swap	Abs. G	Swap G-Y	Abs. Y	Abs. Y
YY		Give G	Swap Y-Y	Swap G-Y	Abs. Y	Abs. Y
YGG		Swap Y-G	Abs. G	Swap G-G	Abs. G	Abs. G
YG		Give Y	Abs. G	Give G	Swap G-G	Abs. G
Y		NA	Give Y	NA	Give G	Same

		$c(\widehat{T}_{vx_i})$		
$c(T_{k+1})$		RG	YGG	YG
GGG		Swap R-GG	Swap Y-G	Abs. G
GG		Swap R-G	Give Y	Swap Y-G
G		Give R	NA	Give Y

		$c(\widehat{T}_{vx_i})$		
$c(T_{k+1})$		GGG	GG	G
GGG		Swap G-G	Abs. G	Abs. G
GG		Give G	Swap G-G	Abs. G
G		NA	Give G	Same

Table 3: Modification to  $\widehat{T}_{vx_i}$ . The top table is for when  $vx_i$  is red, the second table is for when  $vx_i$  is yellow, and the bottom two tables are for when  $vx_i$  is green. The rows are determined by  $c(T_{k+1})$  and the columns are determined by  $c(\widehat{T}_{vx_i})$ . “Swap” indicates that  $\widehat{T}_{vx_i}$  swaps a branch of the first color for a branch in  $T_{k+1}$  of the second color. “Abs.” indicates that  $\widehat{T}_{vx_i}$  absorbs a branch of the specified color from  $T_{k+1}$ . “Give” indicates that a branch of  $\widehat{T}_{vx_i}$  of the specified color either becomes its own tree or is absorbed by another tree. “Same” indicates that  $\widehat{T}_{vx_i} = T_{k+1}$ .

**Case 1:**  $c(T_{k+1}) \succeq c(\widehat{T}_{vx_i})$ . Because  $T_{k+1}$  and  $\widehat{T}_{vx_i}$  both contain the branch with  $vx_i$ , it follows that  $d_{T_{k+1}}(v) \geq 2$  as otherwise  $T_{k+1} = \widehat{T}_{vx_i}$  (following from the priority ranking of the color-types). In all such cases, it follows from Lemma 2 that  $\widehat{T}_{vx_i}$  can either swap a branch with or absorb a branch from some element of  $\widehat{\mathcal{T}} - \{T_1, \dots, T_k\}$  to have more branches in common with  $T_{k+1}$ . This violates the extremality of  $\widehat{\mathcal{T}}$ . (See Table 3.1,

**Case 2:**  $c(\widehat{T}_{vx_i}) \succ c(T_{k+1})$ , and  $v$  has no children after  $T_{k+1}$  is pruned. In this case,  $\widehat{T}_{vx_i}$  must have a branch that contains the parent of  $v$ . Because  $T_{k+1}$  is pruned by Algorithm 5, we conclude that at most  $\Delta(T)/3 - 1$  trees are used by the algorithm for the edges joining  $v$  to its children. If  $v$  is in at most  $\Delta(T)/3 - 1$  trees in  $\widehat{\mathcal{T}}$ , then removing the branch from  $\widehat{T}_{vx_i}$  that contains the parent of  $v$  and using it as its own additional element yields an optimal decomposition. In this decomposition, the color-type of the tree that contains  $vx_i$  is lower ranked than  $c(T_{k+1})$ , yielding an instance of Case 1. Otherwise  $v$  is in  $\Delta(T)/3$  trees in  $\widehat{\mathcal{T}}$ , two of which are in  $\widehat{\mathcal{T}} - \{T_1, \dots, T_k\}$ . It is straightforward to check that in all such cases Lemma 2 applies and we can perform branch swaps on  $T'_{vx_i}$  and another element of  $\widehat{\mathcal{T}} - \{T_1, \dots, T_k\}$  that contains  $v$  to obtain a decomposition that either contains  $\{T_1, \dots, T_{k+1}\}$  (violating the extremality of  $\widehat{\mathcal{T}}$ ) or that is an instance of Case 1.

**Case 3:**  $c(\widehat{T}_{vx_i}) \succ c(T_{k+1})$ , and  $v$  has children after  $T_{k+1}$  is pruned. In this case,  $c(T_{k+1}) \in \{R, YY, YGG, GGG\}$ . In all of these cases,  $\widehat{T}_{vx_i}$  has a branch  $B$  that does not include  $vx_i$  such that  $c(T_{k+1}) \succeq c(\widehat{T}_{vx_i} - B)$ . If  $v$  is in less than  $\Delta(T)/3$  elements of  $\widehat{\mathcal{T}}$ , then replacing  $\widehat{T}_{vx_i}$  with  $\widehat{T}_{vx_i} - B$  and  $B$  yields an optimal decomposition of  $T$  that is an instance of Case 1. Thus we assume that  $v$  is in  $\Delta(T)/3$  elements of  $\widehat{\mathcal{T}}$ .

If  $B$  is green, then  $c(T_{k+1}) \in \{R, YY\}$ . By Algorithm 5, all remaining branches at  $v$  are red or yellow, and the green branch of  $\widehat{T}_{vx_i}$  contains the parent of  $v$ . Therefore all remaining elements of  $\widehat{\mathcal{T}}$  that contain  $v$  have color type R, YY, or Y, and by Lemma 2 any one of these can absorb  $B$  from  $\widehat{T}_{vx_i}$ .

If  $B$  is yellow, then  $c(T_{k+1}) \in \{YGG, GGG\}$ . By Algorithm 5, all remaining branches at  $v$  are green. In this case, by Lemma 2 it is possible to perform a branch-swap with  $B$  and a green branch of a tree in  $\widehat{\mathcal{T}} - \{T_1, \dots, T_k\}$  rooted at  $v$  with color type GGG, GG, or G.

If  $B$  is red, then  $c(T_{k+1}) = GGG$  and  $c(\widehat{T}_{vx_i}) = RG$ . By Algorithm 5, all remaining branches at  $v$  are green. Therefore all remaining elements of  $\widehat{\mathcal{T}}$  containing  $v$  have color type GGG, GG, or G. Because these elements of  $\widehat{\mathcal{T}}$  contain two branches of  $T_{k+1}$ , we can assume (using branch absorption if necessary) that one of the remaining elements of  $\widehat{\mathcal{T}}$  that contains  $v$  is of color-type GGG or GG. Thus it is possible by Lemma 2 to swap  $B$  for two green branches of a tree in  $\widehat{\mathcal{T}} - \{T_1, \dots, T_k\}$ . This completes Case 3.

We conclude that for each color of  $B$  it is possible to obtain an optimal decomposition that is an instance of Case 1 or that contains  $\{T_1, \dots, T_{k+1}\}$ , violating the extremality of  $\widehat{\mathcal{T}}$ .

