1-1-2017

On Barrier Graphs of Sensor Networks

Kirk Anthony Boyer
University of Denver

On Barrier Graphs of Sensor Networks

A Thesis

Presented to

the Faculty of the Daniel Felix Ritchie School

of Engineering and Computer Science

University of Denver

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

Kirk A. Boyer

March 2018

Advisor: Mario A. Lopez

Author: Kirk A. Boyer
Title: On Barrier Graphs of Sensor Networks
Advisor: Mario A. Lopez
Degree Date: March 2018

ABSTRACT

The study of sensor networks begins with a model, which usually has a geometric component. This thesis focuses on networks of sensors modeled as collections of rays in the plane whose use is to detect intruders, and in particular a graph derived from this geometry, called the *barrier graph* of the network, which captures information about the network's coverage. Every such ray-barrier sensor network corresponds to a barrier graph, but not every graph is the barrier graph of some network.

We show that any barrier graph is not just tripartite, but perfect. We describe how to find networks which have certain designated graphs as their barrier graphs. We show that the size of a minimum vertex cover (in this context called the resilience) of a given graph can yield information about whether and how one can find a sensor network whose barrier graph is the given graph. Finally, we demonstrate that barrier graphs have certain strong structural properties, as a result of the geometry of ray-barrier networks, which represent progress towards a full characterization of barrier graphs.

ACKNOWLEDGEMENTS

I would like to thank both my thesis advisor, Mario A. Lopez, and mentor, Paul Horn, for not just helping me through the technical parts of working on this thesis, but for supporting me as a student in general, and working to expand my academic horizons as I focused on this work.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Sensor Networks

A wireless sensor network consists of a collection of spatially distributed sensors deployed to monitor various physical conditions in their immediate surroundings, like temperature, air pressure, movement, etc. There are numerous applications, including, for example, region surveillance to detect intruders, communication control for cell phones, disaster management, and many military applications.

A proper understanding of the behavioral characteristics of these networks starts with an appropriate mathematical model. Various models of sensors have been considered. Some use discs in the plane [2], with obvious analogy to cell towers, while others use wedges to model floodlight-like behavior [1] or rays [9] to abstract the behavior of a laser beam design to perform intrusion detection.

Each model is chosen to represent the network's coverage: the total area covered by discs, the length of time certain places are illuminated by rotating floodlights, or locations that cannot be crossed without the culprit being detected. In the last example, a network is intended to present barriers between that intruder's base and the intended destination.

When the barriers are modeled as rays the resulting network is called a *ray-barrier* network, or just a *barrier* network.

A key observation of Kirkpatrick, Yang, and Zilles (hereafter KYZ)[9] was that a collection of rays forms a barrier if and only if some pair of rays forms a barrier. This yields a graph, the ***barrier graph*** for the ray-barrier network, with the rays as vertices and an edge for any pair that, on its own, forms a barrier.

As discussed in [9], these are tripartite graphs, an example of which is seen in Figure 1.1. Note that the straight segment $\overline{\alpha\beta}$ in Figure 1.1a is used to compute the graph (Figure 1.1b) but in no way implies that a path from α to β needs to be straight. A barrier forces *all* paths from α to β to cross at least one ray.

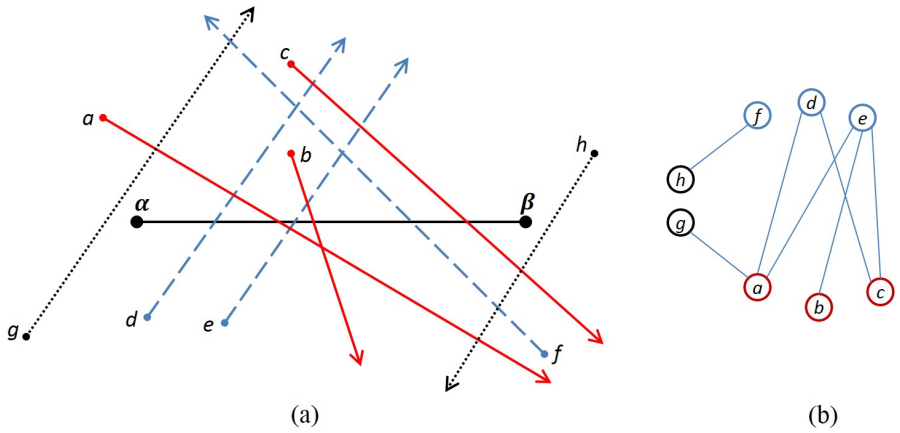


Figure 1.1: (a) A collection of ray sensors with start location α and target β, and (b) their barrier graph.

Starting with a graph G, we can ask whether G is a barrier graph, i.e., whether there is some arrangement A of rays, together with points α and β, so that G is the corresponding barrier graph. If the answer is yes, we say that the tuple $\langle A, \alpha, \beta \rangle$ is a *realization* of G, and we may write $G(A, \alpha, \beta)$. However, because any arrangement of rays with points α and β, can be rotated, scaled, and translated so that α and β are the origin and the point $(0, 1)$ (or any other convenient pair of points) without changing the corresponding barrier graph, we will usually just refer to the arrangement A as the realization of G, where α and β are understood.

Kirkpatrick and Bereg [2] introduced the notion of a sensor network's *resilience*, which is the minimum number of sensors whose removal permits a path (not necessarily straight) between the chosen points. Since barriers are formed by pairs of rays [9], a ray-barrier network's resilience is the size of a minimum vertex cover of the barrier graph. One can also ask whether knowing the size of a minimum vertex cover without having a realization in hand can tell us whether a realization *could* be found, which we explore in Chapter 4.

In general graphs, the minimum vertex cover problem is *NP-hard* [8], but as KYZ have shown, the ray barrier resilience (and in particular, a set of rays which witnesses the resilience) of an arrangement of rays can be found efficiently by making use of geometric information about the arrangement in addition to the barrier graph; this suggests quite strongly that the underlying geometric structure greatly limits the graphs which can arise as a barrier graph. KYZ's algorithm, which takes $O(n^2m)$ time (n is the number of rays and m is the number of barriers they form, i.e. vertices and edges), will be discussed in Chapter 2.

It is ultimately the question of which graphs can be realized as barrier graphs which will be the main focus of this thesis. We will first show that any barrier graph belongs to the special class of graphs called perfect graphs, a class on which some algorithmically difficult problems become efficiently solvable. Then we will show how to find a realization for some of the usual classes of bipartite graphs, including complete bipartite graphs, paths, even-length cycles and trees. In addition, and in spite of how natural these constructions for many well-recognized bipartite graphs are, it turns out that almost all bipartite graphs are not barrier graphs, which we show in Chapter 5.

1.2 Preliminaries

1.2 Definitions, Notation, and Terminology

While certain terms will be defined elsewhere in this thesis, this section will collect the more fundamental or basic ones.

Definition 1.2.1 (Basic Graph Terms)**.**
A **graph** G *is a pair of sets* V *(the* **vertices***) and* E *(the* **edges***) of* G*, so that the elements of* E *are unordered pairs of elements of* V*. If only the name of the graph* G *has been explicitly referenced before, then we write* $V(G)$ *and* $E(G)$ *for the vertices and edges of* G*. When* $(u, v) \in E(G)$*, we say* u *and* v *are* **adjacent***, and we also say that* u *and* v *are* **neighbors***.*

We say that G *is* **finite** *if* $|V(G)|$ *is finite. In this thesis, all graphs are finite. Also,* n *will always refer to the number of vertices,* $|V(G)|$*, and* m *to the number of edges,* $|E(G)|$*.*

A graph H *is a* **subgraph** *of* G *if* $V(H) \subseteq V(G)$ *and* $E(H) \subseteq E(G)$*, or alternatively, if there is an injective function* $\nu : V(H) \to V(G)$ *so that* $(u, v) \in E(H) \implies (\nu(u), \nu(v)) \in E(G)$*. In the case* H *is a subgraph of* G*, we may write* $H \leq G$*.*

If whenever $u, v \in V(H)$ *and* $(u, v) \in E(G)$ *we have that* $(u, v) \in E(H)$*, then we say* H *is an* **induced** *subgraph of* G*. Alternatively,* H *is an induced subgraph if* H *is a subgraph such that* $\forall u, v \in V(H)$ *we have* $(u, v) \in E(H) \iff (\nu(u), \nu(v)) \in E(G)$*.*

The **degree** *of a vertex* $v \in V(G)$*, written* $\deg(v)$*, is the number of edges containing* v*.*

The **neighborhood** *of a vertex* $v \in V(G)$ *is the set of neighbors of* v *and is written* $\mathcal{N}_G(v)$*, or* $\mathcal{N}(v)$ *when the graph* G *is clear from context.*

The **complement** *of a graph G is another graph H with $V(H) = V(G)$ and $(u, v) \in E(H)$ iff $(u, v) \notin E(G)$, for each pair $u, v \in V(G)$.*

Definition 1.2.2 (Graph Classes)**.** *A* **path** *graph is such that the vertices can be labeled $V(G) = \{v_1, \ldots, v_k\}$ in such a way that the edges of G are just $\{(v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_k)\}$, and no vertex is repeated (all $v_i \neq v_j$). We often write paths with the notation $v_1 - v_2 - \ldots v_{k-1} - v_k$. A* **cycle** *graph is like a path graph, but with the extra edge (v_k, v_1). The* **length** *of a path or cycle is the number of edges it contains.*

Let H be a cycle subgraph of G. A **chord** *of the cycle H is an edge $(u, v) \in E(G) \setminus E(H)$, i.e. an edge that goes across the cycle without being an edge of the cycle itself. A* **hole** *in G is an induced cycle subgraph $H \leq G$ with no chords. An* **antihole** *is just the graph complement of a hole, i.e. an induced subgraph missing all the edges of a cycle and containing all other edges (note that antiholes often contain many subgraphs that are also cycles).*

When every pair of vertices $u, v \in V(G)$ is such that there is a path in G with endpoints u and v, we call the graph G **connected***. The* **distance** *between two vertices u and v in a connected graph is the length of the shortest path with endpoints u and v, and is written $d(u, v)$*

If G has no cycles and is connected, then G is called a **tree***. In a tree graph, vertices of degree 1 are called* **leaves***. Often a particular vertex r will be designated as the* **root***, in which case the tree T is said to be* **rooted at** *r, and we may write $T = T(r)$.*

A **subtree** *$T(v)$ of a rooted tree $T(r)$ is another rooted tree, with root $v \in V(T(r))$, formed by taking the induced subgraph of $T(r)$ of all vertices for which the path to the root*

r *must pass through* v*. This is called the* subtree of $T(r)$ rooted at v*. The* t^{th} **level** *of a rooted tree* $T(r)$ *is the set of vertices at distance* t *from the root.*

In a rooted tree, if the shortest path from v *to the root goes through* u*, then we say* u *is an* **ancestor** *of* v*, and that* v *is a* **descendant** *of* u*. If* u *and* v *are also neighbors, we say* u *is the* **parent** *of* v *and that* v *is a* **child** *of* u*. Although it is less common, we also refer to the parent of a parent as a* **grandparent***.*

A **proper coloring** *of the vertices of* G *is an assignment of colors to vertices so that no two adjacent vertices have the same color. The* **chromatic number** $\chi(G)$ *of a graph is the smallest number of colors so that there is a proper vertex coloring of* G *using only* $\chi(G)$ *colors.*

An **independent set** *in* G *is a set of vertices containing no adjacent pair, and a* **clique** *is a set of vertices so that every pair is adjacent. In case we want to specify the size of such a set, we prefix with* k*-, as in* 3*-clique (a triangle graph). The* **clique number** $\omega(G)$ *is the size of the largest clique in* G*.*

A graph is k-partite *if there is a partition of* $V(G)$ *into* k *disjoint independent sets. Colloquially, a* 2*-partite graph is called* bipartite *and a* 3*-partite graph is called* tripartite*. A* bipartition *of a bipartite graph* G *is a particular choice of such a partition into* k *independent sets.*

A **perfect graph** G *is one such that in every induced subgraph* H *of* G*,* $\omega(H) = \chi(H)$*.*

Notation 1.2.1 (Geometric Notation)**.** *If* p *and* q *are two points, then* $\overline{pq}$ *is the line segment between them, and* ℓ_{pq} *is the line between them.* ℓ_{pq}^{-} *is the region of the plane to the left of*

ℓ_{pq}, and ℓ^+_{pq} is the region of the plane to the right of ℓ_{pq}. If either p or q is a ray (or both are), this notation will refer to the line through the anchor *of the ray.*

For any object with a supporting line (notably rays and line segments), which is just the unique line containing the object, we write $\ell(\cdot)$. So, if r is a ray, then $\ell(r)$ is the line containing r.

