Bounding Tree Cover Number and Positive Semidefinite Zero Forcing Number

Sofia Burille   Mentor: Michael Nathanson

September 15, 2014

Abstract

Given a graph, G, with a set of vertices, v, and edges, various properties can be calculated. Two of these properties, or graph parameters, are the tree cover number and the positive semidefinite zero forcing numbers, $T(G)$ and $Z_+(G)$ respectively. Methods of easily bounding $T(G)$ and $Z_+(G)$ are presented. The general procedure is to break a graph G, into multiple smaller subgraphs G_i at specific vertices, separating vertices. Tree cover number of smaller graphs are easier to calculate, so $T(G_i)$ for each subgraph is calculated and combined to estimate the tree cover number of $T(G)$. These estimations and bounds can be more or less accurate depending of the structure of the graph. Different cases are examined to conclude results specific to certain kind of graphs. The same process is applied to positive semidefinite zero forcing, $Z_+(G)$.

Activity Report

To begin this Mathematics research project, I had to first understand what graph theory is and the more recent discoveries relating to the field. So the for the first part of the summer I was reading textbooks and academic papers relating to the topic. Never having read Mathematics research papers I had to learn the various vocabulary, style, and formatting of it. Then I started collecting data. Collecting data consisted of some-what systematically generating graphs, then calculating various parameters of each graph. Essentially I was looking for patterns and insights on how the graphs were behaving in relation to one another. This eventually led me to some observations, which I would develop into propositions and some eventually theorems. I also worked on understanding some proofs in academic papers and even showing the proof held true for an example. Throughout the summer I also tried to develop more efficient ways of collecting data by using computer programs such as Sage or Mathematica (which required also learning the programs themselves). Throughout the summer I would discuss my progress and questions with my mentor, Professor Nathanson. In our discussions, we helped each other with the understanding of the material, a proof, or ideas from the data.

1 Introduction

Graph Theory is the study of graphs. Graphs are a way to model many things from everyday life. Graph parameters are measurable properties of graphs. I investigated methods to bound 2 parameters, tree cover number and the positive semidefinite zero forcing number, which help bound a third more important parameter, minimum semidefinite rank, $T(G)$, $Z_+(G)$, and mr_+ respectively. In [4], cut vertex reduction is formulated. A cut vertex is a vertex such that when removed disconnects the graph. In graphs that contain a cut vertex, the graph can be divided into smaller graphs, called subgraphs, G_i, to more easily calculate $T(G)$ and $Z_+(G)$. We expand this idea to multiple separating vertices, in which removing those vertices disconnects the graph. We make the following definitions $X = \{v_1, v_2, \ldots, v_n\}$ is the set of separating vertices, v_i is a vertex in X, $T_1(G)$ is the size of the minimum tree cover of G which has all v_i in the *same* tree, $T_2(G)$ is the size of the minimum tree cover of G in which the separating vertices are comprised of 2 trees, and $T_k(G)$ is the size of the minimum tree cover of G in which the separating vertices are comprised of k trees.

2 Preliminaries

A graph is made up of vertices and edges between them. Two vertices are neighbors, or connected, of there is an edge between them. The main portion of my research was focused on the graph parameter tree cover number, $T(G)$. In [6], the tree cover property is created and defined. The tree cover number of a multigraph G, denoted $T(G)$, is the minimum number of vertex disjoint simple trees occurring as induced subgraphs of G that cover all of the vertices of G. Essentially what that means is the minimum number of trees, which is a type of graph, needed to cover all of a graph G. An induced subgraph means that one take a portion a graph, but this part of the graph, if it contains any 2 vertices that contain an edge in G, then the edge between them must also be present in the subgraph. So a tree cover of any graph will have a collection of trees, all of which will be induced subgraphs of G. So tree covers are collections of induced subgraphs. Another parameter i investigated was positive semidefinite zero forcing number. The positive semidefinite zero forcing number is given a set of initial black vertices that using the forcing rules can force the other vertices to be black, what is the minimum length of the initial set needed to be able to turn all of the vertices black?

3 Induced Subgraphs

The simplest expansion of the cut vertex reduction is two separating vertices reduction. Even this extra step complicates the problem significantly. Although the tree cover number is the minimal amount of disjoint trees, these trees can often be created in several ways. There are often multiple optimal tree covers. These tree covers can have the separating vertices in many combinations of same or different trees. Since more than a single vertex is included in every G_i we need to verify that they still create a tree covering for G. It is obvious that when starting with a tree cover for G, the resulting tree covers for G_i still cover each vertex in G_i and each vertex of the tree cover. However edges could be missing and result in disconnected trees, causing a general induced forest instead of a specific induced tree cover. We prove the following Lemma to eliminate this problem.