It remains to show that `UNIT_BAR_TREE`( $T$ ) runs in linear time. At a

vertex  $v$ , PRUNE runs in time  $O(d(v))$  and COLOR runs in constant time. Since  $\sum_{v \in V(T)} d(v) = |V(T)| - 1$  and the post-order traversal takes  $O(|V(T)|)$  time, it follows that UNIT\_BAR\_TREE( $T$ ) runs in  $O(|V(T)|)$  time.  $\square$

We note that UNIT\_BAR\_TREE( $T$ ) only provides a decomposition into UBVTs when  $ub(T) = \lceil \Delta(T)/3 \rceil$ . If  $ub(T) = \lceil \frac{\Delta(T)+1}{3} \rceil$ , then modifying the PRUNE routine to allow up to  $\lceil \Delta(T)/3 \rceil$  trees to be pruned at nonroot vertices and  $\lceil \Delta(T)/3 \rceil + 1$  trees to be pruned at the root will allow UNIT\_BAR\_TREE( $T$ ) to produce an optimal decomposition. This adds  $O(|V(T)|)$  additional operations to the algorithm (one more prune at each vertex). Thus an optimal decomposition of such a tree can also be found in linear time, resulting from possibly two iterations of UNIT\_BAR\_TREE( $T$ ). Dean and Veytsel [6] provided a construction of a unit bar visibility representation of a UBVT that is clearly implementable in time linear in the number of vertices. Since the sum of the sizes of the vertex sets in a decomposition of a tree  $T$  into UBVTs is at most  $2|V(T)| - 2$ , it follows that an optimal  $t$ -unit-bar visibility representation of a tree can be found in linear time.

Dean et al. [5] introduced *unit rectangle visibility graphs*, where vertices are assigned to distinct axis-aligned unit rectangles in the plane and edges correspond to uninterrupted vertical or horizontal lines of sight between the rectangles. They proved that a tree  $T$  is a unit rectangle visibility graph if and only if  $\Upsilon_{ub}(T) \leq 2$ . In the conclusion of their paper, they ask if there is an efficient algorithm to determine if an arbitrary graph is a unit rectangle visibility graph, and note that the question is unanswered even for trees. Algorithm 5 answers the question in the affirmative for trees.

**Corollary 7** *There is a linear time algorithm that determines if a tree is a unit rectangle visibility graph.*

Given our results on trees, it is possible to bound the unit bar visibility number of planar graphs based on their maximum degree.

**Theorem 8** *If  $G$  is a planar graph, then  $ub(G) \leq \lceil \frac{\Delta(G)+1}{3} \rceil + 2$ .*

**Proof:** Nash-Williams [10] proved that every planar graph has arboricity at most 3, so there is a decomposition of  $G$  into trees such that every vertex is in three trees (note that some of these trees may be  $K_1$ ). Let  $v \in V(G)$  and let  $T_1$ ,  $T_2$ , and  $T_3$  be the trees in the decomposition containing  $v$ . By Theorem 4, for each  $i \in [3]$  there is a decomposition of  $T_i$  into UBVTs so that  $v$  is in at most  $\lceil \frac{d_{T_i}(v)+1}{3} \rceil$  elements of the decomposition. Therefore there is a decomposition of  $G$  into UBVTs such that  $v$  is contained in at most  $\lceil \frac{d_{T_1}(v)+1}{3} \rceil + \lceil \frac{d_{T_2}(v)+1}{3} \rceil + \lceil \frac{d_{T_3}(v)+1}{3} \rceil$  elements of the decomposition. Since  $d_G(v) = d_{T_1}(v) + d_{T_2}(v) + d_{T_3}(v)$ , it follows that  $\lceil \frac{d_{T_1}(v)+1}{3} \rceil + \lceil \frac{d_{T_2}(v)+1}{3} \rceil + \lceil \frac{d_{T_3}(v)+1}{3} \rceil \leq \lceil \frac{d_G(v)+1}{3} \rceil + 2$ .  $\square$

The *girth* of a graph is the length of its longest cycle. When a planar graph has large girth, we can slightly improve Theorem 8.



**Theorem 9** *If  $G$  is a planar graph with girth at least 7, then*

$$ub(G) \leq \left\lceil \frac{\Delta(G) + 1}{3} \right\rceil + 1.$$

**Proof:** Borodin [3] proved that planar graphs with girth at least 7 are acyclically 3-colorable. That is,  $V(G)$  can be partitioned into three sets such that the union of any two of the sets will induce a forest. Each vertex will appear in exactly two of these forests. Hence there is a decomposition of  $G$  into trees such that each vertex is in at most two trees. Let  $v \in V(G)$  and let  $T_1$  and  $T_2$  be the trees that contain  $v$ . By Theorem 4, there is a decomposition of  $G$  into UBVTs so that  $v$  is contained in at most  $\left\lceil \frac{d_{T_1}(v)+1}{3} \right\rceil + \left\lceil \frac{d_{T_2}(v)+1}{3} \right\rceil$  elements of the decomposition. Since  $\left\lceil \frac{d_{T_1}(v)+1}{3} \right\rceil + \left\lceil \frac{d_{T_2}(v)+1}{3} \right\rceil \leq \left\lceil \frac{d(v)+1}{3} \right\rceil + 1$ , the result follows.  $\square$

## 4 Complete Bipartite Graphs

Dean and Veytsel [6] characterized the complete bipartite graphs that are unit bar visibility graphs; unsurprisingly there are very few.

**Theorem 10** *(Dean-Veytsel [2003]) The complete bipartite graph  $K_{m,n}$ , where  $m \geq n$ , is a UBVG if and only if  $m \leq 3$  and  $n = 1$ , or  $m = n = 2$ .*

In this section we study the unit bar visibility number of complete bipartite graphs. Throughout this section, we let  $K_{m,n}$  have partite sets  $V$  and  $U$  with  $V = \{v_1, \dots, v_m\}$  and  $U = \{u_1, \dots, u_n\}$ . We will also assume that  $m \geq n \geq 2$ . Note that the case when  $n = 1$  is handled in Section 3. We begin with two constructions of  $t$ -unit-bar visibility representations of  $K_{m,n}$ . Both constructions begin with arrangements of bars that represent *most*, but not all, of the edges in  $K_{m,n}$ . There are a small number of edges that are not represented, and we add bars for those edges after.

