WILLIAMS COLLEGE

WILLIAMSTOWN, MASSACHUSETTS

UNDERGRADUATE THESIS

A Characterization of Trees with Convex Obstacle Number 1 or 2

Author:
Philippe Demontigny

Supervisor:
William Lenhart

Submitted in partial fulfillment of the requirements
for the Degree of Bachelor of Arts with Honors in Mathematics

May 23, 2014

Abstract

A convex obstacle representation of a tree T is a drawing of the vertices of T in the plane with a set of convex polygons so that two vertices are connected by an edge if and only if that edge does not intersect any of the polygons. The minimum number of obstacles required to represent a tree in this way is called the convex obstacle number of the tree. This new description of a graph has steadily gained popularity since its introduction in 2009, and is particularly interesting because of its relationship to visibility graphs, which have been studied extensively and have applications in robot motion planning and architecture. So far, it is known that a representation using only 5 convex obstacles exists for all trees, which implies that the upper bound for the convex obstacle number of any given tree is five. However, not much is known about which trees have a convex obstacle number that is less than 5. In this thesis, we begin to fill this gap by providing necessary and sufficient conditions for a tree to have convex obstacle number 1 or 2. We also provide insights into how one could approach the problem of finding all trees with convex obstacle number 3 or 4.

Acknowledgements

First and foremost, I would like to thank my adviser Professor Bill Lenhart for introducing me to the world of graph drawing and computational doodling and for helping me come to terms with the number of cases I needed to consider every step of the way. It was an honor to fill his chalkboard with color during each of our weekly brainstorming sessions. I would also like to thank the many professors in the mathematics and statistics department who supported me along the way, including my second reader Professor Colin Adams, and my SMALL adviser Professor Steven Miller, whose advice on how to succeed in life and research will not be soon forgotten. A special thank-you also goes out to all of my friends at Williams College, who begrudgingly accepted this thesis as an excuse for why I could not spend time with them this year. Lastly, I would like to thank my parents for being my inspiration and for not asking too many questions about the immediate applications of my research.

Contents

1 Introduction

1.1 Background

Obstacle representations of graphs are a relatively new way of representing graphs in the plane that is related to visibility graphs. An *obstacle representation* of a graph G is a choice of locations for the vertices of G in general position along with a set of polygonal obstacles, such that an edge exists in G if and only if the corresponding line segment between the vertices does not intersect an obstacle. Closely related to obstacle representations is the *obstacle number*, which is the minimum number of obstacles needed to represent G in the plane. The first paper to define this concept was Alpert et al. [Al] in 2009, and they produced many interesting results as well as a number of open questions that remain unanswered.

The only graphs that have obstacle number 0 are the complete graphs. Non-complete graphs that have obstacle number 1 are unique in that merely finding a 1-obstacle representation provides proof that the obstacle number is exactly 1. Alpert et al. were the first to study these types of graphs, and they found entire families of graphs that have obstacle number 1, including outerplanar graphs and certain bipartite graphs. Perhaps more interesting, however, are graphs that do not have a 1-obstacle representation. Finding a graph that has obstacle number at least 2 requires a proof that no 1-obstacle representation exists. Alpert et al. were able to find a few examples of such graphs, and these results were then extended in a paper by Pach and Sariöz [PS]. Still other results show that there are graphs with arbitrarily large obstacle number, examples of which can be found in [Al], [PS], and [Mu].

There are many results which show that finding the obstacle number of an arbitrary graph is not an easy task. A paper by Johnson and Sarioz [JS] proved that the general case of finding an obstacle representation using the minimum number of obstacles for a graph G is NP-hard, and in an upcoming paper by Giacomo et al. [Gi], the NP-hard condition was also proven for the problem of determining whether an outside obstacle representation can be drawn for an arbitrary planar graph given a set of obstacles. However, algorithms for more restricted cases have been found. Sariöz [Sa] created a polynomial-time approximation algorithm for the obstacle number of a graph, and Koch et al. [Ko] found a linear time algorithm for finding plane outside-obstacle representations for biconnected graphs.

In this paper, we are particularly interested in what is known as the *convex obstacle number* of a graph. This is the same as the obstacle number, except that all of the obstacles are required to be convex polygons. The motivating problem for us came from a paper by Fulek et al. [Fu], in which they found a construction for any outerplanar graph that requires only 5 convex obstacles. This is one of the first results in the area of obstacle representations that is actually an upper bound, and we know this bound to be tight thanks to Fulek et al., who in the same paper proved that there must exist trees with convex obstacle number at least 4. However, no results currently exist that determine which graphs have convex obstacle number less than or equal to 4. In this thesis project, our objective was to expand upon the results of Fulek et al. concerning outerplanar graphs. For the scope of this paper, we limit ourselves to the study of trees, which are a subset of all outerplanar graphs. The eventual goal of this project is to fully characterize all trees with convex obstacle number 5 or fewer.

1.2 Our Results

In this thesis, we provide necessary and sufficient conditions for a tree to have convex obstacle number 1 or 2. These results were tractable because we did not have to take into account the possibility of edge crossings in our representations (See Lemma 2.2). For convex obstacle numbers 3+, some crossings may be necessary, as they are certainly required for the construction in Fulek et al. to work.

In the 1-convex obstacle case, our result is particularly nice because it revolves around a very well-known family of trees. A *caterpillar graph* is a tree whose vertices are all a distance 1 from a central path, which is referred to as the *spine* of the caterpillar. Using this family of trees, we present the statement of our first Theorem below, and then go through the proof in section 2.2.

Theorem 1. *A tree T has convex obstacle number 1 if and only if T is a caterpillar graph.*

This theorem suggests that caterpillar graphs and the convex obstacle number of a tree are closely related. In fact, this relationship persists through the classification of trees with convex obstacle number 2. We define an *n-caterpillar coloring* of a tree T as a coloring of the vertices and edges of T using n colors so that the subgraph formed by each color is a connected caterpillar graph. In this definition, we will allow all vertices and edges to have multiple colors. That is, if a vertex/edge is marked with k colors, then that vertex/edge is a part of the caterpillar graph formed by *each* of the colors. Next, just like the definition of the obstacle number, we will let the *caterpillar number* of a tree T be the minimum n such that T has an n-caterpillar coloring. Using this definition, we find the following characterization for trees with convex obstacle number 2. Once again, we will save the proof of this Theorem for section 3.2.

Theorem 2. *A tree T has convex obstacle number 2 if and only if T has caterpillar number 2.*

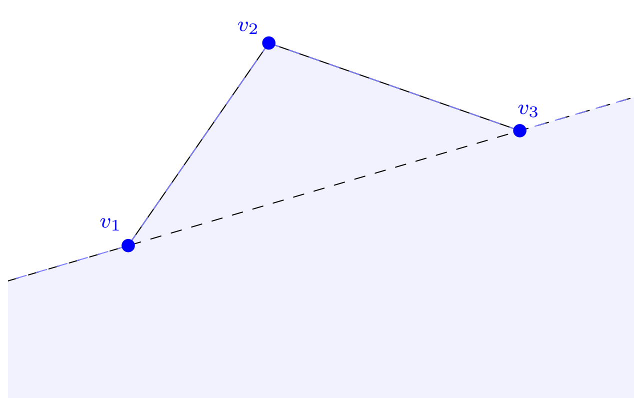
2 Trees with Convex Obstacle Number 1

2.1 Preliminaries

We begin our study of the obstacle number of trees by stating a few Lemmas that will aid in the proof of our two main results. Many of these arguments mirror those used in [Fu], but for ease of understanding we will rigorously prove them here. It is our belief that these Lemmas will continue to be useful beyond the scope of this paper, and we aim to provide a useful starting point for those interested in continuing this research.

First, since our study focuses exclusively on convex polygons, the following observation will be useful because it captures the limitations one faces by using convex obstacles as opposed to general polygonal obstacles.

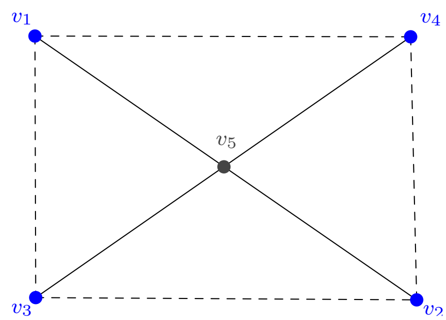
Observation 2.1. *If a convex obstacle is used to represent a path $v_1v_2v_3$ in general position, then the obstacle can only exist within the union of the triangle $\Delta v_1v_2v_3$ and the half-plane defined by the line through v_1v_3, as shown in the figure below.*



In the above diagram, the shaded region denotes the only valid positions for a convex obstacle that intersects the non-edge v_1v_3. This observation is a key component of the following Lemmas, which we will refer to frequently in this paper.

Lemma 2.2. *There are no edge-crossings in any 1 or 2-convex obstacle representation of a tree T.*

Proof. Assume that there are two edges v_1v_2 and v_3v_4 in a tree T that cross in a convex obstacle representation of T. Let v_5 be the point of intersection of these two edges. Then observation 2.1 tells us that we need four distinct obstacles to represent the four paths $v_1v_5v_3$, $v_1v_5v_4$, $v_2v_5v_3$, and $v_2v_5v_4$, as shown in the diagram below.



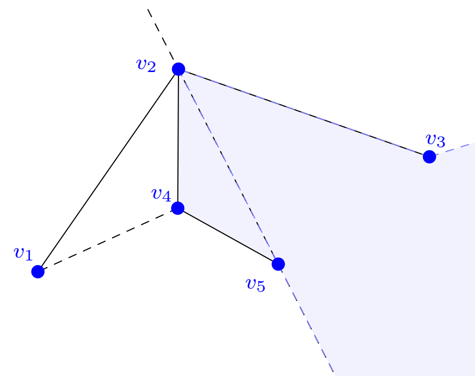
The four pictured non-edges each require a separate obstacle.

However, this is not a true lower bound for the number of convex obstacles in a representation that includes crossings, as we can change at most one of the non-edges in T to an edge. Without loss of generality, let v_1 and v_3 be connected. We now cannot connect any of the other non-adjacent vertices without creating a cycle, which breaks the assumption that T is a tree. Thus we can reduce the number of convex obstacles needed to represent T by at most 1, and it follows that no representation of a tree with at least one edge crossing can have convex obstacle number less than 3. □

This Lemma tells us that there is no 1 or 2-convex obstacle representation of a tree that contains a crossing. Thus for all of the following Lemmas and Theorems, we will assume that any representation of a tree T has no edge-crossings.

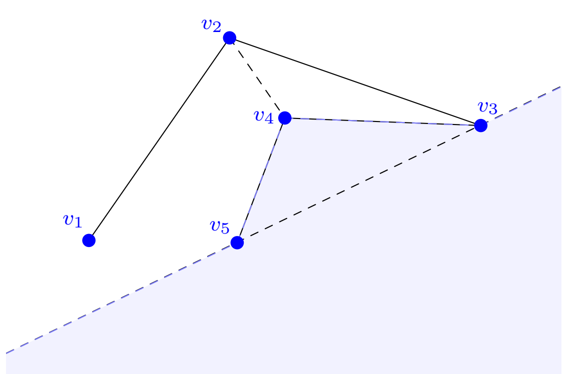
Lemma 2.3. *Given a path $v_1v_2v_3$ of length 2 in general position, any other path of length 2 that is incident to $v_1v_2v_3$ and at least partially contained within the triangle $\Delta v_1v_2v_3$ cannot be represented with a single convex obstacle.*

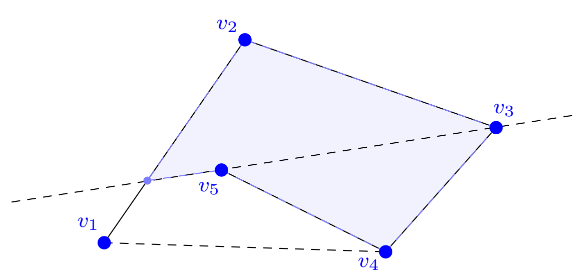
Proof. Let $v_1v_2v_3$ be a path in general position. For this proof, we consider a path of length 2 that starts either at vertex v_1 or v_2, without loss of generality. This gives us a total of three cases to consider. First, assume that we have a path $v_2v_4v_5$. Without loss of generality, we will assume that the non-edge v_1v_4 does not intersect the non-edge v_2v_5, as shown in the diagram below. If these two non-edges did in fact intersect, then we could consider the non-edges v_3v_4 and v_2v_5 instead.