An object (usually a line, ray, or segment) **stabs** *a set of other objects if it intersects each one. When we say an object stabs a* subset *S of a set A of objects, we mean it intersects elements of S and does* not *intersect other elements of A.*

Note that all geometric terms in this thesis are in the context of the Euclidean plane.

1.2 Tripartite Structure of Barrier Graphs

One of the key observations of KYZ is that not only does a collection of ray sensors yield a barrier graph, but this graph must be tripartite. In order to show this, they constructed a 3-coloring of a barrier graph as follows (see Figure 1.1 for an example). Consider any arrangement of rays and a pair of points α, β – the start and target points – and rotate everything so that the segment $\overline{\alpha\beta}$ is horizontal. Rays will be colored red if they intersect $\overline{\alpha\beta}$ from above, blue if they intersect $\overline{\alpha\beta}$ from below, and black if they don't intersect $\overline{\alpha\beta}$.

KYZ showed that with this coloring, barriers can only form from intersecting pairs of differently colored rays. Furthermore, if red and blue rays intersect they always form a barrier, but if one of the rays is black, then a barrier is only formed when the intersection is on the same side of the supporting line of $\overline{\alpha\beta}$ as the other ray's anchor point. Since edges in the barrier graph are exactly these barrier pairs, the colors give three sets of vertices

connected by edges only to elements of a differently colored set. We will refer to this coloring for any barrier graph.

Viewing barrier graphs through the lens of this coloring provides a glimpse of how the graph structure depends on the rays' geometry, and allows us to find properties that prevent a graph from being a barrier graph, starting with the following proposition.

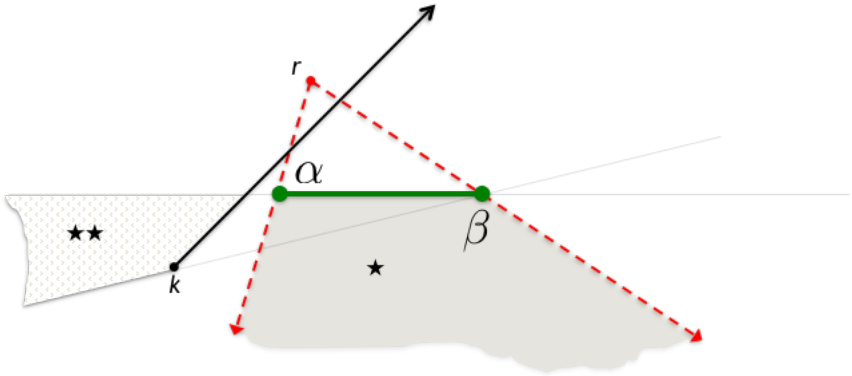


Figure 1.2: A barrier graph may not contain an induced $r - k - b$ path. The bold segment is the segment $\overline{\alpha\beta}$. The region labeled $\star$ cannot contain the anchor of k. The anchor of b must be located in the region labeled $\star\star$ as, otherwise, it will either not be blue or it will intersect k above $\ell(\overline{\alpha\beta})$. This forces b to also intersect r.

Proposition 1.2.1. *Let G be a barrier graph, and fix a realization with a particular set of rays and the path endpoints α and β. Then there is no induced length 2 path $r - k - b$ where r is red, k is black, and b is blue.*

Proof. Fix a realization of G, so that the coloring of the vertices of the barrier graph is fixed, and suppose $r - k - b$ is an induced path of length 2 in G, with r red, k black, and b blue.

Because r is red, it must be anchored above the bolded segment $\overline{\alpha\beta}$, and must also intersect $\overline{\alpha\beta}$. Therefore, its ray r lies somewhere in the wedge bounded by the dashed lines in Figure 1.2. Then, since k forms a barrier with r, it cannot be placed in the shaded region

labeled $\star$ for, otherwise, it would either intersect $\overline{\alpha\beta}$ (thus being blue instead of black) or intersect r below $\overline{\alpha\beta}$ (thus not forming a barrier with r).

k can either start above or below $\ell(\overline{\alpha\beta})$, as long as k intersects r above $\ell(\overline{\alpha\beta})$ and crosses $\ell(\overline{\alpha\beta})$ so that a proper intersection with b is possible. We proceed with the case that has the anchor of k below $\ell(\overline{\alpha\beta})$ and its supporting line to the left of $\overline{\alpha\beta}$ (see Figure 1.2). The argument for each of the other possibilities is similar.

Since b must intersect k below $\ell(\overline{\alpha\beta})$ to form an edge with it, it must be anchored in the region labeled $\star\star$, since were it anchored elsewhere, it would either not intersect $\overline{\alpha\beta}$ (and thus not be blue) or not intersect k below $\ell(\overline{\alpha\beta})$. But any blue ray anchored in $\star\star$ would intersect r, which means that the vertices r, k, and b would induce a triangle instead of a path. So there could not have been such an induced subgraph. □

Proposition 1.2.2. *If G is a barrier graph, then any induced subgraph is a barrier graph.*

Proof. This follows immediately from the fact that any pair of rays forms a barrier independently of all the other rays in the collection. □

Proposition 1.2.1 will be our most basic tool to connect the geometry of ray sensor arrangements to the graph structure of their barrier graphs. Proposition 1.2.2 is more graph theoretic, but together with Proposition 1.2.1 leads to the following stronger structural result about barrier graphs.

Proposition 1.2.3. *No graph containing a chordless cycle of odd order ≥ 5 is a barrier graph.*

Proof. Let $G = (V, E)$ be an odd cycle graph with $|V| \geq 5$ and with no chords, and suppose G is a barrier graph. Fix any realization of G, and let R, B, and K be the sets of red, blue, and black vertices respectively. R, B, and K are all nonempty since otherwise G is bipartite, which odd cycles are not.

For subsets $C, D \subseteq V$, define $C_D := \{c \in C \mid \mathcal{N}(c) \subseteq D\}$, i.e. those elements of C whose neighborhood consists only of elements of D.

Suppose every element of K has two neighbors of the same color.

Then K is the disjoint union $K_R \dot{\cup} K_B$. If we define $R' = R \cup K_B$ and $B' = B \cup K_R$, then R' and B' partition V into disjoint sets.

There are no edges between two elements of R' or between two elements of B', so we have found a bipartition of G, a contradiction.

But then there is an element $k \in K$ with differently colored neighbors $r \in R$ and $b \in B$. So, $r - k - b$ is an induced subgraph (since G is longer than a triangle and is chordless), which contradicts, by Proposition 1.2.1, that G was a barrier graph.

Thus, odd cycles of order ≥ 5 are not barrier graphs and, by Proposition 1.2.2, no graph with such an induced subgraph can be a barrier graph. □

While Proposition 1.2.3 prevents certain tripartite graphs from being barrier graphs, it also leads to an interesting connection to perfect graphs. Recall that a graph is perfect if the chromatic number and clique number of every induced subgraph are equal.

The celebrated strong perfect graph theorem [4], of Chudnovsky, Robertson, Seymour and Thomas, gives an alternate structural characterization of perfect graphs by showing that they are the same as the so-called Berge graphs. A Berge graph is a graph which has no odd hole (a hole with an odd number of vertices) or odd antihole with length longer than 3 as an induced subgraph.

From Proposition 1.2.3 we have that barrier graphs have no odd hole of size ≥ 5. Now, the cycle C_5 has itself as its graph complement, so this also rules out size-5 antiholes in any barrier graph. If we can also rule out antiholes of larger odd size in any barrier graph, then we will have shown that barrier graphs are perfect.

Lemma 1.2.4. *If G is a barrier graph, then G has no antiholes of size $2k+1$ for any $k > 2$.*

Proof. We may exclude odd antiholes of size 5 by the comment above that C_5 is its own complement, and so is an antihole.

First, we'll show that antiholes of size $2k+1$ have chromatic number $k+1$. Label the vertices of an antihole H of size $2k+1$ as $v_1, \ldots, v_{2k+1}$ so that each v_i and v_{i+1} are adjacent in the cycle which is the graph complement of H. Then v_i and v_{i+1} are not adjacent in H, so we can assign the color j to each vertex $2j$ and $2j-1$, for $1 \leq j \leq k$. This leaves one vertex not in a color pair, which needs its own color. So we have used $k+1$ colors to properly color H.

Now, fix any proper coloring of H. If v_1 is colored c, then because v_1 is adjacent to all other vertices except v_2 and v_{2k+1}, only these two may also be colored c. Moreover, these are adjacent, and so only one of them may be colored c. This argument applies to every vertex, for whatever its color is, and thus each color may appear on at most two vertices.

Therefore, the $2k + 1$ vertices of H require at least $k + 1$ colors, and the chromatic number of an antihole with $2k + 1$ vertices is $k + 1$. Since barrier graphs are tripartite, they cannot have induced subgraphs with chromatic number greater than 3, so for $k > 2$, barrier graphs have no antiholes. □

Corollary 1.2.5. *Every barrier graph is a perfect graph.*

Perfect graphs have various properties that distinguish them from tripartite graphs. First, unlike tripartite graphs, we already have algorithms to recognize perfect graphs efficiently[3]. Furthermore, many algorithmically hard problems, such as graph coloring, maximum clique, and maximum independent set have polynomial time algorithms for the class of perfect graphs[6].

We can take advantage of these algorithms in various ways. For instance, since S is an independent set for $G(V, E)$ if and only if $V - S$ is a vertex cover, it follows that a polynomial time algorithm for computing a maximum independent set of a perfect graph can be used, verbatim, to compute the resilience of a barrier graph, also in polynomial time.

This approach provides us with an compelling alternative to the algorithm in KYZ that is also efficient, but does not require explicit use of the geometric information. One could receive a barrier graph and compute its resilience in polynomial time without knowledge of the location and orientation of the sensors involved. This is impossible with the KYZ algorithm, which uses the underlying geometric information very strongly.

With the knowledge that barrier graphs are tripartite and perfect, a natural question is whether the converse is true: are all perfect tripartite graphs barrier graphs?

A natural class of perfect tripartite graphs is the class of bipartite graphs, which have both clique number and chromatic number equal to two. In Chapter 5 we will investigate the rigid structure of neighborhoods in barrier graphs, and will show that, in fact, almost all bipartite graphs are *not* barrier graphs. This is the probabilistic sense of "almost all", meaning that if you chose a bipartite graph (of any size) at random, the probability that what you get is not a barrier graph is 1.

1.2 The Geometric Dual Transformation

We will use a standard notion of geometric duality: the transformation $\mathcal{D}$ maps (non-vertical) lines in the plane to points in the plane and vice-versa. The image of an object r under $\mathcal{D}$ is referred to as the *dual* of r.

For the rest of this thesis we assume that rays, line segments, and lines are in general position (none are vertical, no three coincide in a point, and no two intersect in more than one point).

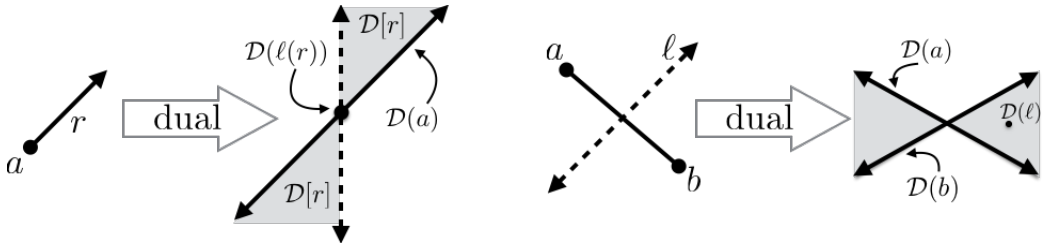


Figure 1.3: Dual transformation of a ray (left) and of a segment (right).