**Construction 1** *The construction begins with  $2 \left\lceil \frac{m}{4} \right\rceil + 1$  horizontal line segments of length  $n$ , each of which will be subdivided into  $n$  contiguous unit bars. From bottom to top, the segments are assigned to  $V$  and  $U$  in alternating fashion; thus both the top and bottom bars are assigned to  $V$ . For  $j \in \left[ \left\lceil \frac{m}{4} \right\rceil + 1 \right]$ , let  $V_j$  be the  $j$ th segment assigned to  $V$  and for  $j \in \left[ \left\lceil \frac{m}{4} \right\rceil \right]$  let  $U_j$  be the  $j$ th segment assigned to  $U$ , indexing from bottom to top. For  $j \in \left[ \left\lceil \frac{m}{4} \right\rceil + 1 \right]$ , the segment  $V_j$  has left endpoint  $(j - 1, j - 1)$ , and for  $j \in \left[ \left\lceil \frac{m}{4} \right\rceil \right]$  the segment  $U_j$  has left endpoint  $(j - \frac{1}{2}, j - \frac{1}{2})$ .*

*In each segment, the unit bars from left to right will correspond to  $n$  consecutive vertices in the (cyclic) ordering of  $V$  or  $U$ . Thus providing the leftmost unit bar in each line segment determines the construction. For  $j \in \left[ \left\lceil \frac{m}{4} \right\rceil + 1 \right]$ , the leftmost bar of the segment  $V_j$  is assigned to  $v_{3j-2}$ . For  $j \in \left[ \left\lceil \frac{m}{4} \right\rceil \right]$ , the leftmost bar of the segment  $U_j$  is assigned to  $u_{2-j}$  (with the index taken modulo  $n$ ). See Figure 4.*

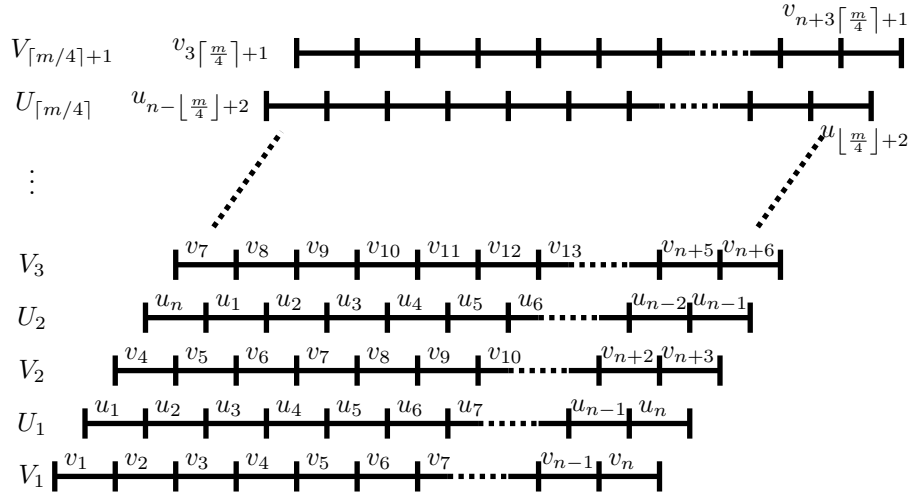


Figure 4: The line segments in Construction 1.

After constructing the line segments, for each pair  $(u_i, v_{i'})$  where no bar of  $u_i$  sees a bar of  $v_{i'}$ , to the right of the line segments we add a unit bar for both  $u_i$  and  $v_{i'}$  that are visible to each other and no other bars.

**Lemma 3** *The unit bar visibility layout of Construction 1 is a  $t$ -unit-bar visibility representation of  $K_{m,n}$  with  $t \leq \lceil \frac{m}{4} \rceil + \frac{2m}{n} + 12$ .*

**Proof:** For  $i \in [n]$  and  $j \in [\lceil m/4 \rceil]$ , let  $b_{i,j}$  be the bar in  $U_j$  that is assigned to  $u_i$ . For  $i \in [m]$  and  $j \in [\lceil m/4 \rceil + 1]$ , let  $\beta_{i,j}$  be the bar in  $V_j$  that is assigned to  $v_i$  (note that  $\beta_{i,j}$  may not exist for certain combinations of  $i$  and  $j$ ). Fix  $j \in [\lceil m/4 \rceil]$  and  $i \in [n]$ . If  $i \not\equiv 2 - j, 1 - j \pmod{n}$ , then  $b_{i,j}$  sees bars  $\beta_{i+4j-4,j}$ ,  $\beta_{i+4j-3,j}$ ,  $\beta_{i+4j-2,j+1}$ , and  $\beta_{i+4j-1,j+1}$ . If  $i \equiv 2 - j \pmod{n}$ , then  $b_{i,j}$  is the leftmost bar of  $U_j$  and sees bars  $\beta_{i+4j-4,j}$ ,  $\beta_{i+4j-3,j}$ , and  $\beta_{i+4j-1,j+1}$ . If  $i \equiv 1 - j \pmod{n}$ , then  $b_{i,j}$  is the rightmost bar of  $U_j$  and sees bars  $\beta_{i+4j-4,j}$ ,  $\beta_{i+4j-2,j+1}$ , and  $\beta_{i+4j-1,j+1}$ . Therefore, given  $i \in [n]$ , the sets of bars seen by  $b_{i,1}, \dots, b_{i,\lceil m/4 \rceil}$  correspond to pairwise disjoint sets of vertices in  $V$ . It also follows that given  $i \in [m]$ , the bars in  $V_1, \dots, V_{\lceil m/4 \rceil + 1}$  assigned to  $v_i$  see bars corresponding to pairwise disjoint sets of vertices in  $U$ .

Let  $u_i \in U$ . Observe that  $u_i$  is assigned to the leftmost bar of at most  $\lceil \frac{\lceil m/4 \rceil}{n} \rceil$  segments. Similarly,  $u_i$  is assigned to the rightmost bar of at most  $\lceil \frac{\lceil m/4 \rceil}{n} \rceil$  segments. Interior bars in the segments see four other bars, and leftmost and rightmost bars see three other bars, so the bars assigned to  $u_i$  in  $U_1 \cup \dots \cup U_{\lceil m/4 \rceil}$  see at least  $4 \left( \lceil \frac{m}{4} \rceil - 2 \lceil \frac{\lceil m/4 \rceil}{n} \rceil \right)$  bars, which are assigned to distinct vertices in  $V$ . Thus  $u_i$  is assigned to at most  $\frac{2m}{n} + 12$  additional bars. Therefore the vertices in  $U$  are assigned to at most  $\lceil \frac{m}{4} \rceil + \frac{2m}{n} + 12$  bars.

Each interior bar of  $V_j$  for  $2 \leq j \leq \lceil m/4 \rceil$  sees four other bars, while the leftmost and rightmost bars see three other bars. The leftmost bar of  $V_1$  and the rightmost bar of  $V_{\lceil \frac{m}{4} \rceil + 1}$  each see one other bar. All other bars in  $V_1$  and  $V_{\lceil \frac{m}{4} \rceil + 1}$  see two other bars. Let  $v_i \in V$ . Observe that  $v_i$  is assigned to the leftmost bar of at most two segments and the rightmost bar of at most two segments. Furthermore, there are at most  $\lceil (m-n)/3 \rceil$  segments that do not contain a bar assigned to  $v_i$ . Suppose that  $v_i$  is not assigned to a bar in  $r$  of the segments. Since  $v_i$  is assigned to at most one bar in  $V_1$  and  $V_{\lceil m/4 \rceil + 1}$ , it follows that the bars of  $v_i$  in the segments see at least  $4(\lceil \frac{m}{4} \rceil + 1 - r) - 8$  bars, which are assigned to distinct vertices in  $U$ . Thus  $v_i$  needs at most  $n - (4(\lceil \frac{m}{4} \rceil + 1 - r) - 8)$  additional bars. Since

$$\begin{aligned} \lceil \frac{m}{4} \rceil + 1 - r + n - (4(\lceil \frac{m}{4} \rceil + 1 - r) - 8) &\leq \lceil \frac{m}{4} \rceil + 3r + n - m + 5 \\ &\leq \lceil \frac{m}{4} \rceil + 3 \left\lceil \frac{m-n}{3} \right\rceil + n - m + 5 \\ &\leq \lceil \frac{m}{4} \rceil + 8, \end{aligned}$$

it follows that  $v_j$  is assigned to at most  $\lceil \frac{m}{4} \rceil + 8$  bars. □