Lemma 3.1. *Given a tree cover $\mathfrak{T}$ of G, let H be an induced subgraph of G. Then,*

$H \bigcap \mathfrak{T}$ *is a collection of subgraphs which serve as a tree cover of H.*

Proof

Start with G. $\mathfrak{T}$ is a collection of subgraphs that contains all of the vertices in G. Then we take the subgraph of $\mathfrak{T}$, $H \bigcap \mathfrak{T}$, which is the restriction of $\mathfrak{T}$. $\mathfrak{T}$ contains all the vertices H could possibly have, therefore the vertices of $H \bigcap \mathfrak{T}$ are exactly those of H. Since the edges of $H \bigcap \mathfrak{T}$ are no more than $\mathfrak{T}$, which is comprised only of trees, then $H \bigcap \mathfrak{T}$ can only be either those trees, or those trees minus some edges, which simply creates more trees. Therefore $H \bigcap \mathfrak{T}$ is still an induced tree cover.

4 Relating $T_k's$

In relating different T_k's to each other some more interesting results arose; however, we have not been able to find an appropriate application for them.

Theorem 4.1. *Limits*
For any graph G, with $1 < k \leq$ the number of separating vertices,

$$T_{k-1}(G) \geq T_k(G) - 1 \tag{1}$$

Proof

If $k = n$ then when looking at T_{k-1}, since there are $n-1$ trees but n vertices, then 1 tree has to have 2 of the v_i together. If $k < n$ then at least 1 tree has to have 2 or more of the v_i together. Let v_1 and v_2 be in the same tree together. There exists a distinct path between v_1 and v_2. If we remove one of the edges in that path then v_1 and v_2 will be disconnected. What was once 1 tree, will now be 2 trees, with v_1 and v_2 being in 2 different trees, which adds exactly 1 to $|\mathfrak{T}|$. So for a particular G, if $T_{k-1}(G) < T_k(G)$, we can make a T_k cover from a T_{k-1} cover by adding exactly one tree.

This theorem was in part created to be able to relate these terms we created to each other and to $T(G)$ in general. Since T_{k-1} is often greater than or equal to T_k, an example being T_1 is usually greater than or equal to T_2, I investigated when is T_1 smaller than T_2? This was in hopes to better understand $T(G)$ in general. However, this question was difficult to answer and is open to further research.

5 Initial Point

The simplest expansion of the cut vertex reduction is two separating vertices reduction. The goal of expanding the cut vertex reduction is that many graphs do not have a single vertex, in which case the theorem cannot be applied. Expanding to two separating vertices, more graphs can be included, with the end goal of expanding to any amount of separating vertices.

5.1 Graphs with Two Separating Vertices

Theorem 5.1. *Let G be a graph with vertex set V. Let X be fixed $\{v_1, v_2\} \in V$, such that X is a separating set for G. Let $\{W_1, \ldots, W_h\}$ be the vertices of the connected components of $G \backslash \{v_1, v_2\}$; and let $G_i = G[W_i \bigcup X]$. Let $h =$ the number of subgraphs.*

$$T(G) \geq \left(\sum_{i=1}^{h} T(G_i) \right) - 2(h-1) \qquad (2)$$

Proof

Start with an optimal tree cover, $\mathfrak{T}$, for G. Break apart G at v_i into subgraphs, G_i, and consequently the optimal tree cover into tree covers for each G_i, which using Lemma 3.1 Trees not containing v_i will be in only one subgraph. Since there are 2 separating vertices in v_i and therefore only 2 vertices that are in multiple subgraphs, the highest amount of trees the v_i can be broken up into is 2 trees. We are calculating a lower bound, we want to subtract off the greatest possible amount of extra trees to get the smallest value. Since 2 is the highest value, we will focus on v_i being in 2 different trees in the tree cover for G to prove our bound. So each G_i will have at least 2 trees. The 2 trees each with a different v_i, will be in each subgraph, meaning they will each be counted h times. We want them counted only once so they are counted $h-1$ extra times. In our final equation we want to subtract this extra. Since we have 2 many different trees then we are subtracting $2(h-1)$, which is the highest we can subtract. The bound is as small as possible. Putting all these trees together and subtracting the extra will give us a size of a tree cover that is a lower bound because this tree covering cannot be made any less optimally. Therefore

$$T(G) \geq \left(\sum_{i=1}^{h} T(G_i) \right) - 2(h-1).$$

□

There is difficulty in building an optimal tree cover, therefore attaining $T(G)$, G from optimal tree covers of each G_i. This is why we have "$\geq$" instead of "=" in our formula. However in certain cases an optimal tree cover of G can be made from optimal tree covers of each G_i. Therefore with some certain assumptions made, we can have "=" in our formula. This means under certain cases, combining for all i,$T(G_i)$ can be used to calculate $T(G)$.