Proposition 1.2.6. *Let $\mathcal{D}$ be the transformation that maps the point $p : (a, b)$ to the line $D(p) : y = ax - b$ and the line $\ell : y = ax + b$ to the point $D(\ell) : (a, -b)$. For a ray or line segment r, let $\mathcal{D}[r] := \bigcup_{p \in r} \mathcal{D}(p)$. Then $\mathcal{D}$ has the following properties:*

(i) *p is above (resp. below) $\ell \iff \mathcal{D}(p)$ is below (above) $\mathcal{D}(\ell)$. [5]*

(ii) *$\mathcal{D}$ preserves incidences between points and lines. [5]*

(iii) *If the dual of a ray (resp. segment) r is defined as the union of the duals of the points comprising the ray (resp. segment), then the dual image $\mathcal{D}[r]$ is a "bow-tie" region, as shown in Figure 1.3.*

Properties (i) and (ii) follow immediately from the inequalities recording what it means for a point to be above or below a line. To see (iii), note that if a and b are the endpoints of a non-vertical segment then every x-value between the x-values of a and b is seen along the segment $\overline{ab}$, and thus every slope between the slope of $\mathcal{D}(a)$ and the slope of $\mathcal{D}(b)$ is the slope of some line in the dual image of $\overline{ab}$.

Since each point on $\overline{ab}$ is incident with the supporting line of $\overline{ab}$, their dual lines all intersect the dual point of this supporting line, which is the center of the bow-tie.

If a line ℓ intersects $\overline{ab}$ at a point p, then because the point $\mathcal{D}(\ell)$ lies on $\mathcal{D}(p)$, it also lies inside of the bowtie $\mathcal{D}[\overline{ab}]$ (see Figure 1.3). If the slope of ℓ is greater than that of $\ell(\overline{ab})$, the supporting line of $\overline{ab}$, then its dual will lie in the right-hand wedge of the bowtie because slopes are translated to x-coordinates, and similarly if its slope is smaller than $\ell(\overline{ab})$ then its dual will lie in the left-hand wedge.

Similarly, the dual of a ray is a bow-tie, whose bounding lines are the dual of the anchor $(\mathcal{D}(a))$ along with a vertical line through the dual of the supporting line of the ray $\mathcal{D}(\ell(r))$. If the ray points right, the bow-tie consists of all lines through $\mathcal{D}(\ell(r))$ with slope at least the slope of $\mathcal{D}(a)$, and otherwise it consists of all lines with slope at most the slope of $\mathcal{D}(a)$. Note that the bounding vertical line is not part of the bow-tie.

CHAPTER 2: PREVIOUS AND RELATED WORK

2.1 Finding Resilience in Polynomial time

As mentioned in Section 1.2.2, because barrier graphs are perfect graphs one can find minimum vertex covers of them in polynomial time; the procedure in general for perfect graphs, however, uses linear optimization, and so, while technically taking polynomial time, can be a very large polynomial in the input.

When we know a realization of a barrier graph $G(A, \alpha, \beta)$, we can take advantage of the geometry of the realization to find a vertex cover very efficiently: the algorithm of KYZ [9] takes $O(n^2m)$ time, where n is the number of ray sensors and m is the number of barriers they form.

The strategy of KYZ's algorithm is to find minimum vertex covers of a sequence of bipartite induced subgraphs of G and combine these to get a minimum vertex cover of G. This strategy relies on being able to directly use the geometric representation of vertices of G.

The proofs of the lemmas below are summarizations of the proofs of KYZ.

Lemma 2.1.1 ([9]). *Let $G(A, \alpha, \beta)$ be a barrier graph and suppose V_c is any vertex cover of G. Then there must exist lines $\ell_{\alpha*}$ and $\ell_{\beta*}$, through α and β, respectively, such that V_c contains all red and blue vertices with anchors lying to the left of $\ell_{\alpha*}$ or to the right of $\ell_{\beta*}$.*

Suppose we have rotated the arrangement so that α is to the left of β on a horizontal line. Also, suppose there is at least one red and at least one blue ray, for otherwise we could take $\ell_{\alpha *}$ and $\ell_{\beta *}$ to both be $\ell_{\alpha\beta}$.

Proof of Lemma 2.1.1. For any line $\ell_{\alpha r}$ through α and the anchor of a red ray r, if b is any ray anchored to the left of $\ell_{\alpha r}$ then r and b must intersect (because they both intersect $\overline{\alpha\beta}$, or otherwise they wouldn't both be non-black), and so must form a barrier together, and one or the other must belong to any vertex cover.

Hence, if we choose r so that it minimizes the angle $\angle\alpha\beta r$, then there will be no red rays anchored to the left of $\ell_{\alpha r}$ and any vertex cover must contain all blue rays anchored to its left. A symmetric argument shows how to choose a line $\ell_{\beta r'}$, this time choosing r' to minimize the angle $\angle\beta\alpha r$. □

Lemma 2.1.2 ([9])**.** *Let $\ell_{\alpha -}$ and $\ell_{\beta -}$ be arbitrary lines through α and β, respectively, and let $\widetilde{RB}$ denote the set of red and blue vertices lying to the left of $\ell_{\alpha -}$ or to the right of $\ell_{\beta -}$. Then the subgraph H of $G(A, \alpha, \beta)$ induced by the vertices $V(G) \setminus \widetilde{RB}$ is bipartite.*

Proof. Suppose H is not bipartite; then it contains an odd cycle. Since H is an induced subgraph of G, it is a barrier graph and is therefore tripartite with partite sets given by the colors red, blue, and black, meaning the odd cycle must contain a black ray k whose neighbors r and b on the cycle are red and blue, respectively.

Since the ray r is anchored between $\ell_{\alpha -}$ and $\ell_{\beta -}$, the segment of r between its anchor and where it intersects $\overline{\alpha\beta}$ is also between $\ell_{\alpha -}$ and $\ell_{\beta -}$. Also, because r and k form a barrier together, they must intersect along this segment, and thus their intersection point r' is above $\overline{\alpha\beta}$ and between $\ell_{\alpha -}$ and $\ell_{\beta -}$.

By a symmetric argument, the point b' at which b and k intersect is below $\overline{\alpha\beta}$ and is also between $\ell_{\alpha-}$ and $\ell_{\beta-}$.

But then k intersects two points which are between $\ell_{\alpha-}$ and $\ell_{\beta-}$ and which are on opposite sides of $\overline{\alpha\beta}$, and therefore k itself intersects $\overline{\alpha\beta}$ itself, contradicting that it was colored black.

So it can't have been that there is such a black ray, and thus it can't have been that H is not bipartite. □

Together, Lemmas 2.1.1 and 2.1.2 give us the tools to use KYZ's algorithm to find a minimum vertex cover of a barrier graph.

Algorithm 1 [9] KYZ's algorithm: BARRIER GRAPH MINIMUM VERTEX COVER

Input: A Barrier graph $G(A, \alpha, \beta)$
Output: Minimum vertex cover of $G(A, \alpha, \beta)$

$\widetilde{RB} \longleftarrow$ {red rays in A}
$VC_{temp} \longleftarrow$ minimum size vertex cover of $G(A \setminus \widetilde{RB}, \alpha, \beta)$.
$VC_{best} \longleftarrow VC_{temp} \cup \widetilde{RB}$
for each red vertex v **do**
  for each red vertex w **do**
    $\widetilde{RB} \longleftarrow$ { red and blue vertices in $\ell^-_{\alpha v} \cup \ell^+_{\beta w}$}
    $VC_{temp} \longleftarrow$ minimum vertex cover of $G(A \setminus \widetilde{RB}, \alpha, \beta)$
    if $|VC_{temp}| + |\widetilde{RB}| < |VC_{best}|$ **then**
      $VC_{best} \longleftarrow VC_{temp} \cup \widetilde{RB}$
    end if
  end for
end for

Suppose $G(A, \alpha, \beta)$ is a barrier graph and V_c is a minimum vertex cover of G. From the proof of Lemma 2.1.1 there are lines $\ell_{\alpha*}$ and $\ell_{\beta*}$ so that V_c contains $\widetilde{RB}$, i.e. all red and blue rays anchored left of $\ell_{\alpha*}$ or right of $\ell_{\beta*}$.

If there is a minimum vertex cover of G that contains all red vertices of $V(G)$, then KYZ's algorithm will find this cover in lines 1-3, because line 2 builds a minimum cover of the edges not involving red rays and on line 3 adds in the red rays. Since any minimum cover containing all red vertices must also cover each of these edges not involving red rays, the result here is minimum. Moreover, because the algorithm records the smallest cover seen through the rest of the algorithm (lines 4-11), this minimum is returned.

If instead there is not a minimum vertex cover of G containing all red vertices, lines 4-11 of the algorithm iterate through each possible vertex cover by looping over all pairs of red rays and constructing the lines through α and β of the proof of Lemma 2.1.2. This lemma is also the reason iterating in this way actually encounters each possible vertex cover, and thus why the algorithm is correct; a minimum vertex cover is just one with minimum size, and every vertex cover appears as the union of the red vertices in $\ell^-_{\alpha v} \cup \ell^+_{\beta w}$ together with a minimum vertex cover of the barrier graph without these red vertices, for some red vertices v and w.

The time complexity of the algorithm as written is actually $O(n^{2.5}m)$ instead of the $O(n^2m)$ promised above, because the critical section at lines 6-9 happen $O(n^2)$ times, i.e. once for each pair v, w of not-necessarily-distinct red rays.

This section of the algorithm (lines 6-9) itself takes $O(m\sqrt{n})$ time because the subgraph $H = G(A \setminus \widetilde{RB}, \alpha, \beta)$ is bipartite (by Lemma 2.1.2) and we can use the Hopcroft-Karp algorithm[7] to find a maximum matching of H and then convert this to a minimum vertex cover in $O(m\sqrt{n})$ time.

KYZ improve the time to $O(n^2m)$ by avoiding recomputing covers of successive bipartite subgraphs: on steps of the inner loop on line 5, these subgraphs differ by a single red

vertex and so have closely related minimum vertex covers that can be found by an augmentation of the previous cover (by adding or removing some vertices to $\widetilde{RB}$), and amortizing over all steps.

CHAPTER 3: REALIZING PARTICULAR GRAPHS

There are many barrier graphs for which a realization is not difficult to find. In this chapter, we present some classes of graphs and schemes for realizing them. Several of these classes are bipartite, but several others are identified instead by the size of their minimum vertex cover.

Note that while many of these constructions are natural, they are usually not unique.

3.1 Complete Bi-/Tripartite Graphs, Paths, and Cycles

A complete tripartite is perhaps the most natural barrier graph to realize.

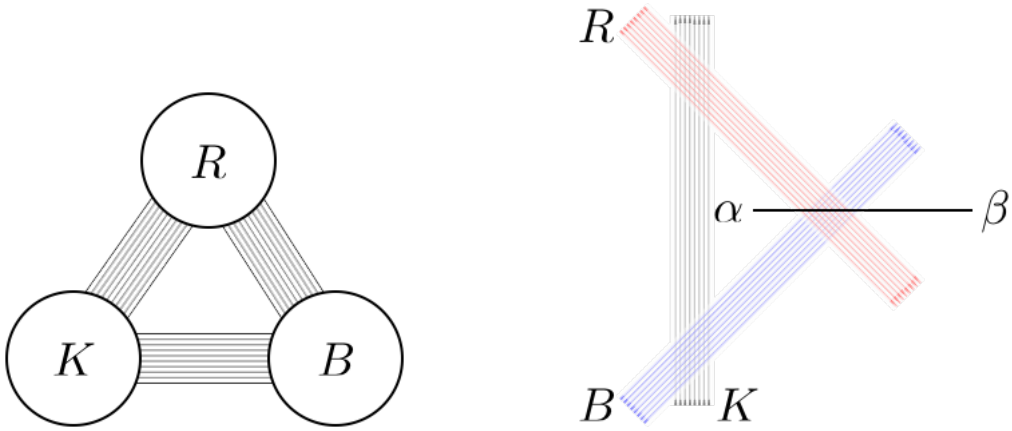


Figure 3.1: A realization (b) of a given complete tripartite graph (a).