**Construction 2** We begin with an array of  $2 \lfloor n/4 \rfloor + 1$  line segments of length  $n \lfloor m/n \rfloor$ . From bottom to top, the segments are assigned to  $V$  and  $U$  in alternating fashion. For  $j \in [\lfloor n/4 \rfloor + 1]$ , let  $V_j$  for be the  $j$ th segment assigned to  $V$  and for  $j \in [\lfloor n/4 \rfloor]$  let  $U_j$  be the  $j$ th segment assigned to  $U$ , indexing from bottom to top. For  $j \in [\lfloor n/4 \rfloor + 1]$ , the segment  $V_j$  has left endpoint  $(j-1, j-1)$ , and for  $j \in [\lfloor n/4 \rfloor]$  the segment  $U_j$  has left endpoint  $(j - \frac{1}{2}, j - \frac{1}{2})$ . The segments assigned to  $U$  consist of  $\lfloor m/n \rfloor$  sets of bars assigned to the vertices of  $U$  in order, up to a cyclic shift. The segments assigned to  $V$  consist of  $n \lfloor m/n \rfloor$  bars assigned to  $\{v_1, \dots, v_{n \lfloor m/n \rfloor}\}$  up to a cyclic shift. For  $j \in [\lceil \frac{n}{4} \rceil + 1]$ , the leftmost bar of the segment  $V_j$  is assigned to  $v_{3j-2}$ . For  $j \in [\lceil \frac{n}{4} \rceil]$ , the leftmost bar of the segment  $U_j$  is assigned to  $u_{2-i}$ . See Figure 5.

After constructing the line segments, for each pair  $(u_i, v_{i'})$  where no bar of  $u_i$  sees a bar of  $v_{i'}$ , to the right of the line segments we add a unit bar for both  $u_i$  and  $v_{i'}$  that are visible to each other and no other bars.

**Lemma 4** The unit bar visibility layout of Construction 2 is a  $t$ -unit-bar visibility representation of  $K_{m,n}$  with  $t \leq \frac{m}{4} + n + 1$ .

**Proof:** Fix  $j \in [\lfloor n/4 \rfloor]$  and  $i \in [n]$ . By construction, if a bar in  $U_j$  that is assigned to  $u_i$  sees a bar that is assigned to  $v_{i'}$ , then  $i' \in [n \lfloor m/n \rfloor]$  and  $i' \equiv i + 4j - 4, i + 4j - 3, i + 4j - 2$ , or  $i + 4j - 1 \pmod{n}$ . The leftmost and rightmost bars of  $U_j$  see three other bars, and all other bars of  $U_j$  see four other bars. Furthermore, the bars in  $U_j$  that are assigned to  $u_i$  see pairwise disjoint sets of bars.

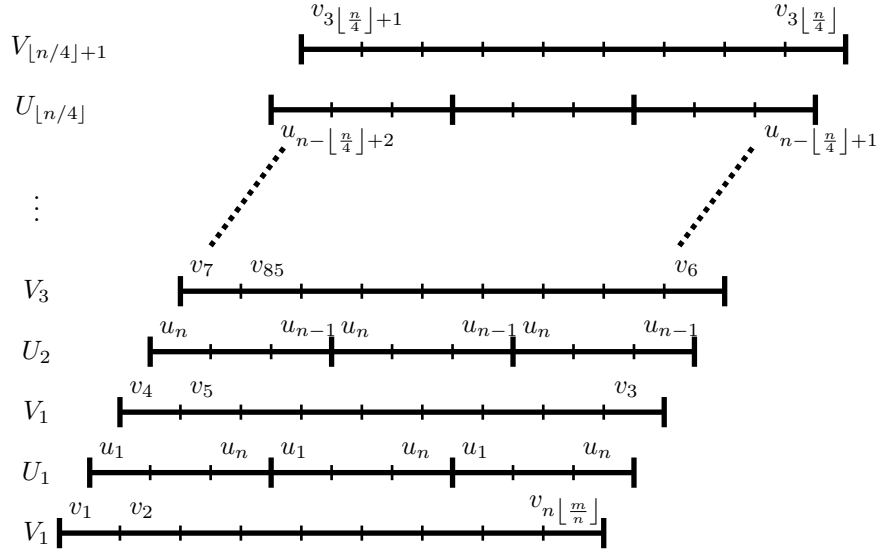


Figure 5: The line segments in Construction 2.

Each vertex in  $U$  is assigned to the leftmost bar of at most one segment and the rightmost bar of at most one segment. Therefore each vertex in  $U$  is assigned at most two additional bars outside of the segments for visibility to vertices in  $\{v_1, \dots, v_{n\lfloor m/n\rfloor}\}$ . Each vertex in  $U$  is also assigned at most  $n - 1$  bars for visibility to vertices in  $\{v_{n\lfloor m/n\rfloor+1}, \dots, v_m\}$ . Thus each vertex in  $U$  is assigned at most  $\lfloor \frac{n}{4} \rfloor \lfloor \frac{m}{n} \rfloor + 2 + n - 1 \leq \frac{m}{4} + n + 1$  bars.

Each vertex in  $V$  is assigned at most  $n$  additional bars outside of the segments. Thus each vertex in  $V$  is assigned at most  $\lfloor n/4 \rfloor + 1 + n$  bars. Since  $m \geq n$ , it follows that each vertex in  $V$  is assigned at most  $\frac{m}{4} + 1 + n$  bars.  $\square$

**Theorem 11** *For positive integers  $m$  and  $n$  with  $m \geq n \geq 2$ ,*

$$ub(K_{m,n}) \leq \min \left\{ \left\lceil \frac{m}{4} \right\rceil + \frac{2m}{n} + 12, \frac{m}{4} + n + 1 \right\}.$$

Therefore  $ub(K_{m,n}) \leq \frac{m}{4} + o(m)$ .

**Proof:** The first statement follows immediately from Lemmas 3 and 4. For the second statement we use  $ub(K_{m,n}) \leq \min \left\{ \frac{m}{4} + \frac{2m}{n} + 13, \frac{m}{4} + n + 1 \right\}$ . If  $n \leq 6 + \sqrt{36 + 2m}$ , then  $ub(K_{m,n}) \leq \frac{m}{4} + (6 + \sqrt{36 + 2m}) + 1$ . If  $n \geq 6 + \sqrt{36 + 2m}$ , then  $ub(K_{m,n}) \leq \frac{m}{4} + \frac{2m}{6 + \sqrt{36 + 2m}} + 12$ . In both cases,  $ub(K_{m,n}) \leq \frac{m}{4} + o(m)$ .  $\square$

We next prove a general lower bound on the unit bar visibility number of  $K_{m,n}$ . We refer to the bars in a  $t$ -unit-bar visibility representation of  $K_{m,n}$  that correspond to vertices in  $V$  and  $U$  as  $V$ -bars and  $U$ -bars, respectively.