Now, observation 2.1 tells us that any obstacle that intersects the non-edge v_2v_5 can only exist in the shaded region above. However, this means that the non-edge v_1v_4 cannot fall within this region, and thus requires a second convex obstacle, as desired.

We now consider the case where we have a path that starts from v_3, namely $v_3v_4v_5$. Within this case, we must separately consider the situations where the non-edges v_2v_4 and v_3v_5 intersect and where they do not. These situations are illustrated in the diagrams below, where observation 2.1 has already been applied to highlight the possible locations for a convex obstacle that touches v_3v_5.





In the left diagram, $\angle v_3v_4v_5$ guarantees that the non-edge v_2v_4 cannot fall within the shaded region, and thus requires a second obstacle. In the right diagram, recall that the path $v_3v_4v_5$ must fall within the triangular convex hull. This means that v_5 must lie within $\Delta v_1v_2v_3$. However, this means that the non-edge v_1v_4 cannot fall within the shaded region. Thus in either situation a second obstacle is required, which completes the proof. It follows that any path of length 2 inside a triangular region cannot be represented by a single convex obstacle, as desired. □

A natural question to ask after seeing the proof of Lemma 2.3 is whether a path of length 2 inside any polygon necessitates 2 convex obstacles. It turns out that it does, and the proof follows immediately from Lemma 2.3. We therefore leave this result as the following Corollary:

Corollary 2.4. *A path of length 2 that is at least partially contained within any polygonal region cannot be represented with a single convex obstacle.*

Proof. This result follows from a triangulation of our polygonal region and Lemma 2.4. Consider any polygonal region that contains a path of length 2 extending from a vertex v_1. Let v_2 and v_3 be the two vertices directly adjacent to v_3 in the polygon. Then we can create a triangle $\Delta v_1 v_2 v_3$ that contains the path of length 2. It then follows by Lemma 2.3 that our polygon cannot have a 1-convex obstacle representation, since this would imply that the subgraph formed by our path and $\Delta v_1 v_2 v_3$ has a 1-convex obstacle representation, which is a contradiction. The desired result follows. □

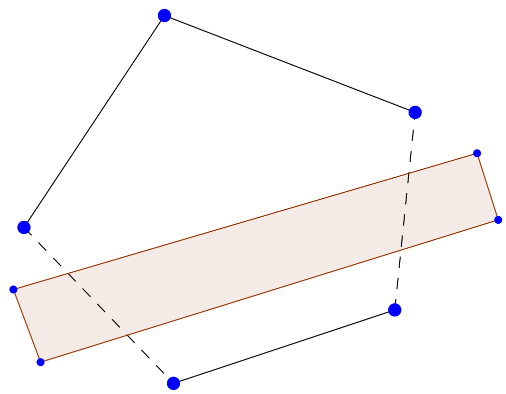
In the previous proof, as well as the proof of Lemma 2.2, we assumed implicitly that the subgraph of a tree T has obstacle number less than or equal to that of T. This fact is easy to prove. Let T be a tree with convex obstacle number n. Then there is some representation of T in the plane that uses exactly n convex obstacles. Notice that deleting vertices of T does not create any new non-edges, and thus all non-edges of T that intersected one or more of the n obstacles will intersect the same obstacles in any subgraph of T. Thus deleting vertices cannot increase the convex obstacle number of a tree T. This idea is easily generalized to the following, which we state as an observation:

Observation 2.5. *If a graph G has obstacle number n, then any induced subgraph of G has obstacle number less than or equal to n.*

Our final preliminary result involves the *convex hull* of a tree T, which is the smallest convex region that contains all of the vertices of T. It turns out that information about the number of non-edges in the convex hull of a tree tells you a lot about the convex obstacle number of that tree. This is the motivation behind the following Lemma:

Lemma 2.6. *In any 1-convex obstacle representation of a tree T, the convex hull of T has exactly 1 non-edge.*

Proof. We will proceed by contradiction. First assume that the convex hull of our tree T contains two non-edges. Since we are working with a single obstacle, we know that the obstacle must pass through both edges. Furthermore, since the obstacle is convex, the interior of the obstacle must pass through the convex hull, connecting these two non-edges. This separates our tree T into two disconnected parts as shown in the diagram below.



But this gives us a contradiction, as we always assume that T is a single, connected tree. It follows that T cannot have two non-edges in its convex hull, and therefore it must have only one, as desired. □

2.2 Proof of Theorem 1

The proof of Theorem 1 can be divided into two main steps. First, we will show that all Caterpillar graphs can be represented using only a single convex obstacle. Then, we will prove that all trees that are not Caterpillars have convex obstacle number 2 or more. For the first step, we will provide a construction that takes a given Caterpillar tree T and creates a representation using only a single convex obstacle. We begin by describing a construction for *star graphs*, which are a subset of the family of Caterpillar graphs that consist of n leaves adjacent to a single central vertex.

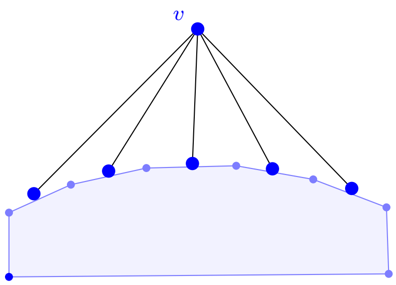
Lemma 2.7. *All star graphs have a 1 convex obstacle representation.*

Proof. Consider a star graph S with central vertex v and n leaf vertices, and consider any rectangular obstacle. The following process will allow us to transform the obstacle so that it blocks all of the non-edges of S:

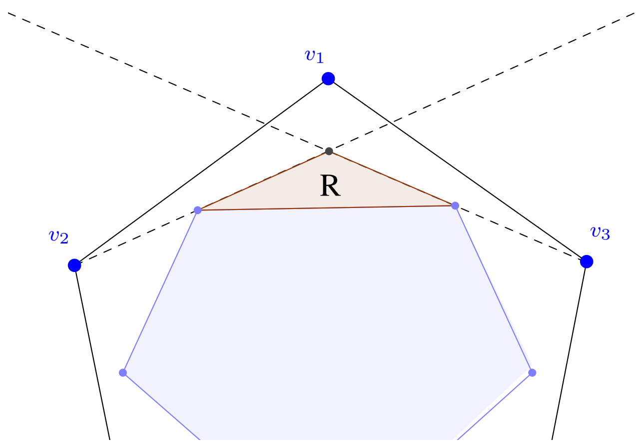
1. Place the central vertex v above one side of the rectangular obstacle. Arrange the n leaf vertices of S along the top of the obstacle so that they are just barely not touching the obstacle.

2. Make very slight perturbations in the top layer of the obstacle so that no two of the leaf vertices can see each other. This process is well-defined, and does not make the obstacle non-convex.

An example of this construction is given below. Notice that all of the non-edges of S fall between the n leaf vertices, all of which are blocked by our convex obstacle. Therefore, this construction creates a valid 1-convex obstacle representation for any star graph, as desired. □

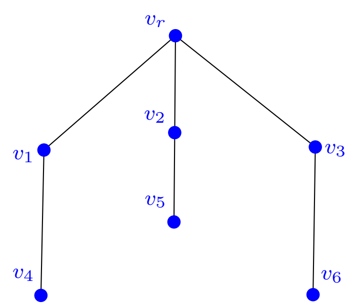


Now, using Lemma 2.7, we can create our desired construction for any Caterpillar graph. Let G be such a graph with m vertices in its spine, and construct an obstacle that is a regular m-gon. Arrange the m vertices in the spine of G so that each one is above exactly one side of the obstacle. It is easy to do this in such a way that two vertices in the spine see each other if and only if they are connected.



Now consider any vertex v_1 in the spine of G that is adjacent to vertices v_2 and v_3. In the diagram above, notice that region R is the area formed by the obstacle that neither v_2 nor v_3 has vision of. Thus we can arrange all other vertices adjacent to v_1 within region R in a manner that mirrors the construction used in the proof of Lemma 2.7, altering the edge of our obstacle as needed. Thus each of the leaves in our Caterpillar graph can only see the vertex in the spine that they are adjacent to, and it follows that all Caterpillar graphs have obstacle number 1, as desired.

For the second portion of the proof, we will define an n-*claw* to be a tree consisting of n paths of length 2 each originating at a common root vertex. Now, recall that a defining characteristic of Caterpillar graphs is that they do not have a 3-claw as an induced subgraph. Therefore, observation 2.5 tells us that proving the 3-claw does not have a 1-convex obstacle representation implies that the Caterpillar graphs are the only trees to have convex obstacle number 1.



In our proof that the 3-claw has obstacle number 2, we will use the labeling scheme from the graph G above, and proceed by contradiction. First assume that G has a 1-convex obstacle representation. Then from Lemma 2.6, we know that the convex hull of G must contain exactly one non-edge. We also know that this non-edge cannot be incident to the root vertex v_r. To see why, assume without loss of generality that this non-edge is $v_r v_4$. Then all of the other 2-paths in G must be completely contained in the triangle $\Delta v_r v_1 v_4$. However this situation is impossible because it contradicts Lemma 2.3. It follows that v_r must be adjacent to both of its neighbors in the convex hull. We will assume without loss of generality that these vertices are v_1 and v_2. Then the path $v_r v_3 v_5$ must fall within the triangle $\Delta v_r v_1 v_2$, which would require 2 convex obstacles to represent by Lemma 2.3. This means that the 3-claw, and therefore any graph that contains the 3-claw as an induced subgraph, cannot have a 1-convex obstacle representation. It follows that a tree T has convex obstacle number 1 if and only if it is a caterpillar graph, as desired.

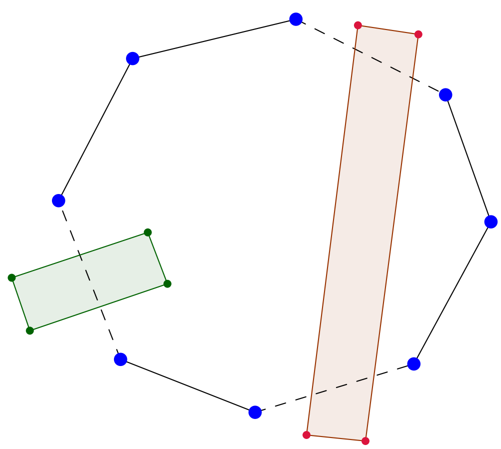
3 Trees with Convex Obstacle Number 2

3.1 Preliminaries

We begin by generalizing a previous result to help us narrow down the cases that we need to consider as we add more obstacles. The first is an extension of Lemma 2.6, which we proved in the previous section.

Lemma 3.1. *In any convex obstacle representation of a tree T with n obstacles, the convex hull of T cannot have more than n non-edges.*

Proof. We will proceed by contradiction. First assume that we have a convex obstacle representation of a tree T with n convex obstacles and $n+1$ non-edges in its convex hull. Now consider coloring each of our obstacles one of n colors. Then since we have $n+1$ non-edges in the convex hull, one colored obstacle, say the red one, must pass through two non-edges. If this is the case, then the interior of the red obstacle must pass through the interior of our tree T, which would separate at least one vertex from the root of T, giving us a contradiction. It follows that the convex hull of T cannot have more than n non-edges, as desired. □

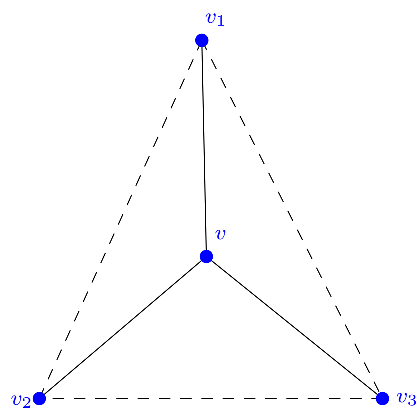


An contradiction in the case of $n = 2$

After studying the convex hull, the next best strategy for approaching problems involving two convex obstacles is to find limitations on the positions of the vertices themselves. The next Lemma is powerful in this regard, although it cannot be generalized beyond the 2-convex obstacle case.