Because induced subgraphs of barrier graphs are also barrier graphs, the construction for any complete bipartite graph is built from that for a tripartite barrier graph with two partite sets of the same size, and discarding the third partite set.

The natural construction for realizing a path graph is just as easy, and only uses two colors: alternate blue and red rays that each intersect two consecutive rays of the opposite color, and stop once you've reached as many rays as vertices in the graph (See Figure 3.2).

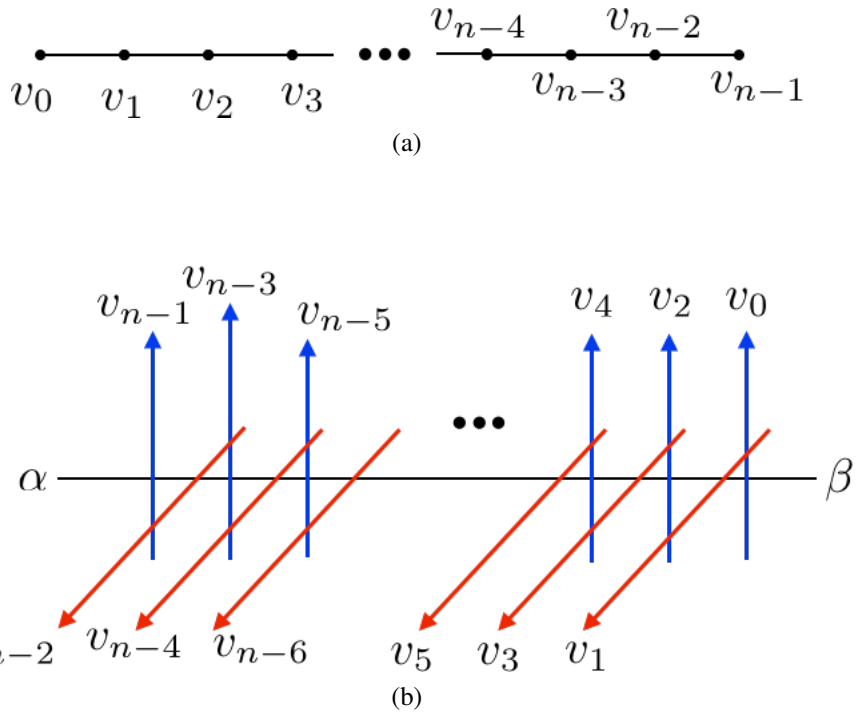
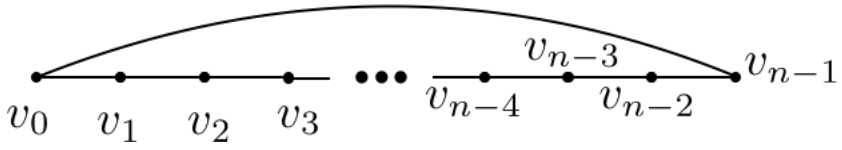


Figure 3.2: A realization (b) of a given path graph (a) with an even number n of vertices. The realization for odd n is the same but without one of the endpoint blue rays.

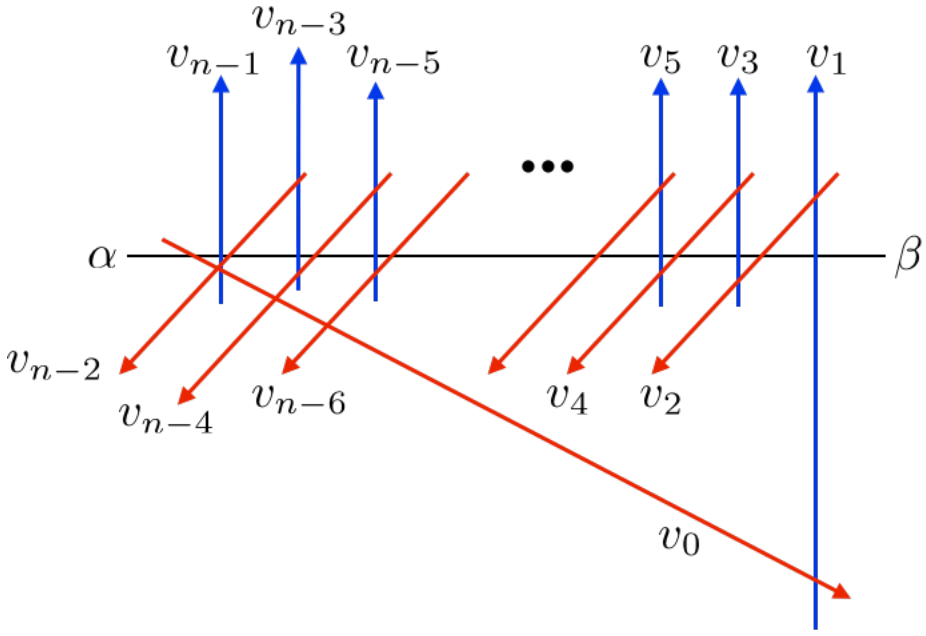
To realize a cycle of even length, a natural construction is (not surprisingly) very similar to that for a path. The main trick is to designate a particular vertex to connect the ends of a path into a cycle. As in Figure 3.3, given an even cycle graph on vertices $v_0, v_1, \ldots, v_{n-1}$, with edges $(v_i, v_{i+1})\forall i$ and (v_0, v_{n-1}), we build a path on the (odd-number many) vertices $v_1, \ldots, v_{n-1}$, and extend the anchor of one of the ends of this path so that there is a straight

line that goes above each of the anchors of the path endpoints and also above α. Placing a ray for v_0 of the opposite color from the path endpoints along this line completes the cycle.

Perhaps most interesting about the simplicity of this construction for arbitrarily long even-length cycles is the fact that there *cannot be* a construction for *odd-length* cycles longer than a triangle, as we saw in Proposition 1.2.3 of Section 1.2.2.



(a)



(b)

Figure 3.3: A realization (b) of a given even-length cycle graph (a).

3.2 Trees

To see how to find a realization of an arbitrary tree, we will first show how to find a realization of an arbitrary full k-ary tree. A k-ary tree of ***height*** h is a rooted tree where the root has degree k, every other non-leaf vertex has degree $k+1$, and where the root is distance h from every leaf. Our notation for the k-ary tree of height h will be T_k^h.

Every tree T is an induced subgraph of T_k^h, where $k = \max_{v \in V(T)} \deg(v)$ and $h = \max_{u,v \in V(T)} d(u,v)$. To see this, designate any vertex r of T as the root and assign it to the root of T_k^h and its children arbitrarily to children of the root of T_k^h. Then, for each neighbor of r, assign each of their respective unassigned neighbors (on the second level) to children of their assigned vertices, and so on. Each step like this is possible because k was chosen to be large enough to accommodate any degree in T and h was chosen to be large enough that the process doesn't reach a leaf of T_k^h until it reaches the end of the longest path in T, and therefore no vertex with unassigned children is ever assigned to a leaf of T_k^h.

By Proposition 1.2.2, induced subgraphs of barrier graphs are also barrier graphs; given a realization for T_k^h we may simply remove vertices not assigned by the process above to arrive at a realization for T.

The construction for a realization of T_k^h is recursive, and proceeds by levels in the tree (i.e. by distance from the root). For brevity, we will refer to a vertex and its corresponding ray (once this is chosen) as if they were the same object. Also, for convenience, we will number the vertices of the ℓ^{th} level from 1 to k^ℓ so that the children of each vertex are numbered consecutively, and children of lower-number vertices in level ℓ have lower number than children of higher-number vertices in level ℓ. Then we may refer to the i^{th} vertex of level ℓ as $v_i^{(\ell)}$.

In the process below, many choices of angle are immaterial; it is the relative location of intersections that matter. Because the process cuts convex polygons into multiple longer and thinner convex polygons in which later recursive steps will work, the description of the process is easier to follow when drawn as in the figures.

First, place a red ray for the root r. The root has k neighbors, so for each neighbor $v_i^{(1)}$ anchor a blue ray along r at even intervals from $\overline{\alpha\beta}$. These rays should not intersect before crossing $\overline{\alpha\beta}$. Each $v_i^{(1)}$ has k children ($v_j^{(2)}$), which will be made with red rays anchored along $v_i^{(1)}$. These red rays are placed so that they intersect the root ray r in the same left-right order that they intersect their parent ray $v_i^{(1)}$. See Figure 3.6 for an illustration of the rays for these first two levels of a 3-ary tree.

Note that placing each of these level-2 rays $v_j^{(2)}$ leaves a convex quadrilateral below it, which is bounded by its parent, by r, by itself, and by its next-highest-index sibling, if it has one. If $v_j^{(2)}$ has no next-highest-index sibling, the convex quadrilateral is completed by a line through the following two points: the intersection of the next-highest-index sibling of the parent of $v_j^{(2)}$ and a point further from r along the parent of $v_j^{(2)}$. For such a quadrilateral, call the side defined by $v_j^{(2)}$ itself the "***left***" side, and the side opposite this the "***right***" side, and call the other two sides the "***top***" and "***bottom***" sides (one is always below the other). For examples of such quadrilaterals, see Figure 3.4.

A quadrilateral like the one above, whose left side is the ray for red (resp. blue) leaf vertex v_j^i, is called the ***Extending Quadrilateral*** for $v_j^{(i)}$ if its bottom side is red (resp. blue), its top side is blue (resp. red), and any ray anchored on its left side (along $v_j^{(i)}$) and intersecting its top or bottom intersects only one color of ray (other than the ray it is anchored along).

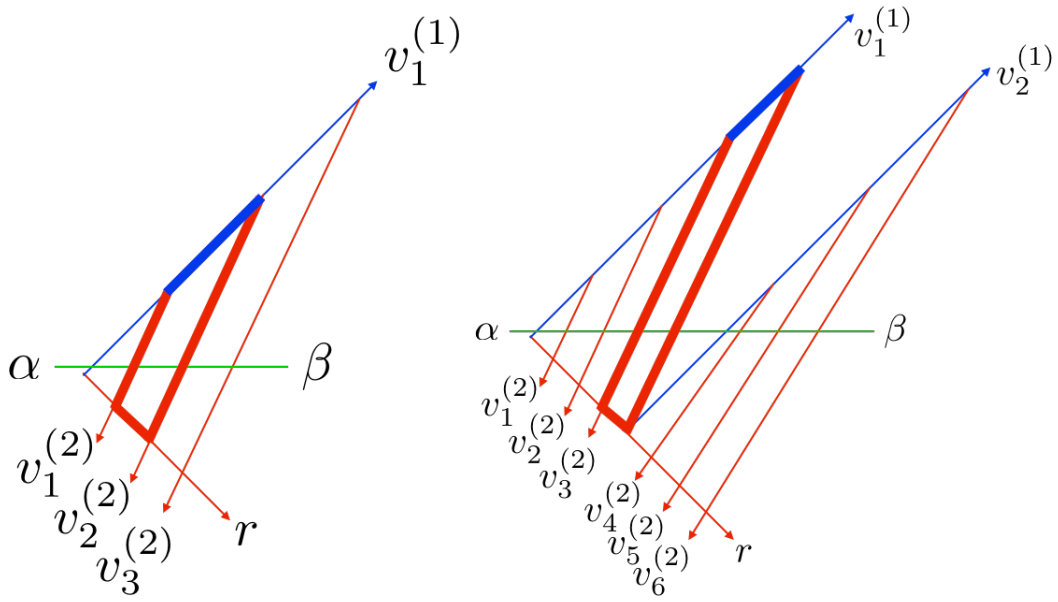


Figure 3.4: A normal Extending Quadrilateral (a) and one for the highest-index child of $v_1^{(1)}$ in a 3-ary tree of height 2 (b).