Theorem 5.2. *Let G be a graph with vertex set V. Let X be fixed $\{v_1, v_2\} \in V$, such that X is a separating set for G. Let $\{W_1, \dots, W_h\}$ be the vertices of the connected components of $G \backslash \{v_1, v_2\}$; and let $G_i = G[W_i \bigcup X]$. For any G such that $v_1 \sim v_2$,*

$$T_1(G) = \left(\sum_{i=1}^{h} T_1(G_i) \right) - (h-1) \tag{3}$$

and

$$T_2(G) = \left(\sum_{i=1}^{h} T_2(G_i) \right) - 2(h-1) \tag{4}$$

Proof

Start with a minimal tree cover, $\mathfrak{T}_i$, for each G_i. For each $\mathfrak{T}_i$ there exists some T_{i_1} such that $v_1 \in V(T_{i_1})$, and some different T_{i_2} such that $v_2 \in V(T_{i_2})$. For $T_i \neq \{T_{i_1}, T_{i_2}\}, T_i$ is contained in only one G_i, therefore appears only once in $\sum_{i=1}^{h} T(G_i)$, and translates simply to be only 1 tree in exactly 1 T_i therefore exactly 1 T_i in $\mathfrak{T}$. Let T_{v_1} be $\bigcup_{i=1}^{h}(T_{i_1})$. Because all T_i's are induced trees and we are just glueing all the trees together at their one common vertex, v_1, T_{v_1} is still a connected tree. Since we are glueing all $T_{i_1}'s$ together at v_1 there is no way for there to be an edge between 1 vertex in one G_i to another except through v_1, therefore there is a unique path from any vertex to another. So T_{v_1} is the induced tree in G that contains v_1. Let T_{v_2} be $\bigcup_{i=1}^{h}(T_{i,2})$, meaning T_{v_2} is the induced tree in G that contains v_2. Then $\mathfrak{T} = (\bigcup_{i=1}^{h}(\mathfrak{T}_i) \setminus \{T_{i,1}, T_{i,2}\}) \bigcup \{T_{v_1} \bigcup T_{v_2}\}$ is a tree cover for G because all T_i are induced trees and $V(\mathfrak{T}) = V(G)$ (All the vertices in G are in $\mathfrak{T}$). Since we found a tree cover the minimum tree cover can be no bigger than the tree cover we just created.
Therefore,

$$T_2(G) \leq \left(\sum_{i=1}^{h} T_2(G_i) \right) - 2(h-1)$$

Combining with 5.1 we get

$$T_2(G) = \left(\sum_{i=1}^{h} T_2(G_i) \right) - 2(h-1)$$

Start with a minimal tree cover, $\mathfrak{T}_i$, for each G_i. Since $v_1 \sim v_2$, and v_1 and v_2 are in the same tree then the tree containing them, because tree covers are collections of induced subgraphs, must go through the edge between them. All the trees in each G_i containing v_1 and v_2 are all glued together at v_1 and v_2. Therefore when glueing all these trees to make one tree, we know it is still a connected tree because there is no way for there to be an edge between 1 vertex in one G_i to another except through v_1 or v_2. Adding this tree to all the other trees gives us a tree cover for G. Since we found a tree cover the minimum tree cover can be no bigger than the tree cover we just created.
Therefore,

$$T_1(G) \leq \left(\sum_{i=1}^{h} T_1(G_i) \right) - (h-1)$$

Combining with 5.1 we get

$$T_1(G) = \left(\sum_{i=1}^{h} T_1(G_i) \right) - (h-1)$$

□

From these we see certain patterns arising and build to multiple vertices.

6 Multiple separating vertices

The 2 vertice general bound conjecture can be expanded to multiple vertices. Each increasing vertex simply adds one more vertex, and therefore tree that can be counted extra times.

Theorem 6.1. *General Bound*
With assumptions and definitions above, for any graph G,

$$T(G) \geq \left(\sum_{i=1}^{h} T(G_i) \right) - n(h-1) \tag{5}$$

Proof
Start with an optimal tree cover, $\mathfrak{T}$, for G. Break apart G at v_i into subgraphs, G_i, and consequently the optimal tree cover into tree covers for each G_i, which using Lemma 3.1 Trees not containing v_i will be in only one subgraph. Since there are n separating vertices in v_i and therefore only n vertices that are in multiple subgraphs, the highest amount of trees the v_i can be broken up into is

n trees. We are calculating a lower bound, we want to subtract off the greatest possible amount of extra trees to get the smallest value. Since n is the highest value, we will focus on v_i being in n different trees in the tree cover for G to prove our bound. So each G_i will have at least n trees. The n many trees each with a different v_i, will be in each subgraph, meaning they will each be counted h times. We want them counted only once so they are counted $h-1$ extra times. In our final equation we want to subtract this extra. Since we have n many different trees then we are subtracting $n(h-1)$, which is the highest we can subtract. The bound is as small os possible. Putting all these trees together and subtracting the extra will give us a size of a tree cover that is a lower bound because this tree covering cannot be made any less optimally. Therefore $T(G) \geq \left(\sum_{i=1}^{h} (T(G_i) \right) - n(h-1)$.