**Theorem 12** For positive integers  $m$  and  $n$  with  $m \geq n \geq 2$ ,  $ub(K_{m,n}) \geq \lceil \frac{m}{5} \rceil$ .

**Proof:** Again, we let  $K_{m,n}$  have vertex set  $U \cup V$ , where  $|U| = n$  and  $|V| = m$ . Let  $R$  be a  $t$ -unit-bar visibility representation of  $K_{m,n}$ . We will assign the edges in  $K_{m,n}$  to the  $U$ -bars in  $R$ . For each edge  $uv$  in  $K_{m,n}$ , pick a line of sight in  $R$  between bars  $b_u$  and  $b_v$  for  $x$  and  $y$ , respectively. Each line of sight defines a maximal axis-aligned rectangle whose interior has empty intersection with all bars; we select the left edge of the rectangle and call this line segment  $\ell_{uv}$ . If  $\ell_{uv}$  contains an interior point of  $b_u$ , we can refer to the line of sight to the left of  $\ell_{uv}$ : this is the line of sight from a point arbitrarily close to the intersection of  $b_u$  and  $\ell_{uv}$  on the left that runs in the same direction from  $b_u$  as  $\ell_{uv}$ . Note that the line of sight to the left of  $\ell_{uv}$  may not be used in the assignment of edges to lines of sight, or may not connect  $b_u$  to another bar. We say that two bars are *aligned* if their left endpoint has the same  $x$ -coordinate (that is, the bars have the same projection onto the  $x$ -axis). We assign the edge  $uv$  to a bar in  $R$  as follows (see Figure 6):

- Type 1:**  $\ell_{uv}$  contains the left endpoint of  $b_u$ . Assign  $uv$  to  $b_u$ .
- Type 2:**  $\ell_{uv}$  contains an interior point of  $b_u$  and the left endpoint of  $b_v$ , and the line of sight to the left of  $\ell_{uv}$  either sees no other bar or sees a bar  $b_{v'}$  where  $b_{v'}$  does not overlap  $b_v$ . Assign  $uv$  to  $b_u$ .
- Type 3:**  $\ell_{uv}$  contains an interior point of  $b_u$  and the left endpoint of  $b_v$ , and the line of sight to the left of  $\ell_{uv}$  connects  $b_u$  to a bar  $b_{v'}$  where  $b_{v'}$  overlaps  $b_v$ . In this case, there is a bar  $b_{u'}$  assigned to a vertex  $u' \in U$  such  $b_{u'}$  and  $b_v$  are aligned, and the continuation of  $\ell_{uv}$  through  $b_v$  is the left edge of a line of sight between  $b_v$  and  $b_{u'}$ . Assign  $uv$  to  $b_{u'}$ .
- Type 4:**  $\ell_{uv}$  contains an interior point of both  $b_u$  and  $b_v$ . It follows that there are bars  $b_{v'}$  and  $b_{u'}$  assigned to vertices  $v' \in V$  and  $u' \in U$  so that  $b_u$  can see  $b_{v'}$ ,  $b_{u'}$  and  $b_{v'}$  are aligned,  $\ell_{uv}$  contains the right endpoints of  $b_{v'}$  and  $b_{u'}$ , and  $\ell_{uv}$  is the right edge of a line of sight between  $b_{v'}$  and  $b_{u'}$ . Assign  $uv$  to  $b_{u'}$ .

For edges of Types 3 and 4, we say that  $b_{u'}$  *blocks* for  $b_u$ , since the presence of  $b_{u'}$  allows  $b_u$  to see two overlapping bars assigned to vertices in  $V$ . For Type 3, we say that  $b_{u'}$  *blocks on the left* and for Type 4 we say that  $b_{u'}$  *blocks on the right*.

It is clear that  $b_u$  is assigned at most two edges of Type 1, since the left endpoint of  $b_u$  can see at most two other bars. It is also clear that  $b_u$  can block on the left for at most one bar and can block on the right for at most one bar, so it is assigned at most one edge of Type 3 and at most one edge of Type 4. If  $b_u$  is able to see the left endpoint of two  $V$ -bars above, then those bars must overlap. Therefore, some bar blocks on the left for  $b_u$ , and that bar is assigned the edge joining  $x$  to the  $V$ -bar that is farther to the right (which is also the

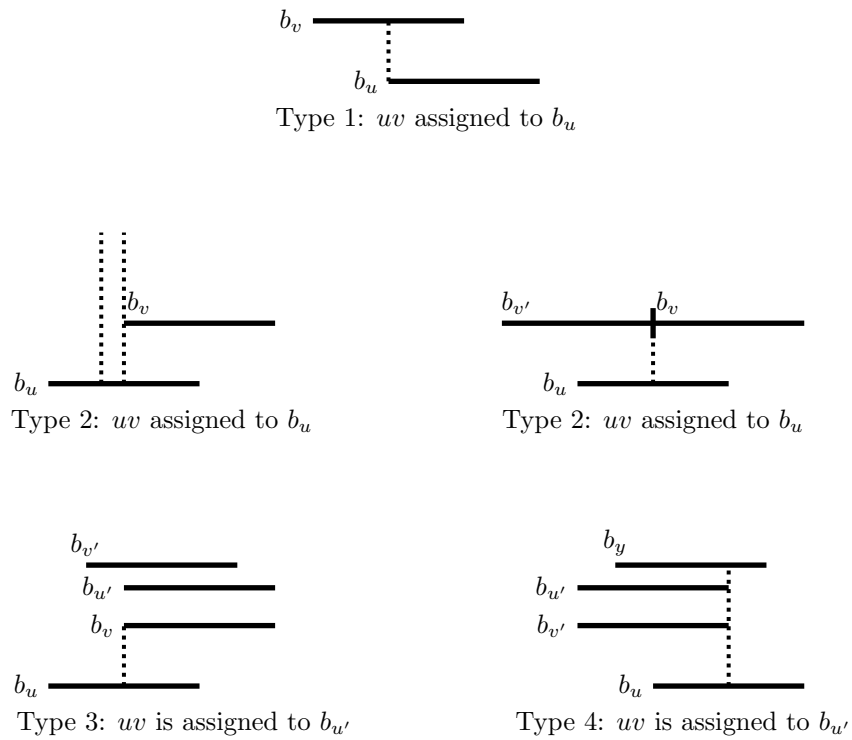


Figure 6: The assignment of edges to bars from the proof of Theorem 12. Each type can also occur with  $b_u$  above  $b_v$ .

lower of the two). Thus at most one of the upward visibilities of  $b_u$  is assigned to it as an edge of Type 2. Similarly, at most one of the downward visibilities of  $b_u$  is assigned to it as an edge of Type 2. Therefore  $b_u$  is assigned at most two edges of Type 2.