For example, the Extending Quadrilateral for $v_2^{(2)}$ in Figure 3.6 is bounded by $v_2^{(2)}$ itself, its sibling $v_3^{(2)}$, its parent $v_1^{(1)}$, and the root r. On the other hand, in the same figure the convex region corresponding to $v_3^{(2)}$ is bounded by itself, the root, its parent, and $v_2^{(1)}$, which is its parent's sibling.

The first tool we will need for our recursive construction is that it is safe to extend the construction by anchoring rays for a vertex v's children along v within this quadrilateral.

Invariant 3.2.1 (Extendable Tree Realization)**.** *Let A be a set of rays that realizes the tree T_k^h. A has the* Extendable Tree Realization *invariant if there is an Extending Quadrilateral for each leaf v of T_k^h.*

Lemma 3.2.1. *Let A be an Extendable Tree Realization for T_k^h, where $h \geq 2$. Then there is a set of rays $A' \supseteq A$ that is an Extendable Tree Realization for T_k^{h+1}.*

Proof. Let $v_j^{(i)}$ be the ray of any leaf of T_k^h and let Q be its Extending Quadrilateral.

Suppose $v_j^{(i)}$ is red. Because Q is an Extending Quadrilateral, each of the k children $u_1, \ldots, u_k$ of $v_j^{(i)}$ in T_k^{h+1} can be created by adding a blue ray anchored along $v_j^{(i)}$ above the segment $\overline{\alpha\beta}$ and through the top (blue) edge of Q; these rays intersect only red rays

In each case the anchor represents the only intersection of the new blue ray with a red one, and thus the only barrier created by the new ray is with it's parent $v_j^{(i)}$, so that we preserve the tree structure of the realized barrier graph. Since this can be done to add children to each leaf of T_k^h, this process will successfully produce a realization of T_k^{h+1}.

Each of the new rays forms the left side of a new quadrilateral, which has as its bottom side a segment of $v_j^{(i)}$, and as its top side a segment of the top side of Q. The right side of this quadrilateral is either $v_{j+1}^{(i)}$ (if $j < k$), or an auxiliary segment like the one in Figure 3.4b.

These new quadrilaterals are Extending Quadrilaterals Q_t for their corresponding rays u_t, the children of $v_j^{(i)}$ (See Figure 3.5). This is true for the following reasons:

- Their top sides are sub-segments of the top side of Q (an Extending Quadrilateral) and the only new rays above this top side are also blue rays, so rays anchored along a child and through the top of its quadrilateral will only intersect other blue rays.

- Their bottom sides are the red ray $v_j^{(i)}$ itself, and because of the angle of u_t, rays pointing downward and anchored along u_t can intersect only the rays below Q, which are guaranteed to all be of the same color (red).

For an example of placing such rays, see Figure 3.6. If instead $v_j^{(i)}$ is blue, repeat the same argument, but exchange the roles of red and blue, and above and below.

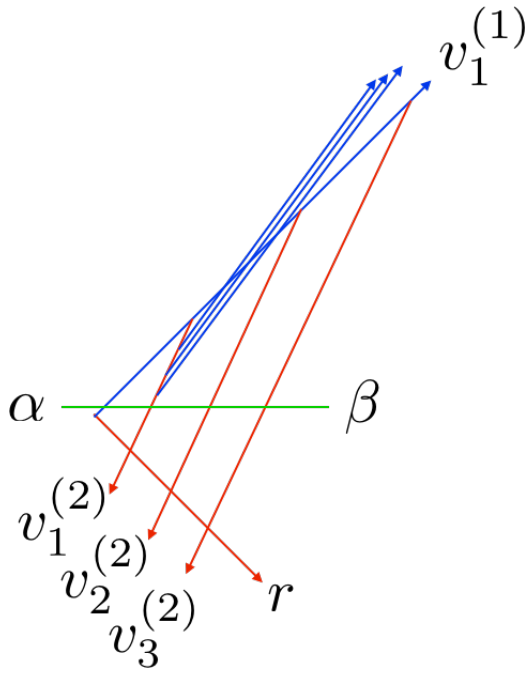


Figure 3.5: Adding the (blue) children of the ray $v_1^{(2)}$ within an Extending Quadrilateral in a way that leaves Extending Quadrilaterals for each child.

□

Lemma 3.2.1 shows that if there is any Extendable Tree Realization for T_k^2, then there is one for T_k^h for any h; the lemma required that $h \geq 2$ simply because otherwise the rays do not form the needed quadrilaterals.

The description of how to construct a realization of T_3^2 shown in Figure 3.6 is easily changed to T_k^2 for any k, and so we have shown that any T_k^h is realizable, and therefore any tree is realizable.

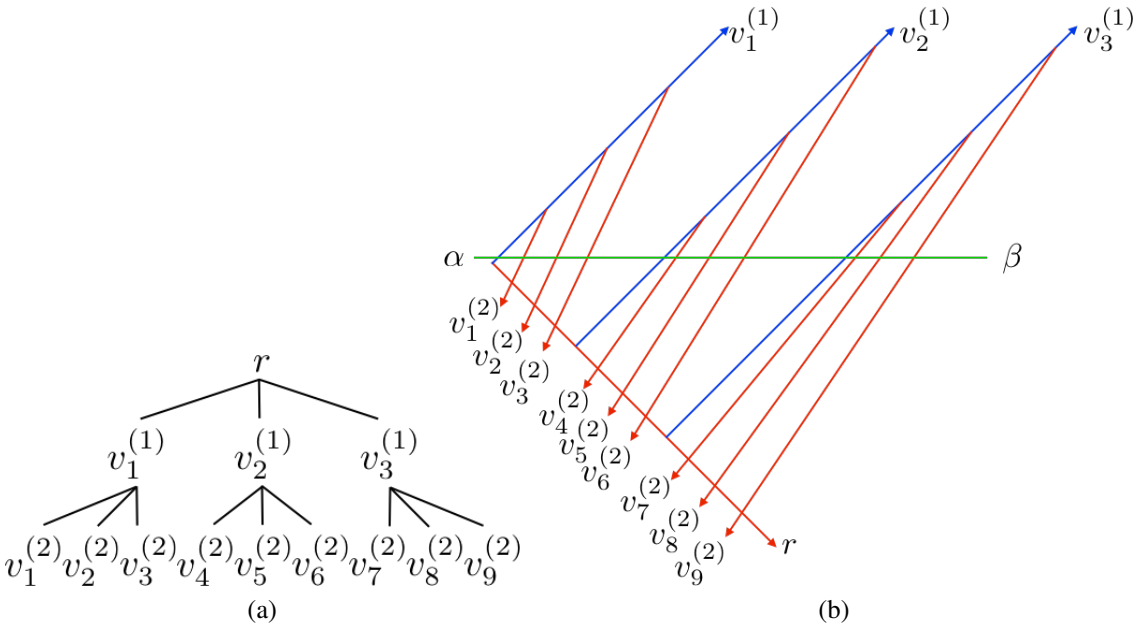


Figure 3.6: A realization (b) of a 3-ary height-2 tree with root r (a).

3.3 Further Realization Questions

One notable aspect of all the realizations of bipartite graphs in this section is that none of them requires three colors of rays, but of course there are alternative realizations of the same graphs that use three colors. For example, in any path, one can exchange the endpoints with black rays and still realize the path.

This raises the question of whether there exists a bipartite barrier graph for which every realization uses rays of three colors. Obviously if our standard coloring of an arrangement of rays only requires two colors, then the corresponding graph is bipartite, but the opposite

implication (though intuitive) is not as obvious, and if true requires proof. Nonetheless, this is our suspicion:

Conjecture 3.1. *If a bipartite graph G is realizable, then there is a realization for G requiring only two ray colors.*

One approach to this question (and others) would be to develop an algorithm which produces a realization of a given graph, if one exists, and reports failure if this is not possible. Analyzing this algorithm might lead to a constructive proof based on assumptions about the input graph. So far, such an algorithm is not known, nor is the complexity of this problem known.

Conjecture 3.2. *There is an algorithm that takes any tripartite graph G as input and either produces a realization of G or a certificate that G is not a barrier graph.*

CHAPTER 4: RESILIENCE

4.1 Introduction

Recall that the *resilience* of a sensor network is the number of sensors whose removal from the network prevents that network from providing its coverage. In the case of the ray-barrier sensor networks considered in this thesis, a network provides coverage if every path from the designated starting location α to the target location β crosses at least one ray sensor.

This idea of the resilience of a network was originally explored in sensor networks whose sensors were modeled as discs [2], which is useful, for example, in cell-tower networks to ensure customers have service unless some unlikely number of towers fail simultaneously. In disc sensor networks, where the network's coverage of a point is represented by the area overlapping at least one sensor, there is not as clear an abstraction to a graph as there is with ray-sensor barriers; because of the correspondence we have between ray-sensor networks and barrier graphs, the resilience of the network *is* the resilience of the graph.

Thus we can also consider the resilience that a network would have, given that it realizes a given graph. Suppose the graph class we consider trying to realize is the class of graphs with a fixed minimum vertex cover size r (which, for the purposes of this discussion, we call resilience even if the graph is not realizable). For which r, if any, are *all* graphs of resilience r realizable? In the general case this proves to be a fairly uninteresting

question as the 5 cycle, for instance, has resilience 3 and is not realizable. However the question seems much more interesting (and much less trivial) when restricted to the class of bipartite graphs.

In particular, it is not difficult to prove the following theorems, which we will prove in the next section.

Theorem 4.1.1. *All bipartite graphs of resilience 2 are realizable as barrier graphs*

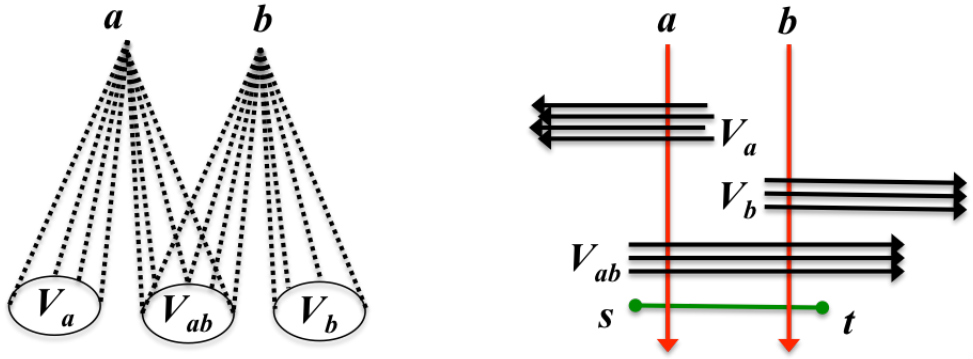
Theorem 4.1.2. *All bipartite graphs of resilience 3 are realizable as barrier graphs.*

4.2 Realizing Graphs of Resilience 2 and 3

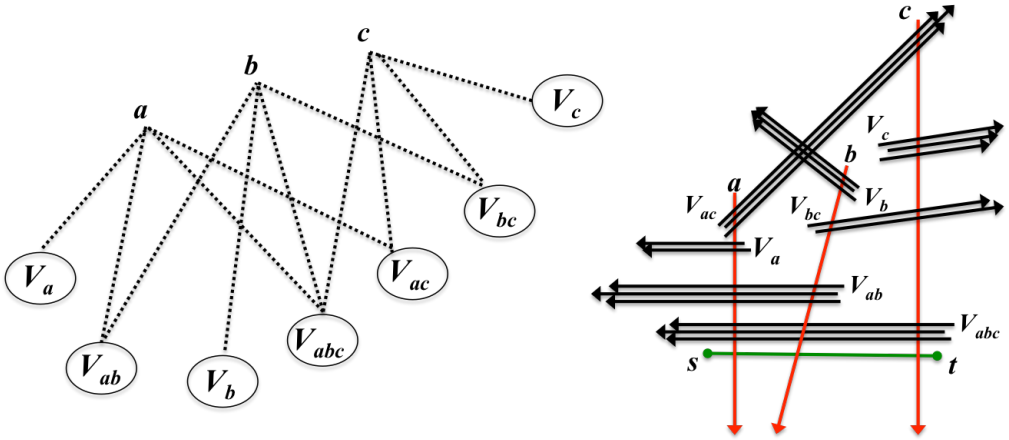
Connected graphs can, to some extent, be described in terms of their vertex covers, because the edges of vertices not in the cover exactly pick out a subset of the vertex cover. When the vertex cover is small, therefore, the set possible descriptions of neighborhoods in the graph also becomes small, as do descriptions of the neighborhoods themselves. For graphs of small resilience, we can thus consider all possible cases at once and build a general construction that works for any graph of that resilience.