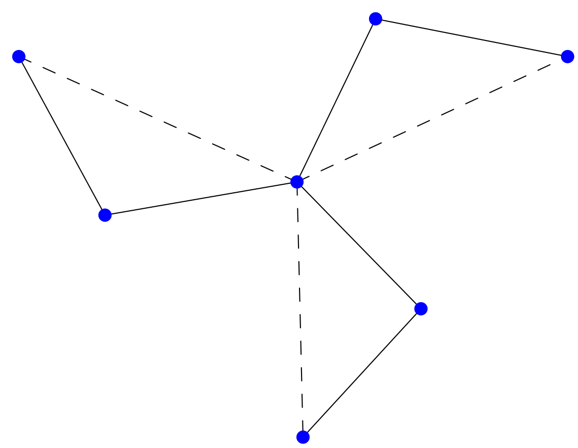
Lemma 3.2. *In any 2-convex obstacle representation of a tree T, if one vertex v of T has degree 3 or more, then all other vertices that are adjacent to v must be on the same side of some line through v.*

Proof. Assume that there was no such line as described above. Let v_1, v_2, and v_3 be neighbors of v. Then we must have a situation in which no angle between two edges has degree measure greater than or equal to 180 degrees, as shown in the diagram below:



But this leaves us with a representation whose convex hull has 3 non-edges. By Lemma 3.1, this is impossible with only 2 convex obstacles, and thus we arrive at a contradiction. It follows that there exists a line that passes through v such that all other vertices of T are on the same side of the line, as desired. □

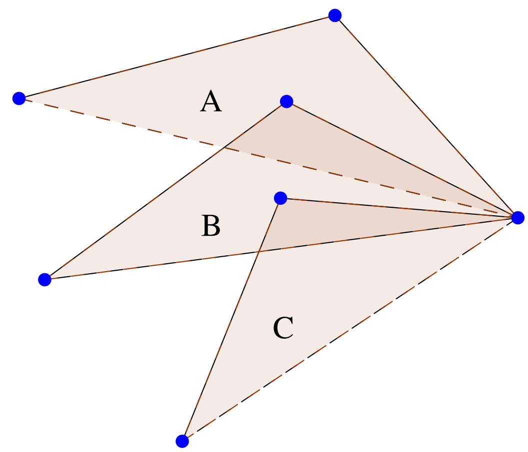
As we have seen in the previous section, n-claws are of central importance when proving our main results. This is still true in the 2-convex obstacle case, only now we need to be even more specific about how our claws can be represented in the plane. The following two Lemmas will be very helpful in this regard. The first is a generalization of an argument used in [Fu], and the second is a new technique introduced in this paper. We begin with some definitions. Consider the tree T that consists of a single n-claw with central vertex v. We will call the vertex in each path that is not the leaf the *joint* if the path. A *pinwheel representation* of T is a drawing of T in the plane such that a line rotated about the root vertex either clockwise or counterclockwise will always touch the joint of a path before its leaf. This gives us a set of n triangular *blades*, which are the triangular regions defined by each claw in T. We will call a pinwheel representation *simple* if no line can be drawn that connects two non-edges both incident to the root vertex without touching an edge of T. An example of a simple pinwheel representation is shown below.



With these definitions, we can now state our first Lemma:

Lemma 3.3. *A pinwheel representation of a tree T with n paths of length 2 (equivalently, n* blades*) requires exactly n convex obstacles to represent it.*

Proof. First, notice that any simple pinwheel representation of a tree T requires n convex obstacles by definition. If the same obstacle were to intersect two of the non-edges incident to the root vertex, then there must be a line through the obstacle that crosses an edge of T, which is a contradiction. Thus we only need to consider the special case when the representation is not simple. This occurs when triangular regions in the pinwheel representation overlap, such as the regions A and B as well as B and C in the image below:



However, in any such situation we can repeatedly apply Lemma 2.3 to show that we need n obstacles for the n paths. To do this, we individually consider all possible choices of 2 paths. In our example, we would look at the pairs (A, B), (A, C), and (B, C), where each letter refers to the path that creates the labeled blade in the above diagram. For each pair, there are exactly two cases: Either the triangular regions formed by each path overlap, or they do not. In the first case, the two paths require separate obstacles as a consequence of Lemma 2.3. In the second case, we have a simple pinwheel representation, which we have shown to requires 2 distinct obstacles. It follows that all pairs of paths require separate obstacles, and thus we need at least n convex obstacles in our representation, as desired. $\square$

Our next Lemma further limits the number of non-edges that can appear in the convex hull when we look at n-claws specifically. In this paper, we provide a slightly stronger result that applies to n-claws with paths of any length. Note that we need to assume no edge-crossings here, as we are no longer protected by Lemma 2.2 when we are working with more than 2 convex obstacles.

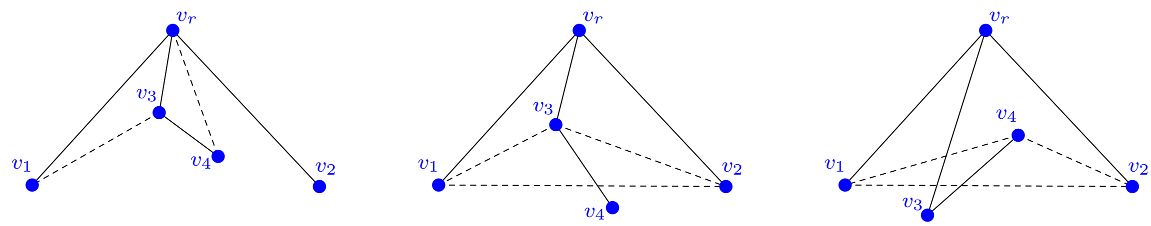
Lemma 3.4. *In an n convex obstacle representation of a tree T consisting of $n+2$ paths of length at least 2 that do not intersect and are incident to a single central vertex, there can be at most $n-1$ non-edges in the convex hull of T.*

Proof. We will proceed by contradiction. First, given a tree T with $n+2$ paths as described above, assume we have a representation of T that includes n obstacles and n non-edges in the convex hull. Let v_r be the central vertex of T, and let $v_1, v_2, \ldots v_{n+2}$ be the vertices in each path that are adjacent to v_r. For each $k \in [0, n]$, consider the $n+1$ triangles $\Delta v_r v_k v_{k+1}$. I claim that there must be a path that passes through one of these triangles. To see why, notice that in the best case scenario each of the n non-edges in the convex hull of T connect two distinct paths of T. This leaves us with $n+1$ paths that touch a non-edge in the convex hull and one that does not. These $n+1$ paths create n disjoint polygonal regions inside the convex hull of T, each corresponding to one of the triangles we defined earlier. If our one remaining path was not at least partially contained within one of these triangles, than it would be part of the convex hull. In this situation, since we do not allow edge crossings, the path would touch a non-edge in the convex hull as well, which is a contradiction. It follows that this path passes through triangle $\Delta v_r v_k v_{k+1}$ for some k, and is therefore completely contained in the polygonal region that contains $\Delta v_r v_k v_{k+1}$. However, since we have exactly n obstacles and n non-edges in the convex hull, only one obstacle can block non-edges within this region, which gives us our contradiction by Lemma 2.4. The desired result follows. □

Lastly, we turn our attention towards finding patterns that appear in every 2-convex obstacle representation of the 3-claw. Clearly, not all representations of the 3-claw allow us to use only two convex obstacles, but knowing which configurations do will be helpful for determining the convex obstacle number of trees that contain the 3-claw as an induced subgraph. Since Theorem 1 implies that all trees with convex obstacle number 2 or more contain the 3-claw as an induced subgraph, the following Lemmas will be useful whenever we must assume that a tree has convex obstacle number 2.

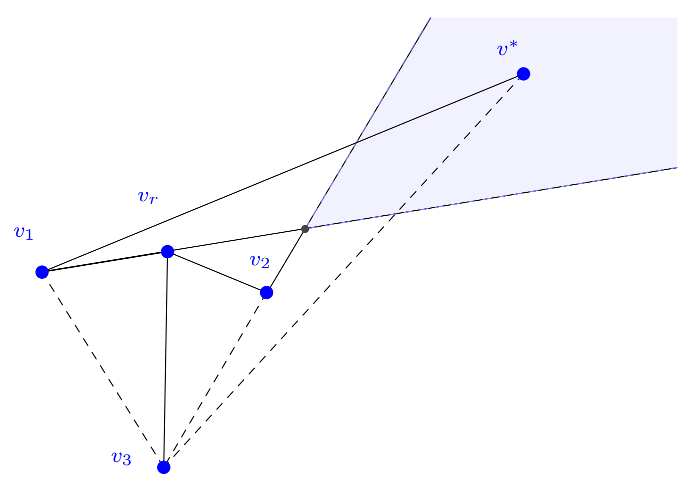
Lemma 3.5. *In any 2-convex obstacle representation of the 3-claw, the root of the claw must be part of the convex hull.*

Proof. Assume that we have a 2-convex obstacle representation of the 3-claw T. Recall from Lemma 3.2 that there must exist a line L through the root vertex v_r of T so that all neighbors of T are on the same side of L. Let v_1, v_2, and v_3 be the three neighbors of v_r, where edge $v_r v_3$ divides angle $\angle v_1 v_r v_2$. We claim that both v_3 and the leaf vertex v_4 adjacent to v_3 must lie outside of triangle $\Delta v_1 v_r v_2$. To see why, refer the three diagrams below.



In the first (leftmost) case, where both v_3 and v_4 are within $\Delta v_1 v_r v_2$, either the non-edges v_1v_3 and v_rv_4 require separate obstacles, or the non-edges v_2v_3 and v_rv_4 do. In the second case, where the path $v_rv_3v_4$ divides triangle $\Delta v_1v_rv_2$, we see that v_1v_3 and v_2v_3 require separate obstacles as well. Lastly, if v_3 lies outside of the triangle but v_4 does not, the non-edges v_1v_4 and v_2v_4 require separate obstacles. Thus in all cases, both obstacles touch non-edges within triangle $\Delta v_1v_rv_2$, and by observation 2.1, we have that v_r must be part of the convex hull of T, since no obstacle can exist above the line that passes through v_r and is parallel to the non-edge v_1v_2.

Now if we are to believe that v_r is not part of the convex hull of T, it must be the case that v_3 and v_4 are both outside of $\Delta v_1v_rv_2$. However, in this situation, the subgraph of T that contains only the vertices v_r, v_1, v_2, and v_3 already has two non-edges in its convex hull. Thus it will be useful to consider only the subgraph $S = T - \{v_4\}$, to take advantage of these two non-edges in our analysis. Doing so limits the area where vertices that are above the line L can be found. In particular, this tells us that there cannot be two leaves of S above this line whose non-edge is part of the convex hull of S. This would leave us with 3 non-edges in the convex hull of S, which contradicts Lemma 3.1. Thus we can only have a single vertex v^* above L that participates in the convex hull of T. In the diagram below, we highlight the only valid region for this vertex, assuming without loss of generality that v^* is adjacent to vertex v_1 (as opposed to v_2). separately).



The shaded region is defined by lines through v_3v_2 and v_1v_r

As long as v^* falls within this shaded region, we see that the convex hull of S will consist of the vertices $v_1, v_3,$ and v^*, and therefore it contains only two non-edges. If we place v^* anywhere else, then either we will have 3 non-edges in the convex hull, or v_r will be part of the convex hull, both of which are contradictions.

We now condition on the location of vertex v_6, which is adjacent to v_2. First, notice that the edge v_2v_6 cannot cross the non-edge v_3v^*, as this situation requires three separate obstacles to block non-edges v_1v_3, v_3v_2, and v_2v^*. If v_6 is inside this convex hull, then either the non-edges v_rv_6 and v_2v_3 intersect, or they do not. If they do not, then they require separate obstacles by observation 2.1. If they do intersect, then the subgraph formed by removing vertex v_3 is a pinwheel with 2 blades, which also requires two distinct obstacles. In either case, neither of the non-edges discussed can share an obstacle with v_1v_3, the other non-edge in the convex hull of T. Thus we have our contradiction, as we have only 2 convex obstacles to work with. It follows that v^* cannot exist in the highlighted region, and thus v_r cannot be part of the convex hull of S. It follows that v_r must be part of the convex hull of S. All that remains to be proven now is that this condition also holds for the larger tree T.