Suppose that  $b_u$  is assigned an edge of Type 2, and that this edge corresponds to a line of sight to a bar  $b_v$  above  $b_u$ . In this case, the left endpoint of  $b_u$  cannot see the left endpoint of a  $V$ -bar above  $b_u$  as such a bar would overlap with  $b_v$ . In this case,  $uv$  would be an edge of Type 3 that is assigned to some other bar. Similarly, the right endpoint of  $b_u$  cannot see the right endpoint of a  $V$ -bar above  $b_u$  since the right endpoint of  $b_u$  sees an interior point of  $b_v$  or a  $V$ -bar that is to the right of and below  $b_v$ . Therefore, if  $b_u$  is assigned an edge of Type 2 with a line of sight above  $b_u$ , then  $b_u$  cannot be assigned an edge of Type 3 or 4 from an  $U$ -bar that is above  $b_u$ . Similarly, if  $b_u$  is assigned an edge of Type 2 with a line of sight below  $b_u$ , then  $b_u$  cannot be assigned an edge of Type 3 or 4 from an  $U$ -bar that is below  $b_u$ . It follows that if  $b_u$  is assigned two edges of Type 2, which requires one above and one below  $b_u$ , then  $b_u$  cannot be assigned an edge of Type 3 or 4. It follows that  $b_u$  is assigned at most five edges.

Because  $R$  is a  $t$ -unit-bar visibility representation of  $K_{m,n}$  and each bar corresponding to a vertex in  $U$  is assigned at most five edges, it follows that  $mn \leq 5nt$ . Therefore  $t \geq \lceil \frac{m}{5} \rceil$ .  $\square$

When  $n = 2$ , the proof of Theorem 12 will actually yield an asymptotically best result.

**Theorem 13**  $ub(K_{2,m}) = \frac{m}{4} + o(m)$ .

**Proof:** The fact that  $ub(K_{2,m}) \leq \frac{m}{4} + o(m)$  is established in Theorem 11. To prove the lower bound we follow the proof of Theorem 12. We claim that each  $U$ -bar is assigned at most four edges. Assume to the contrary that  $b$  is an  $U$ -bar that is assigned five edges. It follows that  $b$  is assigned two edges of Type 1, one edge of Type 2, one edge of Type 3, and one edge of Type 4. Since a bar cannot be assigned an edge of Type 2 above (respectively below) and block for a bar that is above (respectively below), it follows that  $b$  blocks for two  $U$ -bars that are both above or both below  $b$ . Without loss of generality, assume that  $b$  blocks for two  $U$ -bars  $b'$  and  $b''$  that are below  $b$ . It follows that  $b$ ,  $b'$ , and  $b''$  all see the same  $V$ -bar  $\hat{b}$  and that  $b\hat{b}$ ,  $b'\hat{b}$ , and  $b''\hat{b}$  are all edges that are assigned to  $b$ . However, since  $n = 2$ , two of these three edges are the same, contradicting the assumption that each edge is assigned to a single pair of bars. Therefore each  $U$ -bar in a representation of  $K_{2,n}$  is assigned at most four bars and it follows that  $ub(K_{2,m}) \geq \lceil m/4 \rceil$ .  $\square$

We now give another lower bound on the unit bar visibility number of complete bipartite graphs. This bound is better than the bound from Theorem 12 when  $n \geq \frac{2}{3}m$  and is asymptotically best possible when  $n = m - o(m)$ . We note that this bound is an immediate consequence of Lemma 4 from [4]; we include the proof because it is conceptually different from those presented so far.

**Theorem 14** For  $m \geq n \geq 2$ ,

$$ub(K_{m,n}) \geq \frac{n}{2(m+n)}m + \frac{2}{m+n}.$$

Therefore, if  $n = m - o(m)$ , then  $ub(K_{m,n}) \geq \frac{m}{4} + o(m)$ .

**Proof:** Let  $R$  be a  $t$ -unit-bar visibility layout of  $K_{m,n}$ . Represent each visibility in  $R$  as a line segment joining two bars and then contract each bar to a point. This yields a planar graph. Because  $R$  is a representation of a bipartite graph, the resulting planar graph is also bipartite. Since there are at most  $t(m+n)$  bars in the layout, there are at most  $2t(m+n) - 4$  edges in the representation. Since  $K_{m,n}$  has  $mn$  edges, it follows that

$$\begin{aligned} t &\geq \left\lceil \frac{mn+4}{2(m+n)} \right\rceil \\ &\geq \frac{n}{2(m+n)}m + \frac{2}{m+n}. \end{aligned}$$

If  $n = m - o(m)$ , then it follows that

$$\begin{aligned} t &\geq \frac{m^2}{4m - o(m)} - \frac{o(m^2)}{4m - o(m)} + \frac{2}{2m - o(m)} \\ &= \frac{m}{4} - o(m). \end{aligned}$$

□

Theorems 11 and 14 give an asymptotically sharp result for  $K_{m,n}$  when  $m$  and  $n$  are asymptotic.

**Corollary 15** If  $n = m - o(m)$ , then  $ub(K_{m,n}) = \frac{m}{4} + o(m)$ .

## 5 Complete Graphs

Chang et al. [4] proved that  $b(K_n) = \lceil n/6 \rceil$ , using constructions derived from the solution to Heawood's empire problem [7, 8, 12]. They also provided a simpler construction to prove  $b(K_n) \leq \lceil n/6 \rceil + 1$ . Based on this simpler construction we prove the following bound on the unit bar visibility number of  $K_n$ .

**Theorem 16**  $\lceil \frac{n}{6} \rceil \leq ub(K_n) \leq \lceil \frac{n+4}{6} \rceil$ .

**Proof:** The lower bound follows from the facts that  $b(G) \leq ub(G)$  for all graphs and  $b(K_n) = \lceil \frac{n}{6} \rceil$ . For the upper bound, we give a brief description of the construction of an  $(m+1)$ -bar visibility representation of  $K_{6m}$  from [4]. Our only modification is the trivial observation that this construction can be obtained using  $(m+1)$ -unit-bars.



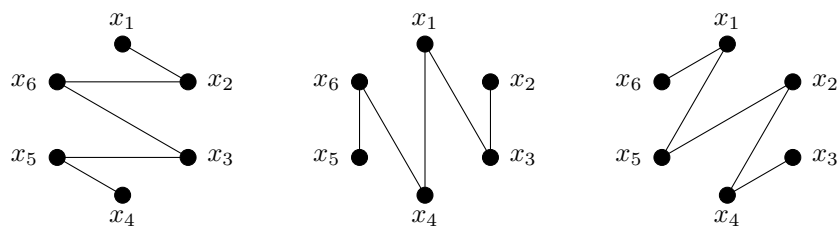


Figure 7: A decomposition of  $K_6$  into zigzag paths.