To that end, we begin with the simplest case that is not a star graph, i.e. graphs with resilience 2.

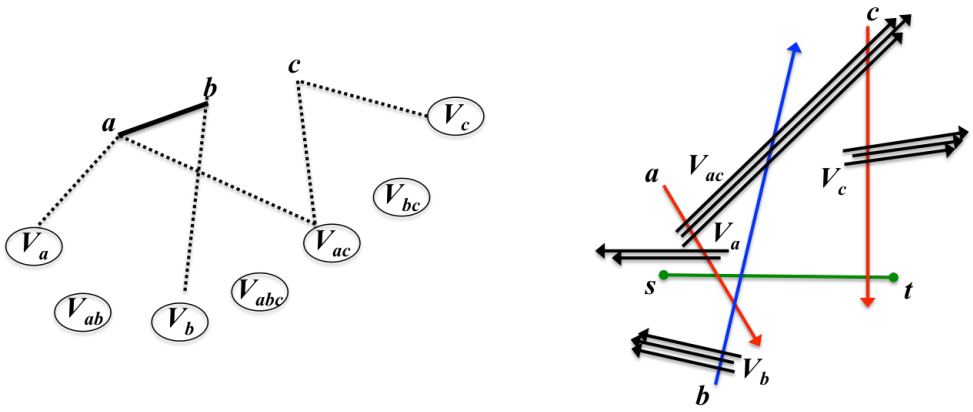
Proof of Theorem 4.1.1. See Figure 4.1a for explicit constructions of most of the cases that may arise. The notation used in the figure is as follows: in a resilience 2 graph, if $\{a, b\}$ is a vertex cover of the graph, we write V_a for the vertices incident on a but not in the cover, V_{ab} for the vertices incident on both a and b but not in the cover, and so on – and similarly for other resiliences.



(a) General resilience 2 bipartite graph with independent-set vertex cover and its realization



(b) General resilience 3 bipartite graph with independent-set vertex cover and its realization



(c) General resilience 3 bipartite graph with one edge between elements of the vertex cover, and its realization.

Figure 4.1: Generalized bipartite graphs with fixed resilience and their realizations

Because each of these sets of vertices V_* are disjoint from a vertex cover of the graph, they are all independent sets; between any two vertices in any V_* (possibly the same one), there is no edge, for otherwise one of these vertices would need to be in any vertex cover.

There are two cases for resilience 2 graphs: the vertex cover $\{a, b\}$ is an independent set (Figure 4.1a); or $(a, b) \in E(g)$, in which case the bipartite graph is a tree, because in this case V_{ab} must be empty (otherwise the graph has a triangle and cannot be bipartite). □

Proof of Theorem 4.1.2. If the resilience is 3, there are only three cases: the vertex cover $\{a, b, c\}$ is independent (Figure 4.1b); or the graph induced by $\{a, b, c\}$ has one edge (Figure 4.1c); or the graph induced by $\{a, b, c\}$ is a path. The construction where the graph induced by $\{a, b, c\}$ is a path is identical to the one in Figure 4.1c, except that the ray for c must be extended so that it also intersects with ray b. Note that in the case where the graph induced by $\{a, b, c\}$ has one edge, it is not necessary to construct both V_{ab} and V_{bc}: the reason for this is that vertices with both neighborhoods cannot be present in the same barrier graph, as otherwise the graph would not be bipartite (and a 5-cycle would be induced). □

The realizations provided in Figure 4.1 and the realizations of Chapter 3 are both natural and simple. On the other hand, Theorem 5.2.2 shows that not all bipartite graphs of large enough fixed resilience are realizable, so clearly there is some largest resilience r such that all bipartite graphs of resilience r are realizable. Hence we propose the following question:

Question 4.2.1. *Find the minimum resilience $r_\star$ such that there exists a bipartite graph G of resilience $r_\star$, so that G is not realizable as a barrier graph.*

We have shown that $r_\star$ exists, and that $4 \leq r_\star \leq 16$, where the upper bound on $r_\star$ follows from Theorem 5.2.2, and ultimately from Theorem 5.2.1. Determining $r_\star$, or even tightening the bounds, would be an interesting step in our understanding of bipartite barrier graphs. Some optimization of the bounds in Lemma 5.1.3 and Theorem 5.2.1 can reduce the upper bound on $r_\star$ slightly, but a substantial tightening seems to require a new idea.

CHAPTER 5: THE RIGIDITY OF BARRIER GRAPHS

5.1 Stabbing Rays and Segments

The next few results connect the geometry of a barrier graph's realization to the kinds of neighborhoods that can appear in the graph. To this end, we give a bound $(O(n^3))$ on the number of subsets of a set of rays or line segments that could be stabbed by another ray. The next lemma is a general geometric result, but it will be particularly helpful in addressing the neighborhood structure of barrier graphs.

Below, we will use the standard geometric dual transformation $\mathcal{D}$ defined in section 1.2.3. We will also use the language that an object A with a supporting line "sees" the objects that it stabs in some particular order from left to right if the intersections of A with the other objects are ordered in the same way by their x-coordinates.

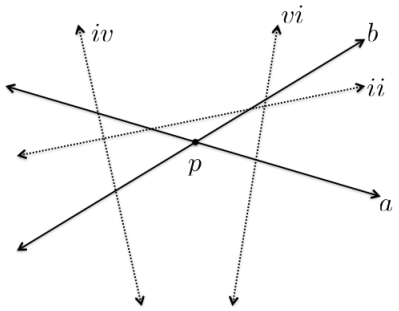
Lemma 5.1.1. *Let X be a set consisting of r rays and s line segments in the primal plane. Then there exists a set X' of lines that divide the dual plane into regions so that any two points lying in the same region correspond to two lines in the primal plane intersecting the same elements of X in the same left-right order. Furthermore, X' can be chosen so that*

$$|X'| \leq \binom{r+s}{2} + 2(r+s).$$

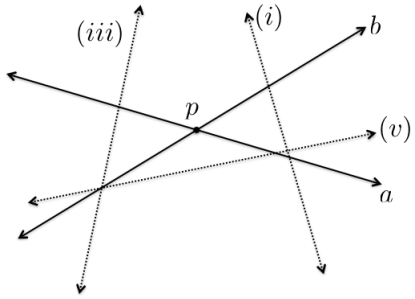
Proof. Let X' consist of the following types of lines in the dual plane:

(I) For each ray in X, the dual of the anchor and a vertical line through the dual of the supporting line, and for each line segment in X, the dual of the endpoints (these are the "bow-tie" bounding lines).

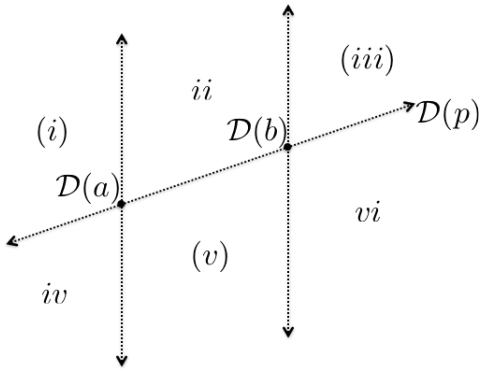
(II) For each pair of elements in X, the dual of the intersection of their supporting lines (if they intersect).



(a) Cases ii, iv, vi see $a \to b$



(b) Cases i, iii, and v see $b \to a$



(c) 6 regions in the dual plane

Figure 5.1: Lines of (a) intersect a, then b from left to right, and lines of (b) intersect b, then a from left to right. All lines which intersect a and b are one of these 6 types, and their duals are in the corresponding regions of (c), determined by Type II lines.

The intuition behind this lemma is that regions from Type I lines tell us which elements of X a primal line intersects, and regions from Type II lines tell us in which order (from left to right) we see them.

Let $n := s + r$. Clearly there are at most $\binom{n}{2} = \binom{s+r}{2}$ points of intersection between elements of X (and hence at most $\binom{s+r}{2}$ lines of Type II), and at most $2(r+s)$ lines of Type I.

Now, consider the partition $\mathcal{P}$ of the dual plane defined by X'. The *regions* of this partition are simply the k-faces of the partition, i.e., the points of intersection of these lines, the open line segments connecting two points, and the open regions in the plane created by removal of these points and segments. We claim this partition has the desired property.

Consider an arbitrary pair of points x, y in the same region of $\mathcal{P}$. First note that the position of x (or y) with respect to Type I lines determines which line segments and rays of X the dual of x (or y) intersects. This is precisely determined by the bowtie regions of Proposition 1.2.6. That is, within a region all points are duals of lines which intersect the same elements of X. Now, fix a pair of elements $a, b \in X$ which are both intersected by the duals lines of x and y. We show they are intersected in the same order. It suffices to assume that a and b are both lines (possibly replacing a ray or segment with their supporting line.)

It is easy to see that the order in which a line intersects two others is determined by both the relative order of the slopes of the lines and whether the first line passes above or below the intersection point of the other two. This is illustrated in Figure 5.1 and is determined by the position of the line's dual with respect to Type I & II lines. In particular, Type II lines carry the data on whether the first line passes above or below, while position with respect to the Type I lines (beyond simply carrying information about whether or not the line intersects the involved ray or segments) carries the information about the slope.

Thus, by our construction of $\mathcal{P}$, being in the same region of $\mathcal{P}$ implies that x and y have the same relation with $\mathcal{D}(a)$ and $\mathcal{D}(b)$, and hence $\mathcal{D}(x)$ and $\mathcal{D}(y)$ intersect a and b in

the same left-right order. Note that a point which lies within one of the line segments might not intersect one or the other (or both) of a and b, or might intersect both at the same time. Nonetheless, the information required to determine this is captured by the region.

Since a and b are arbitrary, this means that $\mathcal{D}(x)$ and $\mathcal{D}(y)$ intersect all elements of X in the same order as desired. □

Let us suppose X has r rays and s line segments, where $r + s = n$, and consider the partition of the dual plane offered by Lemma 5.1.1. Consider a line in the primal plane. It intersects at most n elements of X in some order, and hence there are at most $2n$ subsets that a ray can intersect.

We have just shown that given two lines, if the duals of those lines fall into the same region of the dual plane in our partition, they intersect the same elements of X in the same order. Thus the subsets of X potentially stabbed by segments of a line are completely determined by the region of the dual plane. For such a region A, we denote this set of subsets by $\mathcal{R}_X(A)$, or $\mathcal{R}(A)$ when X is understood. In this language, our observation above is that $|\mathcal{R}(A)| \leq 2n$.

Lemma 5.1.2. *Let $\ell_1 = \mathcal{D}(p)$ and $\ell_2 = \mathcal{D}(q)$ be two intersecting Type II lines in the dual plane, and let A, B, C, and D be the regions adjacent to the intersection of ℓ_1 and ℓ_2 (see Figure 5.2). Then*

$$\mathcal{R}(D) \subseteq \mathcal{R}(A) \cup \mathcal{R}(B) \cup \mathcal{R}(C).$$

Proof. Note that p and q are the intersection points for pairs $x_p, y_p \in X$ and $x_q, y_q \in X$, so the difference between dual points in $\mathcal{R}(A)$ and in $\mathcal{R}(B)$, or for any pair of these regions, is only in the order in which (the supporting lines of) x_p, y_p, x_q, and y_q are seen.