□

In the 2 connected vertices equality proof, when both vertices were in the different trees, the problem was simplified somewhat. This is in part due to how this specific format some some way mirrors the original cut vertex reduction, but instead of one vertex be in all G_i there are more vertices in all G_i but they do not interact with each other. This brings us to a a theorem with a substantial assumption, but very useful result. Recall that $T_k(G)$ is the size of the minimum tree cover of G in which the separating vertices are comprised of k trees. Also, we define n as the number of separating vertices in the graph, G.

An Example

No assumptions: $T(G) \geq \left(\sum_{i=1}^{h} T(G_i) \right) - n(h-1)$, where $n =$ the number of separating vertices and $h =$ the number of subgraphs G_i

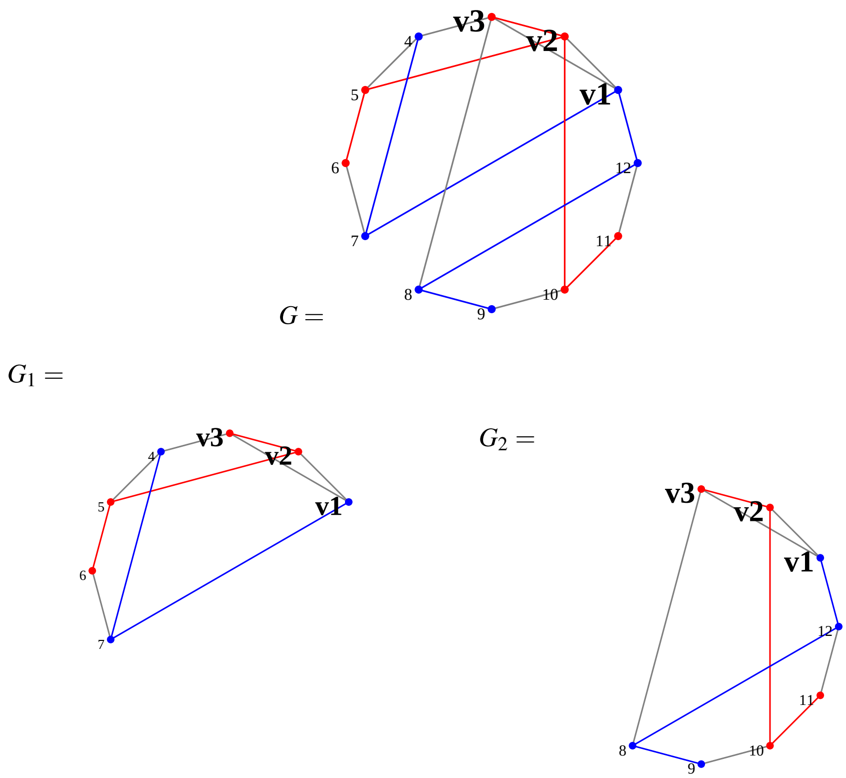


Figure 1: An example of a graph, G, with 3 separating vertices, v_1, v_2, and v_3. and its two subgraphs G_1 and G_2

It was found that $h = 2, n = 3$ because there are 2 subgraphs and 3 separating vertices. It was also found that $T(G) = 2$ and for both i, $T(G_i) = 2$. Using our formula of an estimate of the tree cover number: $T(G) \geq 2 + 2 - 3(2 - 1) = 1$, therefore $T(G) \geq 1$, which is tree because $T(G) = 2$.

Conjecture 6.2. *T_k EQUALITY*
For any graph G, where n is the number of separating vertices and each of the separating vertices are in different trees.

$$T_n(G) = \left(\sum_{i=1}^{h} T_n(G_i) \right) - n(h-1) \tag{6}$$

Proof
Start with a minimal tree cover, $\mathfrak{T}_i$, for each G_i. For each $\mathfrak{T}_i$ there exists some T_{i_1} such that $v_1 \in V(T_{i_1})$, some different T_{i_2} such that $v_2 \in V(T_{i_2})$, ... and some other different T_{i_n} such that $v_n \in V(T_{i_n})$. For $T_i \neq \{T_{i_1}, T_{i_2}, \ldots, T_{i_n}\}, T_i$ is contained in only one G_i, therefore appears only once in $\sum_{i=1}^{h} T(G_i)$, and translates simply