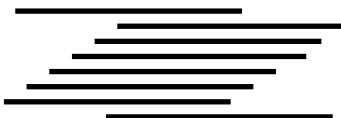


Figure 8: A unit bar visibility representation of  $P_{2m} \vee 2K_1$  with  $m = 3$ .

Partition  $V(K_{6m})$  into three sets of size  $2m$ , which we call  $V_1, V_2$ , and  $V_3$ . The complete graph  $K_{2m}$  has a decomposition into  $m$  copies of  $P_{2m}$ , which can be obtained by rotating a zig-zag path when the vertices are placed on a circle (See Figure 7).

The unit bar visibility representation of  $K_{6m}$  is then obtained by decomposing  $K_{6m}$  into  $3m$  copies of  $P_{2m} \vee 2K_1$  (the *join* of two graphs,  $G \vee H$ , is the graph obtained from the disjoint union of  $G$  and  $H$  by adding all edges joining  $V(G)$  and  $V(H)$ ). Each copy of  $P_{2m} \vee 2K_1$  will consist of a copy of  $P_{2m}$  from the path-decomposition of  $G[V_i]$  and two vertices from  $V_{i-1 \pmod{3}}$  for some  $i \in [3]$ . Provided that the sets of two vertices from  $V_{i-1}$  are pairwise disjoint, each vertex lies in  $m + 1$  of the copies of  $P_{2m} \vee 2K_1$ . A disjoint union of  $3m$  copies of the unit bar visibility representation of  $P_{2m} \vee 2K_1$  in Figure 8 with appropriate vertex-labels for the bars completes the construction.

When  $n$  is not divisible by 6, a representation of  $K_n$  can be obtained from the representation of  $K_{6\lceil n/6 \rceil}$  by deleting the bars assigned to vertices not in  $K_n$ . Therefore  $ub(K_n) \leq \lceil n/6 \rceil + 1$  for all  $n$ . It remains to show that we can improve this bound by 1 when  $n \equiv 1 \pmod{6}$  or  $n \equiv 2 \pmod{6}$ . We will assume that  $n \equiv 2 \pmod{6}$  since a representation of  $K_{6m+1}$  can be obtained from a representation of  $K_{6m+2}$  by deleting the bars assigned to the extra vertex. The reader is advised to consult Figure 9 to see an explicit example of the construction.

Let  $n = 6m + 2$ . We begin with a decomposition of  $K_{6m}$  into  $3m$  copies of  $P_{2m} \vee 2K_1$ . For  $i \in [3]$ , let  $V_i = \{v_1^i, \dots, v_m^i\}$ . Given  $i \in [3]$  and  $j \in [m]$ , let  $H_{i,j}$  be a copy of  $P_{2m} \vee 2K_1$  where

1. the path  $P_{2m}$  is the zigzag path on  $V_i$  with endpoints  $v_j^i$  and  $v_{m+j}^i$ , and
2. the copy of  $2K_1$  has vertex set  $\{v_j^{i-1 \pmod{3}}, v_{m+j}^{i-1 \pmod{3}}\}$ .

For each  $H_{i,j}$  we construct a copy of the representation in Figure 8, where

1.  $v_j^i$  is assigned to the second bar from the bottom (the bottom bar of the path);
2.  $v_{m+j}^i$  is assigned to the second bar from the top (the top bar in the path);
3.  $v_j^{i-1 \pmod{3}}$  is assigned to the bottom bar in the representation;
4.  $v_{m+j}^{i-1 \pmod{3}}$  is assigned to the top bar in the representation.

Note that in Figure 8 the second bar from the bottom is to the left of the top bar, and the second bar from the top is to the right of the bottom bar. Furthermore, it is possible to compress the arrangement so that 1) the second bar from the bottom is to the left of the top bar by exactly  $\frac{1}{12m}$ , and 2) the second bar from the top is to the right of the bottom bar by exactly  $\frac{1}{12m}$ .

To complete the construction, we stack the representations of  $P_{2m} \vee 2K_1$  so that every bar in the representation of  $H_{i,j}$  is below every bar in the representation of  $H_{i',j'}$ , when  $i < i'$ , or  $i = i'$  and  $j < j'$ . Furthermore, for  $i \in [3]$  and  $j \in [m-1]$  arrange the representations of  $H_{i,j}$  and  $H_{i,j+1}$  so that the second bar from the bottom of  $H_{i,j+1}$  is to the right of the topmost bar of  $H_{i,j}$  by exactly  $\frac{1}{12m}$ . Similarly, for  $i \in [2]$  arrange the representations of  $H_{i,m}$  and  $H_{i+1,1}$  so that the second bar from the bottom of  $H_{i+1,j}$  is to the right of the topmost bar of  $H_{i,m}$  by exactly  $\frac{1}{12m}$ . This yields a representation of  $K_{6m}$  in which a single bar can be added above all other bars that will see the highest and second lowest bars of  $H_{i,j}$  for all  $i \in [3]$  and  $j \in [m]$ . Similarly a single bar can be added below all other bars that will see the lowest and second highest bars of  $H_{i,j}$  for all  $i \in [3]$  and  $j \in [m]$ . By construction, these bars will both see bars assigned to each vertex in  $V_1 \cup V_2 \cup V_3$ . Letting  $K_n$  have vertex set  $V_1 \cup V_2 \cup V_3 \cup \{x, y\}$ , we place a bar for  $x$  and a bar for  $y$  in those positions. The addition of an extra bar for  $x$  that sees the bar for  $y$  finishes the construction.  $\square$

## 6 Open Questions

We conclude with some final remarks and open questions. The first questions come from [6] and [5].

**Question 1** *Is there a simple characterization of unit bar visibility graphs? Is there a simple characterization of unit rectangle visibility graphs?*

Regarding planar graphs, we ask if the bounds of Theorems 8 and 9 are sharp.

**Question 2** *Is there a planar graph  $G$  such that  $ub(G) = \left\lceil \frac{\Delta(G)+1}{3} \right\rceil + 2$ ? Is there a planar graph  $G$  with girth at least 7 such that  $ub(G) = \left\lceil \frac{\Delta(G)+1}{3} \right\rceil + 1$ ?*

In light of Corollary 15 and Theorem 13, we conjecture that the upper bound on  $ub(K_{m,n})$  is asymptotically sharp.