Since the root vertex of S must be part of the convex hull of S, the only remaining candidate for v^* in T is v_4, the leaf that is adjacent to v_3. In this situation, we can draw the same region for all possible locations of v^* from above, except this time v_3v^* is an edge while v_1v^* is a non-edge. The same argument from before can then be used to show that this situation also requires 3 convex obstacles. It follows that there exists a line through v_r such that *all* other vertices of T are to one side of the line, and thus it must be the case that v_r is part of the convex hull of any 2-convex obstacle representation of the 3-claw, as desired. $\square$

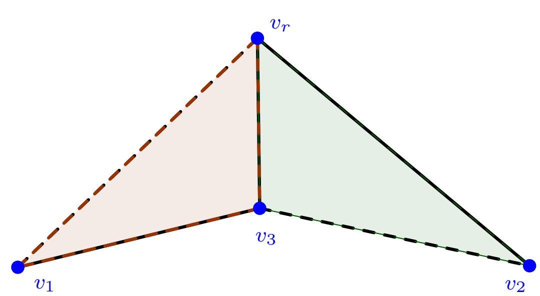
Now that we know where the root is located in any representation of the 3-claw, we turn our attention to our final preliminary Lemma, which provides us with more information about the layout of the edges in the convex hull.

Lemma 3.6. *In any 2-obstacle representation of an n-claw T, the root of T must be adjacent both of its neighbors in the convex hull of T for any $n > 2$.*

Proof. There are two cases to consider here. The first is where the root v_r is adjacent to neither of its neighbors in the convex hull, and the second is where it is adjacent to only one of its neighbors. We will show that both of these cases lead to contradictions when we only have two obstacles to work with.

1. First, assume that the root v_r of T shares non-edges with two vertices v_1 and v_2 in the convex hull of T. Since we have a 2-convex obstacle representaton, these must be the only two non-edges in the convex hull, and thus there is a path along the convex hull from v_1 to v_2. Since v_r is not adjacent to either vertex, there must be a third vertex v_3 such that v_3 is part of the path that contains v_r, v_1, and v_2. However, this path has length greater than 2, which is a contradiction, as no such path exists in the 3-claw.

2. Next, assume that v_r is adjacent to only one of its neighbors in the convex hull. In this case, we let v_rv_1 be the non-edge and v_rv_2 be the edge. Then v_1 must be a leaf of T and there is a third vertex v_3 adjacent to both v_1 and v_r. Now, since we are working with convex polygons, the non-edges v_rv_1 and v_2v_3 must be blocked by different obstacles. However, a third path of our claw must fall within exactly one of the triangular regions $\Delta v_rv_1v_3$ and $\Delta v_rv_2v_3$. This leads to a contradiction by Lemma 2.3, since each region can only contain a single obstacle.



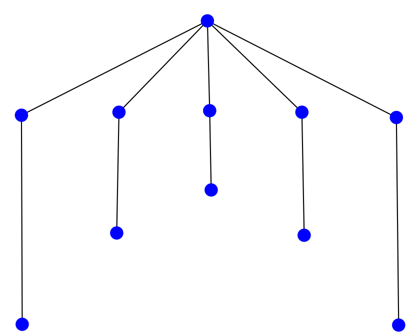
Since both cases above lead to contradictions, it follows that the root must be adjacent to both of its neighbors in the convex hull, as desired. $\square$

We end this section with a definition motivated by the preceding Lemma. We say that a path is *strictly convex* if every pair of edges in the path turns in the same direction. That is, if one were to traverse the path in entirety from one side to the other, then one would only make left turns or only make right turns when passing from edge to edge. Notice that for the 3-claw, Lemma 3.6 ensures that there are two claws which, when considered as a single path of length 4, form a strictly convex path, as otherwise there would be a non-edge in the convex hull that is incident to the root, which we have shown to be impossible.

3.2 Proof of Theorem 2

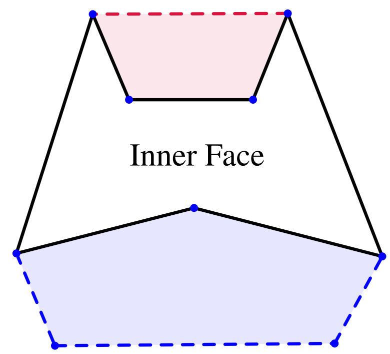
We begin our discussion of Theorem 2 with a motivating example. In the previous section, we saw that the 3-claw played an important role in the proof of Theorem 1. In this section, we will find that it is the 5-claw which is the center of attention, since it is the first n-claw to require 3 convex obstacles, as shown in the following theorem.

Theorem 3.7. *The 5-claw (depicted below) does not have a 2-convex obstacle representation.*



Proof. This proof follows from Lemma 3.3 concerning pinwheel representations. The pigeonhole principle tells us that in any convex obstacle representation of the 5-claw, three of the five claws must form a subgraph that has a pinwheel representation with 3 blades. However, with only two obstacles in our representation, this contradicts Lemma 3.3. It follows that the 5-claw does not have a 2-convex obstacle representation, as desired. □

Now, before we dive into the proof of Theorem 2, it will be necessary to introduce some new definitions. We will refer to the convex hull formed by the vertices of the polygonal obstacles in any obstacle representation as the *obstacle convex hull*. We will then call the *inner face* of an obstacle convex hull the area within the hull that is not occupied by an obstacle. Note that when we are not working with overlapping obstacles, there is exactly one inner face, an example of which is shown below:



In the above situation, we have two edges in the obstacle convex hull that are not a part of either obstacle. In fact, it can be easily shown that with two obstacles that do not intersect, there will always be two such edges, which we will call the *sides* of the inner face. The key insight that makes the inner face so important for the proof of Theorem 2 is the following, which we state as an observation.

Observation 3.8. *Given a tree T and a 2-convex obstacle representation of T, the subgraph S of T that is not contained within the inner face of the obstacle convex hull must be a disjoint unions of caterpillar graphs. In other words, S cannot contain the 3-claw as an induced subgraph.*

The logic behind this observation is that the convex hull of two obstacles is itself a convex polygon. Thus if you replace the two obstacles in a representation with the single convex obstacle that is equivalent to the obstacle convex hull, the subgraph S outside of the inner face must have a valid 1-convex obstacle representation. This follows because S must have been part of a valid representation when the obstacles were separate, and filling in the inner face does not change the positions of any of the vertices in S.

Now that we know what cannot be outside of the inner face, the natural question to ask is what cannot be inside of the inner face. If we understand both, we can get a constraint on the type of trees that can be represented with two convex obstacles. In order to answer this question, we need to make use of the following Lemma, which will continue to be important as we proceed through the proof of Theorem 2.

Lemma 3.9. *In any 2-convex obstacle representation of a tree T in which the convex hull of T contains only a single non-edge, there is some vertex $v \in T$ that is not in the inner face of the two obstacles.*

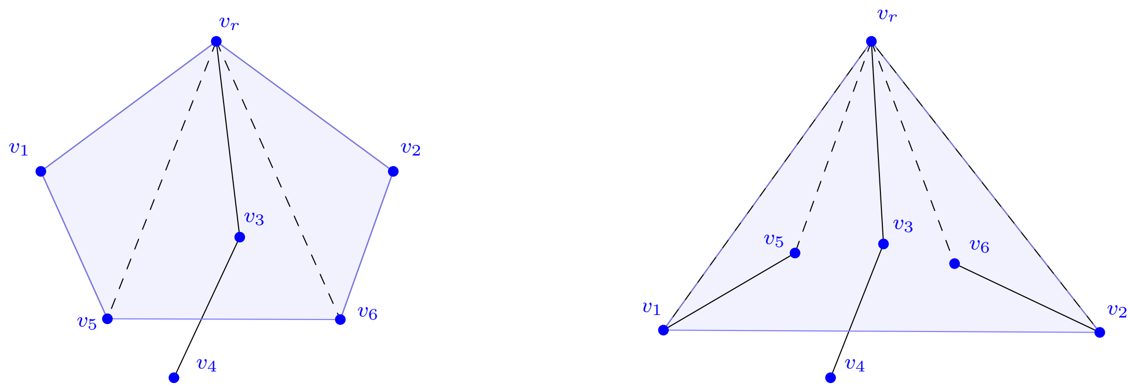
Proof. Consider a tree T as described above, where v_1v_2 is the single non-edge in the convex hull. Since we are working with convex obstacles, extending the segment v_1v_2 to a line L divides the plane into two disjoint regions, call them U and V. All of the vertices of T must fall into one of these regions, say region V without loss of generality. Then the union of the convex hull of T and region U defines the entire space where the obstacles can exist, since we are requiring that the obstacles be convex. We will call this combined region R. Now, consider what happens when we shift our line L in a single direction until moving it any further will cause it to leave region R. At this point, either L intersects R at a single vertex v_3, or at an edge v_3v_4 that is parallel to v_1v_2. Since our obstacles are in a fixed position in any representation of T, we can shift L a distance ϵ in the opposite direction as before so that L separates the two obstacles from the vertex v_3 (and from v_4 if necessary). But then this vertex cannot exist within the inner face of the obstacle convex hull, and it follows that T cannot be completely contained within the inner face, as desired. □

This Lemma allows us to say that, for any representation of a tree T, if we can find a subgraph S of T such that the convex hull of S contains both obstacles and only a single non-edge, then some vertex in S cannot be part of the inner face of the obstacle convex hull. This Lemma gives us everything we need to prove the following, which answers the question posed above about what trees can and cannot be found within the inner face of a 2-convex obstacle representation.

Lemma 3.10. *No tree T that that contains the 3-claw as an induced subgraph can be drawn entirely within the inner face of a 2-convex obstacle representation of T. That is, in any 2-convex obstacle representation of T, the subgraph S of T whose vertices all fall within the inner face must be a disjoint union of caterpillar graphs.*

Proof. First, consider any 2-convex obstacle representation of the 3-claw T. Lemma 3.5 and 3.6 taken together tell us that the root vertex v_r of T must be part of the convex hull of T and adjacent to both of its neighbors. Let v_1, v_2, and v_3 be all vertices adjacent to v_r, where edge v_rv_3 divides $\angle v_1v_rv_2$. We know that there is a subgraph S of T that contains vertices $v_r, v_1,$ and v_2, and whose convex hull has exactly one non-edge and contains all of the vertices of S. In other words, the vertices of S taken together form a convex polygon with a number of sides equal to the number of vertices in S. In the simplest scenario, S would only contain the vertices $v_r, v_1,$ and v_2, and therefore the convex hull of S would contain only the non-edge v_1v_2. Let P be the largest such subgraph of T, that is, the one that contains the most vertices of T and whose convex hull has only one non-edge. Now let v_4 be the leaf adjacent to v_3. We will condition on whether or not v_4 can be found within

P, and show that in either case both obstacles must exist within P. First, if the entire path $v_r v_3 v_4$ is contained within P, then notice that the non-edge $v_r v_4$ will intersect exactly one of the non-edges $v_1 v_3$ or $v_2 v_3$. The one that it does not intersect requires a distinct obstacle by observation 2.1, and thus both obstacles must exist within P. However, if at least part of the path $v_r v_3 v_4$ exists outside of P, we must use a different approach. Let v_5 and v_6 be the vertices adjacent to v_1 and v_2. Then the non-edges $v_r v_5$ and $v_r v_6$ must be within our polygon P, as shown below. If they weren't, then v_r would be adjacent to one or both of them in the convex hull of T, which contradicts Lemma 3.6.



Two possible configurations for T. The shaded region is the polygonal subgraph P

However, as shown above, the path $v_r v_3 v_4$ divides P into two disjoint regions, which makes it impossible for a single convex obstacle to touch both $v_r v_5$ and $v_r v_6$. Therefore both obstacles must exist within P, as in the previous case. It now follows from Lemma 3.9 that the 3-claw cannot exist within the inner face, since both obstacles must fall within the polygonal region P. Thus, the only trees that can exist within the obstacle convex hull are the caterpillars, as desired. □

With these results in hand, we are now in a position where we can prove Theorem 2. Our proof can be divided into three main steps, which are outlined below. Each step corresponds to one of the results from this section, and Theorem 2 is a direct consequence of the three.