to be only 1 tree in exactly 1 T_i therefore exactly 1 T_i in $\mathfrak{T}$. Let T_{v_1} be $\bigcup_{i=1}^{h}(T_{i_1})$. Because all T_i's are induced trees and we are just glueing all the trees together at their one common vertex, v_1, T_{v_1} is still a connected tree. Since we are glueing all $T_{i_1}'s$ together at v_1 there is no way for there to be an edge between 1 vertex in one G_i to another except through v_1, therefore there is a unique path from any vertex to another. So T_{v_1} is the induced tree in G that contains v_1. Let T_{v_2} be $\bigcup_{i=1}^{h}(T_{i,2})$, meaning T_{v_2} is the induced tree in G that contains v_2. Let T_{v_n} be $\bigcup_{i=1}^{h}(T_{i,n})$, meaning T_{v_n} is the induced tree in G that contains v_n. Then $\mathfrak{T} = (\bigcup_{i=1}^{h}(\mathfrak{T}_i) \setminus \{T_{i,1}, T_{i,2}, \ldots, T_{i,n}\}) \bigcup \{T_{v_1} \bigcup T_{v_2} \ldots \bigcup T_{v_n}\}$ is a tree cover for G because all T_i are induced trees and $V(\mathfrak{T}) = V(G)$ (All the vertices in G are in $\mathfrak{T}$). Since we found a tree cover the minimum tree cover can be no bigger than the tree cover we just created.
Therefore,

$$T_n(G) \leq \left(\sum_{i=1}^{h} T_n(G_i) \right) - n(h-1)$$

Combining with the general bound result,

$$T_n(G) \geq \left(\sum_{i=1}^{h} T_n(G_i) \right) - n(h-1)$$

we get:

$$\left(\sum_{i=1}^{h} T_n(G_i) \right) - n(h-1) \leq T_n(G) \leq \left(\sum_{i=1}^{h} T_n(G_i) \right) - n(h-1).$$

Therefore, $T_n = \left(\sum_{i=1}^{h} T_n(G_i) \right) - n(h-1)$, when all v_i are in different trees.

□

The problem with trying to have an exact calculation, instead of just an estimation, of $T(G)$ using $T(G_i)$ is when we are building a tree cover of G using the tree covers of G_i the graphs have to be able to glue together at a distinct part of a tree. This is a problem when two of the separating vertices are in the same tree, but not through an edge between them, or if there is not consistency among the trees with separating vertices. With more assumptions we can get around some of these problems.

Conjecture 6.3. *For any graph G with v_i "tree" connected, with "tree" connected meaning that there is a distinct path between all v_i,*

$$T_1(G) = \sum_{i=1}^{h} (T_1(G_i)) - (h-1) \tag{7}$$

Start with a minimal tree cover, $\mathfrak{T}_i$, for each G_i. For each $\mathfrak{T}_i$ there exists some T_{i_v} such that $v_i \in V(T_{i_v})$. For $T_i \neq \{T_{i_v}\}, T_i$ is contained in only one G_i,

therefore appears only once in $\sum_{i=1}^{h} T(G_i)$, and translates simply to be only 1 tree in exactly 1 T_i therefore exactly 1 T_i in $\mathfrak{T}$. Let T_v be $\bigcup_{i=1}^{h}(T_{i_v})$. Since all the v_i are in the same tree together then all the edges between them will be in all $\mathfrak{T}_\mathrm{i}$. Each vertex that is not v_i, but is in a tree with v_i, will have a unique path to v_i. Because all T_{i_v}'s are induced trees and we are just glueing all the trees together at their one common root, the v_i tree, and glueing them together nowhere else, T_v is still a connected tree. Since we are glueing all $T_{i_v}'s$ together at v_i there is no way for there to be an edge between 1 vertex in one G_i to another except through v_i, therefore there is a unique path from any vertex to another. So T_{v_i} is the induced tree in G that contains all v_i. Then $\mathfrak{T} = (\bigcup_{i=1}^{h}(\mathfrak{T}_i) \setminus \{T_{i,v}\}) \bigcup \{T_v\}$ is a tree cover for G because all T_i are induced trees and $V(\mathfrak{T}) = V(G)$ (All the vertices in G are in $\mathfrak{T}$). Since we found a tree cover the minimum tree cover can be no bigger than the tree cover we just created.
Therefore,

$$T_1(G) \leq \left(\sum_{i=1}^{h} T_1(G_i) \right) - (h-1)$$

Combining with the general bound result,

$$T_1(G) \geq \left(\sum_{i=1}^{h} T_1(G_i) \right) - (h-1) \tag{8}$$

we get:

$$(\sum_{i=1}^{h} T_1(G_i)) - (h-1) \leq T_1(G) \leq (\sum_{i=1}^{h} T_1(G_i)) - (h-1).$$

Therefore, $T_1 = (\sum_{i=1}^{h} T_1(G_i)) - (h-1)$, when all v_i are "tree" connected. □

The various results and specific cases are summed up in a few tables.