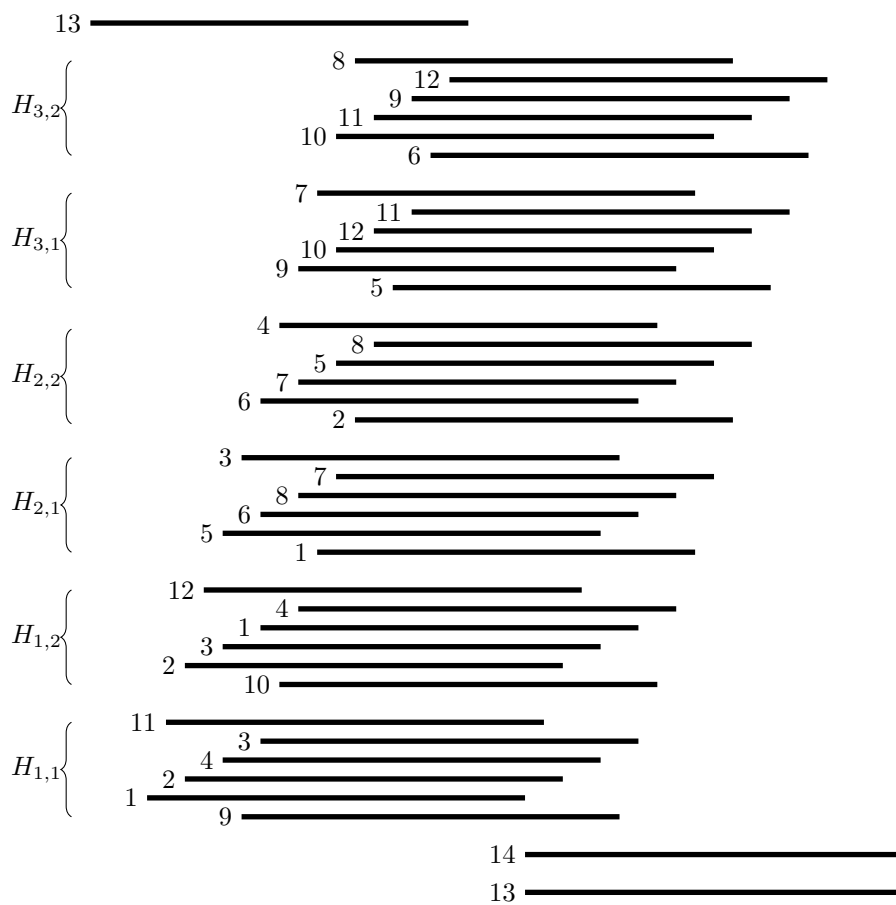


Figure 9: A 3-unit bar visibility representation of  $K_{14}$ . Bars assigned to  $v_j^i$  are labeled by the number  $4(i - 1) + j$ . Bars for  $x$  are labelled 13, and the bar for  $y$  is labelled 14.

**Conjecture 17** For  $m \geq n \geq 2$ ,  $ub(K_{m,n}) = \frac{m}{4} + o(m)$ .

Chang et al. [4] proved that if  $G$  is an  $n$ -vertex graph, then  $b(G) \leq \lceil n/6 \rceil + 2$ . Lovász [9] proved that every  $n$ -vertex graph can be decomposed into at most  $\lfloor n/2 \rfloor$  paths and cycles, and since paths and cycles are both unit bar visibility graphs, we have the following general bound.

**Proposition 18** If  $G$  is an  $n$ -vertex graph, then  $ub(G) \leq \lfloor n/2 \rfloor$ .

Among  $n$ -vertex graphs, the largest unit bar visibility number that we have found is for  $K_{1,n-1}$ , where  $ub(K_{1,n-1}) = \lceil (n-1)/3 \rceil$ . This leads to the following question.

**Question 3** What is the largest value that  $ub(G)$  can take when  $G$  is an  $n$ -vertex graph? In particular, is there an  $n$ -vertex graph  $G$  for which  $ub(G) > \lceil (n-1)/3 \rceil$ ?

We note that Algorithm 5 does not try to minimize the number of trees used in the decomposition. It is natural to seek a decomposition into unit bar visibility forests that is in some sense more efficient.

**Question 4** Is there an algorithm to efficiently determine the minimum number of elements in a decomposition of  $T$  into UBVTs in which no vertex lies in more than  $ub(T)$  of the elements?

Moving in a direction analogous to the idea of total interval numbers (see [1]), we might seek unit bar visibility representations of graphs in which the total number of unit bars is minimized, as opposed to the maximum number of bars that a single vertex requires.

**Question 5** Given a graph  $G$ , what is the minimum number bars in a unit bar visibility layout where the bars can be labeled by  $V(G)$  giving a visibility representation of  $G$ ?

## References

- [1] T. Andreae and M. Aigner. The total interval number of a graph. *J. Combin. Theory Ser. B*, 46(1):7–21, 1989. doi:10.1016/0095-8956(89)90003-8.
- [2] M. Axenovich, A. Beveridge, J. P. Hutchinson, and D. B. West. Visibility number of directed graphs. *SIAM J. Discrete Math.*, 27(3):1429–1449, 2013. doi:10.1137/120864234.
- [3] O. V. Borodin. A proof of B. Grünbaum’s conjecture on the acyclic 5-colorability of planar graphs. *Dokl. Akad. Nauk SSSR*, 231(1):18–20, 1976.
- [4] Y.-W. Chang, J. P. Hutchinson, M. S. Jacobson, J. Lehel, and D. B. West. The bar visibility number of a graph. *SIAM J. Discrete Math.*, 18(3):462–471 (electronic), 2004/05. doi:10.1137/S0895480198343455.
- [5] A. M. Dean, J. A. Ellis-Monaghan, S. Hamilton, and G. Pangborn. Unit rectangle visibility graphs. *Electron. J. Combin.*, 15(1):Research Paper 79, 24, 2008. URL: [http://www.combinatorics.org/Volume\\_15/Abstracts/v15i1r79.html](http://www.combinatorics.org/Volume_15/Abstracts/v15i1r79.html).
- [6] A. M. Dean and N. Veysel. Unit bar-visibility graphs. In *Proceedings of the Thirty-Fourth Southeastern International Conference on Combinatorics, Graph Theory and Computing*, volume 160, pages 161–175, 2003.
- [7] P. Heawood. Map colour theorem. *Quart. J. Math.*, 24:332–338, 1890.
- [8] B. Jackson and G. Ringel. Solution of Heawood’s empire problem in the plane. *J. Reine Angew. Math.*, 347:146–153, 1984. doi:10.1515/crll.1984.347.146.
- [9] L. Lovász. On covering of graphs. In *Theory of Graphs (Proc. Colloq., Tihany, 1966)*, pages 231–236. Academic Press, New York, 1968.
- [10] C. S. J. A. Nash-Williams. Decomposition of finite graphs into forests. *J. London Math. Soc.*, 39:12, 1964.
- [11] R. Tamassia and I. G. Tollis. A unified approach to visibility representations of planar graphs. *Discrete Comput. Geom.*, 1(4):321–341, 1986. doi:10.1007/BF02187705.
- [12] W. Wessel. A short solution of Heawood’s empire problem in the plane. *Discrete Math.*, 191(1-3):241–245, 1998. Graph theory (Elgersburg, 1996). doi:10.1016/S0012-365X(98)00110-1.
- [13] L. Wiglesworth. *A study of unit bar-visibility graphs*. PhD thesis, University of Louisville, 2008.
- [14] S. Wismath. Characterizing bar line-of-sight graphs. In *Proceedings of the First Association for Computing Machinery Symposium on Computational Geometry*.