(1) First, we will prove that, if a tree has a 2-caterpillar coloring, then it must have a 2-convex obstacle representation. See Theorem 3.11.

(2) We will then prove that a tree has a 2-caterpillar coloring if and only if it cannot be reduced to the 5-claw through a series of edge contractions. See Theorem 3.14.

(3) Lastly, we prove that any tree that can be reduced to the 5-claw through a series of edge contractions does not have a 2-convex obstacle representation. See Theorem 3.15.

Theorem 3.11. *If a tree T has a 2-caterpillar coloring, then it has a 2-convex obstacle representation.*

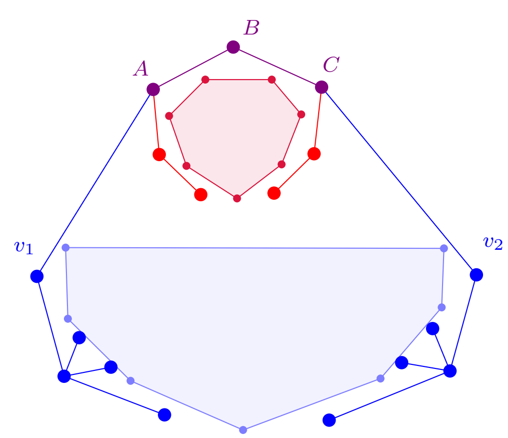
Proof. We will prove this claim by construction. Assume that we have a 2-caterpillar coloring of a tree T using the two colors red and blue. Our construction proceeds as follows:

1. Without loss of generality, start by creating the standard 1-convex obstacle representation from section 2.2 for the red Caterpillar subtree.

2. Next, for each vertex in the blue group that is adjacent to a vertex that is both red and blue (there can be at most two, v_1 and v_2), draw them so that there exists a line that puts all red vertices on one side of the line, and both v_1 (and v_2, if applicable) on the other.

3. Now, if there are n vertices in the spine of the blue caterpillar that are not also red, draw the second obstacle as the bottom-half of a regular $2n$-gon. The top-half of this obstacle should fall directly on the line mentioned in step 2, and it should ensure that v_1 (and v_2) cannot see any red vertex except for the ones adjacent to it.

4. Lastly, return to the technique from section 2.2 to wrap the blue vertices around the second obstacle, thus completing the construction. An example created by this process is given below:



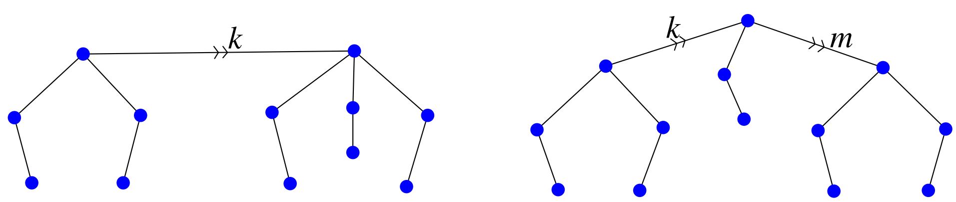
Note: vertices A,B, and C are colored both red and blue.

□

For part (2) of the proof of Theorem 2, we will introduce the process of *pruning* a tree using edge contractions. This process consists of two steps: First, all edges incident to a leaf and a vertex of degree 3 or more are contracted away. Second, all edges that are incident to the remaining leaves are removed unless doing so would put a leaf vertex adjacent to another vertex of degree greater than 2. The second step is then repeated until no more edges can be contracted while satisfying the desired condition. A key observation here is that the edge contractions in both steps of the pruning process can be replaced by a series of leaf deletions with the same end result. Thus any pruned tree will be an induced subgraph of the original tree.

Let T be the tree that we started with, and let T' be the tree we get after pruning. Then by definition, all leaves in T' will be adjacent to a joint vertex, which is in turn adjacent to a vertex of degree 3 or higher. We will refer to these paths of length 2 as *claws*, mirroring the term we used in the definition of an n-claw. This leads us to the following Lemma, which provides a classification for all trees that reduce to the 5-claw.

Lemma 3.12. *Any tree T that is reducible to the 5-claw through a series of edge-contractions either has the 5-claw, or one of the following two families of graphs as an induced subgraph.*



where the marked edges denoted paths of length k or m, where $k, m > 0$. We will refer to the family on the left as the (4,3) family, and the one on the right as the (3,3,3) family, after the degree of the largest vertices in each.

Proof. We begin with the following observation about the pruning process defined above.

Observation 3.13. *The set of all edges removed from a tree T during the pruning process is a subset of the edges we must contract when reducing T to the 5-claw.*

This observation makes sense due to the conformation of edges and vertices in the 5-claw. In the first part of the pruning process, we remove all leaves that are adjacent to vertices of degree greater than 3, which are not present in the 5-claw. Furthermore, in part 2 we reduced all paths that end in leaves to paths of length 2, which are the only paths that exist in the 3-claw. Thus in both steps, we never removed an edge that would have appeared in the final reduced version of our tree.

Now assume that we have a tree T that is reducible to the 5-claw. Let T_1 be the tree we get after pruning T, and assume that T_1 has n vertices of degree 1. Notice that n must be greater than or equal to 5, since the 5-claw requires exactly 5 leaves. Furthermore, all of the leaves of T_1 are parts of claws. Thus to reduce T_1 to the 5-claw, we must remove all of these claws except for 5 (which can be accomplished either through edge contractions or vertex deletions). Let T_2 be the subgraph of T_1 that has only 5 of these claws.

We now claim that T_2 must either be the 5-claw, a tree in the (4,3) family, or a tree in the (3,3,3) family. To see why, first notice that there can be at most three vertices with degree greater than 2 in T_2. All 3 of these vertices $v_1, v_2,$ and v_3 must be connected, which gives us a single path that contains all three. We will now build a valid pruned tree T_3 from this path, and show that T_3 must be part of the (3,3,3) family. Let v_2 be in between v_1 and v_3 in this path. Now, since we want T_3 to be a tree reachable through pruning, the only way to increase the degree of a vertex is to attach a claw to it. Then we can attach a single claw to both v_1 and v_2 in T_3, which makes all three vertices have degree 2. But we want v_1, v_2 and v_3 to all have degree 3. This can only be accomplished by attaching one more claw to each, and it follows that the (3,3,3) family of trees is the only possible family with 3 vertices of degree greater than 2. A similar argument can be made for the situation where only 2 vertices have degree 3 or higher. In this case, we once again begin with a single path connecting two vertices v_1 and v_2. Adding claws onto each in order to make them both have degree 3 leaves us with a tree with four leaves. Thus we can only attach one more claw to either v_1 or v_2, since we assumed that our tree has at most 5 vertices. Both choices are equivalent by symmetry, and thus the (4,3) family is the only possible family with 2 vertices of degree greater than 2. It follows that any tree T that is reducible to the 5-claw through a series of edge contractions either has the 5-claw, or a tree in the (4,3) or (3,3,3) families as an induced subgraph, as desired. □

Theorem 3.14. *A tree T has a 2-caterpillar coloring if and only if it cannot be reduced to the 5-claw by edge contractions.*

Proof. (=>) First, assume that a tree T has a 2-caterpillar coloring with red and blue colors. Let T' be the subtree generated by the pruning process described above. Then it must be the case that T' has no more than four vertices of degree 1. This is because pruning a Caterpillar graph will leave us with a path, as the only leaves that remain after pruning are the endpoints of the spine. Thus T' will be the concatenation (with overlap) of two paths, specifically, the spine of the red Caterpillar and the spine of the blue Caterpillar, which means it can have at most four leaves. But the 5-claw has 5 leaves, and thus T' cannot be reduced to it through edge contractions alone. It follows by observation 3.13 that T cannot be reduced to the 5-claw, as desired.

$(<=)$ Now assume that we have a tree T that does not reduce to the 5-claw by edge contractions, and assume that it does not have a 2-caterpillar coloring. Notice that we only need to consider the case where T has a 3-caterpillar coloring, as we could reduce any tree T with caterpillar number $n \geq 3$ to a subtree with caterpillar number 3. Now, once again we will let T' be the tree generated by pruning T, and note that none of the vertices removed during pruning could have been part of the spine of any colored caterpillar. Thus the caterpillar number of T' will be the same as the caterpillar number of T, which is 3. This implies that T' will have at least 5 leaves. If it didn't, then it is easy to show that we could construct a 2-caterpillar coloring using the 4 leaf vertices as the endpoints of each spine. For this proof, we will assume that T' has exactly 5 leaves. If it had more, we could once again reduce T' to 5 leaves through a series of edge contractions. But we know by Lemma 3.12 that the only trees that have 5 leaves after pruning are the 5-claw itself, or the trees in the (4,3) and (3,3,3) families. However, all of these trees are reducible to the 5-claw, which is a contradiction. It follows that any tree that is not reducible to the 5-claw through edge contractions must have caterpillar number at most 2, as desired. □

Finally, we arrive at part (3) of our proof of Theorem 2. This portion of the proof is a lot more involved than the previous ones, but is nevertheless pretty straightforward once we reduce the number of cases that need to be considered.

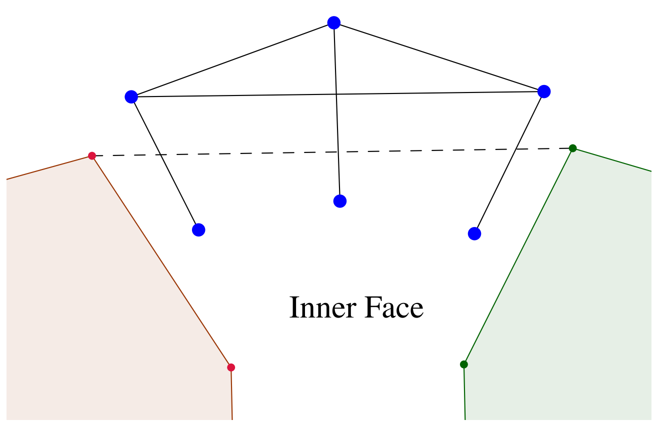
Theorem 3.15. *Any tree that reduces to the 5-claw through edge-contractions has convex obstacle number at least 3.*

First, recall that the only trees that reduce to the 5-claw have either the 5-claw itself as an induced subgraph, or a tree in either the (4,3) or the (3,3,3) families as an induced subgraph. As we have already proven that the 5-claw does not have a 2-convex obstacle representation, we only need to show that neither the (4,3) nor the (3,3,3) families can be represented using only two convex obstacles. To do this, we will return once again to the idea of the obstacle convex hull and the inner face of a set of two obstacles. Recall that the inner face is the area of the obstacle convex hull that is not covered by the obstacles.

Also recall that the obstacle convex hull must have exactly two sides, as otherwise the obstacles would overlap. For this proof, we assume that both the (3,3,3) and (4,3) families have a 2-convex obstacle representation, and find a contradiction in each case. We begin by recapping what we already know. First, we know that there cannot be a 3-claw contained entirely within the inner face. We also know that a 3-claw cannot be entirely outside of the convex hull either. For general trees, this is all we know so far, but in the particular case of the (4,3) and (3,3,3) families, we know more about the structure of the trees we want to represent, which allows us to be more specific about what cannot be found within the inner face. In particular, we will prove the following Lemma:

Lemma 3.16. *In any 2-convex obstacle representation of a tree T in the (4,3) or (3,3,3) families defined above, no vertex of degree 3 or more in T can be inside the inner face of the obstacles.*

Proof. To prove this claim, first notice that it is sufficient to prove the case where a vertex has degree 3, since we can reduce a vertex's degree through a series of vertex deletions. We will let v_3 be the vertex of degree 3 in the tree T that we are considering. Now, notice that if v_3 is in the inner face, then there must be at least one edge crossing through a side of the obstacle convex hull, as otherwise all of T would be inside the inner face, which is not possible. Furthermore, there cannot be more than 3 paths that start at v_3 and exit the inner face, as then we would have a cycle. This is because any two vertices that are on the outside of the obstacle convex hull must see each other if they are incident to vertices within the inner face, as shown in the diagram below:

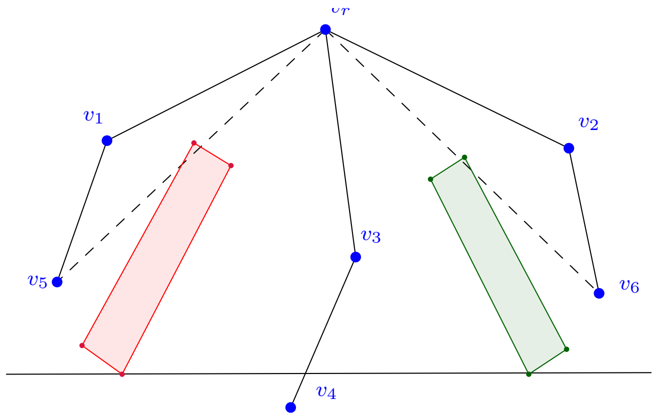


A similar argument can be made if we have only two paths, both of which start at v_3 and then pass through the same side of the inner face. Thus there can be at most two paths from v_3 that exit the inner face, and no two of these paths can exit the inner face through the same side of the obstacle convex hull. Thus we either have only a single edge leaving the inner face, or a pair of edges that exit through opposite sides. We will show that no matter which of these situations we are in, we will run into a contradiction.