v_i	E	T_1	T_2
	1	$=$	$=$
	0	$\geq$	$=$

Table 1: Summary of results when there are 2 separating vertices, when $n = 2$

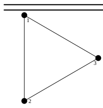
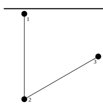
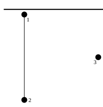
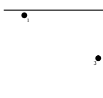
v_i	E	T_1	T_2	T_3
	3	DNE	$\geq$	$=$
	2	$=$	$\geq$	$=$
	1	$\geq$	$\geq$	$=$
	0	$\geq$	$\geq$	$=$

Table 2: Summary of results when there are 2 separating vertices, when $n = 3$

$$T_k(G) \geq \left(\sum_{i=1}^{h} T_k(G_i) \right) - n(h-1), \; n = \text{number of separating vertices}$$

v_i : Example configuration of how v_i could be arranged in G.
E : The number of edges between v_i
$\geq$: Case with only inequality proven in formula
$=$: Case with equality proven in formula

There can be a variety of assumptions made so similar results are attained. One way to enforce consistency without restricting the tree cover to such specific types of configuration, we simply enforce consistency by starting with a tree cover of v_i.

Conjecture 6.4. *MN CONJECTURE 4*
Let G be a graph with vertex set V such that $X \subset V$ is a separating set for G, as usual, and let $\mathfrak{S}$ be a fixed tree cover of X.

For any graph H which contains X, define $T_{\mathfrak{S}}(H)$ to be the size of the minimum *tree cover of H which reduces to $\mathfrak{S}$ on X. Then*

$$T_{\mathfrak{S}}(G) = \sum_{i=1}^{h} T_{\mathfrak{S}}(G_i) - |\mathfrak{S}|(h-1) \tag{9}$$

Proof

Start with a tree cover, $\mathfrak{T}$ for G, break it into tree covers, $\mathfrak{T}_{\mathfrak{i}}$ for each G_i. Trees not containing v_i will only be in 1 G_i and remain as induced trees, as known from Lemma 3.1. Since we started with $\mathfrak{S}$, we extend those trees to the rest of G and consequently to each G_i. Each G_i will have $\mathfrak{S}$ as an induced subgraph. For any two vertices that are in the same tree, if the path between them passes through another G_i, it does so through $\mathfrak{S}$, which is a tree cover of X that is in G_i. Therefore $\mathfrak{T}_{\mathfrak{i}}$ is still an induced tree.

There are at least $|\mathfrak{S}|$ trees in each G_i. So the trees in $\mathfrak{S}$ will be counted for each G_i, we want them counted only once, so they are counted extra $h-1$ times. This happens for each tree in $\mathfrak{S}$, so we have $|\mathfrak{S}|(h-1)$ total extra. Therefore,

$$T_{\mathfrak{S}}(G) \geq \sum_{i=1}^{h} T_{\mathfrak{S}}(G_i) - |\mathfrak{S}|(h-1) \tag{10}$$

Starting with a tree cover for each G_i we build them up to get a tree cover for G. Trees not containing v_i will only be in 1 G_i and are just one tree in G. Each G_i will have the trees that are in $\mathfrak{S}$. These trees can all be glued at each tree in $\mathfrak{S}$, which are in every G_i. Since we are glueing all these trees at a single common base vertex, they form an induced tree covering, which covers all the vertices not in v_i trees. Adding these trees to the trees not containing v_i, gives a tree cover for G. The optimal tree cover cannot be any smaller than this tree cover.
Therefore,

$$T_{\mathfrak{S}}(G) \leq \sum_{i=1}^{h} T_{\mathfrak{S}}(G_i) - |\mathfrak{S}|(h-1) \tag{11}$$

Combining (10) and (11) we get,

$$\sum_{i=1}^{h} T_{\mathfrak{S}}(G_i) - |\mathfrak{S}|(h-1) \leq T_{\mathfrak{S}}(G) \leq \sum_{i=1}^{h} T_{\mathfrak{S}}(G_i) - |\mathfrak{S}|(h-1)$$

Therefore,

$$T_{\mathfrak{S}}(G) = \sum_{i=1}^{h} T_{\mathfrak{S}}(G_i) - |\mathfrak{S}|(h-1)$$

7 Relating $T(G)$ to $Z_+(G)$

Zero forcing, $Z(G)$, and positive semidefinite zero forcing, $Z_+(G)$, are the minimum amount of initial vertices, which make up the Forcing Set, which then "force" other vertices so that the whole graph is covered. $Z(G)$ and $Z_+(G)$ are comparable to $P(G)$ and $T(G)$ respectively. From a forcing set, one can produce a forcing tree cover, which is similar to trees in tree covers. We try to establish connections between $T(G)$ and $Z_+(G)$. Since $Z_+(G)$ is more restrictive then $T(G)$, not every tree cover can be made into a forcing tree cover. I created a algorithm to find out if a specific tree cover could be turned into a forcing tree cover, and if yes, how to do so. The algorithm was never completely finished and is open to further investigation.