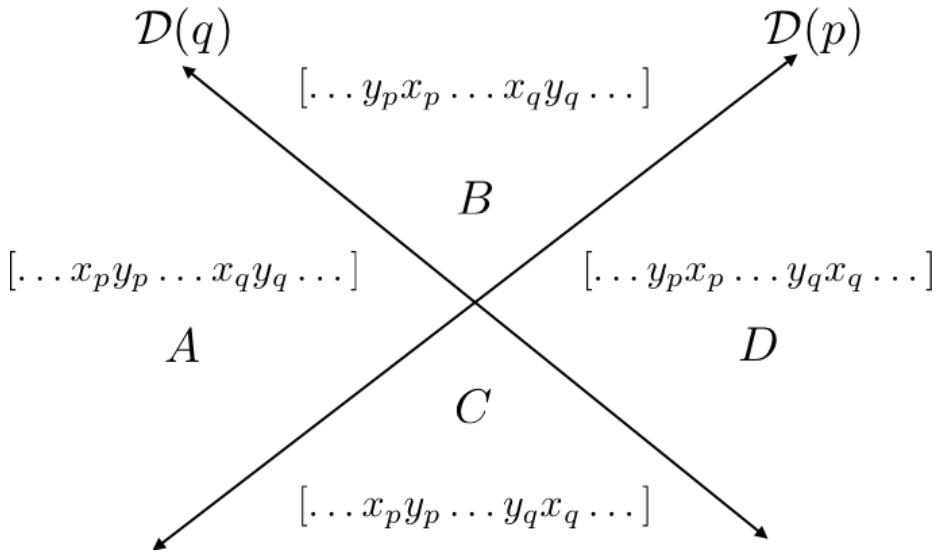


Figure 5.2: Two intersecting Type II lines in the dual plane and the order in which the surrounding regions see the corresponding elements of X

Suppose that the dual of a point in the region A sees elements of X in the order $[\ldots, x_p, y_p, \ldots, x_q, y_q, \ldots]$, B sees them as $[\ldots, y_p, x_p, \ldots, x_q, y_q, \ldots]$, C sees them as $[\ldots, x_p, y_p, \ldots, y_q, x_q, \ldots]$, and D sees $[\ldots, y_p, x_p, \ldots, y_q, x_q, \ldots]$.

Since a ray sees a prefix or a suffix of the order that lines see in a region, the containment above is clear. □

What Lemma 5.1.2 gives us is a tool to avoid over-counting subsets. It says that the four regions around any crossing of Type II lines cannot all correspond to distinct subsets that can be stabbed by a ray.

Lemma 5.1.3. *Let X be a set of n rays or line segments in the plane. Let S_X denote the set of all subsets $A \subseteq X$ so that there exists a ray intersecting exactly the elements of A, and no others. Then*

$$|S_X| \leq 6n^3.$$

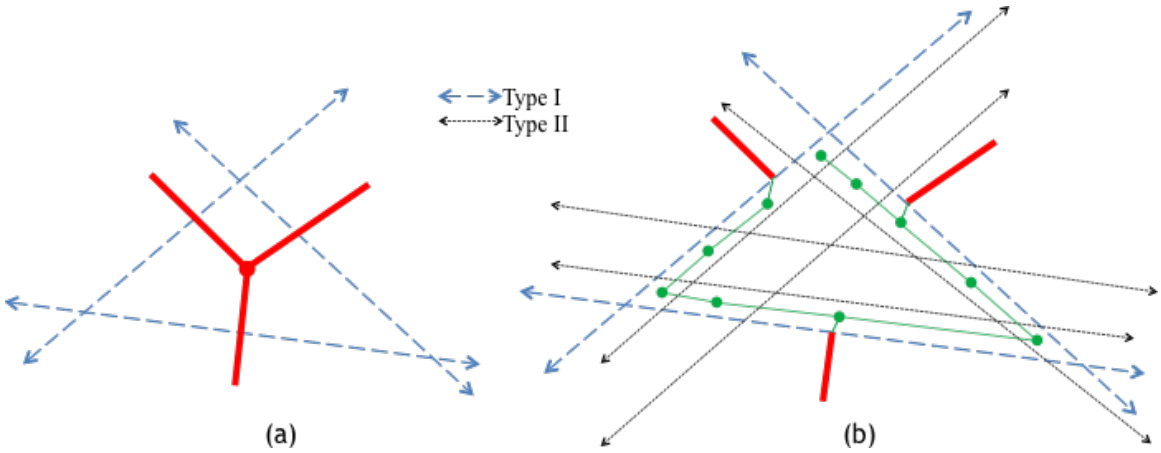


Figure 5.3: (a) The primary spanning tree of 2-faces for Type I lines, which are blue and dashed, and (b) the secondary tree obtained by refining it for a particular 2-face using Type II lines, which are black and dotted.

Proof. We will again use the Type I and Type II descriptions of the lines of the dual-plane partition in Lemma 5.1.1.

For adjacent 2-faces A and B separated by a Type I line $|R(A) \setminus R(B)| \leq n$. This follows as two points on different sides of a Type I line in the dual plane correspond to one line intersecting that element of X, and one line not intersecting it. On the other hand for adjacent 2-faces A and B separated by a Type II line, $|R(A) \setminus R(B)| \leq 2$ since passing through such a line either induces a transposition of the order of two order-adjacent intersected elements of X or has no effect at all.

We now proceed to bound the total number of subsets encountered. Partition the dual plane first by the Type I lines, and then refine this partition by including the Type II lines. We will first (carefully) fix any spanning tree of the 2-faces in the dual plane according to the Type I partition. This tree is shown in red in Figure 5.3 (a). We then fix a root, and note that the total number of subsets encountered by rays within the 2-faces is at most $2n$

for the root, plus n times the number of Type I lines crossed by the spanning tree, plus a yet-undetermined amount for the Type II crossings.

In order to do this as efficiently as possible, we do the following: First, consider the regions of the dual plane determined by the (at most) $2n$ different Type I (bow-tie) lines. There are at most $\binom{2n+1}{2} + 1$ such regions, so the primary spanning tree has at most $\binom{2n+1}{2}$ edges.

Next, we consider the regions defined by all the lines together. This refines the Type I partition of the plane. Within each 2-face A according to the Type 1 partition, we claim that once we account for $\mathcal{R}(B)$ for all refined 2-faces B inside of A and sharing a boundary with A, then we have also accounted for $\mathcal{R}(C)$ for any other refined 2-face C inside of A.

This follows from iterated applications of Lemma 5.1.2. Indeed, suppose some subset of faces including the boundary faces are accounted for. Then the union of the remaining 2-faces (if any) consist of a collection of polygonal faces. Each of these must have a convex vertex around which three faces are already accounted for, and applying Lemma 5.1.2 to those faces, one easily observes that all four incident faces are accounted for.

We use this information to refine the (red) spanning tree so that it also connects these Type II boundary 2-faces by first connecting each set of border 2-faces with a path, and then connecting these paths at any boundary 2-face for each edge in the original spanning tree. This spanning tree crosses Type I lines at most $\binom{2n+1}{2}$ times, and in general has one less than the number of 2-faces as edges.

Now we need to count the number of these boundary 2-faces. For each intersection of a Type I line with a type II line, there are four such 2-faces, but since each border 2-face

has at least two Type II lines bordering it, each is counted at least twice. So there are at most $2(2n)\binom{n}{2}$ of these border 2-faces.

Thus the total number of subsets encountered within the 2-faces is at most

$$\begin{aligned} 2n + 2 \cdot \left(2n^2(n-1)\right) + n \cdot \binom{2n+1}{2} \\ = 6n^3 - 3n^2 + 2n \leq 6n^3 \end{aligned} \tag{5.1}$$

Finally, we must determine the contribution from lower dimensional regions in the dual plane. A point in the dual plane which occurs in a Type II line corresponds in the primal plane to a line through the intersection point. It is clear these account for fewer subsets of S_X as if a segment contains one of the intersecting lines it contains both. But also, these subsets are easily seen to be realizable by lines occurring in the neighboring 2-faces of the dual plane. Thus, for the purposes of differentiating between corresponding collections of subsets of X, we can ignore Type II lines.

Similarly, we can ignore Type I lines: points in the dual plane occurring in these lines correspond to lines through the 'endpoints' (where a point at infinity counts as an endpoint for rays) and the subsets realized by segments of these lines are captured in the neighboring 2-faces.

Thus the bound (5.1) suffices to complete the proof of the lemma. □

5.2 Rigidity

With the results of the previous section in hand, it is finally possible to state and prove our main theorem about the structure of barrier graphs. This next result strongly limits adjacencies within a barrier graph to any fixed subset. In a general graph, when a subset of vertices of size t is fixed, other vertices may potentially have any of the 2^t different adjacencies within the subset. This result states that in a barrier graph G, for any subset $X \subseteq V(G)$, only polynomially many subsets of X can appear as the neighborhood of vertices outside of X.

This is a strong rigidity theorem in the sense that it implies there are few distinct neighborhoods of vertices outside of X into X and, instead, there must be many vertices whose neighbors in X are the same. Quite remarkably, this is true even for fairly large sets X (for an n vertex graph G; there must be many neighborhood clones even into sets of size nearly $n^{1/3}$). The proof, however, follows quite easily from Lemma 5.1.3.

Together, Lemmas 5.1.1 and 5.1.3 give that the number of subsets of sets X of n rays that can be stabbed by another ray is $\Theta(n^3)$.

Theorem 5.2.1 (Barrier Rigidity)**.** *Suppose G is a barrier graph, and $X \subseteq V(G)$ with $|X| = t$. Let*

$$S_X = \{\mathcal{N}_X(y) : y \in V(G) \setminus X\}.$$

Then

$$|S_X| \leq 12t^3.$$

Proof. Let $f(t) := 6t^3$.

Fix a realization of G, and vertex $y \in V(G) \setminus X$. The vertex y corresponds to a ray in the realization. Note that $\mathcal{N}_X(y)$ is determined by the intersection of that ray, along with rays and line segments determined by X.

In particular, if y is red in our coloring, then the neighborhood of y is determined by its intersections with the blue rays determined by X and with the parts of black rays above the line $\alpha\beta$. Suppose there are r red, b blue, and k black colored vertices in X. By Lemma 5.1.3, there are at most $f(b+k)$ subsets of X that can arise from such intersections. If y instead blue, by similar reasoning we see there are most $f(r+k)$ subsets of X which could be the neighborhood of y.

Similarly, if y is colored black, the neighborhood of y is determined by the intersection of the y-ray with the line segments given by red rays above the $\alpha\beta$-line and blue rays below the $\alpha\beta$ line, giving a bound of $f(r+b)$.

In total, we arrive at an upper bound on the number of subsets of X occurring as some vertex's neighborhood of

$$f(r+b)+f(r+k)+f(k+b)=f(t-k)+f(t-b)+f(t-r)$$

where we are subject to the constraint that $r+b+k=t$. Convexity of f implies that this quantity is also convex, and so is maximized when $r=t$, $b=0$, and $k=0$, yielding a bound of $2f(t)=12t^3$ as claimed. □

Theorem 5.2.1 implies strong conditions on the neighbors of vertices within a barrier graph. This is sufficient to show that barrier graphs are rare amongst bipartite graphs, and hence tripartite graphs as well. Indeed, we show an even stronger theorem: barrier graphs are rare for any size of graph, even when one of the sides of the bipartite graph is not large.