1. First, assume that there are two paths that start at v_3, and then exit the inner face on opposite sides. Then these paths divide the inner face into two disjoint regions, each of which can only contain one of the two obstacles. Now, since T is a member of the (3,3,3) or (4,3) families, there must be a third path of length 2 that starts at v_3 and never leaves the inner face. Let v_1 and v_2 be the vertices adjacent to v_3 that are not part of this path. Then the triangle $\Delta v_1 v_2 v_3$ must lie in exactly one of the disjoint sections of the inner face. Furthermore, Lemma 3.2 ensures that the path of length 2 is contained in the same region as our triangle. But this means that we have only one obstacle to represent a path of length 2 that is at least partially contained within the triangular region $\Delta v_1 v_2 v_3$. This contradicts Lemma 2.3, and tells us that we would need a third obstacle in our representation, as desired.

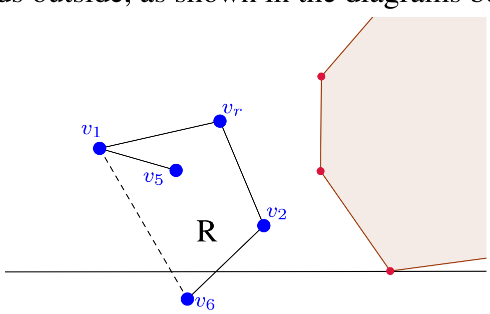
2. Now, assume that only a single edge exits the inner face of our obstacles. Again we can infer that the remaining two paths that start at v_3 have length 2, and are contained entirely within the inner face. But the structure of the (3,3,3) and (4,3) families of trees implies that the path which exits the inner face also contains at least 2 edges, and thus we have the 3-claw as an induced subgraph. By deleting all other vertices in our tree T except for those in these three paths of length 2, we should be left with a valid 2-convex obstacle representation of the 3-claw. We will now show that this cannot be the case, no matter how we arrange the vertices of our claw. The set of all such configurations can be partitioned into two cases, which we describe below.

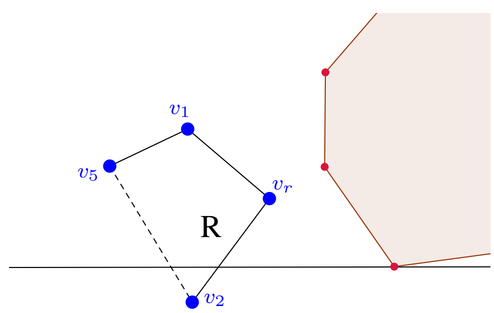
   a. First, assume that the path which leaves the inner face is not incident to the root vertex v_r in the convex hull. Let v_1, v_2 and v_3 be the vertices directly adjacent to the root vertex v_r, and let v_5, v_4, and v_6 be the leaf vertices adjacent to v_1, v_2 and v_3, respectively. If the path $v_r v_3 v_4$ is the one that originates inside of the convex hull, then by assumption this is the path that exits the obstacle convex hull. This implies that v_3 and v_4 are the only vertices that may be outside of the inner face. That is, all other vertices of T must be inside the inner face. We will show that this cannot be the case. To see why, notice that the path $v_r v_3 v_4$ divides the convex hull of T into two sections, each of which must contain exactly one of the convex obstacles. This follows because no line can be drawn from a point on the non-edge $v_r v_5$ to a point on the non-edge $v_r v_6$ without intersecting the path $v_rv_3v_4$, as shown in the figure below, where the horizontal line that divides v_3v_4 denotes a non-edge in the obstacle convex hull.



Thus the convex hull of the tree $S = T - \{v_3, v_4\}$ contains both obstacles, and from Lemma 3.9, we have that some vertex of S is not contained within the inner face of the obstacles. But we assumed that all of the vertices of S were inside the inner face, and thus we have a contradiction.

b. Next we consider the case where the path that leaves the inner face is incident to the root vertex in the convex hull. That is, if the path $v_rv_2v_6$ from the labeling above now leaves the inner face, then the edge v_rv_2 is part of the convex hull of T. If the path $v_rv_1v_5$ is the other such path, then by Lemma 3.6 the combined path from v_5 to v_6 is a strictly convex path that starts inside the inner face, and ends outside, as shown in the diagrams below:

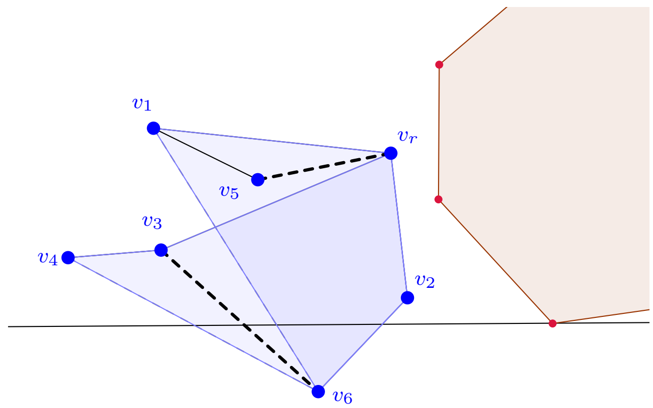




In the figure on the left, we see that it is the edge v_2v_6 that leaves the inner face, while in the figure on the right the edge v_rv_2 leaves the inner face, and v_2v_6 is not pictured. We will condition on which of these two edges is the one that actually exits the inner face, and the argumentation in each situation will be very similar. In the first case, consider the ray L formed by extending the edge v_2v_6 in the direction of v_2. Next, let region R be the area of the convex hull of our path formed by sweeping ray L from the direction of v_2 either clockwise or counterclockwise along the edges of this strictly convex path until moving L any further would cause it to not intersect any of the edges in our path. The non-edges drawn in the above diagrams highlight this region R in either of the two cases. Notice that the vertex v_5 need not be on the convex hull of this path, as the left figure shows.

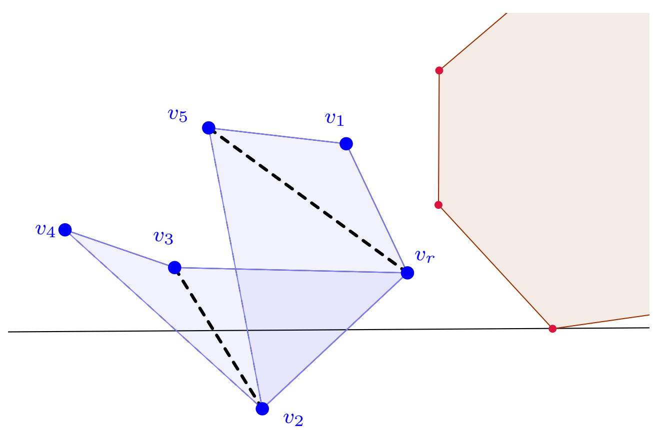
We now claim that only one obstacle can be found inside the highlighted region. To see why, let e be the non-edge in the convex hull of our obstacles that crosses v_2v_6. Then both obstacles must intersect e. Let p_1 be the point of intersection for the first obstacles and let p_2 be the point of intersection for the second. since the path from v_5 to v_6 is strictly convex, if there is a line from p_1 to R that does not cross any edges of our tree, then there cannot be such a line from p_2 to R. It follows that there cannot be any two non-edges inside R that require separate obstacles, as only one of the two obstacles can exist within this region. This means that the third path $v_rv_3v_4$ cannot be entirely contained within R, as then a familiar application of observation 2.1 gives us two non-edges that require different obstacles. Likewise, if the path $v_rv_3v_4$ leaves region R, but one of the vertices (say v_3, but this argument also works for v_4) remains in R, then the two non-edges v_1v_3 and v_2v_3 in R will require distinct obstacles.

Thus we are left with the situation where both v_3 and v_4 are outside of R. Here we cab define a second region R_2 as the area formed by taking our original ray L and sweeping it along the path from v_6 to v_3. Following the argument for region R above, we know that only one obstacle can be found in region R_2, and this obstacle must be the same as the one in R. However, the path $v_rv_3v_4$ separates the two regions, and since our obstacles are convex it is impossible for the same obstacle to occupy both regions, which gives us our contradiction. This situation is illustrated in the figure below:



The highlighted non-edges require separate obstacles

We now return to the second case mentioned above, where the edge that leaves the inner face is $v_r v_2$ instead of $v_2 v_6$. Notice that this situation is almost identical to the previous one, except we now extend ray L along the edge $v_r v_2$ in the direction of v_r. But this still allows us to define our regions R and R_2, and thus following the logic from the preceding argument gives us the same contradiction, which we illustrate below.



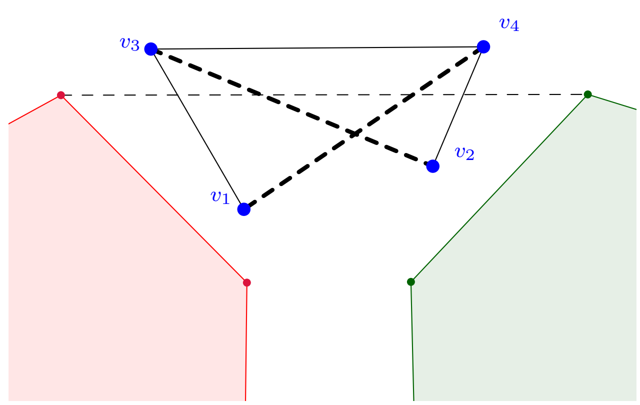
The highlighted non-edges require separate obstacles

Since both of these cases lead to contradictions, this completes the proof of Lemma 3.16. □

We now return to the proof of Theorem 3.15. We know from Lemma 3.16 that all of the vertices of degree 3 or greater must be outside of the inner face. However, we also know that there must be something inside of the inner face, as otherwise we would have a 1-convex obstacle representation of our tree, which is not possible. Thus the only possible candidates for what can be inside the inner face are leaves and joints.

Now, it is easy to see that in both the (4,3) and (3,3,3) families of trees, having only one edge entering the inner face will force a 3-claw to exist either outside or inside of the obstacle convex hull. This is because it is impossible to remove a single edge in either family and be left with a forest of two Caterpillars. Similarly, no two edges can be removed to leave us with a forest of three Caterpillars. It follows that there must be three edges in T that cross into the inner face of our two obstacles. By the pigeonhole principle, we know that two of these edges enter the inner face through the same side. Thus for the proof of Theorem 3.15, we will condition on these two edges, and show that no matter how they are arranged in a given configuration, we will always run into a contradiction.