However, in trying to understand the previously stated question of when is $T_1(G) < T_2(G)$, in a different way, I was able to relate $T(G)$ and $Z_+(G)$ in a very specific way in certain cases.

There is known relationship between $T(G)$ and $Z_+(G)$ that $Z_+(G) \leq T(G)$. From this, it can be seen that forcing tree covers are similar in some ways to tree covers. So saying two vertices are in the same tree is similar to saying to vertices are in the same forcing tree, which means they are not in the same forcing set for that particular forcing set. If two vertices are in different forcing trees, however, we cannot say whether or not they can be in the same forcing set together.

Proposition 7.1. *Fix $\{v_1, v_2\}$ and let $v_1 \sim v_2$. Let G be a graph such that $T(G) = Z_+(G)$. If $T_1(G) < T_2(G)$ then no optimal zero forcing set contains both v_1 and v_2.*

Proof

Let G be a graph such that $T_1(G) < T_2(G)$. The edge v_1v_2 will be included in every single optimal tree cover of G. Every forcing tree cover can be created from a tree cover. Because $T(G) = Z_+(G)$, every optimal forcing tree is also an optimal tree cover. Since $v_1 \sim v_2$, one v always has to force the other v. Therefore v_1 and v_2 can never be in the same optimal forcing set.

□

8 Positive Semidefinite Zero Forcing

Since positive semidefinite zero forcing has similarities to tree cover number, I was able to make similar conclusions with positive semidefinite zero forcing.

Lemma 8.1. *Let $\mathfrak{T}$ be a forcing tree cover of G, let H be an induced subgraph of G. $H \bigcap \mathfrak{T}$ is the restriction of $\mathfrak{T}$ onto H, making it a subgraph of $\mathfrak{T}$. Then,*

$H \bigcap \mathfrak{T}$ is a collection of subgraphs which serve as forcing tree cover of H.

Proof

Because $\mathfrak{T}$ is a forcing tree cover and $H \bigcap \mathfrak{T}$ is a induced subgraph of $\mathfrak{T}$, then $H \bigcap \mathfrak{T}$ must be a forest cover or tree cover. When restricting $\mathfrak{T}$, if trees become disconnected, what was once 1 tree simply becomes multiple trees that cover the same vertices. Since $H \bigcap \mathfrak{T}$ is an induced subgraph, the degree of a vertex in a component can only decrease, meaning u cannot be prevented from forcing because of too many neighbors. So if $u \rightarrow v$ and both u and v are still present, then $u \rightarrow v$. If u is no longer present then v is made it be part of the initial forcing set and forcing will continue as before. This happens for every uv pair in $H \bigcap \mathfrak{T}$. Therefore $H \bigcap \mathfrak{T}$ is still a collection of trees serving as an induced forcing tree cover.

This allows further results, similar to the $T(G)$ results.

Theorem 8.2. *Fix $v_1, v_2, \ldots, v_n$ to be our separating set Then,*

$$Z_+(G) \geq \left(\sum_{i=1}^{h} Z_+(G_i) \right) - n(h-1) \tag{12}$$

Proof

Start with an optimal forcing set $\mathfrak{F}$ for G. The optimal forcing set $\mathfrak{F}$ creates an optimal forcing tree cover, $\mathfrak{T}$, of G. There are at most n trees in $\mathfrak{T}$ intersecting our separating set, one containing v_1, one containing v_2, ..., another containing v_n. Because $\mathfrak{F}$ is an optimal forcing set we know that $|\mathfrak{T}| = Z_+(G)$. We break down this optimal forcing tree cover into covers for each G_i. Using our result from Lemma 8.1, we know these covers, $\mathfrak{T}_\mathfrak{i}$, are comprised of forcing trees for each G_i. These forcing trees give us forcing sets for each G_i. So we have a forcing tree cover, we just do not know if it is the optimal one, so $Z_+(G)$ cannot be any bigger than that. For any $\mathfrak{T}_\mathfrak{i}$, $|\mathfrak{T}_\mathfrak{i}| \geq Z_+(G_i)$. Then, sum up all the forcing trees. Of all the forcing trees in all G_i, only n at most, one for each v_i, can be counted for each G_i.They are counted h times, meaning they are counted extra $h-1$ times. Since there are at most n, we get at most $n(h-1)$ extra counted trees. So from our past proofs we know that

$$|\mathfrak{T}| \geq \left(\sum_{i=1}^{h} |\mathfrak{T}_\mathfrak{i}| \right) - n(h-1) \tag{13}$$

Since $|\mathfrak{T}_\mathfrak{i}| \geq Z_+(G_i)$ then

$$|\mathfrak{T}| \geq \left(\sum_{i=1}^{h} |\mathfrak{T}_\mathfrak{i}| \right) - n(h-1) \geq \left(\sum_{i=1}^{h} Z_+(G_i) \right) - n(h-1) \tag{14}$$

and since $|\mathfrak{T}| = Z_+(G)$, therefore,

$$Z_+(G) \geq \left(\sum_{i=1}^{h} Z_+(G_i) \right) - n(h-1) \tag{15}$$

Taking an optimal forcing set, making an optimal forcing tree cover, then splitting that into forcing tree covers for each G_i we find the minimum value of $Z_+(G)$. □

Using this theorem, in certain cases even more can be said about $T(G)$.