Theorem 5.2.2. *Suppose G is chosen uniformly at random from all bipartite graphs with bipartition (X,Y) satisfying $16 \le |X| \le |Y|$, and $|X|+|Y|=n$. Then the probability that G is a barrier graph is at most $2^{16}e^{-n2^{-17}}=o(1)$.*

Proof. Fix (arbitrarily) $t \ge 16$ vertices in X, and call these t vertices X'. We will prove that if n is sufficiently large, with probability at least $1-2^t e^{-n2^{-(t+1)}}$,

$$|\{\mathcal{N}_{X'}(y) : y \in Y\}| = 2^t.$$

In other words, we will show that with high probability, every subset of X' is exactly the neighborhood of some element of Y. On the other hand, our Barrier Rigidity theorem (Theorem 5.2.1) tells us that the number of subsets of X' which could be the neighborhood of any vertex in Y is no more than $12t^3$, which by our choice of t is smaller than 2^t.

Let Z denote the number of subsets of X' which do *not* appear as a neighborhood of a vertex in Y. In this language, what Theorem 5.2.1 says is that if G is a barrier graph, then $Z \geq 2^t - 12t^3$, or more simply, Z is large and positive. In particular, when n is large enough (and t is at least 16), if G is a barrier graph then Z is certainly at least 1.

Now, for any $y \in Y$, the probability that any fixed subset of X' is the neighborhood of y is just $\frac{1}{2^t}$ because in a uniformly random bipartite graph each edge is present with probability $\frac{1}{2}$, and therefore the probability that any fixed subset of X' is *not* the neighborhood of y is just $1 - \frac{1}{2^t}$.

Hence, for $S \subseteq X'$, the probability that S is not $\mathcal{N}_{X'}(y)$ for all $y \in Y$ is $(1 - \frac{1}{2^t})^{|Y|}$, since these events are independent. Let I_S be the indicator variable for when S is not the neighborhood of any y; then Z is just the sum $\sum_{S \subseteq X'} I_S$.

By linearity of expectation,

$$\begin{aligned}
\mathbb{E}[Z] &= \mathbb{E}\left[\sum_{S \subseteq X'} I_S\right] \\
&= \sum_{S \subseteq X'} \mathbb{E}[I_S] \\
&= \sum_{S \subseteq X'} (1 - \frac{1}{2^t})^{|Y|} \\
&= 2^t (1 - \frac{1}{2^t})^{|Y|}
\end{aligned}$$

Since $|X| \leq |Y|$ and $|X| + |Y| = n$, we have $\mathbb{E}[Z] = 2^t \left(1 - \frac{1}{2^t}\right)^{|Y|} \leq 2^t \left(1 - \frac{1}{2^t}\right)^{n/2}$.

Because Z is nonnegative and integer-valued, we can use Markov's inequality to bound the probability that it is nonzero: $\mathbb{P}(Z \geq 1) \leq \mathbb{E}[Z]$.

Putting together these last two inequalities and using the identity $(1+a)^b \leq e^{ab}$,

$$\begin{aligned}\mathbb{P}(Z \geq 1) &\leq 2^t \left(1 - \frac{1}{2^t}\right)^{n/2} \\ &\leq 2^t e^{-(n2^{-t})/2} \\ &= 2^t e^{-n2^{-(t+1)}}\end{aligned}$$

This probability can thus be made arbitrarily small by choosing large enough n, which implies that the probability that G is a barrier graph is also arbitrarily small.

□

An almost immediate corollary of this is the following:

Corollary 5.2.3. *Almost every bipartite graph is not a barrier graph.*

The main reason the corollary isn't immediate is that choosing a bipartite graph on n vertices uniformly at random in general is not quite as easy as it is when the bipartition is fixed.

Proof. Let G be chosen uniformly at random from the set of (unlabeled) bipartite graphs on n vertices. We say that G admits an s-decomposition if there exists $X, Y \subseteq V(G)$ with $|X| = s$ and such that (X, Y) is a bipartition of G. Note that G admits an s-decomposition iff it admits an $(n-s)$-decomposition.

Let $\mathcal{A}_s$ denote the event that G admits an s-decomposition and $\mathcal{B}$ denote the event that G is a barrier graph. The content of Theorem 5.2.2 is that

$$\mathbb{P}(\mathcal{B}|\mathcal{A}_s) \leq 2^t e^{-n2^{-(t+1)}},$$

if $s \geq t$, where $t = 16$ is as in Theorem 5.2.2.

On the other hand, there is some $0 < s \leq \lfloor n/2 \rfloor$ so that $G \in \mathcal{A}_s$, because there must be some bipartition witnessing that G is bipartite, and we may take s to be the size of the smaller half of the partition.

Thus, summing over possible values of s, we have $\mathbb{P}(\mathcal{B}) \leq \sum_{s=1}^{\lfloor n/2 \rfloor} \mathbb{P}(\mathcal{B} \cap \mathcal{A}_s)$, with inequality instead of equality because the events $\mathcal{A}_s$ do not *partition* the space.

Rewriting this and using the inequality that follows from Theorem 5.2.2, we have

$$\begin{aligned}
\sum_{s=1}^{\lfloor n/2 \rfloor} \mathbb{P}(\mathcal{B} \cap \mathcal{A}_s) &= \sum_{s=1}^{\lfloor n/2 \rfloor} \mathbb{P}(\mathcal{B}|\mathcal{A}_s)\mathbb{P}(\mathcal{A}_s) \\
&\leq \sum_{s=1}^{t-1} \mathbb{P}(\mathcal{B}|\mathcal{A}_s)\mathbb{P}(\mathcal{A}_s) + \sum_{s=t}^{\lfloor n/2 \rfloor} \mathbb{P}(\mathcal{B}|\mathcal{A}_s)\mathbb{P}(\mathcal{A}_s) \\
&\leq \sum_{s=1}^{t-1} \mathbb{P}(\mathcal{A}_s) + 2^t e^{-n2^{-(t+1)}} \sum_{s=t}^{\lfloor n/2 \rfloor} \mathbb{P}(\mathcal{A}_s),
\end{aligned}$$

Finally, we estimate $\mathbb{P}(\mathcal{A}_s)$ rather crudely. There are $2^{s(n-s)}$ graphs (not necessarily bipartite) which admit an s-decomposition because there are $s(n-s)$ possible edges in such a graph. There are at least $2^{\lfloor n/2 \rfloor \cdot \lceil n/2 \rceil}$ bipartite graphs by the same reasoning, and so

$$\mathbb{P}(A_s) \leq \frac{2^{s(n-s)}}{2^{\lfloor n/2 \rfloor \cdot \lceil n/2 \rceil}} = 2^{s(n-s) - \lfloor n/2 \rfloor \cdot \lceil n/2 \rceil},$$

This is increasing in s for $s \leq \lfloor n/2 \rfloor$ and is always at most one. While the $\mathcal{A}_s$ do not partition the space, and hence we cannot simply bound $\sum_s \mathbb{P}(\mathcal{A}_s) = 1$, these bounds suffice. By estimating the $\mathbb{P}(\mathcal{A}_s)$ in each of the sums by the largest s appearing,

$$\begin{aligned}
\mathbb{P}(\mathcal{B}) &\leq \sum_{s=1}^{t-1} \mathbb{P}(\mathcal{A}_s) + 2^t e^{-n2^{-(t+1)}} \sum_{s=t}^{\lfloor n/2 \rfloor} \mathbb{P}(\mathcal{A}_s) \\
&\leq \sum_{s=1}^{t-1} 2^{t(n-t)-\lfloor n/2 \rfloor \cdot \lceil n/2 \rceil} + \sum_{s=t}^{\lfloor n/2 \rfloor} 2^t e^{-n2^{-(t+1)}} \\
&\leq (t-1)2^{t(n-t)-\lfloor n/2 \rfloor \cdot \lceil n/2 \rceil} + 2^t e^{-n2^{-(t+1)}} \cdot \frac{n}{2} \\
&= o(1).
\end{aligned}$$

completing the proof. □

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

Theorem 5.2.1 is a geometric result that says that the way neighborhoods look, from within any fixed set of vertices of a barrier graph, is very restricted. This is progress toward an ultimate goal of a complete classification of barrier graphs.

In Figure 6.1 is a visualization of the state of the classification as of this thesis, in terms of known graph classes. Barrier graphs are always tripartite (as shown in [9]) and always perfect (as we showed in Corollary 1.2.5), but not *all* bipartite, tripartite or perfect (e.g. K_4) graphs are barrier graphs.

We have conjectured that any bipartite graph has a realization that is two-colored, which is intuitive but has turned out to be surprisingly difficult to prove. We have also conjectured that it is possible to find an algorithm which produces a realization for a graph, given that it is tripartite, or else gives a certificate that the graph is not a barrier graph. An

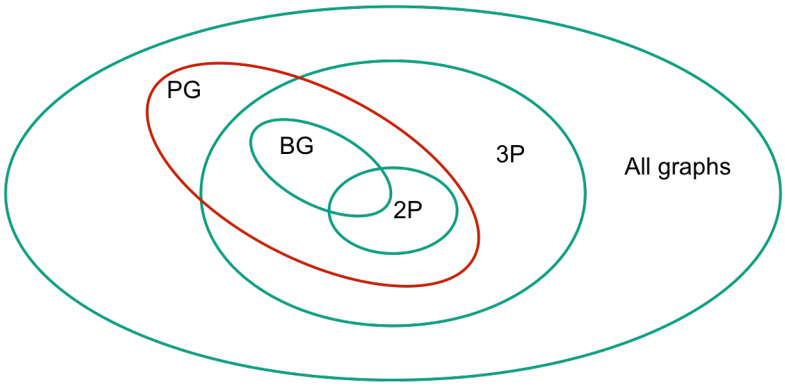


Figure 6.1: Barrier graphs in the context of well-known graph classes. PG = Perfect Graphs, BG = Barrier Graphs, 2P = Bipartite Graphs, 3P = Tripartite Graphs

easier problem (that is also still unsolved) would be to find an algorithm which produces a realization for a graph, given the promise that the graph actually is realizable. An answer to any of these conjectures would represent real progress on the topic of barrier graphs.

We also considered realizing graphs from the perspective of resilience (size of a minimum vertex cover); we know how to realize any bipartite graph with resilience up to 3, and we know that for bipartite graphs of resilience is at least 16 there is no guarantee the graph will be realizable at all. It is thus an interesting question to explore what is the smallest resilience $r_\star$ for which there is a bipartite graph of resilience $r_\star$ that is not a barrier graph.

This question of recognition of barrier graphs is one of the most interesting algorithmic questions remaining for ray-barrier graphs. In general, even the recognition of tripartite graphs the problem is *NP-hard*. On the other hand, both recognition of perfect graphs [3, 4], and determining the chromatic number of perfect graphs is polynomial [6]. Thus it is possible to recognize whether a given graph is both tripartite and perfect in polynomial time. So while these results suggest that efficient recognition of barrier graphs might be possible, more work is needed to find a way to determine whether a given graph is a barrier graph.

BIBLIOGRAPHY

[1] S. Bereg, J.M. Díaz-Báñez, M. Fort, M.A. Lopez, P. Pérez-Lantero, and J. Urrutia. Continuous surveillance of points by rotating floodlights. *Int. J. Comput. Geom. Appl.*, 24(3):183–196, 2014.

[2] S. Bereg and D.G. Kirkpatrick. Approximating barrier resilience in wireless sensor networks. In *Proc. of the 5th Internat. Workshop on Algorithmic Aspects of Wireless Sensor Networks*, LNCS 5804, pages 29–40, 2009.

[3] M. Chudnovsky, G. Cornuéjols, X. Liuand P. Seymour, and K. Vušković. Recognizing berge graphs. *Combinatorica*, 25(2):143–186, 2005.

[4] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. The strong perfect graph theorem. *Annals of Mathematics*, pages 51–229, 2006.

[5] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag New York, Inc., New York, NY, USA, 1987.

[6] M. Grotschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimizations*. Springer-Verlag, 1993.

[7] John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

[8] Richard M. Karp. Reducibility among combinatorial problems. In R. Miller, J. Thatcher, and J. Bohlinger, editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Springer US, 1972.

[9] David Kirkpatrick, Boting Yang, and Sandra Zilles. A polynomial-time algorithm for computing the resilience of arrangements of ray sensors. *International Journal of Computational Geometry & Applications*, 24(03):225–236, 2014.