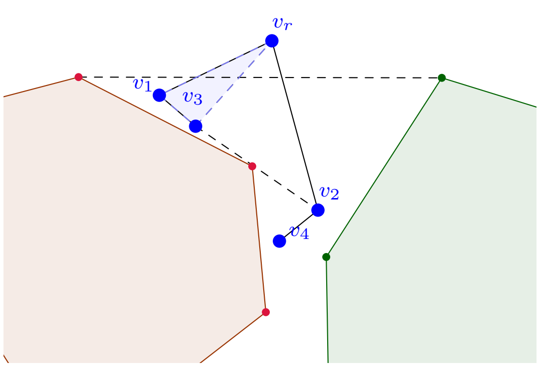
Proof of 3.15. We begin by considering the situation where our two edges are incident to distinct vertices outside of the obstacle convex hull. Let v_1 and v_2 be the vertices inside the inner face that are adjacent to v_3 and v_4 on the outside. Assume without loss of generality that v_1 and v_3 are to the left of v_2 and v_4 in our 2-convex obstacle representation. Let e be the non-edge in the obstacle convex hull that crosses the edges v_1v_4 and v_2v_3. Then both convex obstacles must touch e, and by observation 2.1, the left obstacle cannot touch the non-edge v_1v_4, while the right obstacle cannot touch the non-edge v_2v_3, as shown in the diagram below:

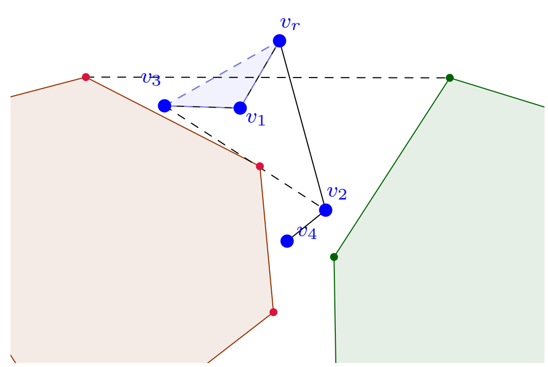


This means that the left obstacle must cross v_2v_3 and the right obstacle must cross v_1v_4. However, if the right obstacle crosses v_1v_4, then there is no line from the intersection of the left obstacle and e that passes through v_2v_3 without touching v_1v_3. Likewise, if the left obstacle touches v_2v_3, by symmetry, there is no line from the intersection e and the right obstacle that touches v_1v_4 without crossing the edge v_3v_4. It follows that the two edges that cross the same side of the inner face must both be incident to the same vertex.

Now let v_r be the vertex outside of the inner face to which both edges are incident. We will continue the labeling scheme from above by letting v_1 and v_2 be the vertices adjacent to v_r within the inner face. Now we know that v_1 and v_2 cannot both be leaves, since no such configuration exists in either the (4,3) or (3,3,3) families. Thus one of these two vertices must be a joint, and without loss of generality we will assume that v_4 is a leaf adjacent to v_2, and that v_1 is a leaf. Then again by the configuration of the (3,3,3) and (4,3) families of trees, it must be the case that v_r is a joint vertex. But this means that v_2 must be a vertex of degree greater than 2, which contradicts Lemma 3.16.

It follows that neither v_1 nor v_2 can be leaf vertices, and so there must be two vertices v_3 and v_4 that are adjacent to v_1 and v_2, respectively, so that v_3 and v_4 are leaves while v_1 and v_2 are joints. Then without loss of generality, we know that exactly one obstacle can touch the non-edge v_3v_2. This follows because observation 2.1 ensures that only one of the obstacles can be inside triangle $\Delta v_rv_2v_3$, as both v_rv_2 and v_rv_3 cross one side of the inner face. However, the convexity of our obstacles implies that the same obstacle that blocks v_2v_3 must also touch the non-edge v_rv_3. But no matter where this non-edge is located, observation 2.1 once again tells us that v_rv_3 cannot be blocked by the same obstacle that blocks v_2v_3, as illustrated in the following two diagrams.





Thus we have shown that two paths of length 2 cannot enter the inner face through the same side of the obstacle convex hull for any tree in the (4,3) or (3,3,3) families. However, this means that there must be a 3-claw outside of the inner face, which is a contradiction. It follows that neither the (4,3) nor the (3,3,3) families of trees have a valid 2-convex obstacle representation, as desired. This completes the proof of Theorem 3.15. □

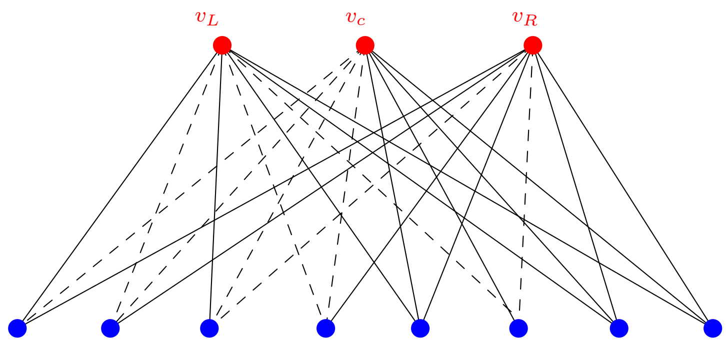
4 Additional Results

4.1 An Infinite Family of Bipartite Graphs

One topic that has been investigated by many in the world of obstacle representations is the question of which graphs have obstacle number 1. As explained in the introduction, this question is well understood for many families of graphs. However, constructions for such graphs with obstacle number 1 have not been addressed directly. In this section, we study the particular case of bipartite graphs, and we identify two infinite families of bipartite graphs that have obstacle number exactly 1. In our results, we also take the next step and identify a construction that will generate a 1-obstacle representation given any bipartite graph in our families. Note that in this section alone, we allow our obstacles to be non-convex.

Theorem 4.1. *Any bipartite graph with a partite group of size 3 has obstacle number 1.*

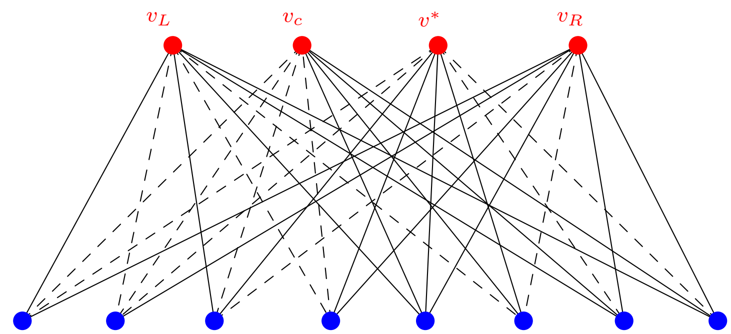
Proof. Consider a bipartite graph G with independent sets S and T that is a subset of the complete bipartite graph $K_{3,n}$. Let S denote the independent set of size 3 and T denote the independent set of size n. Now consider an arrangement of the vertices in G in which all vertices in the same independent set are positioned on the same line in a plane, and the lines for each set are parallel to each other. Notice that in this configuration, the complete graph $K_{3,n}$ has obstacle number 1, as all non edges between vertices in the same partite set can be blocked with a single obstacle in the outside face of the graph. We will prove that this obstacle can be altered so as to generate a 1-obstacle representation for any choice of G. First, it is easy to see that any non-edge that is incident to the right or leftmost vertex in a partite group must touch the outside face. Since our obstacle exists in the outside face, it follows that we can use this obstacle to block any of these non-edges. This just leaves the non-edges incident to the center vertex v_c in S. To show that one obstacle is sufficient to cover these, we will order the vertices in T so that every vertex that is adjacent to v_c is to the right of all vertices that are not adjacent to v_c. With this division, we see that all of the non-edges will be the leftmost edges incident to the center vertex in S, and therefore they can all be blocked by our one obstacle. This completes the proof that the obstacle number of G is 1, for any choice of G. □



Note: All non-edges touch the outside face of G

Theorem 4.2. *Any bipartite graph with a partite group of size 4 has obstacle number 1.*

Proof. Consider a bipartite graph G with independent sets S and T that is a subset of the complete bipartite graph $K_{4,n}$. Let S denote the independent set of size 4, and let T denote the independent set of size n. We will provide a configuration for these vertices in the plane that must have obstacle number 1. To begin, choose 3 of the 4 vertices in S and create the configuration from Theorem 4.1. We will let $v_L, v,$ and v_R denote the vertices of S we chose in order from left to right. Notice that this subgraph of G will have obstacle number one. Let v^* denote the one vertex in S that was left out from our construction. We now arrange the vertices of T into two groups: group 1 contains the vertices that are adjacent to v, and group 2 contains all vertices that are not adjacent to v. Within each group, we can rearrange the vertices without changing the number of obstacles required, so we can move all edges from group 1 to the rightmost legal position, and we can move the edges from group 2 to the leftmost legal position (where a legal position here is one in which the construction conditions from Theorem 4.1 are met). In this updated configuration, we see that the non-edges of v^* are incident to the rightmost and leftmost vertices in group 1 and 2, respectively. That is, the neighbors of v^* are all next to each other in our representation. It follows that we can extend our outside obstacle on both sides of v^* to cover these non-edges, and thus G has obstacle number 1, for any choice of G. $\square$



Note: All non-edges touch the outside face of G

4.2 An Extension of Outside Obstacle Representations

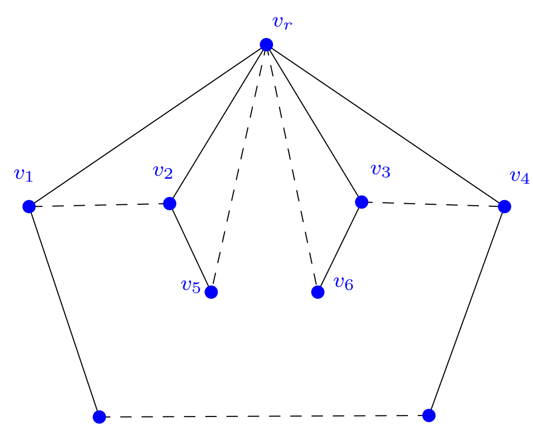
An interesting question that remains open is whether or not all planar graphs require an *outside obstacle representation*, that is, a representation in which all obstacles touch the outside face in the drawing of the graph. With respect to the convex obstacle number of trees, the concept of outside obstacles does not directly translate, as without crossings any obstacle will touch the outside face since its the only face of the tree.

We will instead consider a related problem, which is whether every convex obstacle in the representation of a tree T exists outside the convex hull of T. In other words, does there exist a tree T that requires an obstacle entirely contained within the convex hull of T. Clearly when we have only one obstacle, it must touch outside the convex hull, as the hull must contain a non-edge that is blocked by the obstacle. When we consider two obstacles, however, we get the following result:

Theorem 4.3. *In any 2 convex obstacle representation of the 4-claw, one obstacle must be entirely contained within the convex hull.*

Proof. We begin by appealing to Lemma 3.4, which tells us that our tree T must have only a single non-edge in its convex hull (since $n = 2$ which is the number of convex obstacles we are working with). Furthermore, we can conclude that the root vertex must be on the convex hull. This follows from repeated applications of Lemma 3.5 on all possible 3-claw subgraphs of the 4-claw. Finally, Lemma 3.6 proves that v_r must be adjacent to both of its neighbors in the convex hull of T. This tells us that there must be two paths of length two completely contained within the convex hull, which significantly reduces the number of cases we must condition on.

We now consider the arrangement of these two interior paths. First, notice that we can eliminate any situation in which a subgraph of T has a 3-pinwheel representation, as such a situation would require three obstacles. We also know that deleting a single claw in our representation of T must give us a valid representation of the 3-claw (which, by Lemma 3.6, is a representation in which the root is adjacent both of its neighbors in the convex hull). Thus the only valid 2-convex obstacle representations of the 4-claw are those in which a line through the root vertex v_r separates the 4-claw into two pinwheel representations, such as the configuration in the figure below.



We will now show that there is no way to arrange two convex obstacles in a representation of T so that both of them touch the one non-edge in the convex hull. We condition on whether or not the non-edges v_rv_5 and v_3v_6 are blocked by a single obstacle. The key observation is that, in either case, the obstacle divides the interior of the convex hull by touching the non-edge in the convex hull. This necessitates that the other obstacle is either entirely to the right, or entirely to the left of this first obstacle, since it must also touch the same convex hull non-edge.

1. If a single obstacle touches both the non-edges v_rv_5 and v_rv_6, we would need two convex obstacles to cover both the non-edges v_1v_2 and v_3v_4, which gives us our contradiction.