Theorem 8.3. *Fix v_i to be our separating set and let G have subgraphs G_i such that all v_i can be made as part of an optimal forcing set together for each G_i. Then,*

$$Z_+(G) = \left(\sum_{i=1}^{h} (Z_+(G_i) \right) - n(h-1)$$

Proof

Start with an optimal forcing set $\mathfrak{F}_\mathfrak{i}$ for each G_i in which $\{v_1, v_2, \ldots, v_n\} \in \mathfrak{F}_\mathfrak{i}$. Since each G_i have v_i as part of the optimal forcing set, then when we glue all the forcing sets together, defining $\mathfrak{F} = \cup_{i=1}^{h} \mathfrak{F}_\mathfrak{i}$, $\{v_1, v_2, \ldots, v_n\} \in \mathfrak{F}$. Also, since all $\{v_i\}$ are in a separating set then they will separate G into components, which are the subgraphs that we started with and we will still have all the initial vertices that we started with. Therefore the set is a forcing set for G. We know the optimal forcing set can only be as big as the size of the entire sum (right hand-side of the equation).
Therefore,

$$Z_+(G) \leq \left(\sum_{i=1}^{h} (Z_+(G_i) \right) - 2(h-1) \tag{16}$$

Combining this result with Theorem 8.2 we get

$$\left(\sum_{i=1}^{h} (Z_+(G_i) \right) - 2(h-1) \leq Z_+(G) \leq \left(\sum_{i=1}^{h} (Z_+(G_i) \right) - 2(h-1)$$

Therefore,

$$Z_+(G) = \left(\sum_{i=1}^{h} (Z_+(G_i) \right) - 2(h-1)$$

when given G_i such that all v_i are in an optimal forcing set together for each G_i. □

More may be able to be said about the bounds of $Z_+(G)$ in general or in specific cases, but these are unanswered questions/

9 Open Questions

As mentioned, one unanswered question that arose was when there are two separating vertices, when is T_1 smaller than T_2? This would allow the $T(G)$ to be calculated much more efficiently in many graphs. When we know which is smaller we know how to construct the tree cover, and when T_2 is smaller, then $T_2(G) = T(G)$, which makes calculating $T(G)$ much easier and efficient.

Another question asks what exactly are the connections between tree covers and zero forcing tree covers. Can they be related in a more substantial way? Are are there ways to turn tree covers into forcing tree covers?

Lastlt, are there more precise bounds that can be made on $T(G)$ using $T(G_i)$?

10 Conclusion

I worked on breaking a big graph, G, into smaller graphs, G_i, at the separating set, in order to bound $T(G)$ and $Z_+(G)$. This brought up a variety of special cases in which the bound became an exact calculation. I also related $T(G)$ to $Z_+(G)$ in more specific ways. Since there are many complicated graphs, whenever a graph can be broken down into simpler parts and those smaller parts are analyzed, it makes solving the problem much easier.

References

[1] X. Li, R. Phillips. *Minimum Vector Rank and Complement Critical Graphs*. arXiv:1304.3751 [math.CO], 2013.

[2] L Hogben. *Orthogonal representations, minimum rank, and graph complements*. Linear Algebra and its Applications, **Vol. 428**, 2007.

[3] M. Booth, et al. *On the Minimum Rank Among Positive Semidefinite matrices with a Given Graph*. Society for Industrial and Applied Mathematics, **Vol. 30**, 2008.

[4] Hogben et al. *Positive Semidefinite Zero Forcing Number*. Electronic Journal of Linear Algebra, **Vol. 23**, 2012.

[5] F. Taklimi, S. Fallat, K. Meagher. *On the Relationships between Zero Forcing Numbers and Certain Graph Coverings*. Special Matrices, **Vol. 2**, 2014.

[6] F. Barioli, et al. *Minimum Semidefinite Rank Of Outerplanar Graphs And The Tree Cover Number*. Electronic Journal of Linear Algebra. **Vol. 22**, 2011.

[7] F. Barioli, et al. Parameters Related To Tree-Width, Zero Forcing, And Maximum Nullity Of A Graph. Journal of Graph Theory, **Vol. 72**, 2012.

[8] J. Ekstrand, et al. Note on positive semidefinite maximum nullity and positive semidefinite zero forcing number of partial 2-trees. Electronic Journal of Linear Algebra, **Vol. 23**, 2012.