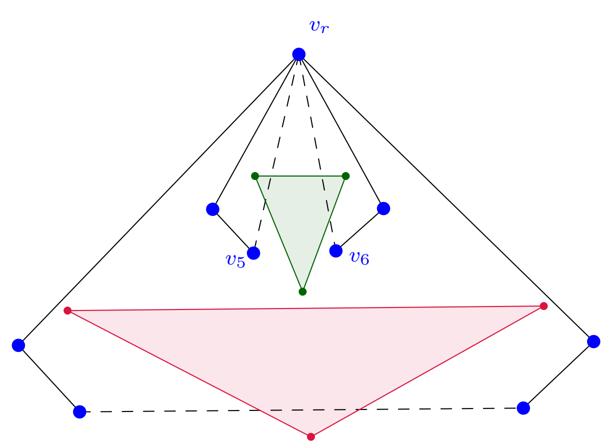
2. If each obstacle touches exactly one of these non-edges, then we also get a contradiction. We'll say without loss of generality that the first obstacle intersects the non-edge v_rv_6, but not v_rv_5. Then we see that the other obstacle must intersect both v_rv_5 and v_3v_4, which is impossible.

Thus we have eliminated all cases where both obstacles touch outside the convex hull, and it follows that there must be a single obstacle completely contained within the hull, as desired. □

The structure of this proof allows us to take a step further, and make the following claim.

Corollary 4.4. *In any 2-convex obstacle representation of the 4-claw from Theorem 4.3, the obstacle that is completely contained within the convex hull must touch both the non-edges $v_r v_5$ and $v_r v_6$.*

Proof. First consider what happens when the interior obstacle touches only one of the non-edges, say $v_r v_5$ without loss of generality. Then the other obstacle must touch both the non-edge $v_r v_6$ and the non-edge in the convex hull. However, by case 2 in the proof of Theorem 4.3, we see that this leads to a configuration that requires 3 obstacles, which is a contradiction. A similar argument can be made if we say that the interior obstacle does not touch either non-edge, as then we enter case 1 in the proof of Theorem 4.3, in which the other obstacle must touch both of these non-edges as well as the one in the convex hull. If we allow the interior obstacle to touch both $v_r v_5$ and $v_r v_6$, however, then we can create the following legal 2-convex obstacle representation:



This completes the proof of Corollary 4.4. □

4.3 Alternate Proofs of Theorem 1

After we introduced the new preliminaries in section 3, it became clear that there were simpler proofs of Theorem 1 that made use of the Lemmas from that section. For example, our original proof that the 3-claw has convex obstacle number 2 can be reduced to the following:

New Proof of Theorem 1. Consider any 1-convex obstacle representation of a 3-claw T. By Lemma 3.2, we know that all neighbors of the root vertex v_r must be on one side of a line through v_r. Let v_1, v_2, and v_3 be these neighbors in counterclockwise order about v_r, and let v_4 be the leaf vertex adjacent to v_3. Then $v_r v_3 v_4$ is a path of length 2 that must be at least partially contained in triangle $\Delta v_1 v_r v_2$, and it follows from Lemma 2.3 that the 3-claw does not have convex obstacle number 1, as desired. □

With this argument and our construction of a 1-convex obstacle representation for Caterpillar graphs in section 2.2, we have a more succinct and elegant proof of our first main result. However, it turns out that we can derive a direct proof of Theorem 1 that does not require a construction or consider the 3-claw at all. This proof follows nicely from the following pair of observations, which are useful restatements of earlier Lemmas.

Observation 4.5. *In any 1-convex obstacle representation of a tree T, If v_1, v_2, v_3, and v_4 are any 4 vertices of T in non-convex position (say v_4 is in the convex hull of v_1, v_2, and v_3) then v_4 is adjacent to at most one of v_1, v_2, or v_3.*

This new observation follows from a familiar observation in section 2. If v_4 were adjacent to two other vertices, say v_1 and v_2, then observation 2.1 tells us that the non-edges $v_1 v_2$ and $v_3 v_4$ require separate convex obstacles, which is a contradiction. Our second observation is similar to this last one, except now we consider what happens when all four vertices are in convex position.

Observation 4.6. *In any 1-convex obstacle representation of a tree T, If v_1, v_2, v_3, and v_4 are any 4 vertices of T in convex position, ordered clockwise about a point within their convex hull, then neither diagonal $v_1 v_3$ nor $v_2 v_4$ forms an edge in T.*

Once again this observation is easy to verify, as it follows from Lemma 2.6. If one of these diagonals, say $v_1 v_3$, were an edge in our tree T, then the convex hull of the four vertices would contain four non-edges, namely $v_1 v_2$, $v_2 v_3$, $v_3 v_4$, and $v_4 v_1$. By Lemma 2.6, all but one of these non-edges must actually be an edge, but there is no choice of three edges in this set that does not create a cycle with $v_1 v_3$. This gives us our contradiction, and it follows that neither diagonal can be an edge in T, as desired.

With these two observations, we can now formalize our proof as follows.

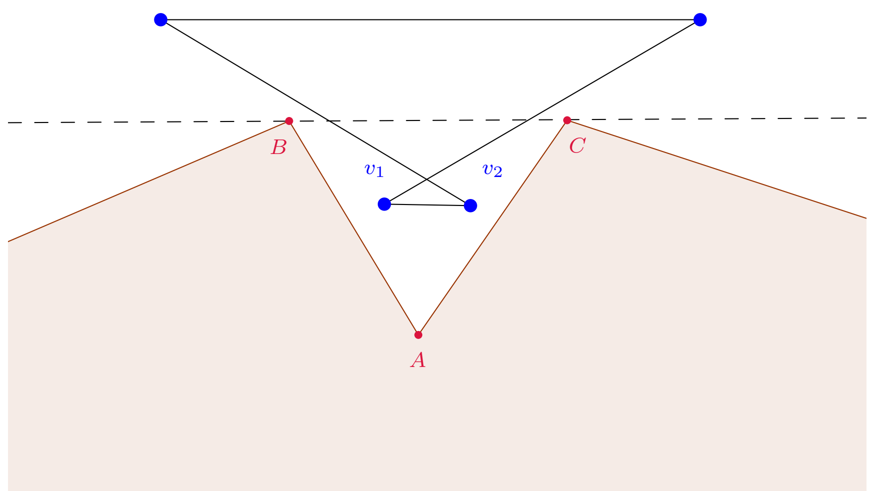
Direct Proof of Theorem 1. Consider any 1-convex obstacle representation of a tree T. By Lemma 2.6, there must be exactly one non-edge in the convex hull of T, which implies that the vertices $v_1, v_2, \ldots, v_k$ on the hull form a path. Now for all $k \in [2, k)$, consider the triangular region formed by v_{k-1}, v_k, and v_{k+1}. By observation 4.6 above, we know that no edge that is incident to a vertex in $\Delta v_{k-1} v_k v_{k+1}$ can cross the non-edge $v_{k-1} v_{k+1}$. Furthermore, observation 4.5 tells us that there cannot be a path of length 2 inside $\Delta v_{k-1} v_k v_{k+1}$. Thus every vertex v_k in the convex hull of T can only be adjacent to leaves and other vertices in the convex hull. It follows that T must be a Caterpillar graph, as desired. □

4.4 Obstacles with Dents

Our final section of additional results was born from questioning of the power we gain by moving slightly further away from the convex obstacles we have been using almost exclusively in this paper. To formalize this idea, we let a *dent* in a convex obstacle be a triangular region in the convex hull of the obstacle in which the triangle shares exactly two vertices with the polygonal obstacle, while the third vertex can be found inside the convex hull. We say that an obstacle has n dents if it has n such triangles within the obstacle's convex hull such that none of them overlap. We are interested in studying what kinds of trees have representations that require a single dented convex obstacle. We define such an *n-dented obstacle representation* as a representation of a tree T in which all non-edges are blocked by a single convex obstacle that contains n dents. To this end, we have the following Theorem:

Theorem 4.7. *A tree T has a 1-dented obstacle representation if and only if it is a Caterpillar graph, or a Caterpillar graph plus a path of length 2.*

Proof. First, notice that any vertices inside of a triangular region that does not contain any obstacles must form a clique. This means that the total number of vertices that can be inside the triangular region formed by a dent is two, since we are only considering graphs without cycles. We now want to prove that these vertices must form a path, that is, one of the vertices is a leaf, and the other has degree 2. We proceed by contradiction. Let the dent of our obstacle be formed by the vertices A,B, and C in the polygonal obstacle, with A being the vertex on the interior of the convex hull of the obstacle. Now assume that the two vertices v_1 and v_2 inside the dent both have neighbors outside the indented triangle. First, notice that it must be the case that v_1 and v_2 do not share a neighbor, as if it did we would have a cycle. Furthermore, the neighbors must both be above the line BC, as otherwise they would not be visible to v_1 and v_2 inside the dent. However, as the following figure shows, this means that the neighbors of v_1 and v_2 can see each other, which means they must be connected. This also creates a cycle, which leads to a contradiction. A similar argument can be made if we assume that a single vertex inside the dent has two neighbors outside the dent, and it follows that the dent can only contain a path of length 2.



To complete the proof, consider the subgraph $S = T - v_1, v_2$. Notice that, since v_1 and v_2 are the only vertices in triangle ΔABC, it must be the case that S has a 1-convex obstacle representation, since all non-edges of S must be blocked by the convex hull of our obstacle. It follows that S is a Caterpillar graph, and that T is a Caterpillar graph with at most one extra path of length 2, as desired. $\square$

5 Conclusions

In this thesis, we found necessary and sufficient conditions for a tree to have convex obstacle number 1 or 2, but there is a lot of work left to be done. First, categorizations for all trees with convex obstacle number 3 and 4 still need to be found, an endeavor that is likely to prove challenging due to the possibility of including edge-crossings in representations with more than 2 convex obstacles. A natural conjecture to make is that convex obstacle number 3 implies Caterpillar number 3, and vice versa. However, this is unlikely to be the case, as most of the results used in the paper do not generalize to situations where edge-crossings are allowed. This means that a proper investigation along this line of research would require almost an entirely new set of Lemmas and restrictions.

Another question that arose during this project that may be worth exploring is the effect of edge contractions and edge deletions on the convex obstacle number of a tree. In our research, we noted that reduction to the 5-claw through edge-contractions was a necessary condition for a tree to have convex obstacle number 3 or greater. Earlier in the investigation, however, it was hypothesized that having the 5-claw as a graph minor was the proper characterization, but we were unable to prove anything in general about edge contractions and deletions, even though the effect of vertex deletions is well understood.

Lastly, a few interesting questions can be drawn from the additional results portion of this thesis. Of particular interest is the consequences of allowing dents in a convex obstacle. So far we only understand the power of triangular dents, but what about dents of other shapes? That is, dents in which more than one vertex can be found on the inside of the obstacle convex hull. At what point does allowing such dents make a single obstacle more powerful than k convex obstacles? Since we know that a single polygonal obstacle can represent any tree, there must be some minimum number of perturbations that need to be made to a convex obstacle before it can represent a given tree T. We believe that the study of dented obstacles may provide some insight into what types of perturbations are required to make trees have obstacle number 1.

References

[Al] H. Alpert, C. Koch, and J. Laison, *Obstacle numbers of graphs*, Discrete & Computational Geometry 44(1) (2010), 223-244.

[Fu] R. Fulek, Deniz Sariöz, N. Saeedi, *Convex obstacle number of outerplanar graphs and bipartite permutation graphs*, Thirty Essays in Geometric Graph Theory, Algorithms and Combinatorics Series, Springer, 2012.

[Gi] E. Giacomo, Jan Kratochvíl, W. Lenhart, G. Liotta, T. Mchedlidze, S. Wismath, and A. Wolff, *Outside Obstacle Representations of Planar Graphs*, preprint.

[JS] M. P. Johnson and D. Sariöz, *Computing the obstacle number of a plane graph*, CoRR, abs/1107.4624, 2011.

[Ko] A. Koch, M. Krug, I. Rutter, *Graphs with Plane Outside-Obstacle Representations*, preprint.

[Kr] J. Kratochvíl, *A special planar satisfiability problem and a consequence of its NPcompleteness*. Discrete Appl. Math., 52(3):233252, 1994.

[Mu] P. Mukkamala, J. Pach, and D. Sariöz, *Graphs with large obstacle numbers*, Dimitrios M. Thilikos, volume 6410 of Lecture Notes in Computer Science, pages 292303, 2010.

[PS] J. Pach, D. Sariöz, *Small (2,s)-colorable graphs without 1-obstacle representations*, CoRR, abs/1012.5907, 2010.

[Sa] D. Sariöz, *Approximating the obstacle number for a graph drawing efficiently*, CCCG, 2011.